

Research Article

Health App Recommendation System using Ensemble Multimodel Deep Learning**Deepak Chowdary Edara, Venkatramaphanikumar Sistla* and Venkata Krishna Kishore Kolli***Department of Computer Science & Engineering, VFSTR Deemed to be University, Vadlamudi, 522213, Guntur, Andhra Pradesh, India.*

Received 7 April 2020; Accepted 20 October 2020

Abstract

Nowadays, mobile devices and apps are meant to fulfill the needs of various people in society. But, mobile app Stores are facing major challenges in recommending proper apps for users. Recommending mobile apps for users according to personal preference and various mobile device limitations is therefore important. In this scenario, there is a huge need for developing recommender systems (RS) for the user's community in enabling critical mobile apps such as Health based Apps. Recommendation Systems perform an extensive survey on the collection of user reviews, preferences and opinions to discover recommendations of suitable applications to the users' community. In this paper, we have designed an aspect-based recommendation framework by performing three tasks: such as identifying the mentions associated with item aspects in user reviews, extracting the sentiment related opinions using Latent Semantic Analysis of such aspects in the reviews, and perform the opinion mining from all of the aspects to generate enhanced recommendations with Ensemble Multimodel Deep Learning (EMDL). EMDL comprises of two state-of-the-art classifiers such as Deep Neural Networks (DNN) and Long Short Term Memory (LSTM). In contrast to the prior work, we conducted a series of experiments with several state-of-art deep learning models to extract useful recommendations. The achieved results show that classification with outperforms in all the aspects based on various evaluation metrics when compared to the rest of the models.

Keywords: Opinion Mining, Mobile Apps, Deep Learning, Ensemble Methods, Analytics, Recommendation Systems

1. Introduction

In the past few years, electronic gadgets such as mobile phones have evolved rapidly and have become a necessity of the present generation. The ascendancy of mobile phones has increased the dependence of people on portable contrivances in their lives. Now millions and millions of people all over the world are mobile phone holders. Intriguingly, few countries have more mobile phones than the number of residents in that area based on recent studies. Moreover, in developing countries, these mobile phones have given the underprivileged section of society access to numerous health and legal services even with minimal telecommunication infrastructure. This has helped service providers to offer a wide variety of mobile phone models, that the public can choose from. According to a news article[1], approximately 165 thousands of medical apps are presently accessible on Apple and Android mobile phones. The usage of these applications in conventional mobile and wireless technologies promotes health goals known as mHealth (Mobile Health). MHealth applications in mobile phones encourage healthcare professionals to work together having instant access to images and sample outcomes provided by patients as and when they request. Patients also prefer to become effective decision-makers. This change in notion is often called patient empowerment. Medics, students, and patients are connected globally to social networking. It is advantageous in sharing knowledge, feedback, and awareness of health. To avoid the uncertainty

of doctor-patient interaction, the adequateness of the communication system and the amount of data that the doctor is willing to disclose must be taken into consideration. For this reason, the General Medical Council has exclusively released a social network platform called Doximity. This allows doctors to promote professional communication and networking and, in rare cases, to work together to share insights into the medical industry. In addition to this, WHO estimates that by 2035, 12.9 million additional health professionals, including sister-friendly health care workers, and doctors, are needed worldwide for learning using electronic media (e-learning). The best way of using this technology is to create awareness in universities, public health organizations, NGOs, and private firms. The Department of Global Health at the University of Washington urges for learning from a distance in inadequate resources. Their Global Health E-Learning (eDGH) department, which had started in Kenya and Haiti, now operates in 30 countries [2]. Being able to use tablets and mobile phones in the training of doctors has helped to train more physicians in countries of poor and average income. The higher the number of physicians who are trained, the better is the care for their patients worldwide. People who bond with such connections give support that would take part in their survival. However, irrelevant and overload information are significant obstacles to conclusions about the status of personal health and adequate action [3]. Users habitually become hesitant after self-analysis when confronted with a lot of health-related information on various platforms (social apps, web forums, news sites, etc.). Therefore, Health Recommender Systems (HRS) [4] is intended to centralize the health data of a person and provide

*E-mail address: svrphanikumar@yahoo.com

ISSN: 1791-2377 © 2020 School of Science, IHU. All rights reserved.

doi:10.25103/jestr.135.03

access to the concerned individual and authorized professional healthcare professionals as a solution.

Recommendation Systems (RS) uses artificial intelligence (AI) methods to make product recommendations to users. Based on feedback-driven information obtained from user's, RS can be categorized into i. Content-Based (CB), ii. Collaborative Filtering (CF), and iii. Hybrid filtering (HF) [5]. To create recommendations, user data must be collected through RSs employing a collaborative or hybrid filtering process. This can either be achieved explicitly or implicitly. When users are conscious of their information, an explicit compilation of user data happens. Example: Users generally fill user details to apply for a new online service. Another form of explicit user data collection occurs when users convey their liking and dislike based on product rating using categorical variables or personal views such as "like" Facebook. Implicit user information indirectly gathers user information. For example, the online store server is used to swap information with a computer that allows RS to know the user's browser as well as the user's location while visiting an online store. More advanced user clicks and keystroke records are being tracked. As well as the common process of recommending that users present items of interest, the recommendations can be attempted to make differently. Trust-based recommendations [6] make another study about the trust-based user relationship. A trust relationship is a direct connection to a friend in a social network. Recommendations based on friends are good enough to justify more than those without trusted on the user's present location, the information is collected based on the context and the conduct of the user (idle, work, and sleeping). The variety of processed context information is tremendous, making context-aware recommendations an extensive field of research. Risk-Aware Recommendation Systems are a collection of context-aware recommendations and will only identify the context from sensitive data like user signs. It is risk-aware, because a user's life may be threatened by wrong choices.

This study mainly focuses on the task of recommending various mobile phone health applications to users based on their health conditions. There is extensive semantic data is available in app stores such as app details from developers and user reviews. User reviews are straight reactions showing the knowledge of the application. Reviews not only serve developers helpful opinions to enhance their apps but often suggest recommendations for possible users in the future. The main purpose of the study is to analyze different user reviews obtained from various health applications and to recommend the specific application required for the user's health condition. To understand and distinguish the approaches by studying literature on this aspect and the techniques they use to make the recommendation system work as necessary in the field of health care. We use the topic modeling techniques for the topics that are analyzed by most of the users and their opinions on topics. These topics emphasize applications which are significant to users and uncover the application's features. The user behavior of apps is better learned to meet the user requirements in recommending applications for the target user. We use web-crawlers to collect real-world review data from mobile app stores, such as metadata apps, user descriptions, and reviews. Before modeling a specific topic, the crawled data is pre-processed. The topic modeling technique in user reviews can generate latent topics and show every application as a distribution of probability for topics. We

identify the topics that have been reached as the app's significant features or functions.

Additionally, a standard analysis of user opinions and views on unstructured large-scale data has been proposed in this study. We develop a user setup history profile that provides recommendations to suggest suitable applications by considering or preventing specific topics to identify polarity about a particular topic. User expectations and the topic distribution of each request predict recommendations based on contingent probability, contributing to the intended user's identified list of recommended applications. For this analysis, we also used Apache Spark Framework because extracting useful knowledge from large amounts of unstructured data is a very tedious task. The analysis is put into effect by several machine learning methods on user review data collected from various medical apps from Apple IOS and Android mobile stores. We have also demonstrated that our approach is relevant for faster analysis, measurement, and recommendation. Finally, the performance of the proposed approach is analyzed by calculating the average recall for all users. The other sections of this paper are structured as follows: a detailed review of the literature is described in Section II. The proposed methodology is summarized in Section III. Section IV describes the results of the experiment and concludes in Section V.

2. Related Work

The integration of big data with data mining improves trust in understanding product requirements and the user needs to be based on the probability of their interest. Accurate Models predict the correct trends by making the use of available resources in all sectors like retail, education, technology, etc. For these scenarios, recommendation systems are more frequently used to expect user needs and their behavior to make recommendations including facts. Below is a compilation of several models of prediction and recommendation using various ML techniques found in the literature.

Due to the available and accessible of a huge amount of information, social media analytics have become evident, and firms around the world found it is a highly tedious situation to manage the outcomes at the same time. Various Analytical Models including Sentiment Analysis were therefore recognized to establish business relationships among different organizations. Sentiment Analysis (SA) or Opinion Mining (OM) focuses on discovering attitudes, feelings, and emotions expressed in text form. In other words, they refer to the handling of text analysis, natural language processing, and biometrics to evaluate, obtain, quantify, analyze affective states and even provide user perspective subjective information about the product or events. These approaches help to determine the feeling of the users about their products or services, which lets them develop and market. Mobile Health apps are not only used for business but also utilized by different types of people for multiple theories in the field of education and medicine. However, the majority of these will rely upon the view of our social environment., and it can only persist until people have experienced it over the period.

