International Journal of Advance Research in Science and Engineering
Volume No.06, Issue No. 12, December 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

Mobile Application Analysis and Classification Using Data Mining -A Survey

J. Jeyaboopathiraja[1], Dr.G. Maria Priscilla[2]

[1]*Research Scholar, Sri Ramakrishna College of Arts and Science, Coimbatore, Tamilnadu, (India)*

[2]*Professor and Head, Sri Ramakrishna College of Arts and Science, Coimbatore, Tamilnadu, (India)*

ABSTRACT

Data mining plays a great role in the current scenario, where every application and domain need the support of data mining. The most popular application of data mining is web mining, mining web contents and analyses teem for a knowledge discovery is the main task of web mining. In this paper, we surveyed several tools and techniques used to extract and analyse data from several websites and pages. In specific, we analysed the techniques, which are available to classify the Facebook and android applications from the web. Web app classification is the task of grouping the application based on its features and functionalities. This process enables the application to find duplicate, redundant, related mobile applications from a huge set of app dataset. The classification involves in data classification based on complete training samples. It also plays an important role in information retrieval and extracting useful information from huge amount of mobile applications. There are n numbers of algorithms and techniques are available for feature extraction and classification. This survey explores the popular mobile application classification as well as the general feature based classification techniques along with its pros and cons.

Keywords: ***Data Mining, Text Mining, Classification, Pattern discovery, Mobile app classification.***

I.INTRODUCTION

In the recent scenario all tasks and functionalities are rehabilitated into smart phone. Every user owns a smart phone and performs all tasks using that. These processes are performed using several mobile applications, which are available in the website for free of cost [1]. So classifying and analysing such mobile application is very important. Machine learning techniques for App Classification aim at extracting those features which are helpful in identifying the relevancy of an application to each category. The following two broad Machine Learning techniques are usually employed.

A. Supervised Learning: Supervised learning is a technique to guide the process of classification by providing prior information such as labeled applications and features to train the classifier. The input app data is parted into the two sets for these phases: the Training set that comprises labeled applications and the Testing set that comprises applications to be classified. At the time of training process the classifier learns by categorizing mobile applications to the appropriate categories by using labeled information in the Training samples. retaining thus tuned its decision parameters, it enters the Testing process, when the classifier assigns categories to the applications of testing set by utilizing its previously acquired knowledge. Some popular generic methods of

supervised ML classifiers are Naïve Bayes (NB), Decision Trees (DT), Support Vector Machines (SVM) and Neural Networks (NN) [2].

B. Unsupervised Learning: Here the classifier does not have any labeled information of applications for learning. K-Means clustering is the most popular technique for unsupervised technique. It classifies a given dataset by initializing a pre-decided number of clusters with seed data and then assigning the remaining data instances to one of them by using a suitable distance metric to calculate their similarity to each of the clusters.

C. Semi-Supervised Learning: the third and customized part of information labeling is semi-supervised learning, which doesn't need complete training. Several data mining applications recently follow the third type of learning framework with active learning paradigm.

Mobile App features are semi structured in nature, i.e., it is neither completely un-structured nor completely structured. A feature may include a few structured data's about the mobile app, such as title, provider and other basic details about the application etc., and it also contains some huge unstructured textual descriptions. In recent research on Mobile App feature analysis and classification, several studies have been done to build and develop semi-structured data. To handle the un-structured features, text indexing and IR (Information Retrieval) techniques have been developed. But those IR and other traditional techniques are not sufficient to handle vast amount of data [2]. This paper discusses the various techniques and tools have been used for malicious and duplication app classification and general feature classification algorithms. And finally provides an outline about the problems of the existing works. This survey also handles the important process of feature extraction and mobile app classification, which is known as feature discovery process. This relevance feature discovery process helps to identify the useful features available in the Mobile App descriptions at the time of training [3].

