International Journal of Aquatic Science
ISSN: 2008-8019
Vol 12, Issue 02, 2021

Mobile App Categorization Based On App Descriptions And Api Calls

Priya Kalaivani K[1], Arulanand N[2]

[1]*Research Scholar, Department of CSE, PSG College of Technology, Coimbatore, Tamil Nadu, India 641004.*
[2]*Professor, Department of CSE, PSG College of Technology, Coimbatore, Tamil Nadu, India 641004.*

ABSTRACT: Mobile applications in the app store have increased rapidly over the years and with the increasing popularity, the mobile app developers resist getting visibility for their product. An important factor that influences the visibility of an app is how it gets categorized in the app market. A study was made to identify misclassified apps and categorize them to help app users. To uncover the misclassified mobile apps in the app, store a new approach to categorize the related apps together based on their description and API calls has been proposed. A dataset containing 25,000+ mobile apps mined from the Google Play Store were used. The initial step involves grouping the applications into various categories based on the technical description of the mobile applications. Pre-processing of descriptions was done using natural language processing techniques and feature extraction using Latent Dirichlet Allocation (LDA). The work can be split into two halves. Work I focus on clustering which was again carried out in two methods by varying the parameters, Model A was using the features extracted from app descriptions as a parameter, and Model B was using the features extracted from descriptions and API calls as parameters. K-Mean clustering was used as a clustering technique due to its hard-clustering nature. Both the clustered outputs were evaluated and an efficient one was identified. Work II focuses on classifying the app based on app description. Popular machine learning and deep learning models were used for classification and a comparative study was made.

Keywords: Latent Dirichlet Allocation, K-Mean Clustering, Machine learning, Deep learning, Mobile Application

1. INTRODUCTION

App market ecosystem is a place where the mobile application (app) developers can host their apps for public visibility. Statistics show that there are nearly 2 million apps each in Google play store and Apple app store. These mobile apps are used in our regular day-to-day life. At the time of app release, the developer needs to specify what they consider as the most appropriate category for their app to be present in. Frequently the play store needs to be refined as there may be the possibility of apps getting miscategorized. Mis-categorization is one of the important problems that need to be addressed by the app market.
Some of the implications of mis categorization are:

1. It damages the truthfulness of existing categories.
2. It allows some app developers to get an unjust pro over others.
3. It makes auditing and ensuring quality or regulatory control more difficult.

4. It might give the wrong impression to users and tempt them to pay for apps that do not provide the expected utilities.

Thus, it is imperative to have a strong categorization and misclassification recognition system in the app market to guard users and maintain a healthy competitive ecosystem.

In this paper, the app descriptions and API calls provided by the app were used for categorization. The topic modeling technique was used to extract the features from the app description. Topic Modeling extracts the topics for a given corpus (set of documents) based on the keywords present in the entire corpus. In this proposed work, Latent Dirichlet Allocation (LDA) [5] which extracts topics that correspond to a probability distribution over words has been used as a topic modeling technique. K-means document clustering was carried out with two different sets of parameters and the evaluation results of both the clusters were analyzed. The apps were also categorized based on classification algorithms in machine learning and deep learning. The efficient classification model with higher accuracy was identified for the given dataset.

2. RELATED WORK

[1]. Alessandra Gorla et al (2014) in this paper classify the apps based on the descriptions and topics extracted. Clustering is carried out based on topics of the descriptions, and then outliers in each cluster concerning their API usage are identified. The techniques used in this paper are Latent Dirichlet Allocation for topic modeling, Document clustering is accomplished on the app descriptions using K-Mean document clustering, and then app descriptions are checked against the API usage in the app implementation by dissembling the application code. One-Class Support Vector Machine is the anomaly Classification algorithm used.

[2]. A. A. Al-Subaihin et al (2016) in this paper raw description of the mobile application from the Google Play Store and BlackBerry store is used. Feature extraction is done using N-gram model. Feature clustering was done and represented based on the App-Feature Matrix (AFM) where the dimensionality was reduced to Feature-Term Matrix (FTM). Features were represented using the Term Frequency – Inverse Document Frequency (TF-IDF). Clustering of Application will be done based on its similarity obtained from the Cosine Similarity measurement. Finally, the Agglomerative Clustering technique was done in conjunction with Cosine Similarity as a distance measure.