Authors [7] carried out sentiment analysis by proposing a linear regression method to predict iPhone sales from their tweets with a strong correlation. Twitter can analyze people's sentiments about specific healthcare or public health

measures linked contagion estimate with Twitter data and stated that the intended use of medicines might be indicated for a specific syndrome of infection or substantial injury. It can also be used to extract and examine the impact of negative reactions to medicines that help improve drugs and make social media and other health forums popular[8]. Therefore, they found that an unprotected community of adverse vaccine sentiments results in an intensified outbreak of diseases. Authors [9] have developed an Appjoy recommendation system that includes User App records to develop a user preference matrix, rather than using certain user ratings. An Eigenapp model [10] was developed by considering item-rating matrix, for recommending apps by solving sparse records obtained from user records. This system improved the visibility of less popular applications and suggested the more diversified recommended list. A uniform custom context-aware recommendation method to consider both independence and dependence assumptions. The model can determine user-specific preferences from mobile user context logs for the recommendation of mobile apps. They also reviewed a Mobile Context-Aware Recommendation survey, which developed the amount of new mobile app services. To detect an application's kernel functionality, authors [11] used metadata from apps to measure the comparison values of apps with the proposed algorithm. Similar applications can then be recommended or searched using the similarity measure. used an external data source (Twitter), the official account mobile app's supporters, to deduce user liking to correct the user-item matrix abnormality in CF, which handles the cold-start issue of fully released applications without user scores.

Before building models, many of the works employ feature selection strategies. Examples from related fields involve spam detection[12], fraud detection[13] and mechanical fault diagnosis [14]. We also use LDA topic modeling of reviews along with sentiments to develop an app recommendation system. SA is a common processing method for natural language, which is used to evaluate subjectivity, feelings, and views in the textual content. Sentiments have been used in textual reviews in the development of recommendation systems. [15]have obtained user feature pairs for customized recommendations obtained from review documents to improve the matrix factorization model. As the number of features obtained from the word intensity can be quite large, the feature is limited to the top of the list according to user comments on the features. We use topical sentiment model[16], [17]to evaluate text to obtain features and user opinions that provide an overview of all features on a wide range of topics. [18] examined user reviews in application stock to identify differences among review text and ratings using LDA[19]to detect topics that users complain about. [20]described the sentiments in the web sites reviews to create virtual ratings for carrying out the CF-based recommendation without considering item ratings. [21]considered user interdisciplinary behavior. Many mobile applications have software variants, but device characteristics could lead to distinct user behavior and attitude. The topic was modeled on cross-platform user reviews and user ratings to generate an application recommendation matrix factorization model [22] using LDA to obtain latent topics based on rating factors to improve the framework for predicting target ratings for CF items. Furthermore, considering the challenges of spam review and rating detection is also essential for App services. For instance, [23] have identified several representative attitudes of spammers and have shaped them to detect spam reviews.

Authors [24] developed a novel unsupervised framework to discover fake opinions and spammers using their cognitive footprints[25] [26] also investigated the problem of analyzing hybrid shilling attacks [27] rating data based on the algorithm of semi-supervised learning. Simultaneously, they use restricted Boltzmann machines are used to enhance collaborative filtering performance. The findings indicate that Restricted Boltzmann Machinery exceeds single value decomposition (SVD) in the Netflix data set. Moreover, other data mining techniques were also implemented by researchers to significantly improve recommender systems using optimization algorithms. For example, Rendle has introduced Factorization Machines (FMs), which combines SVMs with Factorization Models [28].

A recommendation system's scalability problem also creates difficulties and makes it even more difficult for researchers and practitioners to provide efficient and effective services to users. There have been many resolutions to overcome the problem [29]. One of the most common alternatives is parallel computing [30], used to deploy a collaborative filtering system. The effectiveness of recommendation techniques is improved in parallel computation than in single-machine operation. Adopting the distributed computing approach improves the recommended system's performance qualitatively. For example, Hadoop could help to achieve linear speedup with the collaborative filtering technique [31]. And large data sets could speed up better than smaller ones [32]. While Hadoop decreases the scalability of recommendation techniques to some expansion, MapReduce does not support collaborative filtering algorithms[33]. The reason is that in the computation of similarities, CFneedscontinuous reading, and writing of data. However, Hadoop is a disk-based structure and a computational bottleneck is the continuous reading and writing of data. Therefore, Spark memory-based model has become a practical option for large scale recommender systems[34].

K-Means clustering algorithm with Alternating Least Square (ALS) [35] is applied on the Spark platform to prevent computational complexity and data sparsity of collaborative processing algorithms. Multi-criteria CF utilizing the Spark framework has been introduced by [36]. The findings of the experiment showed that with the number of Spark clusters, the effectiveness of algorithms has improved. Spark, therefore, needs to be used in recommending systems to achieve greater computational performance. The authors presented a new framework for improving user recommendations on social networks. They use user priorities to develop an MF model based on past user similar interests. The authors had developed a multi-graph ranking recommendation model to identify latent items for users. They have extended prior efforts by collaborative filtering techniques through the integration of various types of user relations like relationships, neighborhoods, and colleagues, which reduce data scarcity. Through the development of online social networks, such interactions are possible.

Similarly, the authors used hyper graphic ranking to propose a solution that can solve challenging and changing user e-commerce demands. Due to its nature, conventional single-objective recommendation models are inadequate, e.g. users are not always permitted to go to the same hotel despite being "best fit," their companions and time-location contexts may affect their choice. Therefore, the authors suggest a multi-objective model of recommendation that can solve these conditions. Despite the complexity in designing

a real-time recommendation engine, it is often helpful to develop hybrid approaches for problem-solving. In [37] authors have developed a hybrid job recommendation model using statistical relation learning [38]. They also considered algorithm tuning to fulfill the requirement to make the classes (difficult classes) more desirable. Contextual information [39] was used to recognize human actions and, consequently, to recommend music streaming [40].

3. Proposed Health App Recommendation system using Ensemble Multimodel Deep Learning Classification

Online reviews help users to identify liking and disliking an application. We categorize the features of the app as user priorities and topic distributions of user reviews based on features installed on various apps. CF [41], [42] and CB [43], [44] approaches may compete for each other's deficiencies, demonstrating that the recommender system is effective and consistent. When comparing each of these techniques, content-based methods may compensate for the

collaborative filtering method with low scalability. Collaborative filtering can cover the inability to customize the content-based approach. Usually, a hybrid recommendation approach is first used to produce a preliminary recommendation list based on user information and mobile app information. Then, OM is performed to optimize the preliminary list and obtain the final list of recommendations. Moreover, this study completely examines the performance of the recommending system on the premise of the hybrid recommendation implementation. In the process of recommending health apps, this study also tends to focus on user reviews on mobile health applications. Based on the herding effect, users are expected to select products or facilities preferred by most individuals. Thus, health apps with more positive reviews are prioritized to be recommended to users compared with health apps with many negative reviews. Following optimization, the final recommendations were generated as described in the following Fig 1. At the first stage, User reviews and application metadata from Mobile App Store are being crawled by using Heedzy and iTunes API to collect user reviews of health apps.

