Mining Opinions from Google App Reviews-A Deep Learning Approach

Journal:	*Transactions on Data Science*
Manuscript ID	TDS-2020-0075
Manuscript Type:	Research Paper
Date Submitted by the Author:	14-Sep-2020
Complete List of Authors:	RANJAN, SAKSHI; Utkal University Post Graduate Departments, COMPUTER SCIENCE & ENGINEERING Mishra, Subhankar; National Institute of Science Education and Research, School of Computer Sciences
Computing Classification Systems:	Computing methodologies, Natural Language Processing, Computer System Organization

Mining Opinions from Google App Reviews-A Deep Learning Approach

SAKSHI RANJAN, Utkal University, India
SUBHANKAR MISHRA, National Institute of Science Education and Research, India

Google app market captures the school of thought of users from every corner of the globe via ratings and text reviews, in a multi-linguistic arena. The critique's viewpoint regarding an app is proportional to their satisfaction level. The potential information from the reviews can't be extracted manually, due to its exponential growth. So, Sentiment analysis, by machine learning and deep learning algorithms employing NLP, is one of explicitly uncovers and interpret the emotions. This study performs the sentiment classification of the app reviews and identify the university students' behavior towards the app market. We applied machine learning algorithms using TP, TF and TF-IDF text representation scheme and evaluating its performance on Bagging, an ensemble learning method. We used word embedding, GloVe on the deep learning paradigms. Our model was trained on Google app reviews and tested on Students' App Reviews(SAR). The various combination of these algorithms were compared amongst each other using F-score and accuracy and inferences were highlighted graphically. SVM, amongst other classifiers, gave fruitful accuracy(93.41%), F-score(0.89) on bi-gram+TF-IDF scheme. Bagging enhanced the performance of LR and NB with accuracy 87.88% and 86.69% and F-score 0.86 and 0.78 respectively. Overall, LSTM on Glove embedding recorded the highest accuracy(95.2%) and F-score(0.88).

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; *Supervised learning by classification*; • **Natural Language processing** → *Information extraction, Document Classification, Text mining and Sentiment Analysis*; • **Computer System organization** → **neural networks**.

Additional Key Words and Phrases: Sentiment analysis,Natural Language processing, NLP, Machine learning, Deep learning, University students reviews, Google Playstore apps., app reviews, visualization, ensemble methods, Word embedding.

ACM Reference Format:
Sakshi Ranjan and Subhankar Mishra. 2020. Mining Opinions from Google App Reviews-A Deep Learning Approach. 1, 1 (September 2020), 31 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

We are living in an era where technology and Internet have redefined social norms. There is no denying that mobile apps have changed every aspect of our lives completely[1]. Irrespective of what we want or need to do; everything is at our fingertips, just by discovering the relevant apps and reading the reviews and ratings posted by others. This helps in generation of profit for the developers, giving bug reports, request for new features, documentation of experience to analysts[2] and designers[3]. It gives information related to products, services, organizations, individual's issues, events, satisfaction or dissatisfaction with new features or business relevant information. Hence, the market intelligence via the purchasing inclination and attitudes of users are easily explicated by the software developers. Whether we are travelling[4], communicating[5], watching movie[6], ordering products[7], performing bank transaction, there is an

Authors' addresses: Sakshi Ranjan, Utkal University, Vani Vihar-751004, Bhubaneswar, Odisha, India; Subhankar Mishra, National Institute of Science Education and Research, 752050, Bhubaneswar, Odisha, India.

Manuscript submitted to ACM

app for everything, and so is the review. The proliferation of Google apps have helped us realise the rich interplay concerning the users-, trading-, and technologically concentrated traits[8].

The motivation for studying the Google app reviews and conducting this study is as follows. In the past decades, smart phones were uncommon so there were fewer interactions worldwide. Mainly the source of information was news, Internet and other sources of media. But nowadays, they are our inseparable companions. Internet, mobile technology and networking infrastructures, has brought inception and explosive growth of Google Play store and Apple store apps into being (like Facebook, Twitter, Instagram, Kindle, Amazon, Google pay etc.) [9]. People write and publicize their reviews and ratings from across the globe, based on the apps on their devices, satisfaction rate and likings[10]. These have coincided with social media on Web (reviews, discussions forum, blogs, micro-blogs, Twitter) and provided us with rich sources of data for researching[11]. Google Play store app market captures countless responses per month, for example Facebook captures 4000 reviews per day. It could be an unfeasible option to read millions of reviews and have an unbiased and consistent measure of the user's sentiment[12]. Therefore, there is a need of having a tool capable of analyzing the reviews. To this aim, using sentiment analysis and mechanizing this process, we can benchmark how users feel about apps without having to read thousands of user comments at once[13]. Particularly, just by studying the ratings for a given app, the criteria of understanding the mindset of a person, cannot be fulfilled. This is because ratings do not provide tangible statistics. The practicality of star ratings is confined solely to app developers as it combines positive and negative appraisals. So studying and analyzing the real time reviews is also a necessity. This study concatenated the huge repository of reviews and ratings(i.e., instantaneously available on Internet) with real time data collection and thereby giving other users an unprecedented platform to download and purchase apps.

New apps are rolling out every day with technical and multifaceted information available in the description; and ordered in terms of latest reviews, ratings, download strategy[14]. This helps in the qualitative and quantitative analysis of users' viewpoint for sizing and pricing strategy, technical claims, and features of apps. This intensifies a fierce competition amongst apps with similar features. We need to efficiently analyze the technical and users' aspect of the app market because sometimes users' may intentionally or unintentionally leave a review that might be false regarding the technical claims made. Natural Language Processing(NLP), a buzzword in recent researches, mines the technical information from reviews. It is one of the trending applications of Artificial Intelligence(AI) It comprehends the features in an interesting manner. Many pioneer researchers are exploiting algorithmic approaches to understand the relationship between the claimed features[15].

Nevertheless, the observations from the entire app market may prove to be robust. However, the problem with the app market is its abundance of reviews that takes extra efforts and longer time in manual computations. One of the bottleneck is information overload problem and its noisy nature[16]. Secondly, the quality of reviews vary tremendously from essential and innovative advice to offensive comments. Thirdly, filtering the negative and positive comments in the reviews and extracting feedback from them are sometimes tricky. Also, the unstructured nature of reviews is troublesome to parse and analyze. Furthermore, it is not always possible to devise a rational set of linguistic measure for mining opinions from characterized languages used in a dialect continuum. For example, app users might use languages that is often explicated as a set of diversities that are mutually intelligible. Apparently, this study only focuses on English reviews given by users and not on multilingual sentiment analysis or resource poor languages. To address the aforesaid problems several works are proposed in literature in subsequent section.

Sentiment analysis help to mine the people's opinions, sentiments, behaviors, emotions, appraisals and attitudes towards products or services, issues or events, topics[17]. There are three types of people's opinions namely, positive, negative and neutral which identify the entire knowledge of the domain. It is an integral part of the NLP and enables

text mining and information retrieval[18]. In field of education, sentiment analysis refines the international education institutions by e-learning techniques[19] and perceptions[20]. It is a computative field utilized to gather a structured and comprehensive knowledge from indistinguishable contents, hence fostering decision support systems. In recent years, it has extended to fields like marketing, finance, political science, communications, health science, education using a coherent framework[21]. If one wants to buy a consumer product[22] or apps, one is no longer limited to asking friends and family for opinions because there are many user reviews and discussions about the product or apps in the comments section, resulting in information repository on Web. We can extract opinions using sentiment analysis tools, process the results and come to valuable conclusions[2]. The resulting model from this study sets a new state-of-the-art to focus only a bunch of university students and crawl their reviews regarding the play store apps they use and using NLP to introspect the sentiment associated with it[23].

Machine Learning based techniques[24] as well as lexicon based methods are used in sentiment analysis[25]. Lexicon based approach is an approach that considers the semantic order of the words and doesn't include labelled data. Dictionary is created manually and includes words and phrases in a document[26]. Sentiment orientation of text corpus are easily pointed out by this approach, by assigning a positive and negative values to texts. The words used are not associated to a generic topic. Goal of sentiment analysis through machine learning approach deals with labelled data and helps to create model using supervised learning algorithms namely, Naïve Bayes(NB), Support Vector Machine(SVM), and K-nearest neighbor(KNN)[27]. The related works of sentiment analysis figured out that the predictive performance was enhanced using classification algorithms. Also ensemble learning methods played a vital role as it combines the predictions from different algorithms. Deep learning paradigm[28], an interdisciplinary of machine learning algorithms, based on fine-tuned layers, has outperformed major classification algorithms[29]. It has yielded fruitful results in speech recognition, computer vision and sentiment analysis. Researchers and practitioners of other specializations are fascinated by its acclaimed reputation. When used with word embedding, it scales well with fine grained opinions and tunes itself with the hyper parameters. Longer sentences are mined easily and used for features extractions[30].

In our study, we had collected 10,841 Google app reviews with 13 fields to train our model[31]. While for the sake of testing of our model, we collected 400 reviews with 6 fields from amongst the Utkal university students via local survey, department wise. This in turn, can be used as a measure for sentiment analysis and understanding local trends of the app market by other students. In addition, university student reviews can be utilized in the administrative related decisions. Real time data collection from students, enhanced disclosures and self-motivation. The evaluation platform provided a basis for pronounced privilege of statements amongst students. Specifically, this paper presents the correlation between Students App Reviews(SAR) and the Google app reviews via an exploratory analysis and visualization of sentiment polarity, subjectivity versus other features like price, installs, type, size, category, ratings.

Towards this end, we initiate a methodical approach to mine opinions from Google app reviews and hence the contribution of our paper includes:

- Several Research Questions(RQ) were designed and evaluated on the corpus through visualization using charts and make intuitive judgements.
- Use of multiple machine learning algorithms and deep learning algorithms and comparing them for sentiment analysis on Google reviews dataset.
- The text representation scheme namely, TP(Term Presence), TF(Term Frequency) and TF-IDF(Term Frequency-Inverse Document Frequency) were implemented on uni-gram, bi-gram and tri-gram strategy.

- The supervised machine learning methods(such as, NB, SVM, logistic regression(LR), KNN, and Random Forest(RF)) were implemented on the text representation scheme and compared amongst each other with respect to its performance metrics.
- The ensemble learning method(namely, bagging) was used with the classification algorithm namely, LR and NB and its performance was evaluated on text representation scheme.
- Fine-tuned Deep leaning models like Long Short Term Memory(LSTM), Convolution Neural Network(CNN), Recurrent Neural Network(RNN) were implemented layer by layer on word embedding(GloVe) and its performance was noted.
- Inferences were drawn from the performances of algorithms graphically.

The organization of this paper comprises of five sections. The study begins with a brief introduction presented in Section 1. Section 2 throws light on the related works in sentiment analysis. Section 3 describes the methods utilized in the paper. Section 4 highlights the empirical analysis, results. Finally, Section 5 wraps up with conclusions of the study and future scope in this context.

2 BACKGROUND

The main realm of text mining, to discover admissible information or knowledge that is possibly unknown, hidden or non-trivial in the terms of other information, is accomplished by employing NLP. It been one of the thrust areas to tag the machine learning and deep learning approaches and also the statistical methods to implement a predictive model. This section briefly describes the wonderful results obtained by researchers of different fields.

2.1 Related Works in Sentiment Analysis using Machine Learning approaches

(1) Lima et al.[32] have used majority voting scheme on the twitter dataset. They have combined the machine learning based paradigms and lexicon based methods. In their work, the tweets are a part of the labelled training data only when it consists of 5% of words or emoticons otherwise it is considered a part of test data. Novak et al.[33] have explained us about emoji based sentiment analysis and the 750 frequently used emojis were also analyzed in the twitter dataset. Lately, Onan et al.[34] have collected instructors reviews from students for opinion mining using machine learning and deep learning paradigm. A comparison between different machine learning and deep learning algorithms were made and the inference was GloVe with Recurrent Neural Network - Attention Mechanism(RNN-AM) algorithm has outperformed others.

(2) While, Adekitan and Noma-Osaghae[35] have predicted about the performance of the university students using machine learning algorithms in their work. Linear and quadratic regression models were used for validation. Almasri et al.[36] have predicted the performance of students using ensemble tree-based model. While Adinolfi et al.[5] evaluated student satisfaction on different learning e-platforms of online courses using sentiment analysis.