[3]. Babatunde Olabenjo et al (2016) in this paper two variations of the Naive Bayes classifier using open metadata from top developer apps on Google Play Store are built to classify new apps on the store. These classifiers were evaluated using various evaluation methods and their results were compared against each other. This paper helps to understand the Machine Learning categories which are classified as three broad types Supervised Learning, Unsupervised Learning, and Reinforcement Learning. Two major Naive Bayes algorithms i.e., Multinomial and Bernoulli Naive Bayes classifiers were explained in detail in this paper with the demonstration.

[4]. Ruizhang Huang et al (2014) in this paper a study was made on Latent Dirichlet Allocation. The proposed approach's performance was explored on a synthetic and realistic document dataset. Using this LDA document clustering was made on labeled and unlabeled instances. Based on this clustering using LDA the quality of the clusters was identified and found that when inserted with supervised information to the LDA model, the positive impact of labels was reinforced. A comparison on both the datasets was done and their efficacies

were analyzed. Complete mathematical logic behind the Latent Dirichlet Allocation has been explained apparently in this paper.

[5]. David M. Blei et al in this paper a complete study on Latent Dirichlet Allocation was explained elaboratively. Interior processing of the LDA model was explained with understandable architecture for explaining the flow of the model. The Relationship of LDA with other latent variable models for texts like the unigram model, a mixture of unigrams, and the Probabilistic latent semantic indexing (pLSI) model. A geometric interpretation of all the models was done. All the Inference and parameters of the LDA model were estimated. The document modeling was done using LDA to achieve high likelihood and perplexity was computed to evaluate the model.

[6]. Chengpeng Zhang et al discussed the way of identifying malware apps based on the Third-Party Libraries whereas various researches were considering only the app descriptions and app behaviors for identifying the malicious apps. The impact of Third-Party Libraries was also removed to pinpoint the malicious behavior of custom code.

[7]. Siqi Ma et al proposed an active semi-supervised approach for detecting malware. Both benign and malicious apps were made use of to predict other future malicious apps. An active approach was achieved by labeling the apps as benign or malicious. The Labeled and unlabeled data were used for training the classification model. Description and API were pre-processed separately based on these features were extracted and the classification algorithm is run on the extracted features.

3. PROPOSED METHODOLOGY

The proposed system shown below in Figure 1 was constructed to analyze the set of mobile application descriptions and extract inherent knowledge about the categories of each application.

The algorithms used in the implementation include Latent Dirichlet Allocation for extracting the topics from the descriptions, K-Mean Document Clustering, and various Machine Learning and Deep Learning models for Document Classification to categorize the apps based on its descriptions.

DATASET DESCRIPTION

Descriptions of various android apps were extracted from Google Play Store. The app descriptions and API calls were the features considered for the work. These data collected contains app descriptions in different languages like French, German, etc.., but in the proposed work the app descriptions in the English language alone were considered for the next step. The API calls collected consist of around 860 including the permission used by the mobile apps. The dataset consists of around 33,000 app descriptions which have been reduced to 25,799 descriptions.