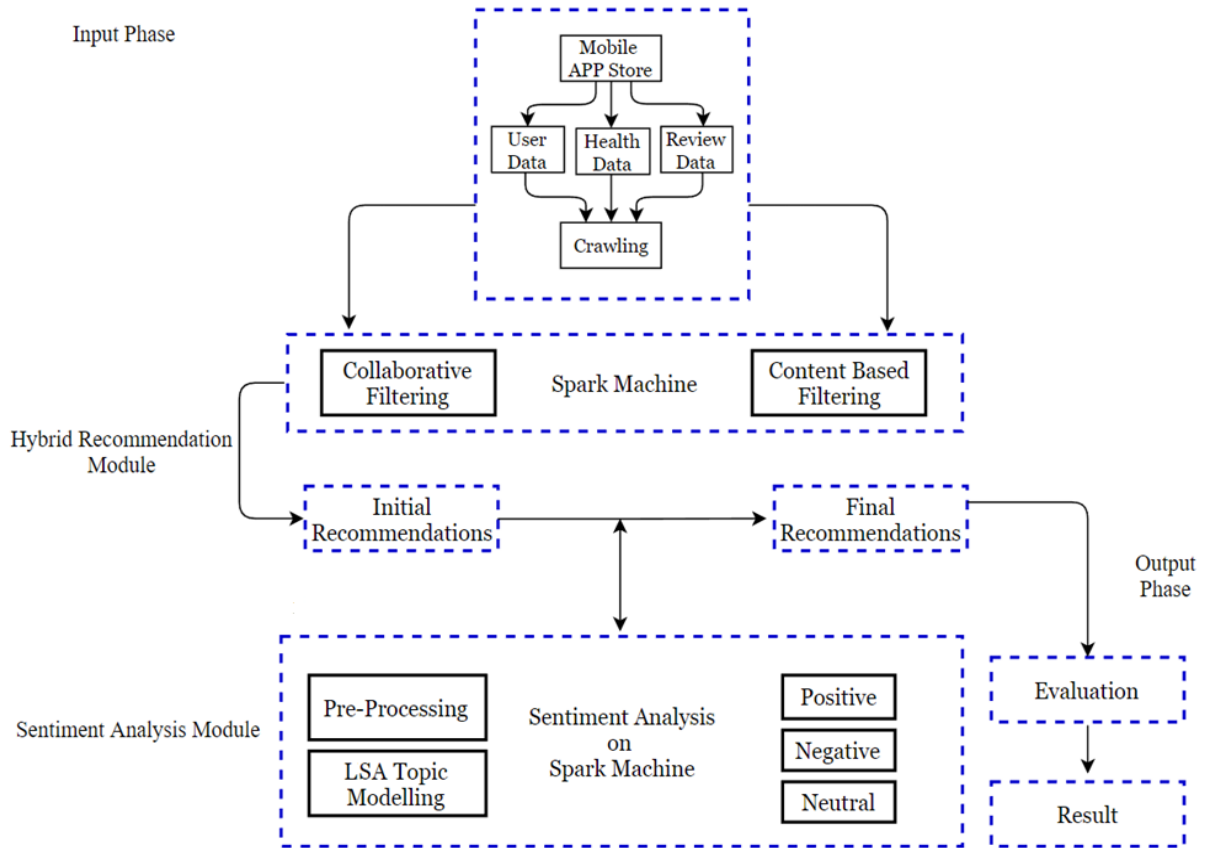


Fig. 1. Proposed Research Workflow for Health App's Recommendation System

The application metadata contains details from the developer's application like Source, Date, Title, Content, Name, Rating, Compatibility, and Version. In contrast to the text-based descriptions of user reviews, the user ratings that follow each user review are also considered in the recommendation. In this study, we assess the efficiency of the app recommendation based on user reviews. The following Tab. 1 lists statistics of crawled data with all the application metadata.

Table 1. Statistics about the Data

Characteristics	Count
Unique Words	942310
Stop Words	724100
Positive Words	662012
Negative Words	520268
Neutral Words	914306
Total Terms	3762996

In the second stage, the Collaborative filtering approach is incorporated to analyze the user's opinions and choices, so how to gather the user's choices becomes the starting point of this process. Users have several options of delivering their own system choices like ranks and clicks. Our approach analyzes user's reviews based on their ratings on health applications which are taken into consideration. Before importing the data into the CF process, we need to preprocess the data. We use the Natural Language Toolkit (NLTK) for pre-processing tasks. We first eliminate punctuation, non-English letters, stop words and words less than three characters. NLTK offers a list of English stop words from its corpus. The Porter stemming algorithm [45] is then performed to remove the inflectional ending of a word to result in its root form. All the pre-processed app reviews are incorporated into a single document to generate the data set for the topic modeling task. Each document contains all the text reviews written for the app, and every document represents a corpus for modeling a topic. Further, the normalization of input data is required. This approach can be more effective by optimizing the data. From the above steps, we get a two-dimensional table where the user list has one dimension, and the other is the health app list, while the user app rating is the value. The priority data is transformed into the user app resilient distributed data set (RDD), which can be processed by Spark. We can learn some disciplines from the behavior of the user and preferences to assist the following recommendation. Because of the high timeliness of mobile services, the calculation efficiency has to be improved. The data are stored in Spark's memory during the computing of user preferences. If the CB analysis of recommendations is carried out after the data has been entered into disks where unnecessary operations were faded. Therefore, simultaneous operations like the computation of user preferences are performed by both CB and CF strategies. The Spark platform has calculated the total number of apps preferred by the user and the number of apps recommended concurrently by both users to manage the data on the distributed platform. The two types of statistics are the computation of nodes of the Spark platform in s distributed process and storing the processed results in the form of RDD's.

At the third stage, identical apps were determined based on user preferences by analyzing their behavior. To find identical users, similarities were computed by Normalized Minkowski distance. Hence, the similarity (S) between the user's u_x, u_y can be determined by

$$S(u_x, u_y) = \left(\sum_{i=1}^n w_i |x_i - y_i|^\lambda \right)^{1/\lambda} \quad (1)$$

where x_i, y_i denotes the ratings given by the user's on an app i .



Fig. 2. Word cloud representation for informative aspects extracted from reviews of various Mobile Health Applications

In the above steps, all users can be rated by assessing the importance $S(u_x, u_y)$. To recommend user apps, top K most similar users are acknowledged based on the similarities and preferences of their apps where a recommended list of user u_x apps is computed. Moreover, the item representation vectors and similarities among the preference of user u_x are also to be considered. Apps that are irrelevant for the user u_x will be discarded from the list and hence the initial recommendation list becomes the basis for our proposed framework. The scores (SC) obtained from both recommendation models which were computed is represented as.

$$SCR_{CF,a} = \sum_i sim(u_i, u_c) R_{i,a} \quad (2)$$

$$SC_{CB,a} = R_{t,a} \sum_j sim(a, a_{j,i}) \quad (3)$$

where S_{CF} represents the score of the app a in collaborative filtering. $sim(u_i, u_c)$ denotes the similarity between user u_i and candidate user u_c . $R_{c,a}$ is the rating from candidate user u_c on app a . $S(CB, a)$ represents the app score as a recommended method based on app's content. $S(a, a_{j,i})$ denotes the similarity between app a and app which user u_i has already installed. $R_{t,a}$ is the rating from user u_c on an app a .

Topic Modelling with Latent Semantic Analysis (LSA):

Generally, user reviews are in an unstructured format and contain rich information. A statistical linguistic framework is essentially needed to interpret the semantic significance of unstructured textual content and to summarize the valuable information. After computing the similarities among different user's, the user preferences were modeled into various topics associated with reviews using a generative statistical model called Latent Semantic Analysis (LSA) [46], [47] on a specified corpus. The reason was that reviews were frequently shared with a simultaneous representation of latent space, and although they had no prevalent terms in the data. Moreover, the identification of term similarities between reviews can be more accurate in the optimal representation than in the actual representation. LSA overcomes all noise terms and discovers such terms and synonyms that are connected to a certain topic in this regard. In this study, PLSA [48] from the LSA model had been used to extract topics from the co-occurrence matrix. As shown in the above Fig. 2. The PLSA framework is one of the statistical methods derived from a collection of topic models and also known as the aspect model. This PLSA approach comprises of a latent variable which is linked with observed concepts by hidden concepts. In our data with the PLSA model, the hidden concepts were determined with three sets of parameters:

$$r \in R = \{r_1, \dots, r_n\} \quad (4)$$

$$f \in F = \{f_1, \dots, f_m\} \quad (5)$$

$$t \in T = \{t_1, \dots, t_k\} \quad (6)$$

where the above eq (1) represents the reviews corpus with n number of observed variables, eq (2) describes the observed variables with m number of distinct aspects from the review corpus, and eq (3) represents k number of hidden topics. Fig. 3 describes the conceptual process for reviews with all three

sets of parameters associated to an aspect model that identifies t topics with observed (r, f) pairs. The following observations indicate the conceptual process for all reviews:

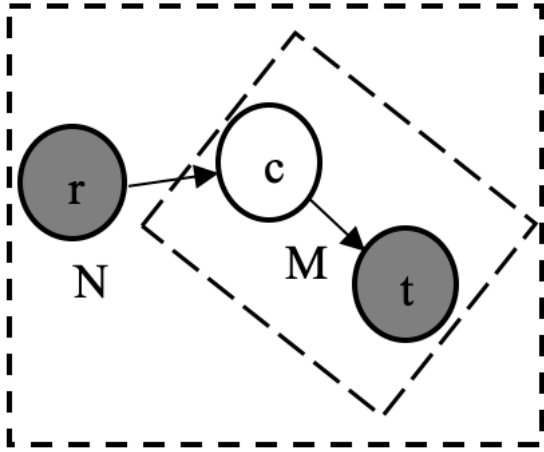


Fig. 3. Graphical Model of PLSA

A review r_n is selected with a probability $P(r)$. For each feature f_i in the review r_n , generate a topic t_i with a probability $P(t/r_n)$. where $i \in \{1, \dots, M_f\}$. Select a feature f_i from the previously generated topic with a probability $P(f/t_i)$.