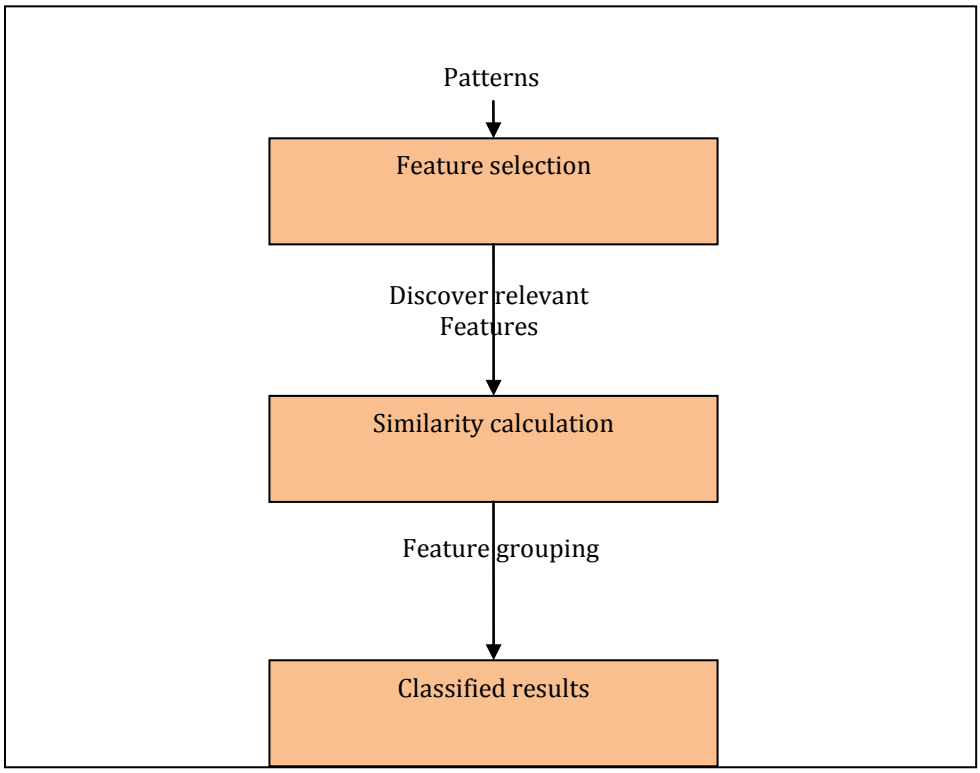

Fig 1.0 Feature discovery and file classification process

The above fig 1.0 represents the basic process involved in App Classification. In order to group the application based on the features, this is important to find the relevant and appropriate features from its descriptions and functionalities. The above process shows the feature selection and feature based application similarity calculation and based on the similarity score, the applications are grouped together.

II.LITERATURE REVIEW

At the present time, people are using smart phones for different purposes, so different and numerous mobile apps are available in the internet. Handling the hefty amount of apps in the internet, the task of how to define their similarities, relationship becomes more and more useful. By performing the similarity calculation, app retrieval and app recommendation are easy to be performed.

Authors in [3] presented a primary study to calculate and analyse spam campaigns launched on Online Social Networks (OSN). The authors have an enormous anonymized dataset of asynchronous Facebook wall messages collected from several Facebook users. The system has identified 200000 malicious wall posts with embedded URLs in the wall message, which initiated from more than 50 thousand user accounts. Authors found that more than 72% of all malicious wall posts advertise phishing sites. To study the distinctiveness of malicious accounts, and see that more than 97% are compromised accounts, rather than "fake" accounts formed solely for the principle of spamming.

Third-party applications capture the attractiveness of web and platforms providing mobile application. Many of these platforms accept a decentralized control strategy, relying on explicit user consent for yielding permissions that the apps demand [4]. Users have to rely principally on community ratings as the signals to classify the potentially unsafe and inappropriate apps even though community ratings classically reflect opinions regarding supposed functionality or performance rather than concerning risks. The advantages of user-consent permission systems are examined through a large data collection of Facebook apps, Chrome extensions and Android apps. The study confirms that the current forms of community ratings used in app markets today are not reliable for indicating privacy risks. This study confirmed with some proofs, which indicating the attempts to mislead or entice users for yielding permissions. This also grants permission for free applications with mature content requests and for "look alike" applications which have similar names as that of popular applications also request more permissions than is typical. Authors find that across all three platforms popular applications request more permissions than average. OSNs such as Orkut, Face book and others have grown-up rapidly, with hundreds to millions of active users. A new feature provided on several sites is social applications and services written by third party developers that supply additional functionality linked to a user's profile [5]. However, present application platforms put users at risk, because it contains numerous applications and lots of predefined recommendations. This may confuse the user to select a best app without malicious and malignant behaviour.