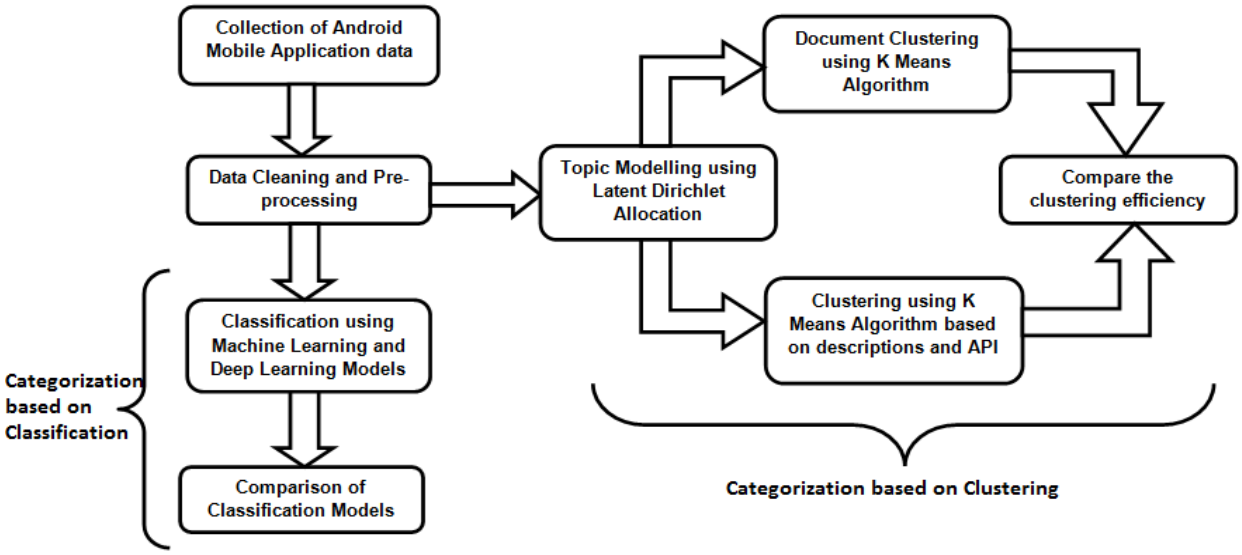

Figure 1: Work Flow Diagram

These 25,799 app descriptions were given as input to the LDA model. The dataset in CSV format was used in the proposed work.

DATA CLEANING AND PREPROCESSING

Cleaning is the most vital step in any of the mining tasks. The app store contained apps with descriptions in the English language as well as in other languages. The English language was considered in the proposed work. To remove other languages from the data collected, a script was implemented and executed with the dataset.

The next step involved extracting topics from the given corpus and hence the app descriptions were pre-processed to make them more meaningful. All the stop words like articles, prepositions, conjunctions, etc.., were removed from the entire corpus. Stemming was done to make the description simple and easy. In the stemming process, each word in the description was trimmed to its stem form by removing the suffix. Lemmatization was done to take into consideration the morphological analysis of each word. From this, the base form of each word was considered as the output of the pre-processing process. Finally, a normalized corpus was obtained for further processing[11].

LATENT DIRICHLET ALLOCATION (LDA) ALGORITHM

Latent Dirichlet Allocation is one of the most popular Topic Modeling techniques. Topic Modeling is one type of statistical modeling for discovering the abstract topics that occur in a corpus [10]. Initially, the LDA model assumes each document to be belonging to a mixture of topics. Based on the term's probability distribution topics were generated. When the LDA model is given with the corpus, it backtracks and finds out the topics that make the particular document more priority. LDA is a kind of Matrix Factorization technique. Any corpus can only be represented as a document-term matrix in the vector space for easy evaluations. LDA model internally converts this document-term matrix into lower-dimensional matrices namely document-topic matrix and topic-term matrix [9]. From these matrices' topic, word, and document topic distributions were acquired. The main goal of LDA is to improve the distribution of the above matrices.

For each topic, two probabilities p1 and p2 were calculated.Where,

- p1 – p (topic/document) = frequency of terms in a particular document that is currently assigned to a particular topic.
- p2 – p (term/topic) = proportion of the allocation to a particular topic over all the documents that arise from a particular word.

The LDA model attains a steady state when the distributions of both the matrices are reasonably good and they were considered as the convergence point of the model.