The joint distribution is determined to develop independently sampled Bag-of-words with Joint Variable (r, f) from the observed data is resolved as :

$$P(R, F) = \prod_{(r,f)} P(r, f) \quad (7)$$

The conditional independence of features and reviews in a given topic is defined as

$$P(f, r/t) = P(f/t)P(r/t) \quad (8)$$

where

$$P(r, f) = P(r)P(f/r) \quad (9)$$

$$P(f/r) = \sum_{t \in T} P(f, r/t) = \sum_{t \in T} P(f/r, t)P(t/r) \quad (10)$$

Based on the conditional independence assumption, we get

$$P(f/r) = \sum_{t \in T} P(f/r, t)P(t/r) \quad (11)$$

$$P(f, r) = \sum_{t \in T} P(t)P(r/t)P(f/t) \quad (12)$$

From the above eq (5), $P(f/t)$ and $P(t/r)$ describes a variety of factors for the PLSA method. This equation probably results in the decomposition of the matrix under the mixed coefficients $P(t/r)$ and $P(f/t)$ topic vectors classified based on probability distributions. The probabilities of $P(f/t)$ and $P(t/r)$ in PLSA method is defined with the highest probability function as represented below:

$$L_M = \prod_{(f,r)} P(f/r) = \prod_{r \in R} \prod_{f \in F} P(f/r)^{n(r,f)} \quad (13)$$

where, $n(r, f)$ indicates the number of times feature f appeared in review r .

Matrix Factorization [49], [50] is also a PLSA approach representation method. The matrix with review terms is quite significant and has a huge number of rows and columns. Here, rows represent Number of reviews and M represents a certain number of column terms. In each review, the aspects with less sparsity were used as per their relevant topic. Most topics in this evaluation matrix consist of zero entries, as they do not provide information. In this case, the dimensionality reduction must be done to improve the dimensionality of a review-word matrix by removing such null entries. This can be achieved by the following:

$$H_0 \approx \widehat{H}_0 = M_1 \cdot M_2 \quad (14)$$

where H_0 indicates the co-occurrence matrix and $M_1 \cdot M_2$ denotes the product of two low-rank matrices M_1 and M_2 . If the size of the matrix M_1 is $i_1 \times j_1$ and M_2 is $i_2 \times j_2$, with $i_2 \ll j_2, i_1$, then the task of dimensionality reduction is fulfilled. This was performed because of $i_1 \cdot j_2 \gg i_1 \cdot j_1 + j_1 \cdot j_2$. Furthermore, Matrices M_1 and M_2 extract the latent structure information from the corpus. As the above eq(5) shows the exact matrix factorization of $P(f/r)$, eq(6) is interpreted to factorize the full co-occurrence of $P(f, r)$. The matrix notation of this can be denoted as:

$$H_0 = M_1 \cdot M_D \cdot M_2 \quad (15)$$

where H_0 shows probabilities of reviews $P(r/t)$, M_D is a diagonal matrix of prior probabilities with the topics $P(t)$ and the corresponding M_2 shows the probability of each word $P(f/t)$. These two matrices ensure the possible outcomes attained without any difficulty from SVD by representing the consistent and non-negative probability distributions.

Initially, each app's pre-processed reviews are treated as a corpus to form the LSA model for each app review to produce the topic distribution. The output is easy to interpret and can be informative as it is possible to identify the probability of an app for specific topics by analyzing the topic's probability distributions. We interpret these topics as the features of applications or user needs and hence each application represents their topic distributions of probability. Because, users tend to adopt particular kinds of applications unless, of their distinct interests or needs, we can acquire data about the user's inclination based on earlier installed applications. Therefore, we analyze the topic distributions of earlier installed applications to analyze the user preferences about the features of mobile health applications. The profile of a user is developed in the following way. A user c installed m applications $A(u) = \{a_1, a_2, \dots, a_m\}$ with topic probability $\{p_{1,1}, p_{1,2}, \dots, p_{1,K}, p_{2,1}, \dots, p_{m,K}\}$ acquired from the results of LSA trained corpus. To compute the preferences of the user of each topic $p(t_k|u)$, we compute the possibility that a user likes at first for every topic z_k as

$$p(t_k|u) = \frac{\sum_{a \in A(u)} p(z_k|a)}{m} \quad (16)$$

where $k = 1, \dots, K$.

Note that reviews $A(u)$ are the user's review set of installed applications and the latent topics $T = \{t_1, \dots, t_k\}$ are derived from the corpus LSA model training. Then we get $U = \{p(t_1|u), p(t_2|u), \dots, p(t_k|u)\}$, which summarizes the user's topic preferences of previously installed applications.

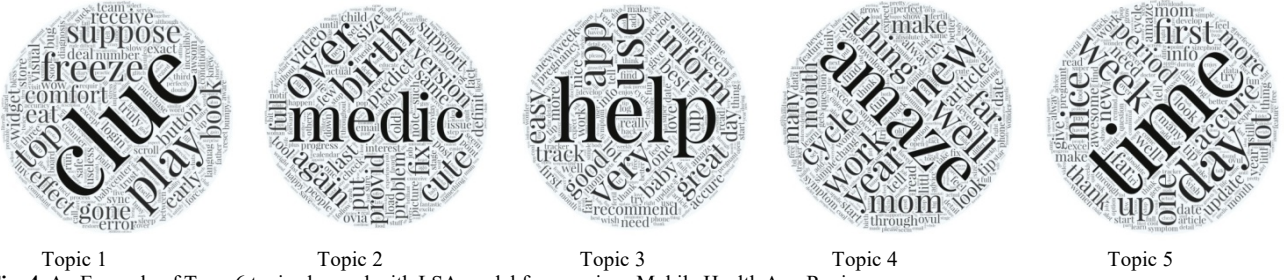


Fig. 4. An Example of Top - 6 topics learned with LSA model from various Mobile Health App Reviews

Opinion Mining & Recommendations through Weighted Ranking:

Each app is given with 5-star rating from the user posted the review in the mobile app stores. The rating shows the user is happy with the app or unhappy with it. An application that serves a user shows the app provides unique features to meet the needs of the user. Every app is given with weights and associated with the user profile to obtain their preferences based on rating pieces of information. These weights are obtained from each device by optimizing the rating for all the applications installed.

$$W_a = \frac{r_a}{\sum_{a' \in A(u)} r_{a'}} \quad (17)$$

which indicates the weight of an application a for u , where r_a refers to the rating of an application provided by u . To employ the weights of applications with the user profile, we apply our weights to applications like

$$p_w(t_k|u) = \sum_{a \in A(u)} W_a p(t_k|a) \quad (18)$$

This method is used to develop the user profile c and is then used to compute each application's recommendation ratings for a target user. The sentiments in the text can show if a user likes certain features. If sentiments are not considered, the obtained topics may contain negative sentiments which combine the semantics in the topics. The LSA model deals with such sentimental topics based on emotions. With the number of predefined K topics associated with the polarities l_s , LSA produces K sentimental topics for each sentiment. Each review r_n has the distribution of polarities $p(l_s)$ of which $l_s = 1, \dots, S$, are the labels of sentiment. Topics of a review in the corpus are based on polarities $p(t_k^l|r_n)p(t_k^l, r_n)$, where $k=1, \dots, K$, $l_s = 1, \dots, S$ in each sentiment each review connected to K topics. To develop the topic modes from app user reviews, we use three variety of polarities, positive, negative and neutral. Each app is associated with distribution of sentiments across positive, negative, and neutral labels. Therefore, the similarities between the target user profile and the profiles of all target applications for each sentiment category are computed separately and all the recommendation scores were estimated. Further, the results of all sentiment categories are summarized to generate the recommendation ratings. Existing users share related opinions with the final user in summing up three sentiment-based ratings. The segregation of sentiment classes prevents the mix of topics with opposing sentiments. Moreover, the neutral category topics are considered to be important because it also contains significant app information though they don't have any polarity. Based on the probability of different sentiment distributions $p(pos|r_n)$, $p(neg|r_n)$, and $p(neu|r_n)$ of an app's review r_n and the topic distributions incorporated with the

sentiment labels of the app $p(t_k^{pos}|pos, r_n)$, $p(t_k^{neg}|neg, r_n)$, and $p(t_k^{neu}|neu, r_n)$, $k = 1, \dots, K$, sentiment-topic for each profile can be denoted as:

$$p(t_k^{pos}|r_n) = p(t_k^{pos}|pos, r_n) \cdot p(pos|r_n) \quad (19)$$

$$p(t_k^{neg}|r_n) = p(t_k^{neg}|neg, r_n) \cdot p(neg|r_n) \quad (20)$$

$$p(t_k^{neu}|r_n) = p(t_k^{neu}|neu, r_n) \cdot p(neu|r_n) \quad (21)$$

Then, we recognize the user's already installed apps to build the user profile, which also consists of Positive (pos), negative (neg) and neutral (neu) sentiment groups as

$$p_{ws}(t_k^{pos}|u) = \sum_{a \in A(C)} W_a p(t_k^{pos}|a) \quad (22)$$

$$p_{ws}(t_k^{neg}|u) = \sum_{a \in A(C)} W_a p(t_k^{neg}|a) \quad (23)$$

$$p_{ws}(t_k^{neu}|u) = \sum_{a \in A(u)} W_a p(t_k^{neu}|a) \quad (24)$$

Hence, the scores of final recommendations of each app is calculated as:

$$\sum_{l_s = S^*} \sum_{k=1}^K p(t_k^{l_s}|r) \cdot p_{ws}(t_k^{l_s}|u) \quad (25)$$

where $S^* = \{pos, neg, neu\}$ and r is the corresponding review of the each app a .