(3) Recently, Jena[37] have used machine learning algorithms(namely, NB, SVM, entropy classifiers) along with conventional text representation schemes(namely, uni-gram, bi-gram, tri-gram) on students data for obtaining the sentiment polarity

(4) Farhan et al.[38] proposed a research paper to mine opinions from twitter data. The performed pre-processing of reviews for sentiment analysis. These include slang and abbreviation identification, correcting spellings, removing stop words, tokenization, stemming and lemmatization. Emoticon identification was also done. They used lexicon based approaches. SentiWordNet was used. Misclassifications of tweets was also handled efficiently

(5) Harman et al[39] proposed that App Store Analysis can be used to understand the relation between user, technical, market and social aspects of app stores. They extended their study to non free app in the Blackberry app market. Also find the correlation between the claimed features, ratings, price, size, downloads.Feature extraction was done from the app descriptions.
(6) Mcllroy et al.[40] have studied the updates strategy of mobile apps in Google playstore apps. They inferred that 1% of apps are updated weekly while 14% apps are updated very often. Ranking of frequently updated apps are done on basis of frequency. New updates are not highlighted in 45% of the frequently updated apps.
(7) A research by Taba et al.[41] showed that users preferred the apps and gave a high ratings to those apps that have simpler user interfaces. They studied 1292 free Android apps across 8 categories.
(8) Huang et al.[42] researched that qualitative and quantitative influence of reviews given by reviewers based on the degree of helpfulness. It included the users' experience and impact towards the domain and revealed its effectiveness.
(9) Gao et al.[43] studied the pattern of the reviews given by reviewers, to understand the consistency of ratings over time towards the apps. The inferred users ratings and wrt their behaviour towards the apps or products are consistent over time.

2.2 Related Works in Sentiment Analysis using Deep Learning approaches

(1) Deep learning techniques have been used in for opinion mining and emotion recognition and used for educational tasks employing data mining. In a study, Bustillos et al[44] examined supervised algorithms and LSTM and CNN algorithms showing accuracy of 88.26%.
(2) In a similar fashion , Cabada et al[45] showed deep learning architectures for sentiment analysis based on educational system. The emphasis was on CNN architecture and LSTM attained an accuracy of 84.32%.
(3) A comparative survey between machine learning and deep learning algorithms was presented by Sultana et al[46] on educational data. In their study they showed the highest predictive performances was claimed by SVMs and multi layer perceptron with accuracy of 78.75% and 78.33%, respectively.
(4) In another instance, Nguyen et al[47] emphasized on Vietnamese students' reviews using machine learning and deep learning techniques. NB was used; LSTM and bidirectional LSTM were included in empirical analysis. Unigram and bigram feature were calculated for the corpus while word2vec word embedding scheme was calculated on deep learning algorithms. They got a crystal clear inference that deep learning-based architectures yield higher predictive performance in comparison with conventional machine learning classifiers. Bidirectional LSTM showed an accuracy of 89.3%
(5) Zhou et al[48] figured out sentiment analysis of movie reviews using Stanford Sentiment Treebank(SST) corpus by employing deep learning techniques. They inferred that CNN and LSTM outperformed CNN and RNN models. Positive negative(2-class) reviews achieved 87.7% accuracy while the 5-class reviews(very positive, positive, neutral, negative, very negative) attained 49.2% accuracy. When GloVe was used on corpus accuracy had risen up to 88%.
(6) Zhang et al.[49] proposed a sentence level neural model approach to overcome the weakness of pooling functions that don't uncover tweet- level semantics. They used two gated neural networks, namely a bi-directional gated neural network and three-way gated neural network to model the interaction between target text and surrounding contexts. The bias of RNN is also reduced. Moreover, words were connected in the tweets so as to apply pooling functions over the hidden layers of texts.

(7) Lei et al.[50] implemented neural network approach to extract pieces of texts as reasons for review ratings, by using generator and decoder. Generator maps the distribution of extracted texts an decoder maps those texts to target-specific vectors. A response is generated in context with the target vector based on responses or ratings.
(8) Kandhro et al.[52] used LSTM for analysis of the sentiments expressed by students through reviews for their teachers. The corpus used for this study was built through student's feedback and then divided into 70% and 30% for training and testing purpose.
(9) Chen et al.[51] researched the user and product related information using classification techniques at the word and document level for sentiment analysis. Deep learning was used by Dou et al[53] for the aforesaid analysis. Implementation is done in two fold manner, using LSTM at document level and also deep memory network for predictions.
(10) Sarcasm analysis was also researched by Zhang et al.[54] using deep learning models of tweets. They used bidirectional Gated Recurrent Unit(GRU) model to understand the sentiment of tweets. a pooling network was implemented, for features extraction. Also Joshi et al.[55] found out features from twitter dataset using word embeddings with outstanding reults. Moreover, Poria et al.investigated a CNN based model for sarcasm detection in tweets. They modelled features and emotions associated with the tweets.
(11) Dahou et al[57] performed a research in resource poor language and multilingual sentiment analysis using word embedding and a CNN-based model for some Arabic text sentiment analysis of sentences. Joshi et al.[58] used LSTM for the sentiment analysis of Hindi - English texts corpus.
(12) Multimodal data for sentiment analysis was researched by Bertero et al.[59] describes emotion and sentiment analysis of data from dialogues systems. While Wang et al.[60] studied the visual sentiment analysis using CNN network namely Deep Couple Adjective and Noun(DCAN). These model figure out the adjective and noun descriptions in texts to learn the intermediate representations. Zhu et al.[61] used CNN-RNN model to extract different features for visual emotion recognition using the concepts of multiple layers.
(13) Tang et al.[62] examined the twitter dataset for sentiment analysis in a two-way manner, all the linguistic features were examined and also word embeddings were implemented on the corpus.
(14) Hu et al.[63] mined opinions from reviews of products, movies and hotels using deep learning models.Linguistic features were extracted efficiently.
(15) Severyn et al.[64] used CNN models to mine opinions from messages and phrases used.
(16) Glorot et al.[30] and Pang et al.[65] studied sentiment analysis using deep learning models tagged with auto encoders and rectifier units.

2.3 Observations from Literature Survey

There are several prospects which have been surveyed in context with Google app reviews. A comprehensive literature survey is highlighted in preceding Sections 2.1 and 2.2. The use of conventional text representation schemes with machine learning algorithms and also deep learning algorithms have drawn research attention, lately. However, according to our study there are very limited works based on predictive performances of algorithms using Google app reviews in conjugation with university students reviews related to Google apps. Apparently, there are no past reports on NLP in sentiment analysis of user reviews regarding the Google apps in conjugation with SAR. Precisely, the state-of-the-art methods does not capture any of the comparisons between the machine learning algorithms and deep learning algorithms. And to the best of our knowledge, the combination of techniques used in our study is a bit unique.

Table- 1 captures a comparison of existing literature in context with Sentiment analysis based on different reviews crawled in different languages and domains.

For bridging this gap, our literature survey was inspired by instructors review paper approach[34] that threw light on multiple combinations of machine and deep learning algorithms. The latest trend observed from our work is that it does not emphasise on count vectorizer method of splitting the data set rather aggregates a new university data set. We incorporated data analysis along with modeling. Moreover, basic research questions, in context with domain of Google app reviews and SAR were answered via charts.

Table 1. Comparison between Existing Literature of NLP for Instructor Reviews [34]

Reference	Methods	Accuracy
Sultana et al[46]	Multilayer perceptron	78.33
Sultana et al[46]	Support vector machines	78.75
Nguyen et al[47]	Unigram features + Naive Bayes	85.30
Nguyen et al[47]	Bigram features + Naive Bayes	87.50
Nguyen et al[47]	word2vec + LSTM	87.60
Nguyen et al[47]	word2vec + Bi-LSTM	92.00
Kandhro et al[52]	word2vec + LSTM	89.00
Bustillos et al[44]	Bernoulli Naive Bayes	76.77
Bustillos et al[44]	CNN + LSTM	88.26
Cabada et al[45]	Multilayer perceptron	90.42
Cabada et al[45]	CNN	92.46
Cabada et al[45]	LSTM	90.92
Cabada et al[45]	CNN + LSTM	92.15
Onan et al[34]	Glove + RNN-AM	98.29

Table 2. Sample students' reviews and Sentiment Characteristics from Students' dataset.

Students' review	App	Orientation	Polarity	Subjectivity
It's helpful to learn at home.Highly recommendable	Unacademy	Positive	0.04	0.135
It's amazing and works well.	PhonePay	Positive	0.3	0.725
Horrible. Keeps crashing my phone.	Subway Surfers	Negative	-0.104	0.43
It' annoying due to adds.	JioSaavn	Negative	-0.033	0.388
Very well designed. Many updates present.	WPS Office	Positive	1	0.75

Table 3. Assessing the Research Questions.

Serial	Research Question	Figure	Answer
RQ1	Do the apps which get a higher rating in the training dataset tend to be more popular among the students as well?	Fig.6a, 6b.	The Google app market breakdown showed prominent downloads in Social and Games categories. On the contrary, Weather and Comics were of least interest among students. The average ratings shooted up to 4.17 across major categories. Interestingly, Shopping, Food and Drinks, News and Magazine are also catching up. Expensive apps may make students disappointed, if they are not good enough and consequently get low ratings. Students from Mathematics and Sanskrit department participated fairly well while Women Studies and Geography showed least participation. Other departments showed an acceptable participation.
RQ2	Do the priced and free apps get the same ratings and popularity from the students as compared to the training dataset?	Fig. 6c, 6d.	This jointplot visualization depicts the sizing strategy(small vs huge). We got a clear conclusion that, small sized app(0-60 Mb) and free apps(81.7%) are predominant for downloads among students. This enhances the ratings. Average rating turning out to be 4-5. On the contrary, larger, paid apps have least ratings and less preferable.
RQ3	What is the correlation between price, rating, popularity amongst the university students when compared with the training dataset?	Fig.8.	The installs and reviews are positively correlated amongst students. While, installs and pricing are negatively correlated.
RQ4	What is the sentiment polarity could be analyzed by the reviews of students when compared with the training dataset?	Fig. 6e.	The scatter plots are heavily clustered towards positive side rather than on negative. Specifically, we can say students weren't so harsh while give reviews, instead gave genuine and lenient feedback.
RQ5	How size of apps affect the installs amongst the students as compared to the training dataset?	Fig. 6f.	The points in the jointplot are heavily clustered where the price for apps are 0. This gives us an inference that students prefer free apps rather than paid and an average rating between 3.5 to 5 is shown.
RQ6	What confusion matrix can be obtained from students' reviews as compared to the training dataset?	Fig. 9a.	The confusion matrix when applied with LR gave us an accuracy of 90.8%.

Table 3. Assessing the Research Questions.

Serial	Research Question	Figure	Answer
RQ7	What positive words were used by the students as compared to the training dataset?	Fig. 7a.	The bold and highlighted words(good, great, love)were highly used amongst students while smaller and less distinct words(little, much, back) were least used.
RQ8	What negative words were used by the students as compared to the training dataset?	Fig. 7b.	The bold and highlighted words(load, log, work, take-time) were highly used amongst students while smaller and less distinct words(problem, open, login) were least used.
RQ9	What confusion matrix can be obtained from students' reviews as compared to the training dataset when GloVe scheme is used?	Fig. 9b.	The confusion matrix when applied with GloVe gave us highest accuracy of 81.6 on Falsely classified reviews and second highest accuracy of 80.7% on truly classified reviews
RQ10	What ratings are obtained for various app based on Content Rating from students' reviews as compared to the training dataset when GloVe scheme is used?	Fig. 6g.	The free and paid apps showed a predominance in ratings i.e., 4-4.5 amongst students that were accessible to Everyone. Free apps that were accessible to Adult only 18+ showed a rating of 4.5. The free and paid apps under the section Teens, Everyone 10+, Mature 17+ showed an average rating of 3.2-4.8.

Table 4. Accuracy values for Machine Learning Algorithms.

Algorithms	Unigram+ TP	Unigram+ TF	Unigram+ TF-IDF	Bigram+ TP	Bigram+ TF	Bigram+ TF-IDF	Trigram+ TP	Trigram+ TF	Trigram+ TF-IDF
SVM	91.5	92	92.89	93.4	93	93.41	93	93	93.37
KNN	90	91	91.01	91	91.5	90.9	89.5	89	88.39
LR	84.36	84.99	84.08	84.77	84.96	84.61	85	84	84.48
RF	83	84.15	83.42	85.47	84.23	85.11	84.5	85	84.16
NB	78.56	79.25	80	81.2	82.09	82.14	81.27	80	82.21
LR(Bagging)	86.47	86.5	86.5	86.5	86.77	86.5	86.5	87	87.88
NB(Bagging)	85.14	85.69	85.5	85	86.69	85.11	85.68	85	84

3 METHODOLOGY

This section briefly describes the methods used in our study i.e., Corpus used, conventional text representation schemes, word embedding, text representation schemes, deep learning and machine leaning algorithms. Fig.1 explains the proposed methodology of our study which comprises the machine learning based and deep learning based sentiment analysis framework.