K-MEANS CLUSTERING ALGORITHM

K-Means is one of the most popular Unsupervised Machine Learning Algorithms. The main reason for using the K-Means algorithm is due to its hard clustering nature and simplicity. The proposed work clustering was carried out in two ways:

Model A - K means the algorithm works only with a numeric value. The input app descriptions were converted into numbers. Instead of converting each word in the description to numbers, the topics extracted from the entire corpus were converted into numerical values. After execution of the LDA model, Document Topic Matrix was generated and that has to be inputted to K-means algorithm for clustering the app descriptions based on their similarity. The output of this K-mean algorithm was clustered that contain descriptions belonging to the same categories.

Model B – Clustering was done by providing two parameters that were App Descriptions and API calls used by every app considered for the proposed work. The Document Topic Matrix generated by the LDA model was combined with the API calls of each app and the combined data frame was provided as input to the K-Means clustering algorithm. Based on the similarity score the apps were clustered and the final clustered data was stored for processing.

A comparison of both models was done to identify the efficient way of clustering. The number of clusters was unable to predict at the initial stage so for predicting the number of clusters there are various methods like the elbow method and silhouette coefficient available. As there are 30 different categories in Google play store the same value was considered extracting the topics from the descriptions and clustering.

DOCUMENT CLASSIFICATION

Document Classification was carried out using various Machine Learning and Deep Learning models. Machine Learning is similar to that of Data Mining and Predictive Modeling. The Machine Learning algorithms are generally categorized into two Supervised and Unsupervised. Supervised algorithms involve a data analyst with machine learning skills to provide both input and preferred output, in addition to furnishing feedback about the accuracy of predictions during algorithm training[8]. Unsupervised algorithms do not require to be trained with desired outcome data. Deep Learning is a subfield of machine learning alarmed with algorithms enthused by the structure and function of the brain called artificial neural networks. Deep learning programming can create multifaceted statistical models directly from its iterative output and it can create accurate analytical models from a large quantity of unlabeled data. One of the main reasons for using deep learning algorithms over traditional algorithms was that they provide good accuracy value for a large amount of data.
The models used for the work were,

- Stochastic Gradient Descent (SGD)
- Convolution Neural Network (CNN)
- Long-Short Term Memory (LSTM)
- Gated Recurrent Unit (GRU)

Based on the output of the classification model the incorrect predictions can be identified by manual inspection of the data.

STOCHASTIC GRADIENT DESCENT (SGD)

SGD is widely regarded as one of the best text classification algorithms. The pre-processed descriptions along with the categories were converted into vectors and provided as input to the SGD model for classification. It is similar to SVM but it treats the data in batches and performs a gradient descent to minimize the expected loss concerning the sample distribution.

CONVOLUTION NEURAL NETWORK (CNN)

CNN is most commonly used in Image Classification and Image Prediction but here it was used for Document Classification. In Image Classification using CNN, images will be converted into a pixel vector matrix and given as input to the convolution layer. Likewise, In Document Classification the encoded app descriptions and embedding vectors considered were converted into a matrix and provided as input to the convolution layer.

LONG SHORT-TERM MEMORY (LSTM)

It is an extension of the Recurrent Neural Network which enables one to remember its input over a long period. LSTM contains three gates to regulate the flow of information. Those gates are the input gate, forget gate, and output gate. LSTM makes it easier for inputs to be repeated without much alteration.

GATED RECURRENT UNIT (GRU)

A Gated Recurrent Unit is another form of Recurrent Neural Network. Instead of the LSTM layer, it was changed with the GRU layer. It also has two gates, a reset gate, and an update gate. This GRU contains fewer tensor operations so it was a little speedier to train than LSTM. For all these document classification models the labeled output from the dataset was given as input. The input shape of these neural networks was provided based on the binning concept which counts the number of words in the corpus and finds the approximate maximum size of the input text.

4. EXPERIMENT RESULTS

K-MEANS CLUSTERING RESULT ANALYSIS

The common behaviors of the mobile applications were compared based on the words in the app descriptions. The similarities of the documents were considered based on the probability provided in the matrix that was inputted to the K-Means clustering algorithm. The work comes under unsupervised learning as the features from the descriptions were not known in advance. All the application that has been considered was non-payable apps. The general benefits of all the models were established in the result.