The hybrid recommendation lists include health apps rated by their scores (SC) obtained computing CFAs shown below:

$$S_{hybrid,a} = S_{CF,a} + S_{CB,a} \quad (26)$$

where $S_{hybrid,a}$, denotes the rank of health app a in the hybrid recommendation system.

The Opinion Mining significantly improves the final recommendation list by incorporating the sentiment polarity on the score of each app. Therefore, the score of each app is computed as follows:

$$S_{final,a} = W_{hybrid} S_{CF,a} + W_{S^*} S_{CB,a} \\ S_{final,a} = W_{hybrid} S_{hybrid,a} + W_{S^*} S_{S^*,a} \quad (27)$$

where W_{hybrid} and W_{S^*} indicates the weights of two recommendation models and $S_{CB,a}$ denotes the score of app a obtained from SA. $S_{S^*,a}$ is the summation of all l_s of reviews for the mobile apps. The final recommendations are obtained based on the new score computed and will be provided without any order. Therefore, to choose suitable health apps, more apps are chosen by the proposed hybrid recommendation approach and some are eliminated based on obtained final scores.

Predictive Model Development using Machine Learning Approaches:

Multinomial Logistic Regression: A Multinomial Logistic Regression (MLR) (Hamdan, Bellot&Bechet, 2015; Vinodhini & Chandrasekaran, 2014) is one of the most popular statistical classification models in the pool of supervised classification algorithms. This algorithm uses the linear regression model as represented in eq (28) to compute the score and then trains the classifier by estimating the scored with various classes based on the target classes.

$$w \cdot X + b \quad (28)$$



(a) Positive Aspects (b) Negative Aspects
Fig. 5. Example list of Positive and Negative Opinion Aspects obtained from Preliminary Recommendation List

Where w denotes a weighted matrix with values w_1, w_2, w_3, X represents set of inputs with X_1, X_2, X_3 . It also interprets a Softmax function to solve the problems occurred in performing multi-classification. Initially, the workflow of the MLR has been split into the number of phases starting from input to output. The input phase of an MLR contains the number of recommendations obtained through the SA task. For example, if we are going to predict the top N-recommendations for mobile health apps, the features will then be based on its polarity and will be considered as inputs for the MLR. In the second phase, the outputs from the previous phase will be considered as final scores obtained by computing w and X from a linear model as follows:

$$w_1 \cdot X_1 + w_2 \cdot X_2 + w_3 \cdot X_3 \quad (29)$$

In the next phase, the higher and lower probabilities were computed for the obtained scores based on the softmax function defined as follows:

$$S(Z) = \frac{e^{z_j}}{\sum_{i=1}^i e^{z_i}} \text{ for } j = 1, 2, 3, \dots, i \quad (30)$$

Cross-Entropy (CE) is computed at the final phase to estimate the similarities between the probabilities obtained from $S(Y_i)$ based on the one-hot-encoding matrix. One-Hot-Encoding is an approach matrix used to calculate the target class similarity distance and to store the input in a binarized form.

Multinomial Naive Bayes: Multinomial Naive Bayes (MNB) [51]–[53] is special type of Naïve Bayes Classifier which is used to expect independence between the features once their corresponding class is identified. Moreover, MNB computes the higher probability of most frequent terms from reviews belonging to a particular class based on the following equation (25).

$$P(c|t_i) = \frac{P(c)P(t_i|c)}{P(t_i)} \quad (31)$$

where $c \in C$ and C represents a set of classes and $P(c|t_i)$ represents the highest probability of a test review t_i that belongs to a particular class c . $P(t_i)$ is considered as normalization factor that can be calculated as:

$$P(t_i) = \sum_{k=1}^{|C|} P(k)P(t_i|k) \quad (32)$$

Hence, the prior class $P(c)$ can be computed by dividing the number of reviews belonging to class c by the total number of reviews. $P(t_i|c)$ is the probability of acquiring a review like t_i in the class c is computed as:

$$P(t_i|c) = \frac{(\sum_n f_{ni})! \prod_n \frac{P(w_n|c)^{f_{ni}}}{f_{ni}!}}{\quad} \quad (33)$$

where $P(w_n|c)$ represents the probability of word n of class c and f_{ni} represents the word count of test review t_i . The latter probability of count of aspects F_{xc} from all the training data belonging to class c can be computed as:

$$\hat{P}(w_n|c) = \frac{1 + F_{nc}}{N + \sum_{x=1}^N F_{xc}} \quad (34)$$

where x is the word and the laplacian estimator is used to evade the zero-frequency problem [87]. $(\sum_n f_{ni})!$ and $\prod_n f_{ni}!$ are considered as computationally excessive terms and can be avoided if necessary without affecting the results as they were based on class c . Therefore, it can be defined with a constant parameter α as:

$$P(t_i|c) = \alpha \prod_n P(w_n|c)^{f_{ni}} \quad (35)$$

Linear Support Vector Machine: Linear Support Vector Machine (LSVC) or Support Vector Machines (SVM)[54] is a well known supervised learning method that has been commonly utilized for SA or OM task. For the first time Vapnik, introduced this algorithm in 1998 to analyze the data for classification and regression tasks. The primary objective of this classifier is to separate training data points from various classes. The traditional SVM is suitable only for binary classification. For this study, we employed a classical SVM algorithm with One-Versus-One and One-against-all method based on LibSVM [55] which is a popular library for SVM. The one-Versus-One method in SVM has a limitation as the number of SVM's with many classes grows superlinearly, the classification of data samples becomes inefficient [56], [57]. For this reason, we considered the One-against-all as the second method in SVM for classifying data samples with multiple classes like Positive, Negative and Neutral. The below equation defines the Multiclass SVM with One-against-all strategy [58].

$$D_i(x) = w_i^t x + b_i \quad (36)$$

where b_i is a scalar vector and w_i indicates the m -dimensional vector. x represents the input and $D_i(x)$ is the hyperplane which forms an optimal separating plane when it is set to be "0". When the trained data samples are linearly separable, the hyperplane $D_i(x)$ will be equal to "1" and the support vectors belong to class i . w_i indicates the m -dimensional vector.

Predictive Model Development with Deep Neural Network and Ensemble Classification Approaches:

In this work, authors have extended the concepts of Deep Neural Network Models to the problem of classifying

hierarchical recommendations for Health apps. These models serve effective computational models by combining non-linear layers of processing elements. Simple components can, therefore, generalize the overall network [59]. We utilized different deep neural network models to propose an ensemble approach which helps in generating valuable user recommendations from various Mobile Health Applications. The main aim of using ensemble approaches in this study is to achieve more accurate recommendations by integrating a set of classifiers than using a single classifier.

The usage of ensemble approaches enables to learn the residuals of the base classifiers to obtain a final classifier with more accuracy. Ensemble Multimodel Deep Learning (EMDL) is used for text classification [60]. In this work, we also authors constructed an ensemble classifier by generating the nodes and layers randomly with two deep neural network architectures like DNN and LSTM as shown in Fig 6. Further, the obtained final recommendation list is trained with the EMDL network using majority voting.

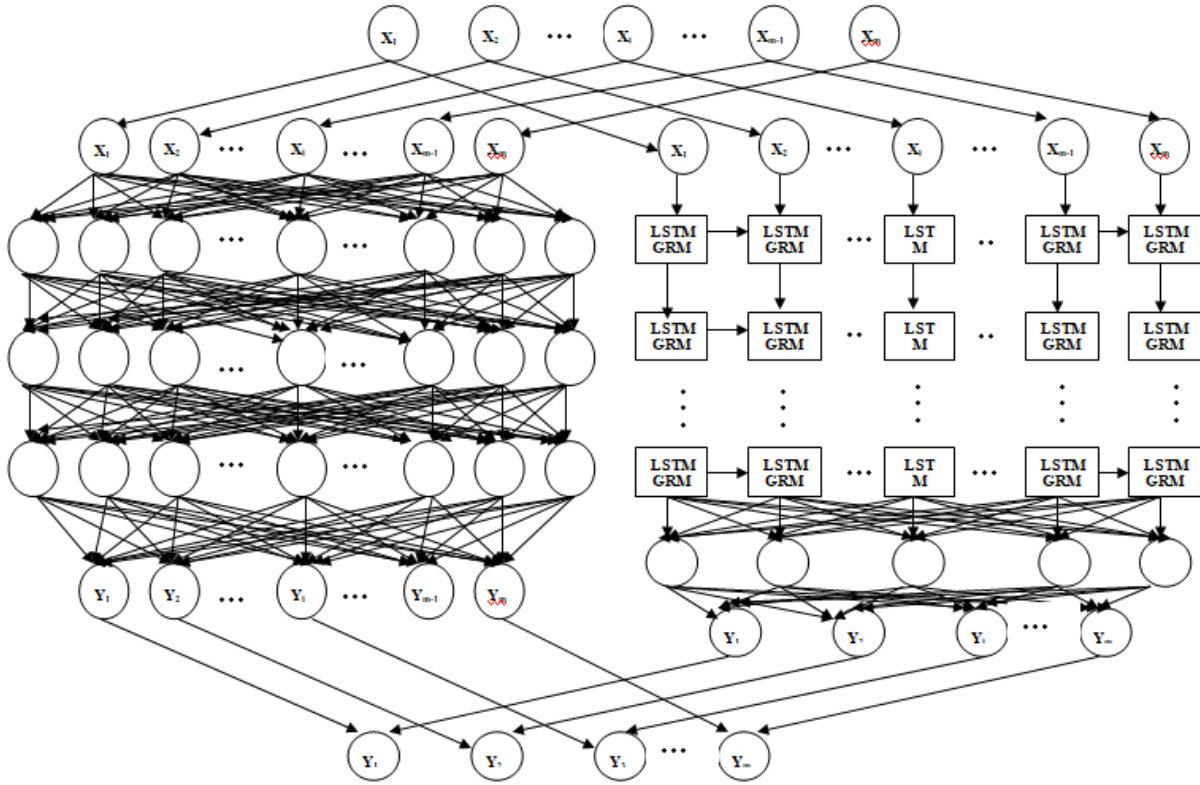


Fig. 6. Architecture of the proposed Ensemble Multimodel Deep Learning for text classification

