International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 11, Issue 07, July 2022)

Survey of Machine Learning Classification Approach for Google Play Store Apps Rating Prediction

Dimple Ghole[1], Sarwesh Site[2]

[1]*M.Tech Scholar,* [2]*Assistant Professor, Department of Computer Science Engineering, All Saint's College of Technology, Bhopal, India*

***Abstract*-- A mobile app is a software application developed specifically for use on small, wireless computing devices, such as smart phones and tablets, rather than desktop or laptop computers. Most mobile devices are sold with several apps bundled as pre-installed software, such as a web browser, email client, calendar, mapping program, and an app for buying music, other media, or more apps. Some pre-installed apps can be removed by an ordinary uninstall process, thus leaving more storage space for desired ones. Where the software does not allow this, some devices can be rooted to eliminate the undesired apps. Rating of apps is descried to success of the Apps in future. This paper presents survey of machine learning classification approach for Google play store apps rating prediction.**

***Keywords*-—— Apps, Classfication, Google, Play Store, Machine Learning, Rating .**

I. INTRODUCTION

The integration of the fifth generation (5G) networks and artificial intelligence (AI) benefits to create a more holistic and better connected ecosystem for industries. User profiling has become an important issue for industries to improve company profit. In the 5G era, smartphone applications have become an indispensable part in our everyday lives. Users determine what apps to install based on their personal needs, interests, and tastes, which is likely shaped by their genders-the behavioral, cultural, or psychological traits typically associated with their sex. It is possible to profile users' gender based simply on a single snapshot of apps installed on their smartphones. With this inference based on easy to access data, we can make smartphone systems more user-friendly, and provide better personalized products and services. In this article, we explore such possibilities through an empirical study on a large-scale dataset of installed app lists from 15000 Android users [1]. Today's smartphone application (hereinafter `app') markets do not provide information on power consumption of apps, which is essential for users.

Continuous sensing apps make this problem more severe because significant power is consumed without the users' awareness [3]. The mobile app market has been surging in recent years. It has some key differentiating characteristics which make it different from traditional markets. To enhance mobile app development and marketing, it is important to study the key research challenges such as app user profiling, usage pattern understanding, popularity prediction, requirement and feedback mining, and so on [4].

Figure 1: Mobile Apps

The Android ecosystem has recently dominated mobile devices. Android app markets, including official Google Play and other third party markets, are becoming hotbeds, where malware originates and spreads. Android malware has been observed to both propagate within markets and spread between markets. If the spread of Android malware between markets can be predicted, market administrators can take appropriate measures to prevent the outbreak of malware and minimize the damages caused by malware. In this work, we make the first attempt to protect the Android ecosystem by modeling and predicting the spread of Android malware between markets. To this end, we study the social behaviors that affect the spread of malware, model these spread behaviors with multiple epidemic models, and predict the infection time and order among markets for well-known malware families [5].

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 11, Issue 07, July 2022)

Next-app prediction is the task of predicting the next app that a user will choose to use on the smartphone. It helps to establish a variety of intelligent personalized services, such as fast-launch UI app, intelligent user-phone interactions, and so on. Since app names only provide limited semantic information, the intrinsic relation among apps cannot be fully exploited. Meanwhile, next-app to be used is largely determined by a sequence of apps that a user used recently. To address these challenging problems, this work first enriches the semantic information of apps by extracting descriptive text of each app from the app store and thus proposes a topic model to transform apps as well as user preferences into latent vectors [7].

II. LITERATURE SURVEY

R. Gomes et al.,[1] present the classifiers to meet the success requirements of the Google Play Store app store. Through the techniques of KNN and Random Forest, a statistical analysis was done performing the regressions of the applications according to some characteristics: as hypothesis test, correlation and regression metrics analysis. This work aims to create inference engines, allowing the prediction of application ratings, using the KNN and Random Forest regression techniques. The Random Forest showed better results than the KNN.

S. Zhao et al.,[2] investigate the following research questions: 1) What differences between females and males can be explored from installed app lists? 2) Can user gender be reliably inferred from a snapshot of apps installed? Which snapshot feature(s) are the most predictive? What is the best combination of features for building the gender prediction model? 3) What are the limitations of a gender prediction model based solely on a snapshot of apps installed on a smartphone? We find significant gender differences in app type, function, and icon design. We then extract the corresponding features from a snapshot of apps installed to infer the gender of each user. We assess the gender predictive ability of individual features and combinations of different features. We achieve an accuracy of 76.62% and area under the curve of 84.23% with the best set of features, outperforming the existing work by around 5% and 10%, respectively. Finally, we perform an error analysis on misclassified users and discussed the implications and limitations of this article.