Cluster analysis was manually done by comparing the clustered data from model A and model B. A python script was implemented for comparing the clustered data with already existing categories by identifying the maximum intersection in each cluster of apps.

CLASSIFICATION RESULT ANALYSIS

MACHINE LEARNING MODEL ACCURACY COMPARISON

The accuracy of the Stochastic Gradient Descent model was found to be 66% on the mobile application dataset. The accuracy across all the categories was illustrated using a confusion matrix. The diagonal represents accurate matches. This confusion matrix will evaluate the classification based on the precision, recall, and F1-scores as displayed in Figure 2 and 3.

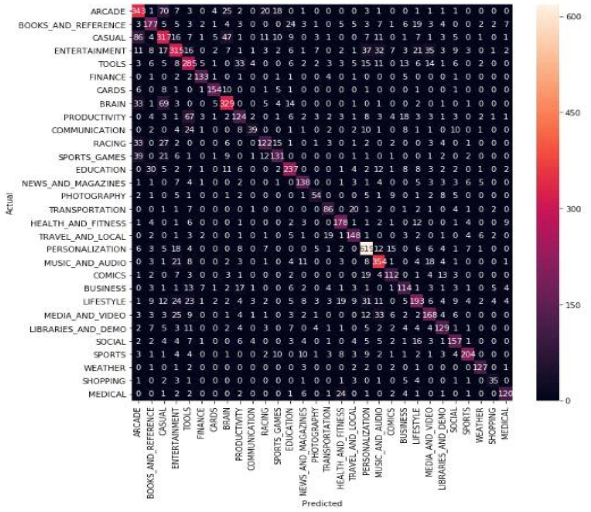

Figure 2: Confusion Matrix for SGD Model

accuracy 0.662673244068593

	precision	recall	f1-score	support
ARCADE	0.59	0.68	0.63	502
BOOKS_AND_REFERENCE	0.64	0.60	0.62	295
CASUAL	0.54	0.57	0.56	555
ENTERTAINMENT	0.63	0.57	0.60	553
TOOLS	0.54	0.65	0.59	439
FINANCE	0.84	0.87	0.85	153
CARDS	0.87	0.81	0.84	190
BRAIN	0.71	0.70	0.71	469
PRODUCTIVITY	0.53	0.46	0.49	272
COMMUNICATION	0.68	0.34	0.45	116
RACING	0.64	0.55	0.59	223
SPORTS_GAMES	0.66	0.57	0.61	229
EDUCATION	0.69	0.68	0.69	347
NEWS_AND_MAGAZINES	0.69	0.73	0.71	189
PHOTOGRAPHY	0.64	0.55	0.59	98
TRANSPORTATION	0.64	0.66	0.65	130
HEALTH_AND_FITNESS	0.70	0.78	0.74	227
TRAVEL_AND_LOCAL	0.68	0.73	0.70	203
PERSONALIZATION	0.76	0.85	0.80	730
MUSIC_AND_AUDIO	0.69	0.79	0.74	448
COMICS	0.66	0.64	0.65	175
BUSINESS	0.56	0.58	0.57	196
LIFESTYLE	0.59	0.48	0.53	403
MEDIA_AND_VIDEO	0.59	0.58	0.59	289
LIBRARIES_AND_DEMO	0.65	0.64	0.65	202
SOCIAL	0.66	0.67	0.67	233
SPORTS	0.84	0.73	0.78	278
WEATHER	0.85	0.89	0.87	142
SHOPPING	0.67	0.59	0.63	59
MEDICAL	0.79	0.71	0.75	169
avg / total	0.66	0.66	0.66	8514

Figure 3: Precision, Recall, and F1-score for SGD model

ACCURACY COMPARISON OF CLASSIFICATION MODELS

The Accuracy and Loss value obtained from the classification models is visualized from Figure (4-9),

Table 1: Accuracy Comparison

The performance graph of various deep learning models was drawn to clearly understand the accuracy of the model. The red line in the graph represents the training model and the blue line indicates the testing model. From the Table 1, it is understood that SGD classification models work better with 66% accuracy and the hyper-parameters were tuned for improvements.