Deep Neural Networks:

Multilayer Perceptron [61] is a feed-forward based artificial neural network classifier embedded with a set of multiple layers named as the input layer, hidden layer, and output layer. Each of these layers is interconnected to form a network with a combination of neurons, weights, and biases assigned to their interconnections. The learning process is performed based on the backpropagation (BP) method on the input and output data through hidden nodes. The BP method follows an error-correction mechanism for comparing the data processed between input and output nodes. The input to a network deals with feature vectors as $x_1, x_2, x_3, \dots, x_n$ which are connected to the nodes in the input layer and the hidden layers deals with computation of nodes assigned by weights $w_{k1}, w_{k2}, w_{k3}, \dots, w_{kn}$. And finally, the linear output a_k is computed at the output layer based on the linear arrangement among weighted inputs with an activation function f to produce anode output as z_k . The weights and biases b_k are fine-tuned to reduce the error calculated based on BP algorithm until this process reaches the lowest error rate which is predefined. Mean Square Error (MSE) is the error function used in this study to compute the error rate. A neuron K in MLP classifier can be described as:

$$z_k = f(a_k + b_k) \quad (37)$$

$$a_k = \sum_{i=1}^N w_{ki}x_i \quad (38)$$

Recurrent Neural Networks (RNN):

Recurrent Neural Networks [62]–[64] is another type of neural network model which has many advantages over text processing. This RNN model allocates more weights to the prior data points of sequence and becomes one of the powerful models for text data classification as we utilized in this work. The prior information of the nodes considered for the neural net is a very complicated method that let for enhanced semantic analysis. The general notation of RNN is represented as follows:

$$x_t = F(x_{t-1}, u_t, \theta) \quad (39)$$

where x_t represents the state at time t , u_t is the input at time t . The weights were also used in RNN and the representation is as follows:

$$x_t = w_{rec}\sigma(x_{t-1}) + w_{in}u_t + b \quad (40)$$

where b is the bias, w_{in} is the weights of the input, σ is the element-wise function and w_{rec} represents weight of the

recurrent matrix. The main limitation of this RNN model is about fading and exploding the gradient.

Long-Short Term Memory Networks(LSTM):

Long-Short Term Memory [65]–[67] is a unique category of RNN model that conserve long term dependency more effectively when judged against the RNN base model. The key advantage of this LSTM network is to overcome the fading and exploding difficulty caused in RNN. The LSTM normalizes the quantity of information and allows such information to each state of the node through multiple gates. The generic formation of LSTM network is explained in a step-by-step procedure as follows:

The input gate of a LSTM network is defined as

$$i_t = \sigma(w_i[x_t, h_{t-1}] + b_i) \quad (41)$$

The Candid Memory cell value is defined as

$$\tilde{C}_t = \tanh(w_c[x_t, h_{t-1}] + b_c) \quad (42)$$

The Activation of Forget Gate is defined as

$$f_t = \sigma(w_f[x_t, h_{t-1}] + b_f) \quad (43)$$

The value of a New Memory Cell is defined as

$$C_t = i_t \times \tilde{C}_t + f_t C_{t-1} \quad (44)$$

The values of Output Gate are represented as

$$o_t = \sigma(w_o[x_t, h_{t-1}] + b_o) \quad (45)$$

$$h_t = o_t \tanh(C_t) \quad (46)$$

From the above equations, $o, f, c,$ and i are the indices for the output gates, forget gates, cell memory and input. x_t is denoted as the memory cell input at the time t . w indicates the weighted matrices and b is the bias.

4. Experimental Results

4.1 Data Collection

The data used in this work was collected from the Android Google Play Store and Apple Store using HeedzyWebcrawler [84]. The collected data were represented in the form of mobile health app reviews and ratings along with a user name, title, content and date where each user can review and rate each mobile health app that was installed. The user-generated reviews are presented to those who show their interest to utilize the particular application. A total of 18,38,023 reviews were crawled from 89,736 users of 637 health apps. Each health app contains an average of about 2885 reviews, and each user makes 20 reviews.

4.2 Hardware Implementation & Evaluation Criterion

This section describes an extensive analysis of the effectiveness and computational complexity of the proposed approach. A cluster was initially arranged with six Linux Operating System computer nodes; 8 GB RAM with a 2.4 GHz and 1 TB hard drive. It is shown that one of the six nodes is a master node and another is a data server. The apps were integrated with the library mounted on top of the Hadoop using the Python API over Spark version 2.3.0. The

proposed method has been implemented in Spark data frames that take the input in tabular values. To evaluate the proposed model, we employed various classifiers like MLR, NB, LSVC, and MLP classifiers which are available on the Spark ML Lib platform. We also call various neural networks and deep learning models CNN, DNN, RNN, LSTM and EMDL externally on Spark Machine for classifying the Mobile health app recommendations. Also, we defined each user with N apps recommended by the proposed approach and then calculated it with the actual user-initiated apps, where N is the total number of apps in the list of recommendations. For instance, if $N = 1$, the recommendation list contains only one mobile app. Given a top- N recommendation list A_N resorted in descending order by computing the scores. The following four metrics were used to evaluate the recommendations based on predicted values.

The measure Accuracy (Acc) estimates the proportion of users whose applications are included in the recommendation list:

$$Acc @ N(\mu_i) = \frac{A_{N,rec} \cap A_{\mu_i,ins}}{A_{\mu_i,ins}} \quad (47)$$

The measure Precision (P) estimates the proportion of predicted applications that are equivalent to the actual apps that were included by users:

$$P @ N(\mu_i) = \frac{A_{N,rec} \cap A_{\mu_i,ins}}{N} \quad (48)$$

The measure Recall (R) estimates the proportion of actually included applications which are equivalent to the applications that were predicted:

$$R @ N(\mu_i) = \frac{A_{N,rec} \cap A_{\mu_i,ins}}{A_{ins}} \quad (49)$$

The measure F-Score (F) is used to compute a weighted harmonic mean of the precision (p) and recall (r):

$$F = \frac{2 * P * R}{P + R} \quad (50)$$

The mobile health app list of a candidate can be optimized by performing the Opinion Mining task. The results of the Opinion Mining used in mobile health app reviews are correlated to the assessment of the initial recommendation list. Our model is enhanced by grouping CB and CF methods with the OM framework on a distributed platform. Deep Learning approaches work better than the traditional machine learning approaches with basic recommendations in terms of accuracy, where the EMDL performs more effectively in identifying suitable health applications for users based on their interest. To evaluate the effectiveness in generating recommendations with the proposed ensemble approach, we performed a chain of experiments and compared the performance with various classifiers as discussed below. First, Spark's computational efficiency increases when the nodes in a computational cluster enhance and the corresponding experimental results show a reduction in running time. Secondly, when our approach is tested on larger data, it is better to speed up computational efficiency. The findings as shown in the following Tables.2,3 and 4 indicates that the EMDL approach is accurate and efficient. Performance evaluation of proposed Collaborative filtering Recommendation approach with various classifiers is evaluated on single and

multi node cluster machines. On a single node cluster machine, precision, recall and f-score values of EMDL are 0.852, 0.85 and 0.851 respectively. On a multinode machine, precision, recall and f-score values of EMDL is 0.908, 0.906

and 0.907 respectively. In both the scenarios, EMDL outperforms other classifiers and performance of EMDL on multi-node cluster outperforms single-node cluster.

Table 2. The Performance of Collaborative filtering Recommendation approach with various existing Machine Learning Models on a Single node Cluster (S)vs Multi-node Cluster (M).