In this paper [6], authors proposed a personalized recommendation system for mobile application software (app) to mobile user using semantic relations of apps. To achieve this, semantic relations between apps consumed by a specific member and his/her social members are analysed using Ontology framework. With the help of app associations, this identifies the most similar social members from the analysis method. The analysis is discovered from measuring the common features between apps used by the specific set of members. The more

features shared by the users, the more similar is their preference for consuming apps. This also developed a prototype of the system using OWL (Ontology Web Language) by defining ontology-based semantic relations among 50 mobile apps in the internet. With the proposed prototype, the authors demonstrated the feasibility of the recommendation algorithm.

In this paper [7], Chen, Lei, Chong Wu, and Yilan Dai proposed an iterative process for app relationship detection. It combines the review similarity and app similarity together. So the authors used the iterative process to dig deep relationship among apps via reviews and feedbacks. It finds the given two apps are similar in their objective or not. This iterative process has two ways to run and only needs to set one initial parameter. By this iterative procedure, deep relationship among apps and topic similarity among reviews can be both attained. But, there are many kinds of relationship among apps which are very tough to identify. So the authors have concluded, that the iterative process can only detect there is relationship and gives its value, however, cannot tell which type of relationship it belongs to. The authors left some app relationship classification process as future process.

In [8] Kim, Junhyoung, Tae Guen Kim, and Eul Gyu Im proposed a method for Android malware similarity and clustering process. The structural information that is extracted from methods in given applications is compared to match the similar apps in the targeted application with various factors, and the number of matched methods and the total number of methods are used for similarity calculation. All the structured information's are used for the comparison process. This used DBSCAN algorithm for clustering. It also suggested clustering mechanism that can provide some feedbacks about the malware apps to the others.

As a summarized view, the usual way to calculate app relationship and similarity is to extract attributes from app's description and other sources to represent apps similarity measurements has several challenging tasks. From the literature, we can summarize that there only few researches made for App similarity and malicious detection problems using data mining. The above methods have the following limitations.

- One is that it can only detect shallow similarity between apps. So it doesn't give deep variations
- Some useful information is unaware and insensible, such as the viewpoint in user's review.

For this reason, this paper proposes a novel approach to combine app relationship calculation and review similarity calculation together for duplicate app and malicious app detection. The proposed work can calculate app similarity accurately. It can find more general relationships between apps, which are collected from Google play store. Additionally, several improvements are made on this proposal. One is obtaining high-quality results and the other is to replace calculation by matrix product to reduce time. And finally the malicious and similar apps are filtered at the time of recommendation.

Due to the significance and essential of detecting and suspending Twitter spammers, many researchers along with the engineers in Twitter Corporation have devoted themselves to keeping Twitter as spam-free online communities [9]. In this paper, the authors made an empirical analysis of the evasion tactics utilized by Twitter spammers, and then design several new and robust features to detect Twitter spammers on most popular and important Internet sites, which are undertaken by the attacker like phishers, fraudsters, and spammers [10]. The main intension of such attacker is to hack the user data and render them with spam content. The attackers have vast resources at their disposal. They are well-funded, with full-time skilled resources, control over

compromised and infected accounts, and access to global botnets. Protecting the users is a challenging adversarial learning problem with extreme scale and load requirements. Over the past several years they have built and deployed a coherent, scalable, and extensible real-time system to protect the users and the social graph. This Immune System achieved the real-time checks and classifications on every action of the application such as application access, read and written contents.

In [11] authors proposed phrase-based content similarity, which helps to compute the pair wise relationship between multiple text contents. The work by the authors is based on the Suffix Tree Content model. In general, phrase in a contents have been considered as a more informative feature. And it improves the effectiveness of text content clustering. The STD model is a phrase based approach, which inherits the tf-idf weighting scheme. The authors have applied group average Hierarchical agglomerative clustering algorithm to develop a new clustering algorithm. The experiments are conducted in RCV1 and corpora datasets. The authors proved that the phrase based content similarity works better than the single word tf-idf measure. The new phrase based content similarity successfully connects the two content models and inherits their advantages. The concept of STD is simple and effective, but the implementation is become very difficult and tough. The STD structure is used n-gram method to identify and extract phrases from the contents.