Table 5. F-score values of Machine Learning Algorithms.

Algorithms	Unigram+ TP	Unigram+ TF	Unigram+ TF-IDF	Bigram+ TP	Bigram+ TF	Bigram+ TF-IDF	Trigram+ TP	Trigram+ TF	Trigram+ TF-IDF
SVM	0.88	0.88	0.89	0.87	0.87	0.89	0.88	0.88	0.88
KNN	0.85	0.85	0.86	0.85	0.85	0.85	0.85	0.84	0.85
LR	0.7	0.7	0.69	0.66	0.66	0.68	0.71	0.7	0.7
RF	0.68	0.68	0.68	0.63	0.6	0.61	0.62	0.63	0.62
NB	0.7	0.7	0.72	0.62	0.6	0.62	0.64	0.63	0.63
LR(Bagging)	0.85	0.85	0.87	0.87	0.86	0.86	0.84	0.85	0.86
NB(Bagging)	0.77	0.78	0.75	0.75	0.75	0.76	0.77	0.77	0.76

Table 6. Accuracy and F-score values for Deep Learning Algorithms.

Algorithms	Vector Size	Dimension of Projection layer	Accuracy	F-score
LSTM(GloVe)	200	100	95.2	0.88
RNN(GloVe)	200	100	93	0.85
CNN(GloVe)	200	100	92.7	0.78

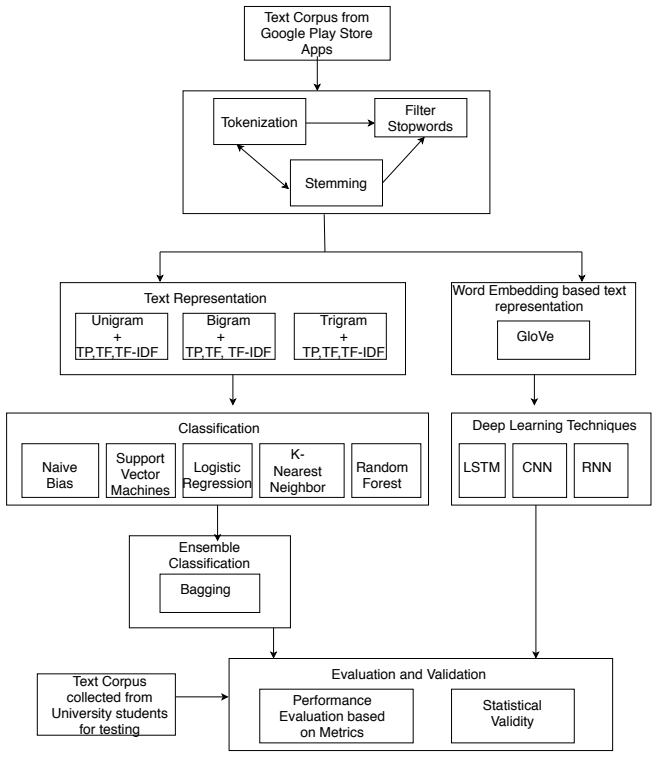


Fig. 1. Architectural Framework for Sentiment Analysis.

3.1 Data Sources

This study had 2 data sets under consideration and are as listed below:

- Training data set, Google app reviews.
- Test data set, SAR.

Google app reviews- The corpus is openly available for research and it was collected in .csv format[31]. There were 9659 apps, 33 categories, 115 genres in the dataset. The columns of the data set are as follows app(name), category(app), rating(app), reviews(user), size(app), installs(app), type(free/paid), price(app), content rating(everyone/teenager/adult), genres(detailed category), last updated(app), current version(app), android version(support).

SAR - In this study, the aim is to understand the trend of Google app market and comparing it with test data i.e., analyzing the students' behavior towards the Google app market. So, we had collected the real-life data from the Utkal university students, department wise. The survey was entirely voluntary in nature and no incentives were offered to perform the survey. If university students did not wish to participate, then they were excluded from the survey. The reviews regarding the frequently used apps, were gathered via a survey for a month. The survey was made on an online platform via a Google form. One student from one department could at max list out seven frequently apps used on their device. They had given their reviews in English. There were 400 data collected from survey with 6 fields including department(name), app(name), reviews(user), ratings(everyone/teenager/adult), type(free/paid), category. Students rated the apps on a 5-point scale where ratings below 3 were labeled as "negative" and that with 3 or greater than 3 is considered "positive".

Furthermore, in the empirical analysis data cleaning, text pre-processing techniques were carried out on the dataset to build efficient learning models and enhance the overall performance. These included:

- removing missing data
- dropping NA values
- removing punctuation, tags, special characters URLs, emojis, digits
- filtering stop words
- Tokenization
- noise removal, spelling correction
- stemming and lemmatization[66]

After cleaning, there were 9,360 and 380 records in the training and test dataset respectively. Table-2 presents some sampled test data reviews along with the sentiment characteristics. Orientation - determines positivity, negativity or neutrality of sentence, Polarity - helps identify the sentiment orientation, Subjectivity - defines person' opinions, emotions or judgment; ranging from 0.0 (objective) to 1.0 (subjective). TextBlob package in Python helps in calculations for sentiment analysis. A sentiment score determines how negative or positive the entire text analyzed is. For eg., the phrase "not a very great app" has a polarity of about -0.3, implies it is slightly negative, and a subjectivity of about 0.6, implies it is fairly subjective.

3.2 Text Representation Schemes

Practical understanding of corpus requires a lot of effort to evaluate the semantics, syntax and structure of sentence. One of the motive to study the text representation schemes(namely, Bag-of-words, TP, TF and TF-IDF) is feature engineering, used especially for NLP applications. Features act as input parameters for the machine learning algorithms to generate some output. Feature engineering is an art and skill generating the best possible features and choosing the

best algorithms for developing NLP applications. Using all the 4 text representation schemes leveraged us to analyze the corpus well and enabled the classifiers to model its performance.

Bag-of-words paradigm[67] is a very commonly used technique to represent all the unique words occurring in the documents. The occurrences of the terms in a document is noted while the order and the sequence of words is not considered. This scheme helps in feature extraction from text documents. The three weighted schemes frequently utilized are based on bag-of-words model i.e., TP, TF and TF-IDF. TP maps the appearance of words in a document, by binary values 1 and 0, indicating its presence and absence respectively. TF counts the number of appearances of words in a document. Commonly used words have a higher count in context with rarely appeared words. TF-IDF scheme is an improvement over TP[68] and uses a normalizing aspect for computations. TF-IDF basically combines two metrics, namely TF and IDF. It helps in ranking the queries in search engines and used widely in information retrieval and text mining. It is used to weight words according to their importance.

Mathematically, TF-IDF is defined as:

$$TF - IDF = TF(w, D) * log(C/df(w)) \tag{1}$$

Here, $TF - IDF(w, D)$ maps the TF-IDF score for a word w in document D. Therefore, it will score higher if the term is not common. TF(w,D) is the frequency of term in a given document(synonymous with bag of words). IDF measures the significance of the word in the corpus of documents. Given a corpus C, the number of documents divided by the frequency of word in the document w. followed by the log transform gives IDF. Highly occurring words across many documents will have a lower weight, and otherwise would have a higher weight.

N-gram model is a collection of words from a text document in which the the words are contiguous and occur sequentially. They may be in the form of phrases or group of words. In n-gram model:

- when n is one (order is one) i.e., it consist of one word, therefore it is termed as an uni-gram model.
- Similarly, bi-gram model indicates n is two (order is two) i.e, it consists of two words.
- Tri-grams indicates n is three (order is three) i.e., it consists of three words and so on. The n-gram model is a supplement of the bag-of-words model.

In this study, we performed an experiment on the Google apps corpus and SAR based on three n-gram model and TP, TF, TF-IDF, and obtained nine different configurations.

3.3 Machine Learning Algorithms

If a machine learning algorithm is trained on a dataset tagged with labels, it is called supervised learning. Labelling basically marks the output on the input parameters. We trained our model on different N-gram models(i.e., uni-gram, bi-gram, and tri-gram models) and TP, TF, TF-IDF based weighting scheme using google apps reviews. As a result, nine different feature sets were obtained. In the next subsections, we briefly describe the details of the five most frequently used classifiers in sentiment analysis.

- **LR** [69] helps to solve a classification problem by analyzing a dataset where the outcome depends on one or more independent features. It is a linear algorithm and the underlying technique is quite similar to Linear Regression. The term "Logistic" is taken from the Logit function that is used in classification. The idea is to come up the model that best describes the relationship between the outcome and a set of independent variables. The dependent variable is binary , i.e., it only contains data coded as 1 (TRUE, success) or 0 (FALSE, failure). However, this

classifier has captured a lot of work in NLP according to our literature survey, namely in teaching evaluation review[70], students' performance[35].

- **SVM**[71] is a supervised machine learning algorithm. It helps to solve classification and regression problems. In SVM, the data points are plotted in N-dimensional space where N denotes number of features and a hyper-plane is found to differentiate the data points. However, this algorithm is computationally expensive but is used when the number of dimensions is high with respect to the number of data points. For instance researchers[37] have worked with SVM classifiers and shown remarkable results in NLP. Teacher evaluation review also used SVM[70]. This indeed motivated us for using SVM on our corpus too.
- **NB**[72] is a probabilistic classifier which uses Bayes theorem. The object with similar features are grouped in one class while others in a different class based on certain probability. In this method, there is a strong independence assumptions between the features. It requires a small training data for classification, and all terms can be pre-computed thus, classifying becomes easy, quick and efficient. For instance, teacher evaluation review also used NB[70], students' performance[35].
- **KNN**[73] solves the classification problem by assigning the object to a class by a plurality vote from its k(positive integer) neighbors. While in regression, the output value for an object is the aggregate values of k nearest neighbors. KNN captures the idea of similarity amongst the object with respect to its neighbors in terms of distance, proximity, or closeness. KNN as such did not capture much compromising results in NLP, but used in instructors' review[34].
- **RF**[74] widely uses bagging, random subspace methods, ensemble learning paradigms. For the purpose of classification decision trees are used. It decomposes the training dataset based on certain conditions on attribute values using random sampling. The data is divided recursively until the leaf node is left with minimum amount of records using random subset features. For instance, ensemble tree based model[36] was efficiently modeled and showed remarkable results in NLP on students performance dataset. Another instance was captured in students' performance[35].

3.4 Ensemble Learning Methods

The base estimators are built on a given learning algorithm and their predictions can be combined to improve the robustness and performance over single estimator[75]. It includes averaging and boosting methods. In averaging methods, several estimators are built independently and their predictions are averaged. The combined estimator performs better due to the reduced variance. Examples Bagging method, Forests of randomized trees. By contrast, boosting methods[76], reduces the bias of the combined estimator by building the base estimators sequentially. Several weak models can be combined to generate a powerful ensemble. Examples AdaBoost, Gradient Tree Boosting. In this study, we have considered Bagging method and the rest of the section describes it.

Bagging It is also called as Bootstrapped Aggregation[77]. It is used for predictive modeling (CART). Random subsets of data are drawn from the training dataset with replacement, and a final model is produced by averaging result from several models. One popular way of building Bagging models is by combining several DecisionTrees with reduced bias that increases the model's prediction than individual Decision Trees. Averaging ensembles with bagging techniques like RandomForestClassifier and ExtraTreesClassifier reduces the variance, avoids over-fitting and increases the model's robustness with respect to small changes in the data.

3.5 Word Embedding

One of the importance of word embedding is to convert the text into vector representation(numerical format) using some statistics as computers can't understand natural language directly. Python with its huge libraries(Keras, TensorFlow) is capable of dealing with the vectorization using some mathematical concepts(matrix and vector form). These word embedding applied using deep learning improvises the machine translation, enabling language modelling. Examples are SpaCy(word2vec), GloVe, Pointcare, Numberbatch, Elmo (Language Model), Flair, FastText. Word embeddings are dense vectors representations with lower dimensionality and overcomes word ambiguities[78]. It provides an improvement over the simplest bag-of-words model, widely used in NLP. Bag-of-word faces two problems:

- Firstly, the word frequency count results in sparse vectors that describe the document but not the meaning of words causing performance issues.
- Secondly, it ignores the words semantics in text document.