Cluster	Metrics	MLR	MNB	LSVC	MLP	CNN	RNN	LSTM	EMDL
S	P	0.742	0.786	0.770	0.723	0.803	0.832	0.853	0.852
	R	0.740	0.784	0.770	0.723	0.802	0.830	0.852	0.850
	F	0.741	0.785	0.770	0.723	0.802	0.831	0.851	0.851
M	P	0.832	0.853	0.842	0.803	0.832	0.877	0.887	0.908
	R	0.830	0.852	0.840	0.802	0.830	0.875	0.888	0.906
	F	0.831	0.851	0.841	0.802	0.831	0.876	0.886	0.907

The precision, recall and f-score values of EMDL on a single node cluster machine as a Content-based filtering Recommendation approach has achieved 0.933, 0.932 and 0.931 respectively, and these results outperforms all other

state-of-art classifiers. Performance of EMDL has higher value of 0.03, when it was evaluated on multimode cluster machines.

Table 3. The performance of a Content-based filtering Recommendation approach with various existing Machine Learning Models on a Single node Cluster (S)vs Multi-node Cluster (M)

Cluster	Metrics	MLR	MNB	LSVC	MLP	CNN	RNN	LSTM	EMDL
S	P	0.779	0.837	0.841	0.740	0.782	0.890	0.924	0.933
	R	0.777	0.836	0.840	0.748	0.783	0.890	0.922	0.932
	F	0.778	0.835	0.842	0.749	0.781	0.890	0.923	0.931
M	P	0.814	0.862	0.876	0.760	0.919	0.922	0.958	0.963
	R	0.813	0.861	0.875	0.768	0.917	0.920	0.956	0.962
	F	0.815	0.863	0.877	0.769	0.918	0.921	0.957	0.961

The precision, recall and f-score values of EMDL on a single node cluster machine as a Hybrid- Recommendation approach has achieved 0.953, 0.952 and 0.952 respectively,

and these results outperforms all other state-of-art classifiers. Performance of EMDL has yielded higher value of 0.02, when it was evaluated on multimode cluster machines.

Table 4. The performance of the Hybrid Recommendation approach with various existing Machine Learning Models on a Single node Cluster (S)vs Multi-node Cluster (M)

Cluster	Metrics	MLR	MNB	LSVC	MLP	CNN	RNN	LSTM	EMDL
S	P	0.850	0.864	0.863	0.835	0.872	0.892	0.948	0.953
	R	0.850	0.863	0.862	0.834	0.871	0.890	0.946	0.952
	F	0.850	0.863	0.860	0.834	0.873	0.891	0.947	0.952
M	P	0.872	0.896	0.902	0.852	0.918	0.935	0.972	0.978
	R	0.870	0.895	0.901	0.850	0.915	0.933	0.971	0.977
	F	0.871	0.894	0.900	0.851	0.916	0.934	0.973	0.979

We compared the performance of the proposed hybrid recommendation approach with EMDL with content-based and Collaborative filtering Recommendation approaches. Performance of the proposed EMDL outperforms all other state-of-art classifiers and has yielded higher accuracy on both single node and multi-node clusters. From the experimental results, the proposed approach EMDL have better recommends health apps to the users.

5. Conclusion

Mobile Recommendation System needs both precision and timeliness. In this study, a novel ensemble framework is proposed for generating helpful mobile health app recommendations. Also, Spark is used to increase the system's timeliness. Our proposed approach makes it quick and convenient to acquire useful recommendations for health apps for users. The health app recommendation is an extensive task involving different types of users and health apps. Because of the useful information hidden in user

reviews, collaborative filtering is the most common and broadly used approach in the recommendation system. Additionally, the user's history is very crucial, so we add a CB recommendation technique to CF to create a hybrid recommendation system with the features derived from the health app recommendation. Moreover, during the analysis, we understand the sentiments of positive and negative information where people generally believe that positive views have a helpful impact and negative views have an unhelpful impact. Hence, opinion mining will assist us to enhance the precision of the results of recommendations. Also, as discussed in our results, the need for a distributed system is essential to resolve the scalability and time complexity issues of the recommendation system. In the immediate future, this research will, therefore, focus on reducing individual qualities hidden from users in the text analysis.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License