In[12] authors proposed a fuzzy similarity-based self-constructing algorithm for feature clustering, because the feature clustering is the most powerful tool for text classification. Based on the words, the contents are grouped in the same cluster. The proposed FFC (Fuzzy Feature Clustering) is an incremental clustering approach to reduce the dimensionality of the features in text classification. Here statistical mean deviation has been used. While grouping the texts based on the words, outliers are considered as new clusters. The similarity measure is calculated using the variance. The authors show the FFC is a feature reduction technique, which facilitates the fast clustering process. The FFC method is only good for text categorization problems due to the suitability of the distributional word clustering concept.

In[13] proposed a new spectral clustering method called CPI, which is abbreviated as correlation preserving indexing. This CPI method is performed in the correlation similarity measure space. The authors are much concentrated on the intrinsic geometrical structure of the content. The work is not sufficient to apply on the huge datasets.

Later **In[14]** developed a two phase cross domain method for text classification. Particularly, a CD-PLSA (Collaborative Dual Probabilistic Latent Semantic Analysis) model is first learned to effectively capture the distinction and commonality from corner to corner numerous domains in a collaborative nature. In that case, the authors further mined the intrinsic composition of desired domains by refining the outputs from CD-PLSA, which also called RCD-PLSA. The work by [14] is based on EM (expectation Maximization) algorithm, and provided various training samples for almost all domains. But the paper suffers from accuracy issues.

In[15] proposed supervised term weighting method, i.e., tf:rf, to improve the terms' discriminating power for text categorization task. This paper utilizes vector space model for text representation, this transforms the content of a text content into a vector. In this study, the authors investigated numerous unsupervised and supervised term weighting methods with SVM and kNN algorithms. Finally the paper shows the supervised term weighting methods are good in performance. The implementation of the tf:rf in text summarization and IR

are left for future work. And the proposal has been experimented in a sample synthetic dataset and the proposal is not efficient for huge dataset like RCV1 benchmark corpus. This doesn't have the ability to handle huge text collections.

In paper [16] authors have proposed a new concept-based mining model, which analyses terms on the sentences. This work bridges the gap between NLP and text mining regulations. A new concept based mining model composed of four components. The authors proposed a method to enhance the text-clustering eminence by exploiting the semantic structure of the sentences in contents. A better text clustering result is achieved by using the above concept based mining model. The first one is sentence based concept analysis, content based concept analysis, corpus level concept analysis and finally concept based similarity detection. The authors used tf method for all the above analysis. This has the ability to calculate pair-wise content similarity accurately. It is very robust and accurate. However, the work affected by several issues, such as this work only handles homogeneous text contents and tough for real time implementation.

In the paper [17] authors presented a novel fuzzy clustering algorithm that operates on relational input data. The relational input data are such as in the form of a square matrix of pair-wise similarities between data contents. The proposed algorithms utilize the graph representation and operate in the EM algorithms as like the previous paper. The likelihood function is created using the graph centrality. In the paper, the concepts present in natural language contents (NLD) usually display some type of hierarchical structures, while the algorithm presented in this paper identifies only flat clusters rather than the hierarchical. So it doesn't support hierarchical structure.

In the paper [18] authors proposed a new similarity measure algorithm for text classification and clustering. This takes many cases for similarity calculation, which are features from both contents; features from a single content and features are not in the given contents. The authors generated the awareness of detecting presence and absence of features, features have non-zero values. This has been applied in hierarchical clustering and KNN clustering algorithms. But the proposed work has been investigated only few clustering algorithm and doesn't provide accuracy in similarity finding.

In the paper [19] authors developed a novel unsupervised feature selection algorithm, named as clustering guided sparse structural learning, which is abbreviated as CGSSL. This algorithm integrates the cluster analysis and sparse structural analysis as a joint framework. The authors used the nonnegative spectral clustering for accurate cluster label detection. The cluster labels are predicted using non-negative analysis. This is suitable for much type of text documents, however the algorithm has more iteration and the accuracy is not much satisfied.

III.CONCLUSION

Finding App relationship and similarity is an iterative process, which meets two imperfections. One is that it needs to run once more when novel apps appear. Hence, it is time consuming. The other is that this iterative process needs to set two initial factors. Certain results denoted that initial parameters deeply affect calculating the outcomes. But, it is difficult to determine which factor is suitable to determine the App similarity and relationship calculation. The research community has paid little attention to Google play store, Apple store apps specifically. Most research related to duplicates, spam and malware on social media has focused on detecting

malicious contents and social spam posting. Google play store has dismantled its app rating functionality recently. A recent work studies how app permissions and community ratings correlate to privacy and security risks of Google play store apps. At last, there are several complications misleads the App usage in the real-time apps on social Medias. So detection and suggestion of valuable apps with the elimination of duplicate and harmful featured apps is important. With the use of data mining approach, data management becomes easier and convenient. Due to digital document process, the size of documents is very huge and very tough to manage. In such environment, the effective and fast grouping is more important because the data should retrieve quickly and effectively. In this survey, the different document clustering techniques and algorithms are discussed. This survey gives the overall summary of the review by different metrics and parameters. In this survey, various techniques in text mining for document management is discussed, this paper shows the pros and cons of several traditional feature discovery algorithms based on different techniques.