We have trained a simple neural network with a single hidden layer to get a more expressive and compact representation for text documents. The goal is to extract the semantic and syntactic meaning among the words with the help of unsupervised texts. It can handle a very large corpus. From our study we can infer word embeddings outperforms conventional text representation schemes in NLP. The next section highlights GloVe i.e., used in this study.

GloVe One of the necessity to study GloVe is to focus on the distributional semantics(develop theories that quantify and categorize semantic similarities between linguistic items based on the distributional properties in large samples). In NLP, developing tools to deal with semantics of words, phrases, sentences is a big deal, and word2vec, GloVe does a great job to figure out the contextually similar words in a vector space(context builder and vocabulary builder). However, GloVe overcomes the shortcomings of one hot vector encoding, FastText and word2vec. For instance, in the research of Onan et al.[34] using instructors reviews from students for text analysis, GloVe was used with all deep learning paradigm(LSTM, RNN, CNN, RNN-AM, GRU). Upon comparison, GloVe with RNN - AM algorithm has outperformed others, this urged us to handpick Glove as one of the architectural component in our study.

It[79] is another kind of word representation from the Stanford NLP Group. It is an unsupervised learning. GloVe is basically an extension of word2vec model. In the vector representation of words similar words are placed together. Both GloVe and word2vec trains a word embedding and provides the same core output: a vector per word. The difference between the two is word2Vec does incremental, 'sparse' training of a neural network, by repeatedly iterating over a training corpus. While GloVe uses an objective function to train word vectors from the global word to word co-occurrence matrix. It goes through the entire corpus and constructing a co-occurrence matrix. While Word2vec learns how to represent words by trying to predict context words given a center word (or vice versa), GloVe learns by looking at each pair of words in the corpus that might co-occur. Pre-trained GloVe vectors are easily loaded and used from Gensim on a large corpus.

The objective function of the model is given by:

$$J(\theta) = \frac{1}{2} \sum_{i,j=1}^{W} f(P_{ij})(u_i^T v_j - logP_{ij})^2 \tag{2}$$

where $J(\theta)$ is the objective function, given (θ). $u_i^T v_j$ is the dot product of the vectors $u_i and v_j$.P_{ij} is count of co occurrence of i and j.

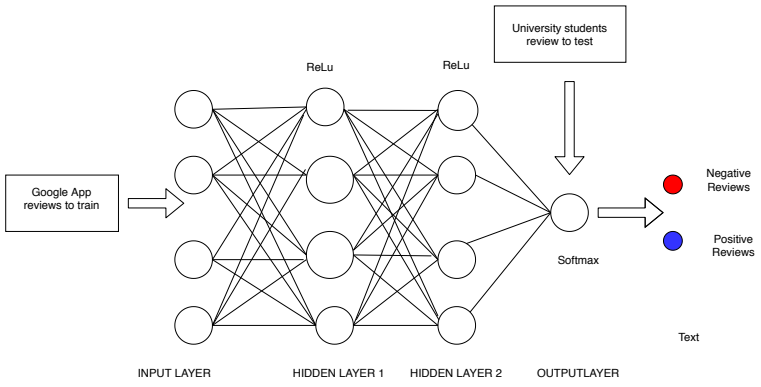


Fig. 2. MLP Architecture.

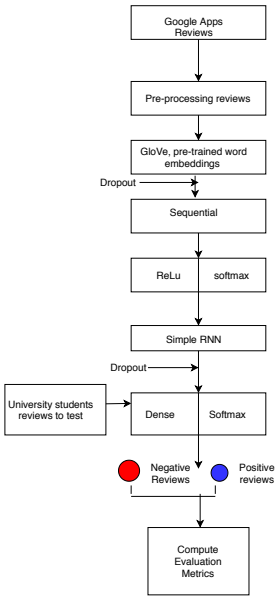


Fig. 3. RNN Architecture.

3.6 Deep Learning

Deep learning, a specialism of machine learning employs adaptation of neural networks. It doesn't need to be explicitly trained, unlike machine learning algorithms which uses feature extraction[80]. It has extended to many applications from computer vision , speech recognition and now produced a state-of-art in NLP too. Deep learning is widely used due to tremendous increase in dataset. Also, researches in the field of machine leaning and data science has shown significant enhancement. In short, deep learning paradigms uses multiple layers of non linear processing units cascaded together to extract features and transform the dataset. Lower layers of the architecture are close to input data and learn simple features while higher layers get data from lower layers and learn complex features thereby picturing a hierarchical scheme. The rest of the section describes the architectures used in our study:

3.6.1 ***Multilayer Perceptron***. One of the fascinating fields of artificial neural network is called multi-layer perceptron (MLP)[81]. The main concern of MLP is to solve robust algorithms and data structures to overcome complex problems. Perceptron is a single neuron model that was a pioneer to complex neural networks.

Neural network representation helps us to relate the training data and also predict the output variables through any of the mapping functions. Furthermore, the predictive capability of MLP comes from the hierarchical structures by combining high order features. Neurons are the basic building blocks of artificial neural network and have associated neuron weights and activation function. The exposed part of the network is called input layers that accepts the dataset, hidden layers are arranged thereafter and are not much exposed. The final hidden layer is the output layer and it is useful for giving a value or vector of values that are in accordance with the computed problem. The choice of activation function in the output layer is strongly constrained by the type of problem that you are modeling. For example: Deep learning are tagged with many hidden layers in our neural network. Fig.2 illustrates MLP architecture.

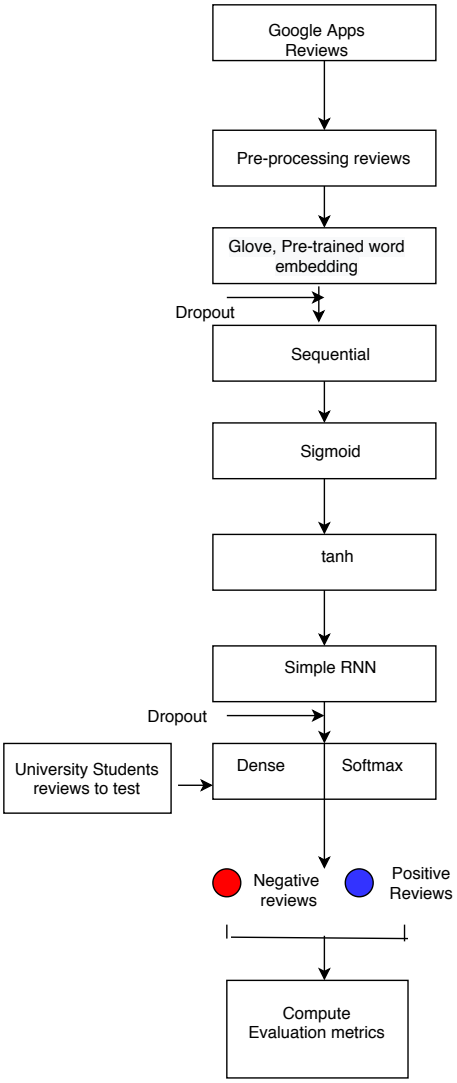


Fig. 4. LSTM Architecture.

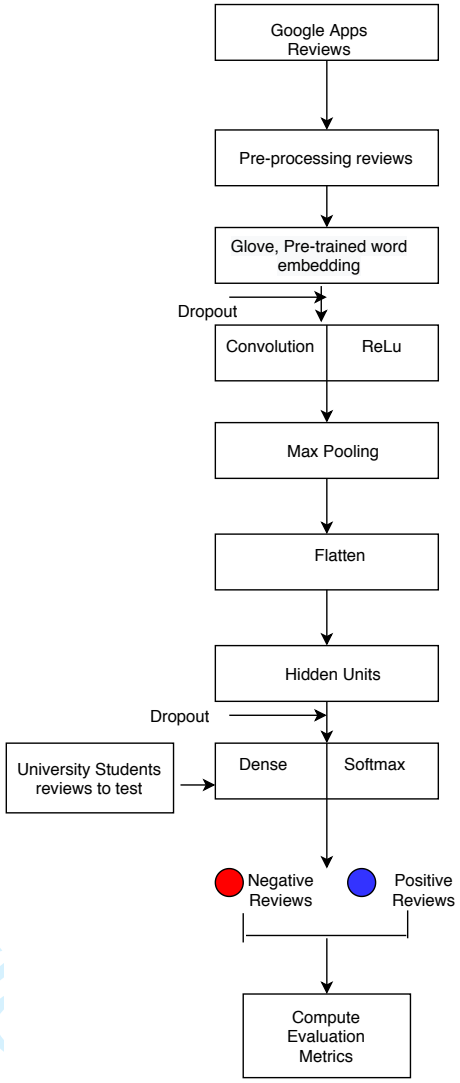


Fig. 5. CNN Architecture.

3.6.2 ***RNN***. It is short for Recurrent Neural Network processes sequential data[82]. All the neurons are connected in a graphical form, resulting in a directed graph. Speech recognition in dealt by RNN as it emphasises sequential data[83]. The task are recurrently processed over the sequences and output can be found out based on previous computations made. The length of input sequences determines the length of time steps also. The equation for RNN is as given below:

$$s_t = f(Ux_t + W_S t - 1) \tag{3}$$

Here, f maps the activation function used and it is namely, ReLu or tanh. U and V denotes the weights assigned. RNN encounters vanishing gradient problem and exploding gradient problem. Its cannot deal with long sequences of input. The shortcomings of RNN is easily dealt with, by using LSTMs or GRU and bidirectional RNN.

The architecture of RNN is depicted in Fig.3

3.6.3 ***LSTM***. It is a modification over RNN. It is short for Long Short Term Memory networks[84]. Long-term dependencies are easily handled by LSTMs and they can overcome vanishing gradient problems also. The core idea of LSTMs is to remember the information stored for long periods. They have feedback connections too. The architecture of LSTM includes an input gate, forget gate and output gate. The flow of information in LSTM is done through cell states, by simple additions and multiplications. LSTM networks are applicable for classify, process and to predict the outcomes of time series dataset. Hence, back propagation of errors is allowed in fixed number of iterations A typical LSTM models the opening and closing of gates as in when the preservation of information is needed and when the access of information is needed via gates. The architecture of LSTM is depicted in Fig. 4 In our study LSTM transitions includes the given equations below:

$$i_t = \sigma(W^i x_t + U^i h_{t-1} + b^i) \tag{4}$$

$$f_t = \sigma(W^f x_t + U^f h_{t-1} + b^f) \tag{5}$$

$$o_t = \sigma(W^o x_t + U^o h_{t-1} + b^o) \tag{6}$$

$$u_t = \sigma(W^u x_t + U^u h_{t-1} + b^u) \tag{7}$$

$$c_t = i_t^o u_t + f_t^o c_{t-1} \tag{8}$$

$$h_t = o_t^o \tanh(c_t) \tag{9}$$

3.6.4 ***CNN***. It is short for Convolution Neural Network, that uses grid based topology[85]. Instead of general matrix multiplication, CNN uses convolution in one or more layers of model. Input layer, output layer and hidden layers are included in CNN architecture. Hidden layers of CNN architecture substitute other layers namely, convolution layers, fully connected layers, normalization layers and pooling layers. Convolution operation is employed on input data. Activation function like ReLu, is used and adds non linearity to architecture. Pooling layers combine output from neurons and control the feature size space[86]. As a result model's over-fitting can be controlled. Maximum value from each cluster is taken and after convolution final output represents the fully connected layers. The architecture of CNN is depicted in Fig.5

3.6.5 ***Activation Functions***. It helps in model evaluation and optimization. By improving the model's hyper-parameters we can improve its performance. These functions that are a part of activation nodes found in the hidden layers of neural networks. In order to introduce non-linearity to the neural network we add activation function, otherwise it would be similar to linear regression. This in turn defeats the purpose of neural networks because they wouldn't be able to learn complex functional relationship that exists within the data. Activation function need to be differentiable to promote backpropagation. Activation function converts the input signal to an output signal. Examples of activation

functions are sigmoid, tanh, softmax, ReLU (Rectified Linear Unit). An activation node calculates the weighted sum of inputs it receives, adding the bias and applies an activation function to this value. An output is generated for that particular activation node, it being used as an input by the proceeding layers. The output is so called as activation value. The proceeding activation node in the next layer will receive multiple activation values from preceding nodes and a new weighted sum is found out. And an activation function is applied to this node. This is how data flows in neural network. The activation function used in our study are described in the subsequent section:

- **Sigmoid** - It is a S- shaped curve. It ranges between 0-1. It is basically used in the cases when we need to predict the output. It is not zero centric. It is used in binary classification problems. The representation of sigmoid is given by:
$$f(x) = \frac{1}{1 + e^{-x}} \tag{10}$$
It being differentiable we can easily find the slopes between two points(smooth gradient). It is more efficient with logistic functions tagged with multi class problem.
- **Softmax** - It is mainly used in neural networks which has multi-class-classifier and the constraints are assigned to more than one classes. It helps to transform the vector scores to class probabilities. It is given by:
$$f(x) = \frac{e^{x_i}}{\sum_{j=0}^{k} e^{x_j}} \tag{11}$$
In our study we have used softmax in the nth layer of the neural network (i.e., the output layer of network). This is because the last layer give some scores that is interpreted by humans. It rather normalizes the scores by finding exponent and dividing by a normalized constant.
- **ReLu** - It is Rectified Linear units. It is used in neural network and has an advantage over tanh i.e., it replaces all the negative values with 0, implying they are mainly sparse. They are denoted by:
$$f(x) = max(0, x) \tag{12}$$
It suffers less from vanishing gradient problem and used mostly in CNN. Leaky ReLUs, Parametric ReLU (PReLU) are the variations of ReLu.
- **tanh** - It is hyperbolic tangent. It is a non linear function and is zero centered. It squashes the output between -1 to 1. The formula is given by:
$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{13}$$
The effect they have on nodes is continuously evaluated. One can use tanh to change how one layer influences the next layer in the chain. They resemble the sigmoid like pattern and penalizes the extreme node values repeatedly causing vanishing gradient problem. Tanh is a good choice, but is generally avoided due to its cost.