Model Used	Accuracy
Convolution Neural Network	**59%**
Long Short-Term Memory	**51%**
Gated Recurrent Unit	**52%**
Stochastic Gradient Descent	**66%**

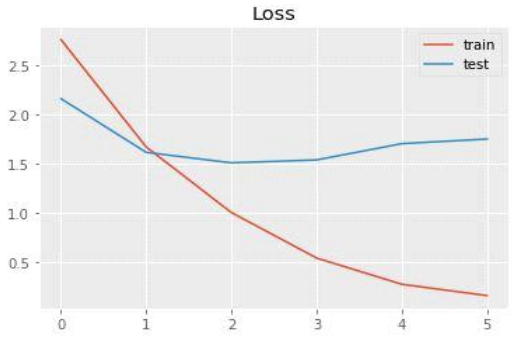

Figure 4: Loss score for Convolution Neural Networks

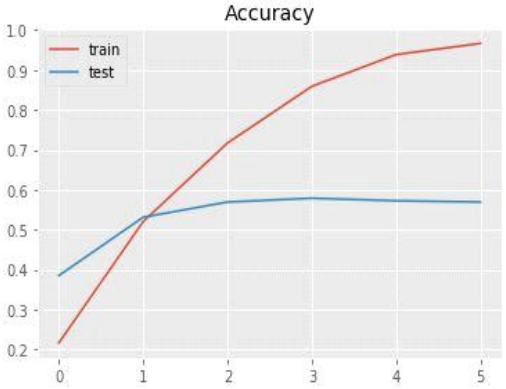

Figure 5: Accuracy score for Convolution Neural Networks

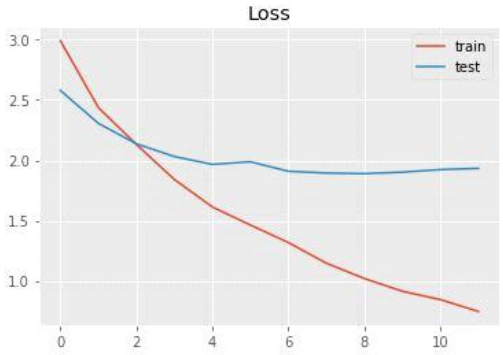

Figure 6: Loss score for Long Short-Term Memory

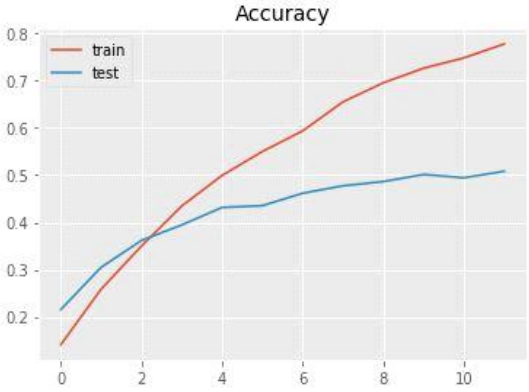

Figure 7: Accuracy score for Long Short-Term Memory

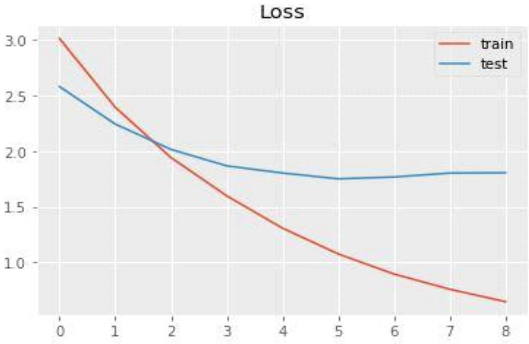

Figure 8: Loss score for Gated Recurrent Unit

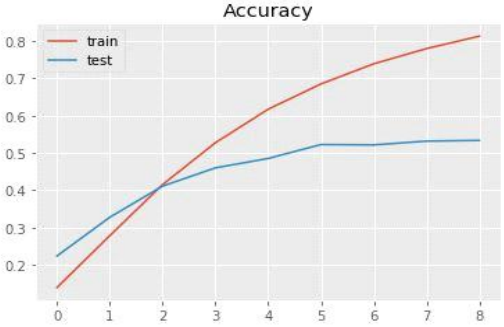

Figure 9: Accuracy score for Gated Recurrent Unit

5. CONCLUSION AND FUTURE WORK

By categorizing the mobile applications based on their description, the applications with common characteristics were found out and grouped under a single cluster this solves the problem that has been produced during the refinement process of the Play Store. This system was efficient in categorizing misclassified apps. As there is a huge business market for mobile applications it has to be categorized for various purposes both for the benefit of users and developers. This work can be used for evaluating the similarity of the applications in a single group or a single category. The topics that were extracted initially have influenced the proposed work to a great level. Using various classification models has also improved the result of categorization.

The result analysis of the work shows the performance of various classification algorithms and a comparative study of all these algorithms was made. It was found that the Stochastic Gradient Descent model performed comparatively better than other models with higher accuracy of 66%.

In the future, the source code of each app can be extracted to examine the behavior of each app against the features described in the app descriptions and permission usage. Malicious apps under each category can be identified and removed. The work can also be extended for IOS apps.

6. ACKNOWLEDGMENT

We thank Dr. Meiyappan Nagappan, Assistant Professor, and his student Lakshmanan Arumugam, David R. Cheriton School of Computer Science, University of Waterloo, Ontario, Canada for helpful discussions and constant support in carrying out our work.