However, the techniques almost concentrated on general text mining process, where the document clustering needs additional concentration and work to improve the following problem. The first problem is discovering appropriate features with less effort and validation on discovered features are not yet studied. And there is a need for a new system to handle the above problem in document management and information retrieval.

REFERENCES

[1].Liu, Ming, et al. "APP Relationship Calculation: An Iterative Process." *IEEE Transactions on Knowledge and Data Engineering* 27.8 (2015): 2049-2063.

[2].https://www.statista.com/topics/1002/mobile-app-usage/

[3].Gao, Hongyu, et al. "Detecting and characterizing social spam campaigns."*Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 2010.

[4].Chia, Pern Hui, Yusuke Yamamoto, and N. Asokan. "Is this app safe?: a large scale study on application permissions and risk signals." *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012.

[5].Besmer, Andrew, et al. "Social applications: exploring a more secure framework." *Proceedings of the 5th Symposium on Usable Privacy and Security*. ACM, 2009.

[6].Kim, Jognwoo, et al. "Recommendation algorithm of the app store by using semantic relations between apps." *The Journal of Supercomputing* 65.1 (2013): 16-26.

[7].Chen, Lei, Chong Wu, and Yilan Dai. "Find relationship among applications."International Journal of Networking and Virtual Organisations 15.1 (2015): 80-98.

[8].Kim, Junhyoung, Tae Guen Kim, and EulGyuIm. "Structural information based malicious app similarity calculation and clustering." Proceedings of the 2015 Conference on research in adaptive and convergent systems. ACM, 2015.

[9].Yang, Chao, Robert Chandler Harkreader, and Guofei Gu. "Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers." *International Workshop on Recent Advances in Intrusion Detection*. Springer Berlin Heidelberg, 2011.

[10].Stein, Tao, Erdong Chen, and Karan Mangla. "Facebook immune system." *Proceedings of the 4th Workshop on Social Network Systems*. ACM, 2011.

[11].Chim, Hung, and Xiaotie Deng. "Efficient phrase-based document similarity for clustering." *IEEE Transactions on Knowledge and Data Engineering* 20.9 (2008): 1217-1229.

[12].Jiang, Jung-Yi, Ren-JiaLiou, and Shie-Jue Lee. "A fuzzy self-constructing feature clustering algorithm for text classification." *IEEE transactions on knowledge and data engineering* 23.3 (2011): 335-349.

[13].Zhang, Taiping, et al. "Document clustering in correlation similarity measure space." *IEEE Transactions on Knowledge and Data Engineering* 24.6 (2012): 1002-1013.

[14].Zhuang, Fuzhen, et al. "Mining distinction and commonality across multiple domains using generative model for text classification." *IEEE Transactions on Knowledge and Data Engineering* 24.11 (2012): 2025-2039.

[15].Lan, Man, et al. "Supervised and traditional term weighting methods for automatic text categorization." *IEEE transactions on pattern analysis and machine intelligence* 31.4 (2009): 721-735.

[16].Shehata, Shady, FakhriKarray, and Mohamed Kamel. "An efficient concept-based mining model for enhancing text clustering." *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010): 1360-1371.

[17]. Skabar, Andrew, and Khaled Abdalgader. "Clustering sentence-level text using a novel fuzzy relational clustering algorithm." *IEEE transactions on knowledge and data engineering* 25.1 (2013): 62-75.

[18].Lin, Yung-Shen, Jung-Yi Jiang, and Shie-Jue Lee. "A similarity measure for text classification and clustering." *IEEE transactions on knowledge and data engineering* 26.7 (2014): 1575-1590.

[19].Li, Zechao, et al. "Clustering-guided sparse structural learning for unsupervised feature selection." *IEEE Transactions on Knowledge and Data Engineering* 26.9 (2014): 2138-2150.

.