3.6.6 ***Dropout***. Machine learning and deep learning model encounters a serious problem of over fitting. This tendency occurs when model performs exceptionally well on training dataset but unable to generalize the model with test dataset. One of the ways to avoid over fitting is to use regularization. Regularization constraints the parameters to 0. Dropout is a common regularization technique, which randomly drops the neurons in forward and backward passes. The probability of dropping a neuron is specified as a parameters. By randomly dropping the neurons ensures the model's flexibility and help us generalize the model better.

3.7 Result Evaluation Metrics

This section briefly discusses about the metrics used in this study for the result computation.

Precision: It measures correctness of a classifier. It is the proportion of number of precisely extracted opinions to the total number of extracted opinions.

$$precision = \frac{TP}{TP + FP} \tag{14}$$

Recall: It measures the sensitivity of a classifier. It is defined as the proportion of number of precisely extracted opinions to the total number of annotated opinions.

$$recall = \frac{TP}{TP + FN} \tag{15}$$

F-Measure : It is a commonly used measure. Precision and recall are combined to give a single criterion called as F-measure. The harmonic mean of precision and recall computes F measure. It rates a system with one unique rating.

$$F - measure = \frac{2 * precision * recall}{precision + recall} \tag{16}$$

Accuracy: It is commonly used performance measure in supervised learning techniques. It is the ratio of truly predicted observation to the total number of observations.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

Here, TP refers to true positive i.e., the positive tuples that were faultlessly characterized by the classifier. TN refers to true negative i.e., the negative tuples that were faultlessly characterized by the classifier. FP refers to false positive i.e., the negative tuples that were inaccurately characterized by the classifier as positive. FN refers to false negative i.e., the positive tuples that were untruly characterized by the classifier as negative.

4 EXPERIMENTS AND RESULTS

The empirical analysis was done to train our model on the Google reviews dataset and tested on SAR. The contribution of paper is three-fold because the experimental demonstration is in sets of three i.e., the visualizations of corpus and evaluating the predictive performance of corpus based on machine learning and deep learning algorithms. The platform used and the results of the empirical analysis are reported in the subsequent sections. Furthermore, the code for the experimental analysis is available at https://github.com/smlab-niser/Google-Reviews-Sentiment-Analysis

4.1 Evaluation Environment

The language used is Python and the platform used in our study was Jupyter Notebook. The basic and vast libraries that we familiarized ourselves with are namely, matplotlib, scikit-learn, TensorFlow, numpy, pandas, seaborn, keras and others. The were extensively used for descriptive and result statistics, statistical tests, plotting functions. These are the open source tools and focuses on reducing the complexity and gives efficient results for modeling our dataset. In this study we have leveraged the power of Google Colab to develop our machine learning and deep learning based sentiment classifier. We can expect a speedup of 10-30 times when compared to running the training session on CPU. It also depends on power of GPU, processing steps and amount of data involved. Google offers upto 12 hours of free GPU usage per day to train models.

The system used is Windows-10 operating system with Intel core i7 8565U, RAM Size 8 Gb, SSD Capacity 512 Gb, processor frequency 1.8 GHz.

4.2 Results based on Visualization

One of the motivation to create RQ's was to analyze the personal skill of students and showcase the fundamental statistics concepts merged with programming skills for corpus analysis. This is because as human beings, people express their thoughts or feelings via language. Whatever we speak, listen or write is in the form of natural language(namely, WhatsApp chat message, movie dialogues etc.). NLP is an intelligent system that uses computational linguistics and technologies to process the natural language like humans. So, we set out to answer the questions in context with basic characteristics of app market like price, category, ratings, genres, size and downloads and look through the prism of statistics. Firstly, an exploratory analysis was initiated on our training dataset i.e., the Google app reviews. The results so obtained, were compared with the SAR through visualizations. The inferences drawn using charts helped us to get a glimpse of students' behaviour towards the distribution of the app market. RQ's were formulated and investigated to understand the correlation between the app market characteristics specifically, price, popularity, sizing, categories, genres and ratings of apps by the students when compared with that of training data set. Table-3 presents the visualizations based on scatter plot, histograms, bag-of-words and other plots .

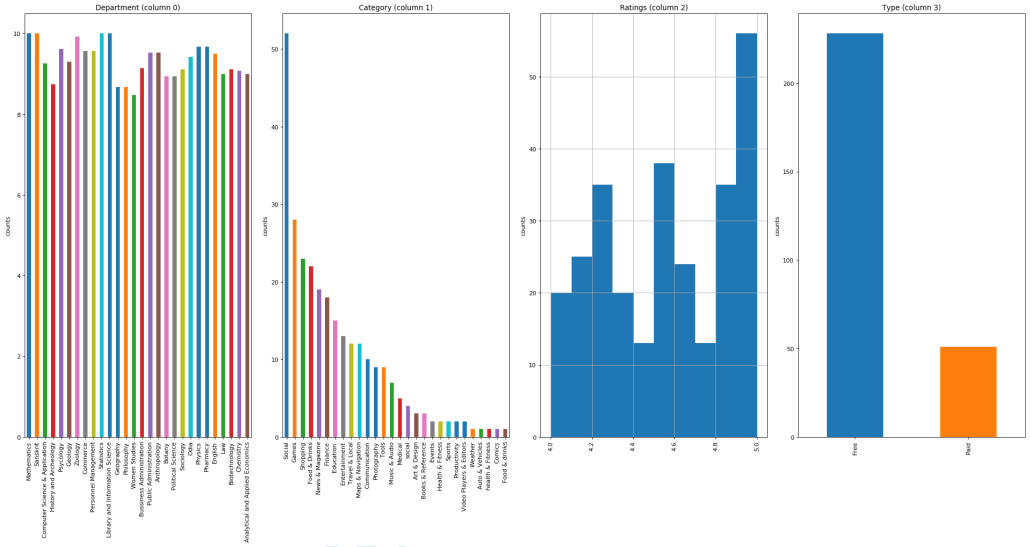
4.3 Results based on Machine Learning

Secondly, we performed several experiments to train our model using the classification algorithms on the conventional text representation schemes. These are briefly described in sections 3.2 and 3.3. The major classification algorithms of our study are NB, LR, KNN, SVM and RF. The conventional text representation schemes include TP, TF, TF-IDF. Ensemble learning method namely Bagging was also used in our empirical analysis. As mentioned in previous sections ensemble learning techniques do enhance the predictive performance of the algorithms. These are emphasized in Section 3.4. We used evaluation metrics namely F-score and accuracy to generate useful intuitions from our corpus. The nine different configurations so obtained, shows a comparison amongst these algorithms. Table-4 outlines the results of Accuracy values obtained from the empirical analysis based on machine learning algorithms. While Table-5 figures out the results of F-score values obtained from the empirical analysis based on machine learning algorithms.

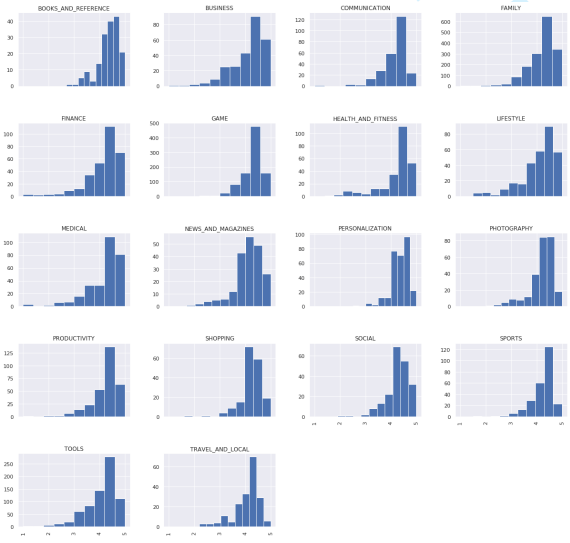
Regarding the performance of classification algorithms on text representation schemes, SVM proved best for our corpus and attained the highest accuracy and F-score value. SVM on bi-gram+TF-IDF got accuracy 93.41% and TF-IDF on bi-gram, tri-gram model got F-score 0.89. NB performed worst on our corpus and didn't turn out to be fit for out study. The least accuracy 78.56% and F-score 0.60 was achieved. The second best algorithm of our study is KNN and is catching up with SVM. TF on bi-gram got highest accuracy of 91.5% and F-score of 0.85 was constant throughout. LR and RF performed averagely on our corpus in terms of F-score and accuracy. LR on uni-gram and TF scheme resulted in accuracy of 84.99%, F-score of 0.70. RF on bi-gram+TP captured highest accuracy 85.47%., F-score 0.68 on uni-gram models for all TP, TF, TF-IDF schemes. Bagging was applied on LR and NB and we got comprehensible results showing an enhancement in accuracy and F-score.

4.4 Results based on Deep Learning

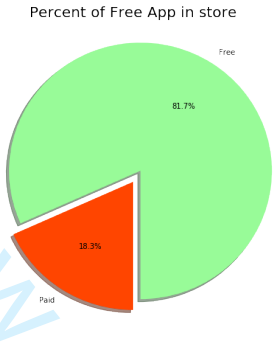
Furthermore, we performed experiments to train our model using the deep learning algorithms also. We used word embedding, namely GloVe along with the algorithms, briefly described in Section 3.5. The Google app reviews were evaluated using the major deep learning algorithms namely, LSTM, RNN and CNN, briefly described in Section 3.6. We used evaluation metrics namely F-score and accuracy to generate useful intuitions based on the algorithms. The batch size and hyper parameters were thoroughly investigated during the experiments. Vector size used in our experiment



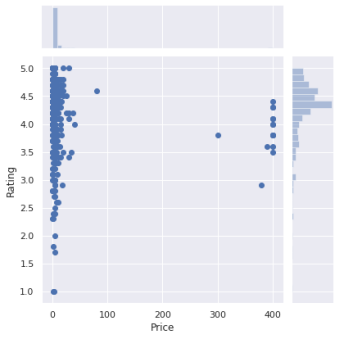
(a) Counts over Category, Ratings, Price.



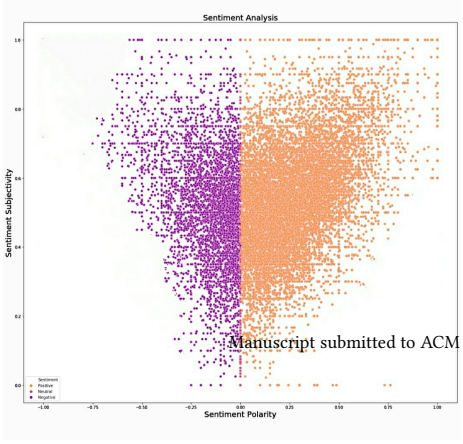
(b) Ratings over Category.



(c) Paid vs Free apps.