7. REFERENCES

[1] Alessandra Gorla, Ilaria Tavecchia, Florian Gross and Andreas Zeller, “Checking App Behavior Against App Descriptions”, International Conference on Software Engineering, 2014.

[2] Al-Subaihin, F. Sarro, S. Black and L. Capra, M. Harman, Y. Jia, and Y. Zhang CREST, “Clustering Mobile Apps Based on Mined Textual Features”, Department of Computer Science, University College London, UK, 2016.

[3] Babatunde Olabenjo, “Applying Naive Bayes Classification to Google Play Apps Categorization”, Department of Computer Science, University of Saskatchewan, Saskatoon, SK, Canada, 2016.

[4] Ruizhang Huang, Ping Zhou, and Li Zhang, “A LDA-Based Approach for Semi-Supervised Document Clustering”, International Journal of Machine Learning and Computing, 2014

[5] David M. Blei, Andrew Y. Ng, and Michael I. Jordan, “Latent Dirichlet Allocation”, University of California, Berkeley.

[6] Chengpeng Zhang, Haoyu Wang, Ran Wang, Yao Guo, and Guoai Xu, “Re-checking App Behavior against App Description in the Context of Third-party Libraries”, Beijing University of Posts and Telecommunications, Beijing, China.

[7] Siqi Ma, Shaowei Wang, David Lo, Robert Huijie Deng and Cong Sun, "Active Semi-Supervised Approach for Checking App Behavior Against Its Description", School of Computer Science and Technology, Xidian University.

[8] Swapnil Hingmire, Sandeep Chougule, Girish K. Palshikar, and SutanuChakraborti, "Document Classification by Topic Labeling", 2013.

[9] P. Anupriya and S. Karpagavalli, "LDA Based Topic Modeling of Journal Abstracts", International Conference on Advanced Computing and Communication Systems, 2015

[10] Padmaja CH V R, S Lakshmi Narayana, and Divakar CH "Probabilistic Topic Modelling and Its Variants – A Survey", International Journal of Advanced Research in Computer Science, Visakhapatnam, AP, India, 2018.

[11] Jain, M.; K, V. Paris Attack in Wireless Ad Hoc Network. IARS' International Research Journal, Vic. Australia, v. 1, n. 1, 2011. DOI: 10.51611/iars.irj.v1i1.2011.2.

[12] BEUNG-HOON, "Two-Stage Document Length Normalization for Information Retrieval", Busan University of Foreign Studies, South Korea, 2015.