References

- Rimal, R. N. & Lapinski, M. K. Why health communication is important in public health. *Bull. World Health Organ.* **87**, 247 (2009).
- Sommerhalder, K., Abraham, A., Zufferey, M. C., Barth, J. & Abel, T. Internet information and medical consultations: Experiences from patients' and physicians' perspectives. *Patient Educ. Couns.* **77**, 266–271 (2009).
- Wiesner, M. & Pfeifer, D. Health recommender systems: Concepts, requirements, technical basics and challenges. *Int. J. Environ. Res. Public Health* **11**, 2580–2607 (2014).
- Wang, Y., Wang, M. & Xu, W. A Sentiment-Enhanced Hybrid Recommender System for Movie Recommendation: A Big Data Analytics Framework. *Wirel. Commun. Mob. Comput.* **2018**, 1–9 (2018).
- Nikzad-Khasmakhi, N., Balafar, M. A. & Reza Feizi-Derakhshi, M. The state-of-the-art in expert recommendation systems. *Eng. Appl. Artif. Intell.* **82**, 126–147 (2019).
- Mohammadi, V., Rahmani, A. M., Darwesh, A. M. & Sahafi, A. Trust-based recommendation systems in Internet of Things: a systematic literature review. *Human-centric Comput. Inf. Sci.* **9**, (2019).
- Broniatowski, D. A., Paul, M. J. & Dredze, M. National and local influenza surveillance through twitter: An analysis of the 2012–2013 influenza epidemic. *PLoS One* **8**, 1–8 (2013).
- Salathé, M. & Khandelwal, S. Assessing vaccination sentiments with online social media: Implications for infectious disease dynamics and control. *PLoS Comput. Biol.* **7**, (2011).
- Yang, C. *et al.* Personalized Mobile Application Discovery How to Find Apps. *Crowd Soft ACM* 49–54 (2014).
- Shi, K. & Ali, K. GetJar mobile application recommendations with very sparse datasets. *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 204–212 (2012). doi:10.1145/2339530.2339563
- Chen, N., Hoji, S. C. H., Li, S. & Xiao, X. SimApp: A framework for detecting similar mobile applications by online kernel learning. *WSDM 2015 - Proc. 8th ACM Int. Conf. Web Search Data Min.* 305–314 (2015). doi:10.1145/2684822.2685305
- Gao, M., Ling, B., Yuan, Q., Xiong, Q. & Yang, L. A Robust collaborative filtering approach based on user relationships for recommendation systems. *Math. Probl. Eng.* **2014**, (2014).
- Zanin, M., Romance, M., Moral, S. & Criado, R. Credit Card Fraud Detection through Parenclitic Network Analysis. *Complexity* **2018**, (2018).
- Wang, G., He, Y., Peng, Y. & Li, H. Bearing Fault Identification Method Based on Collaborative Filtering Recommendation Technology. *Shock Vib.* **2019**, (2019).
- Zhang, Y. *et al.* Explicit Factor Models for Explainable Recommendation based on Phrase-level Sentiment Analysis Categories and Subject Descriptors. *Proc. 37th Int. ACM SIGIR ...* 83–92 (2014).
- Devi, K. A., Edara, D. C., Sistla, V. P. K. & Kolli, V. K. K. Extended correlated principal component analysis with SVM-PUK in opinion mining. *Turkish J. Electr. Eng. Comput. Sci.* **26**, 2570–2582 (2018).
- Sokhin, T. & Butakov, N. Semi-automatic sentiment analysis based on topic modeling. *Procedia Comput. Sci.* **136**, 284–292 (2018).
- Pagano, D. & Maalej, W. User feedback in the appstore: An empirical study. *2013 21st IEEE Int. Requir. Eng. Conf. RE 2013 - Proc.* 125–134 (2013). doi:10.1109/RE.2013.6636712
- Lin, C., He, Y., Everson, R. & Rüter, S. Weakly supervised joint sentiment-topic detection from text. *IEEE Trans. Knowl. Data Eng.* **24**, 1134–1145 (2012).
- Zhang, W., Ding, G., Chen, L., Li, C. & Zhang, C. Generating virtual ratings from chinese reviews to augment online recommendations. *ACM Trans. Intell. Syst. Technol.* **4**, (2013).
- Cao, D. *et al.* Cross-platform app recommendation by jointly modeling ratings and texts. *ACM Trans. Inf. Syst.* **35**, (2017).
- McAuley, J. & Leskovec, J. Hidden factors and hidden topics: Understanding rating dimensions with review text. *RecSys 2013 - Proc. 7th ACM Conf. Recomm. Syst.* 165–172 (2013). doi:10.1145/2507157.2507163
- Liu, B. & Lauw, H. W. Detecting Product Review Spammers using Rating-lim.pdf. 939–948 (2010).
- Mukherjee, A., Liu, B. & Glance, N. Spotting fake reviewer groups in consumer reviews. *WWW'12 - Proc. 21st Annu. Conf. World Wide Web* 191–200 (2012). doi:10.1145/2187836.2187863
- Crawford, M., Khoshgoftaar, T. M., Prusa, J. D., Richter, A. N. & Al Najada, H. Survey of review spam detection using machine learning techniques. *J. Big Data* **2**, (2015).
- Wang, Y., Qian, L., Li, F. & Zhang, L. A Comparative Study on Shilling Detection Methods for Trustworthy Recommendations. *J. Syst. Sci. Syst. Eng.* **27**, 458–478 (2018).
- Hao, Y., Zhang, P. & Zhang, F. Multiview Ensemble Method for Detecting Shilling Attacks in Collaborative Recommender Systems. *Secur. Commun. Networks* **2018**, (2018).
- Rendle, S. Factorization machines. *Proc. - IEEE Int. Conf. Data Mining, ICDM 995–1000* (2010). doi:10.1109/ICDM.2010.127
- Georgiou, O. & Tsapatsoulis, N. Improving the scalability of recommender systems by clustering using genetic algorithms. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* **6352 LNCS**, 442–449 (2010).
- Zhou, Y., Wilkinson, D., Schreiber, R. & Pan, R. Large-Scale Parallel Collaborative Filtering for the Netflix Prize BT - Algorithmic Aspects in Information and Management. *Algorithmic Asp. Inf. Manag.* 337–348 (2008). doi:10.1007/978-3-540-68880-8_32
- Zhao, Z. D. & Shang, M. S. User-based collaborative-filtering recommendation algorithms on hadoop. *3rd Int. Conf. Knowl. Discov. Data Mining, WKDD 2010* 478–481 (2010). doi:10.1109/WKDD.2010.54
- Sun, J., Xu, W., Ma, J. & Sun, J. Leverage RAF to find domain experts on research social network services: A big data analytics methodology with MapReduce framework. *Int. J. Prod. Econ.* **165**, 185–193 (2015).
- Jiang, J., Lu, J., Zhang, G. & Long, G. Scaling-up item-based collaborative filtering recommendation algorithm based on Hadoop. *Proc. - 2011 IEEE World Congr. Serv. Serv. 2011* 490–497 (2011). doi:10.1109/SERVICES.2011.66
- Dahdouh, K., Dakkak, A., Oughdir, L. & Ibriz, A. Large-scale e-learning recommender system based on Spark and Hadoop. *J. Big Data* **6**, (2019).
- Panigrahi, S., Lenka, R. K. & Stitipragyan, A. A Hybrid Distributed Collaborative Filtering Recommender Engine Using Apache Spark. *Procedia Comput. Sci.* **83**, 1000–1006 (2016).
- Wijayanto, A. & Winarko, E. Implementation of multi-criteria collaborative filtering on cluster using Apache Spark. *Proc. - 2016 2nd Int. Conf. Sci. Technol. ICST 2016* 177–181 (2017). doi:10.1109/ICSTC.2016.7877370
- Yang, S., Korayem, M., AlJadda, K., Grainger, T. & Natarajan, S. Application of Statistical Relational Learning to Hybrid Recommendation Systems. (2016).
- Nguyen, A. T. *et al.* API code recommendation using statistical learning from fine-grained changes. *Proc. ACM SIGSOFT Symp. Found. Softw. Eng.* **13-18-Nove**, 511–522 (2016).
- Pichl, M., Zangerle, E. & Specht, G. Towards a Context-Aware Music Recommendation Approach: What is Hidden in the Playlist Name? *Proc. - 15th IEEE Int. Conf. Data Min. Work. ICDMW 2015* 1360–1365 (2016). doi:10.1109/ICDMW.2015.145
- Schedl, M., Zamani, H., Chen, C. W., Deldjoo, Y. & Elahi, M. Current challenges and visions in music recommender systems research. *Int. J. Multimed. Inf. Retr.* **7**, 95–116 (2018).
- Faridani, V., Jalali, M. & Jahan, M. V. Collaborative filtering-based recommender systems by effective trust. *Int. J. Data Sci. Anal.* **3**, 297–307 (2017).
- Yadav, S., Vikesh, Shreyam & Nagpal, S. An Improved Collaborative Filtering Based Recommender System using Bat Algorithm. *Procedia Comput. Sci.* **132**, 1795–1803 (2018).
- Wang, D., Liang, Y., Xu, D., Feng, X. & Guan, R. A content-based recommender system for computer science publications. *Knowledge-Based Syst.* **157**, 1–9 (2018).
- Yang, S., Korayem, M., AlJadda, K., Grainger, T. & Natarajan, S. Combining content-based and collaborative filtering for job recommendation system: A cost-sensitive Statistical Relational Learning approach. *Knowledge-Based Syst.* **136**, 37–45 (2017).
- Porter, M. F. An algorithm for suffix stripping. *Program* **14**, 130–137 (1980).
- Hu, F., Li, L., Zhang, Z. L., Wang, J. Y. & Xu, X. F. Emphasizing Essential Words for Sentiment Classification Based on Recurrent Neural Networks. *J. Comput. Sci. Technol.* **32**, 785–795 (2017).

47. Ren, W. & Han, K. Sentiment Detection of Web Users Using Probabilistic Latent Semantic Analysis. *J. Multimed.* **9**, 1194–1200 (2014).
48. Hofmann, T. Probabilistic Latent Semantic Analysis. (2013).
49. Koren, Y., Bell, R. & Volinsky, C. 09_matrix_factorization_Korean.pdf. 42–49 (2009). doi:10.1109/MC.2009.263
50. Salakhutdinov, R. & Mnih, A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. in *Proceedings of the 25th International Conference on Machine Learning (2008)* **25**, 880–887 (2008).
51. Parlar, T., Özel, S. A. & Song, F. QER: a new feature selection method for sentiment analysis. *Human-centric Comput. Inf. Sci.* **8**, 1–19 (2018).
52. Susanti, A. R., Djatna, T. & Kusuma, W. A. Twitter's Sentiment Analysis on Gsm Services using Multinomial Naïve Bayes. *TELKOMNIKA (Telecommunication Comput. Electron. Control.* **15**, 1354 (2017).
53. Kibriya, A. M., Frank, E., Pfahringer, B. & Holmes, G. Multinomial naive bayes for text categorization revisited. *Lect. Notes Artif. Intell. (Subseries Lect. Notes Comput. Sci.* **3339**, 488–499 (2004).
54. Catal, C. & Nangir, M. A sentiment classification model based on multiple classifiers. *Appl. Soft Comput. J.* **50**, 135–141 (2017).
55. Chang, C. C. & Lin, C. J. LIBSVM: A Library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2**, 1–39 (2011).
56. Abe, S. Analysis of multiclass support vector machines. *Int. Conf. Comput. Intell. Model. Control Autom.* **21**, 385–396 (2003).
57. Kijisirikul, B. & Ussivakul, N. Multiclass support vector machines using adaptive directed acyclic graph. *Proc. Int. Jt. Conf. Neural Networks* **1**, 980–985 (2002).
58. Mathur, A. & Foody, G. M. Multiclass and binary SVM classification: Implications for training and classification users. *IEEE Geosci. Remote Sens. Lett.* **5**, 241–245 (2008).
59. Souma, W., Vodenska, I. & Aoyama, H. Enhanced news sentiment analysis using deep learning methods. *J. Comput. Soc. Sci.* **2**, 33–46 (2019).
60. Kowsari, K., Heidarysafa, M., Brown, D. E., Meimandi, K. J. & Barnes, L. E. RMDL: Random multimodel deep learning for classification. *ACM Int. Conf. Proceeding Ser.* 19–28 (2018). doi:10.1145/3206098.3206111
61. Akhtar, M. S., Kumar, A., Ghosal, D., Ekbal, A. & Bhattacharyya, P. A Multilayer Perceptron based Ensemble Technique for Fine-grained Financial Sentiment Analysis. 540–546 (2018). doi:10.18653/v1/d17-1057
62. Soutner, D. Application of LSTM Neural Networks in Language Modelling.
63. Choudhary, N., Singh, R., Bindlish, I. & Shrivastava, M. Emotions are Universal: Learning Sentiment Based Representations of Resource-Poor Languages using Siamese Networks. *ArXiv e-prints* (2018).
64. Zaremba, W., Sutskever, I. & Vinyals, O. Recurrent Neural Network Regularization. 1–8 (2014). doi:ng
65. Sahu, S. K. & Anand, A. Drug-drug interaction extraction from biomedical texts using long short-term memory network. *J. Biomed. Inform.* **86**, 15–24 (2018).
66. Barry, J. Sentiment analysis of online reviews using bag-of-words and LSTM approaches. *CEUR Workshop Proc.* **2086**, 272–274 (2017).
67. Chowdary, D., Lakshmi, E., Vanukuri, P. & Sistla, V. Sentiment analysis and text categorization of cancer medical records with LSTM. *J. Ambient Intell. Humaniz. Comput.* (2019). doi:10.1007/s12652-019-01399-8