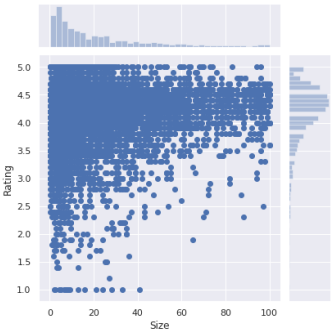


(d) Price over Ratings.

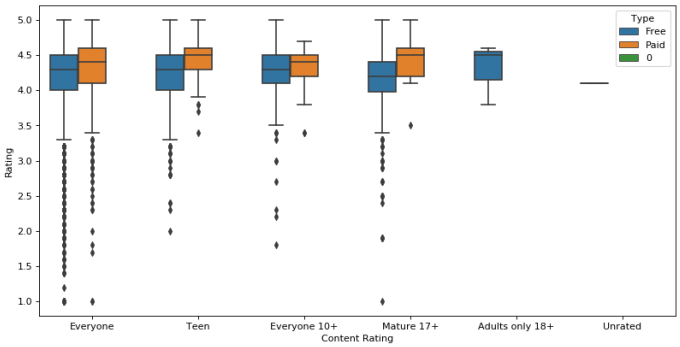


(e) Sentiment Subjectivity over Sentiment Polarity.

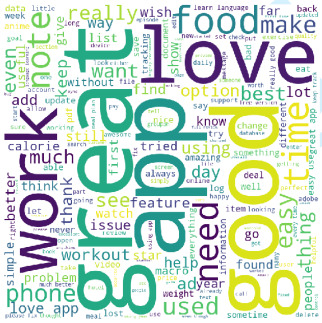
Fig. 6. Visualization of Distribution Graphs.



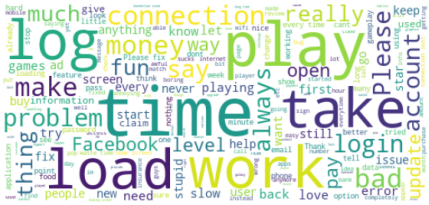
(f) Size over Installs.



(g) Content Rating vs Ratings.



(a) Positive Word Cloud.



(b) Negative Word Cloud.

Fig. 7. Visualization of Word Cloud.

was 200 while dimension of projection layer was 100. The different configurations so obtained, shows a comparison amongst these deep algorithms based on accuracy and F-scores. These are listed in Table-6.

Regarding the performance of algorithms listed in Table-6, LSTM with GloVe attained the highest accuracy of 95.2%, and F-score of 0.88. CNN and RNN with GloVe performed averagely with an accuracy value close to 93% In terms of F-score, CNN with Glove proved worst for our corpus by attaining 0.78.

4.5 Main Effects Plots for Empirical Analysis

The findings of this research has pinpointed strong empirical support to our research contribution which are accordant and robust across the real-time collection of reviews and performance metrics. To evaluate the model and judge the significance of the empirical analysis of the results we have shown various comparison plots.

- In order to encapsulate the key findings of deep learning algorithms on GloVe namely, LSTM, RNN and CNN on the Google app reviews corpus and tested on SAR, we performed the experiments in 10 epochs in specified batch sizes. The accuracy and loss plots for LSTM, RNN and CNN are depicted in Fig. 10a, 10b, 11a, 11b, 12a, 12b. respectively.
- The main findings in terms of loss and accuracy from the test dataset when applied with deep leaning algorithms on GloVe are also plotted. These are listed in Fig. 10c, 10d, 11c, 11d, 12c, 12d. for LSTM, RNN, CNN respectively.

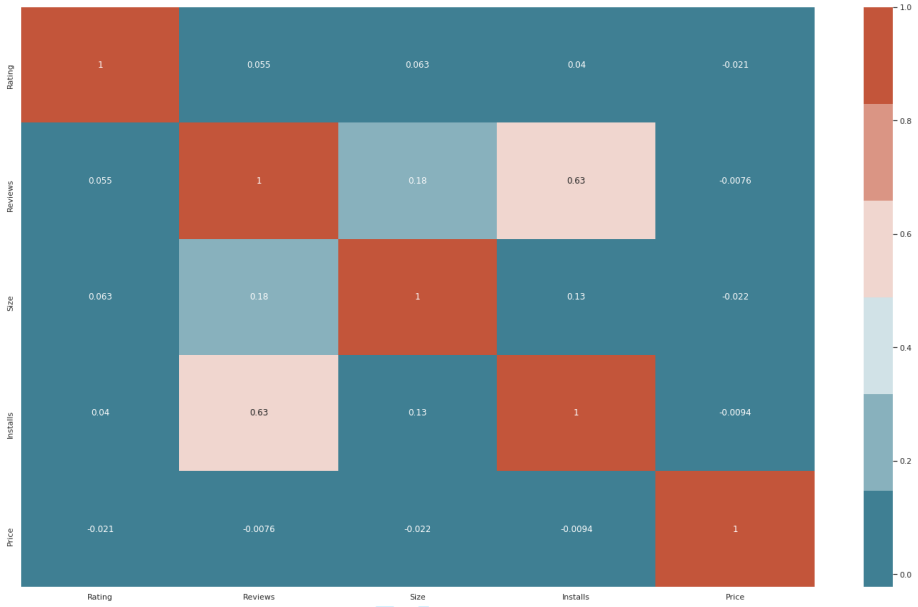
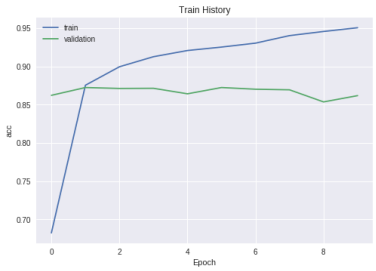


Fig. 8. Correlation between training and test data set.

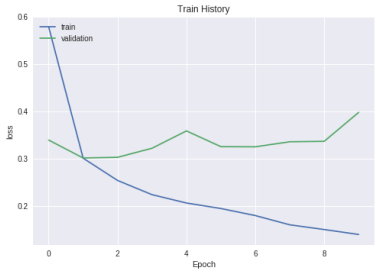
(a) Confusion Matrix with Glove (b) Confusion Matrix

Fig. 9. Visualisation of Confusion Matrix

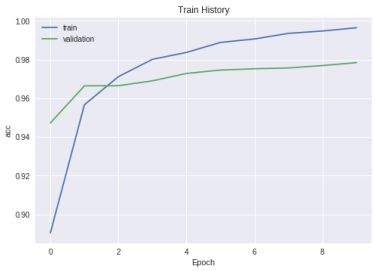
- The accuracy for the test data is found out to be 97.8% for LSTM, 89.5% for RNN and 85.9% for CNN respectively.
- In our study, we have highlighted a pattern followed by the different conventional machine learning classifiers based on accuracy and F-score. Fig. 13a shows a comparison amongst the aforesaid machine leaning algorithms based on accuracy via line chart. And Fig. 13b shows a comparison amongst the aforesaid machine leaning algorithms based on F-scores via line chart.
- The main findings in terms of F-score and accuracy from the Google app reviews corpus when applied with deep leaning algorithms in conjugation with GloVe are also plotted. The comparison plot is shown in Fig. 13c via a ranged scatter plot.



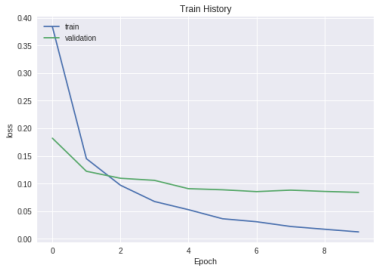
(a) Accuracy Plot for training data for LSTM.



(b) Loss Plot for training data for LSTM.

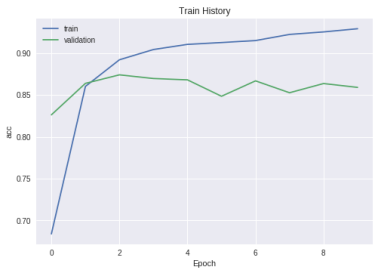


(c) Accuracy Plot for test data for LSTM.

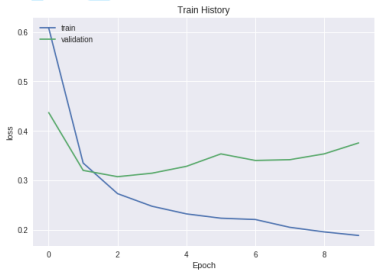


(d) Loss Plot for test data for LSTM.

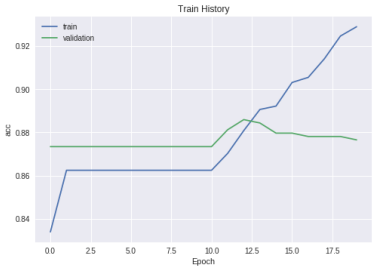
Fig. 10. Comparative Plots for LSTM.



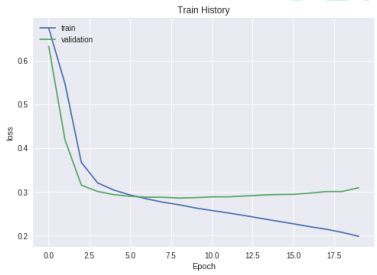
(a) Accuracy Plot for training data for RNN.



(b) Loss Plot for training data for RNN.

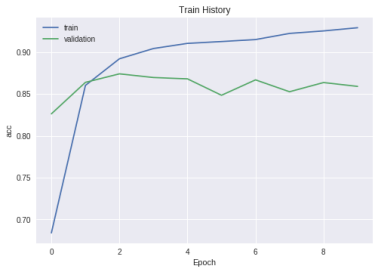


(c) Accuracy Plot for test data for RNN.

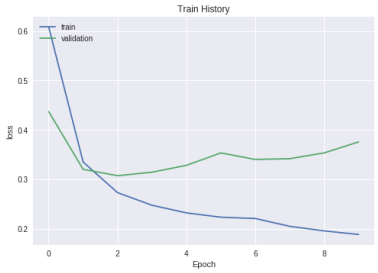


(d) Loss Plot for test data for RNN.

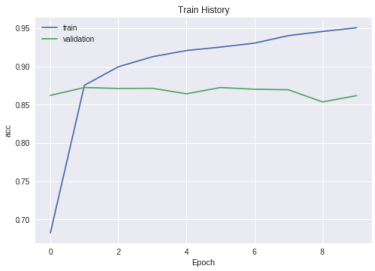
Fig. 11. Comparative Plots for RNN.



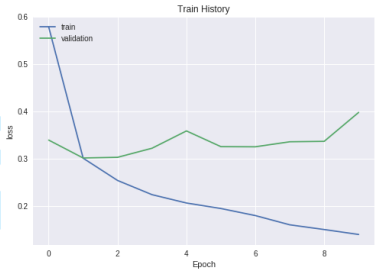
(a) Accuracy Plot for training data for CNN.



(b) Loss Plot for training data for CNN.

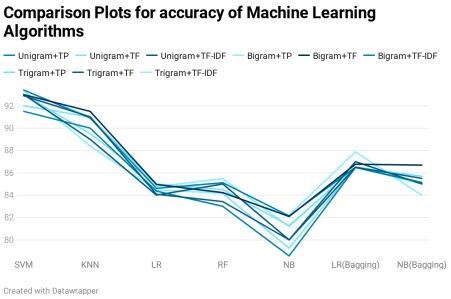


(c) Accuracy Plot for test data for CNN.

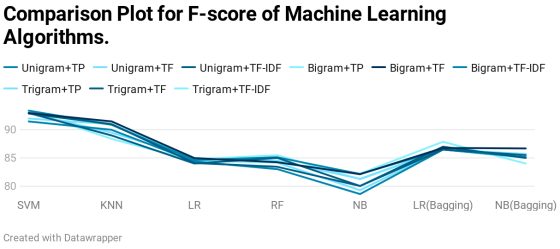


(d) Loss Plot for test data for CNN.

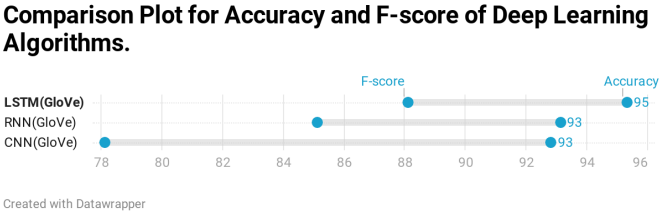
Fig. 12. Comparative Plots for CNN.



(a) Machine Learning algorithms for Accuracy.

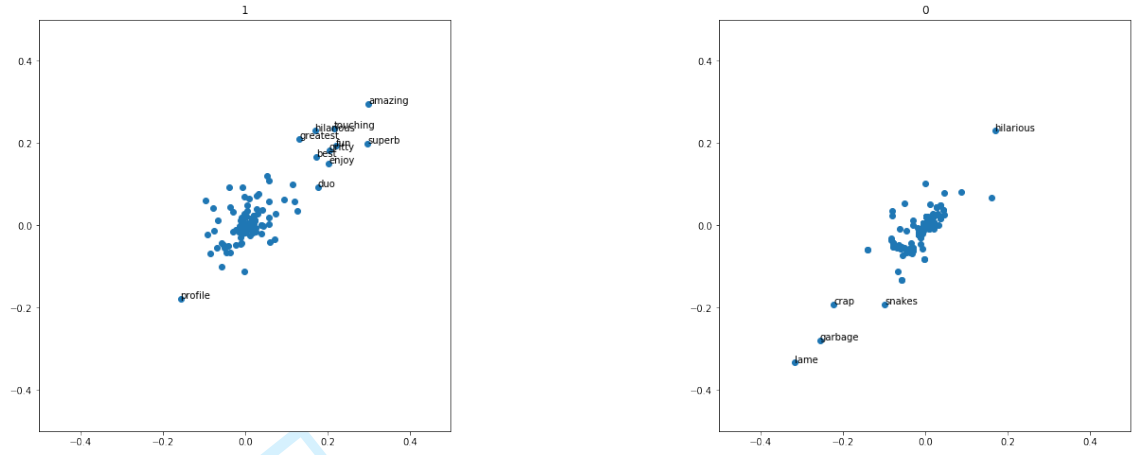


(b) Machine Learning algorithms for F-score.



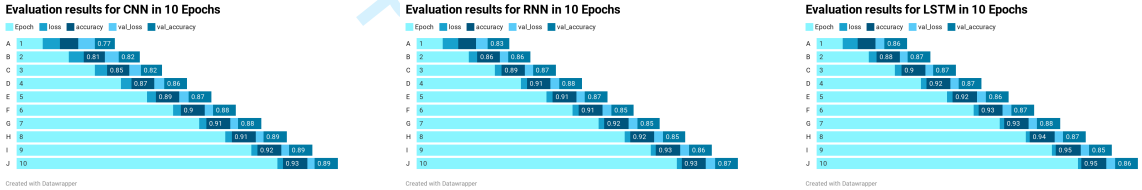
(c) Deep Learning for Accuracy and F-score.

Fig. 13. Visualization of Comparison Plot.



(a) Positive words in reviews wrt Sentiment Polarity. (b) Positive words in reviews wrt Sentiment Polarity.

Fig. 14. Visualization of words through scatterplots.



(a) CNN plot for 10 Epochs. (b) RNN plot for 10 Epochs. (c) LSTM plot for 10 Epochs.

Fig. 15. Visualization of Epochs in Deep Learning.

- From Table-6 of our study, we inferred, LSTM with GloVe attained the maximum accuracy. Using this, we modelled our corpus, Google app reviews and showed a graphical analysis of Sentiment subjectivity vs Sentiment Polarity,over a specified category Entertainment, thereby picturing positive and negative words from the university students reviews. These are illustrated through scatter plots in Fig. 14a, 14b.
- The empirical analysis of Deep learning paradigm, namely LSTM, RNN, CNN, in conjugation with GloVe were trained on the Google app reviews corpus, for 10 Epochs and tested on SAR. The key discoveries are sorted in descending order via a stacked plot in terms of Epochs and are shown in Fig. 15a, 15b, 15c respectively.

4.6 Discussions

Based on our study of Google app reviews and SAR, several insights are highlighted in the following manner:

- From the graphical analysis, we answered 10 RQ's based on university students' viewpoints towards the app market when compared with the training data set. So, we could easily analyze the percentage of students with similarity in mindset and their inclination and liking and diversified knowledge towards app market

- From our study, the best algorithm LSTM along with word embedding, GloVe yielded the maximum accuracy of 95.2% and F-score of 0.88. This algorithm can also be tested for other word embedding namely FastText and Word2vec based on the same data set.
- Text analysis and machine learning techniques can foster the administrations of educational institutes to get a feedback regarding the app market as used by university students. This might generate some learning about the valuable apps that could be made openly accessible to all students, if it is a paid one or if it is important to students for e-learning.
- The presented text mining approach for sentiment analysis of university students reviews crawled via a survey could also be initiated on a website, wherein university students of multiple universities within the same city could share their opinions on a common platform based on the commonly used or unique apps. Evaluating the e-learners reviews, identifying learners emotions, based on text feedback in real-time sentiment analysis could be integrated with deep learning based framework.
- In the empirical analysis, conventional text representation schemes, ensemble methods, machine learning paradigms and deep learning approach have been considered. Ensemble learning techniques generated higher predictive performance when compared with conventional classification algorithms. From our study a crystal clear inference is drawn that deep learning model has outperformed the machine learning classifiers.

5 CONCLUSIONS AND FUTURE SCOPE

AI, the catchphrase of today, in its high pace development has complemented humanity and made human life easier. One of the widely flourished technology in AI, namely text analysis and NLP is bench marked in our study. NLP is an overlap of machine learning and deep learning paradigm. Speech recognition, Natural Language Generation(NLG) and Natural Language Understanding(NLU) are the major capabilities of NLP. This work models the sentiment of the users using the Google reviews dataset and find the university students' behavior towards the Google app market. Usually, k-fold cross validation technique is used for testing, i.e splitting the dataset in ratio of 70:30 or 80:20. Not much research has been done in Sentiment analysis using students' reviews for testing. So, we had collected the real-life dataset from university students to study the proposed model.

The training dataset had approximately 30,000 reviews while the test dataset had approximately 400 reviews. Machine learning paradigm was efficiently employed to perform the Sentiment analysis. Five commonly used classification algorithms namely NB, SVM, KNN, LR, RF were used for performance comparison. Bagging, an ensemble method, was employed to intensify the predictive performance of the classifier. In this study, amongst the classification algorithm, SVM outperformed others in terms of accuracy(93.41%) on the TF-IDF+bi-gram feature, while NB under-performed with an accuracy(78.56%). In terms of F-score also, SVM outperformed other algorithms on TF-IDF on uni-gram, bi-gram schemes. KNN and LR performed significantly well and are fit for our data set. Apparently, RF is also catching up. Bagging was implemented on LR and NB showing apprehensive increment in accuracy and F-score. The corpus when trained using deep learning paradigms with word embedding namely GloVe showed that LSTM is highly suited for our study and some future researches. It marked an outstanding accuracy of 95.2% and F-score 0.88. CNN and RNN performed averagely on GloVe with 93% accuracy.

Despite having favourable results, there are certain limitations of this paper. To sum up, there were certain challenges faced while performing this study. They are:

- Collection of University students reviews was indeed tedious and time-taking.

- Moreover, the survey was localized only to a particular university and we did not get a 100% participation of university students.
- There were constraints on the reviews crawled from students, as they used abbreviations or short form or SMS language, slang words, spelling mistakes and emojis to express their reviews showing disbelief or disappointment.
- Some students were reluctant to participate or gave false opinions or invaluable reviews and hence would create discrepancies as well.

For further improvement the future scope of our study could be listed as follows:

- Empirical analysis of TP, TF and TF-IDF based representation in conjugation with uni-gram, bi-gram and tri-gram model respectively on other ensemble methods like random subspace and boosting.Hence, employing machine learning algorithms to study the predictive performance.
- Analysis of word embedding namely, word2vec and FastText could also be explored on deep learning models. Other deep learning models namely GRU and RNN-AM could be encountered.
- Eventually, we could expand our data set by extending our online survey in other universities as well and more students within the range of the city. Some resource crunched university students should be taken into account.
- Statistical validity of empirical analysis of Google app reviews on our model can be done using ANOVA test.
- Exploring the university students' reviews in a multilingual domain and other resource poor language could be encountered.
- Identification of influential reviewers and text summarizing could be one of the grounds for future research as well.

REFERENCES

[1] Al-Subaihin, A., Finkelstein, A., Harman, M., Jia, Y., Martin, W., Sarro, F. and Zhang, Y., 2015, August. App store mining and analysis. In Proceedings of the 3rd International Workshop on Software Development Lifecycle for Mobile (pp. 1-2).

[2] Carreño, Laura V. Galvis, and Kristina Winbladh. "Analysis of user comments: an approach for software requirements evolution." In 2013 35th International Conference on Software Engineering (ICSE), pp. 582-591. IEEE, 2013.

[3] Prasetyo, Budi Eko, Divi Galih Prasetyo Putri, and Endang Wahyu Pamungkas. "Aspect Extraction using Informative Data from Mobile App Data Review." International Journal of Computer Applications 975: 8887.

[4] Blanco-Fernandez, Yolanda, Martin Lopez-Nores, José J. Pazos-Arias, Alberto Gil-Solla, and Manuel Ramos-Cabrer. "Exploiting digital TV users' preferences in a tourism recommender system based on semantic reasoning." IEEE Transactions on Consumer Electronics 56, no. 2 (2010): 904-912.

[5] Adinolfi, Paola, Ernesto D'Avanzo, Miltiadis D. Lytras, Isabel Novo-Corti, and Jose Picatoste. "Sentiment analysis to evaluate teaching performance." International Journal of Knowledge Society Research (IJKSR) 7, no. 4 (2016): 86-107.

[6] Thet, Tun Thura, Jin-Cheon Na, and Christopher SG Khoo. "Aspect-based sentiment analysis of movie reviews on discussion boards." Journal of information science 36, no. 6 (2010): 823-848.

[7] Cui, Hang, Vibhu Mittal, and Mayur Datar. "Comparative experiments on sentiment classification for online product reviews." In AAAI, vol. 6, no. 1265-1270, p. 30. 2006.

[8] Li, Xinxin, and Lorin M. Hitt. "Price effects in online product reviews: An analytical model and empirical analysis." MIS quarterly (2010): 809-831.

[9] Go, Alec, Lei Huang, and Richa Bhayani. "Twitter sentiment analysis." Entropy 17 (2009): 252.

[10] Devaraj, Sarv, Ming Fan, and Rajiv Kohli. "Antecedents of B2C channel satisfaction and preference: validating e-commerce metrics." Information systems research 13, no. 3 (2002): 316-333.

[11] Abbasi, Ahmed, Hsinchun Chen, and Arab Salem. "Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums." ACM Transactions on Information Systems (TOIS) 26, no. 3 (2008): 1-34.

[12] Chua, Alton YK, and Snehasish Banerjee. "Understanding review helpfulness as a function of reviewer reputation, review rating, and review depth." Journal of the Association for Information Science and Technology 66, no. 2 (2015): 354-362.

[13] Tian, Fang, Hashim M. Al-Hashimi, John L. Craighead, and James H. Prestegard. "Conformational analysis of a flexible oligosaccharide using residual dipolar couplings." Journal of the American Chemical Society 123, no. 3 (2001): 485-492.

[14] Seyff, Norbert, Florian Graf, and Neil Maiden. "Using mobile re tools to give end-users their own voice." In 2010 18th IEEE International Requirements Engineering Conference, pp. 37-46. IEEE, 2010.

[15] Malik, M. S. I., and Ayyaz Hussain. "An analysis of review content and reviewer variables that contribute to review helpfulness." Information Processing Management 54, no. 1 (2018): 88-104.

[16] Liu, Jingjing, Yunbo Cao, Chin-Yew Lin, Yalou Huang, and Ming Zhou. "Low-quality product review detection in opinion summarization." In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 334-342. 2007.

[17] Liu, Bing, and Lei Zhang. "A survey of opinion mining and sentiment analysis." In Mining text data, pp. 415-463. Springer, Boston, MA, 2012.

[18] Fersini, Elisabetta, Enza Messina, and Federico Alberto Pozzi. "Sentiment analysis: Bayesian ensemble learning." Decision support systems 68 (2014): 26-38.

[19] Santos, Carolina Leana, Paulo Rita, and João Guerreiro. "Improving international attractiveness of higher education institutions based on text mining and sentiment analysis." International Journal of Educational Management (2018).

[20] Hwang, Wu-Yuin, Yung-Hui Li, and Rustam Shadiev. "Exploring effects of discussion on visual attention, learning performance, and perceptions of students learning with STR-support." Computers Education 116 (2018): 225-236.

[21] Pak, Alexander, and Patrick Paroubek. "Twitter as a corpus for sentiment analysis and opinion mining." In LREc, vol. 10, no. 2010, pp. 1320-1326. 2010.

[22] Anderson, Michael, and Jeremy Magruder. "Learning from the crowd: Regression discontinuity estimates of the effects of an online review database." The Economic Journal 122, no. 563 (2012): 957-989.

[23] Mudambi, Susan M., David Schuff, and Zhewei Zhang. "Why aren't the stars aligned? An analysis of online review content and star ratings." In 2014 47th Hawaii International Conference on System Sciences, pp. 3139-3147. IEEE, 2014.

[24] Bermingham, Adam, and Alan F. Smeaton. "Classifying sentiment in microblogs: is brevity an advantage?." In Proceedings of the 19th ACM international conference on Information and knowledge management, pp. 1833-1836. 2010.

[25] Medhat, Walaa, Ahmed Hassan, and Hoda Korashy. "Sentiment analysis algorithms and applications: A survey." Ain Shams engineering journal 5, no. 4 (2014): 1093-1113.

[26] Achtermann, Jeffrey M., Indrajit Bhattacharya, Kevin W. English Jr, Shantanu R. Godbole, Sachindra Joshi, Ashwin Srinivasan, and Ashish Verma. "Cross-domain clusterability evaluation for cross-guided data clustering based on alignment between data domains." U.S. Patent 8,229,929, issued July 24, 2012.

[27] Onan, Aytuğ, Serdar Korukoğlu, and Hasan Bulut. "A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification." Expert Systems with Applications 62 (2016): 1-16.

[28] Prabowo, Rudy, and Mike Thelwall. "Sentiment analysis: A combined approach." Journal of Informetrics 3, no. 2 (2009): 143-157.

[29] Deng, Li, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Michael Seltzer et al. "Recent advances in deep learning for speech research at Microsoft." In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 8604-8608. IEEE, 2013.
[30] Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. "Domain adaptation for large-scale sentiment classification: A deep learning approach." In ICML. 2011.
[31] Prakash,G. (2019, April). "R vs. Python: Google Play Store Apps, Version 1. Retrieved April 5, 2019 from https://www.kaggle.com/gauthamp10/google-playstore-apps
[32] Lima, Ana Carolina ES, Leandro Nunes de Castro, and Juan M. Corchado. "A polarity analysis framework for Twitter messages." Applied Mathematics and Computation 270 (2015): 756-767.
[33] Novak, Petra Kralj, Jasmina Smailović, Borut Sluban, and Igor Mozetič. "Sentiment of emojis." PloS one 10, no. 12 (2015): e0144296.
[34] Onan, Aytuğ. "Mining opinions from instructor evaluation reviews: A deep learning approach." Computer Applications in Engineering Education 28, no. 1 (2020): 117-138.
[35] Adekitan, Aderibigbe Israel, and Odunayo Salau. "The impact of engineering students' performance in the first three years on their graduation result using educational data mining." Heliyon 5, no. 2 (2019): e01250.
[36] Almasri, Ammar, Erbug Celebi, and Rami S. Alkhawaldeh. "EMT: Ensemble meta-based tree model for predicting student performance." Scientific Programming 2019 (2019).
[37] Jena, R. K. "Sentiment mining in a collaborative learning environment: capitalising on big data." Behaviour Information Technology 38, no. 9 (2019): 986-1001.
[38] Khan, Farhan Hassan, Saba Bashir, and Usman Qamar. "TOM: Twitter opinion mining framework using hybrid classification scheme." Decision support systems 57 (2014): 245-257.
[39] Harman, Mark, Yue Jia, and Yuanyuan Zhang. "App store mining and analysis: MSR for app stores." In 2012 9th IEEE working conference on mining software repositories (MSR), pp. 108-111. IEEE, 2012.
[40] McIlroy, Stuart, Nasir Ali, and Ahmed E. Hassan. "Fresh apps: an empirical study of frequently-updated mobile apps in the Google play store." Empirical Software Engineering 21, no. 3 (2016): 1346-1370.
[41] Taba, Seyyed Ehsan Salamati, Iman Keivanloo, Ying Zou, Joanna Ng, and Tinny Ng. "An exploratory study on the relation between user interface complexity and the perceived quality." In International Conference on Web Engineering, pp. 370-379. Springer, Cham, 2014.
[42] Huang, Albert H., Kuanchin Chen, David C. Yen, and Trang P. Tran. "A study of factors that contribute to online review helpfulness." Computers in Human Behavior 48 (2015): 17-27.
[43] Tang, Jiliang, Huiji Gao, Xia Hu, and Huan Liu. "Context-aware review helpfulness rating prediction." In Proceedings of the 7th ACM conference on Recommender systems, pp. 1-8. 2013.
[44] Oramas Bustillos, Raúl, Ramón Zatarain Cabada, María Lucía Barrón Estrada, and Yasmin Hernández Pérez. "Opinion mining and emotion recognition in an intelligent learning environment." Computer Applications in Engineering Education 27, no. 1 (2019): 90-101.
[45] Cabada, Ramón Zatarain, María Lucía Barrón Estrada, and Raúl Oramas Bustillos. "Mining of educational opinions with deep learning." Journal of Universal Computer Science 24, no. 11 (2018): 1604-1626.
[46] Sultana, Jabeen, Nasreen Sultana, Kusum Yadav, and Fayez AlFayez. "Prediction of sentiment analysis on educational data based on deep learning approach." In 2018 21st Saudi Computer Society National Computer Conference (NCC), pp. 1-5. IEEE, 2018.
[47] Nguyen, Phu XV, Tham TT Hong, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen. "Deep learning versus traditional classifiers on vietnamese students' feedback corpus." In 2018 5th NAFOSTED Conference on Information and Computer Science (NICS), pp. 75-80. IEEE, 2018.
[48] Zhou, Chunting, Chonglin Sun, Zhiyuan Liu, and Francis Lau. "A C-LSTM neural network for text classification." arXiv preprint arXiv:1511.08630 (2015).
[49] Zhang, Meishan, Yue Zhang, and Duy-Tin Vo. "Gated neural networks for targeted sentiment analysis." In Thirtieth AAAI Conference on Artificial Intelligence. 2016.
[50] Li, Cheng, Xiaoxiao Guo, and Qiaozhu Mei. "Deep memory networks for attitude identification." In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, pp. 671-680. 2017.
[51] Chen, Huimin, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. "Neural sentiment classification with user and product attention." In Proceedings of the 2016 conference on empirical methods in natural language processing, pp. 1650-1659. 2016.
[52] Kandhro, Irfan Ali, Shaukat Wasi, Kamlesh Kumar, Malook Rind, and Muhammad Ameen. "Sentiment analysis of students' comment using long-short term model." Indian J. Sci. Technol. 12, no. 8 (2019): 1-16.
[53] Dou, Zi-Yi. "Capturing user and product information for document level sentiment analysis with deep memory network." In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 521-526. 2017.
[54] Zhang, Meishan, Yue Zhang, and Guohong Fu. "Tweet sarcasm detection using deep neural network." In Proceedings of COLING 2016, The 26th International Conference on Computational Linguistics: Technical Papers, pp. 2449-2460. 2016.
[55] Joshi, Aditya, Pushpak Bhattacharyya, and Mark J. Carman. "Automatic sarcasm detection: A survey." ACM Computing Surveys (CSUR) 50, no. 5 (2017): 1-22.
[56] Majumder, Navonil, Soujanya Poria, Haiyun Peng, Niyati Chhaya, Erik Cambria, and Alexander Gelbukh. "Sentiment and sarcasm classification with multitask learning." IEEE Intelligent Systems 34, no. 3 (2019): 38-43.

[57] Dahou, Abdelghani, Shengwu Xiong, Junwei Zhou, Mohamed Houcine Haddoud, and Pengfei Duan. "Word embeddings and convolutional neural network for arabic sentiment classification." In Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers, pp. 2418-2427. 2016.
[58] Kamble, Satyajit, and Aditya Joshi. "Hate speech detection from code-mixed hindi-english tweets using deep learning models." arXiv preprint arXiv:1811.05145 (2018).
[59] Bertero, Dario, and Pascale Fung. "A long short-term memory framework for predicting humor in dialogues." In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 130-135. 2016.
[60] Zhang, Daoqiang, Yaping Wang, Luping Zhou, Hong Yuan, Dinggang Shen, and Alzheimer's Disease Neuroimaging Initiative. "Multimodal classification of Alzheimer's disease and mild cognitive impairment." Neuroimage 55, no. 3 (2011): 856-867.
[61] Zhu, Guangming, Liang Zhang, Peiyi Shen, and Juan Song. "Multimodal gesture recognition using 3-D convolution and convolutional LSTM." Ieee Access 5 (2017): 4517-4524.
[62] Tang, Duyu, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. "Learning sentiment-specific word embedding for twitter sentiment classification." In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1555-1565. 2014.
[63] Wang, Jindong, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. "Deep learning for sensor-based activity recognition: A survey." Pattern Recognition Letters 119 (2019): 3-11.
[64] Severyn, Aliaksei, and Alessandro Moschitti. "Learning to rank short text pairs with convolutional deep neural networks." In Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, pp. 373-382. 2015.
[65] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up? Sentiment classification using machine learning techniques." arXiv preprint cs/0205070 (2002).
[66] Miller, George A. "WordNet: a lexical database for English." Communications of the ACM 38, no. 11 (1995): 39-41.
[67] Hackeling, Gavin. Mastering Machine Learning with scikit-learn. Packt Publishing Ltd, 2017.
[68] Brownlee, Jason. "Machine learning mastery with python." Machine Learning Mastery Pty Ltd (2016): 100-120.
[69] Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. The elements of statistical learning: data mining, inference, and prediction. Springer Science Business Media, 2009.
[70] Kaewyong, Phuripoj, Anupong Sukprasert, Naomie Salim, and Fatin Aliah Phang. "The possibility of students' comments automatic interpret using lexicon based sentiment analysis to teacher evaluation." In 3rd International Conference on Artificial Intelligence and Computer Science (AICS2015), pp. 179-189. 2015.
[71] Vapnik, Vladimir, and Vlamimir Vapnik. "Statistical learning theory Wiley." New York 1 (1998).
[72] Lewis, David D. "Naive (Bayes) at forty: The independence assumption in information retrieval." In European conference on machine learning, pp. 4-15. Springer, Berlin, Heidelberg, 1998.
[73] Aha, David W., Dennis Kibler, and Marc K. Albert. "Instance-based learning algorithms." Machine learning 6, no. 1 (1991): 37-66.
[74] Breiman, Leo. "Random forests." Machine learning 45, no. 1 (2001): 5-32.
[75] Onan, Aytuğ, Serdar Korukoğlu, and Hasan Bulut. "A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification." Information Processing Management 53, no. 4 (2017): 814-833.
[76] Freund, Yoav, and Robert E. Schapire. "Experiments with a new boosting algorithm." In icml, vol. 96, pp. 148-156. 1996.
[77] Breiman, Leo. "Random forests." Machine learning 45, no. 1 (2001): 5-32.
[78] Rezaeinia, Seyed Mahdi, Rouhollah Rahmani, Ali Ghodsi, and Hadi Veisi. "Sentiment analysis based on improved pre-trained word embeddings." Expert Systems with Applications 117 (2019): 139-147.
[79] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543. 2014.
[80] LeCun, Yann. "Generalization and network design strategies." Connectionism in perspective 19 (1989): 143-155.
[81] Block, H. D. "A review of "perceptrons: An introduction to computational geometry." Information and control 17, no. 5 (1970): 501-522.
[82] Dong, Qi, Yu Chen, Xiaohua Li, and Kai Zeng. "Explore Recurrent Neural Network for PUE Attack Detection in Practical CRN Models." In 2018 IEEE International Smart Cities Conference (ISC2), pp. 1-9. IEEE, 2018.
[83] Zhang, Lei, Shuai Wang, and Bing Liu. "Deep learning for sentiment analysis: A survey." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8, no. 4 (2018): e1253.
[84] Rojas-Barahona, L. M. "Deep learning for sentiment analysis language and linguistics. Compass 10: 701–719." (2016).
[85] Ciregan, Dan, Ueli Meier, and Jürgen Schmidhuber. "Multi-column deep neural networks for image classification." In 2012 IEEE conference on computer vision and pattern recognition, pp. 3642-3649. IEEE, 2012.
[86] Elman, Jeffrey L. "Finding structure in time." Cognitive science 14, no. 2 (1990): 179-21.