C. Min et al.,[3] propose Power Forecaster to break through such an exhaustive cycle. It provides users with personalized estimation of sensing apps' power cost at pre-installation time. It is challenging to provide such estimation in advance because the actual power cost of a sensing app varies depending on user behavior such as physical activities and phone use patterns. To address this, we develop a novel power emulator as a core component of Power Forecaster. It achieves accurate, personalized power estimation by reproducing users' behaviors and emulating the target app's power use. We optimize the system to make the power emulation fast and its trace collection energy efficient. We further address the problem of dealing with large-scale emulation requests from worldwide deployment. We develop a novel selective emulation approach to minimize the server-side resource cost. We performed extensive experiments and the experimental results show that Power Forecaster achieves the power estimation accuracy of 93.4 percent and saves on 60 percent of the emulator instance usage.

B. Guo et al.,[4] This work reviews CrowdApp, a research field that leverages heterogeneous crowdsourced data for mobile app user understanding and marketing. We first characterize the opportunities of the CrowdApp, and then present the key research challenges and state-of-the-art techniques to deal with these challenges. We further discuss the open issues and future trends of the CrowdApp. Finally, an evolvable app ecosystem architecture based on heterogeneous crowdsourced data is presented.

G. Meng et al.,[5] To achieve an accurate prediction of malware spread, we model spread behaviors in the following fashion: 1) for a single market, we model the within-market malware growth by considering both the creation and removal of malware; 2) for multiple markets, we determine market relevance by calculating the mutual information among them; and 3) based on the previous two steps, we simulate a susceptible infected model stochastically for spread among markets. The model inference is performed using a publicly available well-labeled dataset AndRadar. To conduct extensive experiments to evaluate our approach, we collected a large number (334,782) of malware samples from 25 Android markets around the world. The experimental results show our approach can depict and simulate the growth of Android malware on a large scale, and predict the infection time and order among markets with 0.89 and 0.66 precision, respectively.

Y. Ouyang et al.,[6] this work aim to forecast the popularity contest between Mobike and Ofo, two most popular bike-sharing apps in China. We develop CompetitiveBike, a system to predict the popularity contest among bike-sharing apps leveraging multi-source data. We extract two novel types of features: coarse-grained and fine-grained competitive features, and utilize Random Forest model to forecast the future competitiveness. In addition, we view mobile apps competition as a long-term event and generate the event storyline to enrich our competitive analysis. We collect data about two bike-sharing apps and two food ordering & delivery apps from 11 app stores and Sina Weibo, implement extensive experimental studies, and the results demonstrate the effectiveness and generality of our approach.

C. Fang et al.,[7] a set of nearest neighbors can be constructed based on the similarity of latent vectors and it is employed for training the prediction model. Furthermore, our prediction scheme is built on the temporal sequential data and is modeled by using the chain-augmented Naive Bayes model. Experimental results with a real smartphone application log data have demonstrated that our method achieves higher recall and DCG values compared with several baseline next-app prediction methods.

E. Liotou et al.,[8] work takes advantage of recent standardization trends in SDN and proposes a programmable QoE-SDN App, enabling network exposure feedback from MNOs to VSPs towards network-aware video segment selection and caching, in the context of HAS. The video selection problem is formulated using Knapsack optimization and relaxed to partial sub-problems that provide segment encodings that can mitigate stallings. Furthermore, a mobility prediction mechanism based on the Self-similar Least-Action Walk model is introduced, toward proactive segment caching. A number of use cases, enabled by the QoE-SDN App, are designed to evaluate the proposed scheme, revealing QoE benefits for VSPs and bandwidth savings for MNOs.

Y. Kwon et al.,[9] present Mantis, a framework for predicting the computational resource consumption (CRC) of Android applications on given inputs accurately, and efficiently. A key insight underlying Mantis is that program codes often contain features that correlate with performance and these features can be automatically computed efficiently. Mantis synergistically combines techniques from program analysis and machine learning. It constructs concise CRC models by choosing from many program execution features only a handful that are most correlated with the program's CRC metric yet can be evaluated efficiently from the program's input.

We apply program slicing to reduce evaluation time of a feature and automatically generate executable code snippets for efficiently evaluating features. Our evaluation shows that Mantis predicts four CRC metrics of seven Android apps with estimation error in the range of 0-11.1 percent by executing predictor code spending at most 1.3 percent of their execution time on Galaxy Nexus.

J. C. Ferreira et al.,[10] presents a mobile information system denominated as vehicle-to-anything application (V2Anything App) and explains its conceptual aspects. This application is aimed at giving relevant information to full electric vehicle (FEV) drivers by supporting the integration of several sources of data in a mobile application, thus contributing to the deployment of the electric mobility process. The V2Anything App provides recommendations to the drivers about the FEV range autonomy, location of battery charging stations, information of the electricity market, and also a route planner, taking into account the public transportations and car or bike sharing systems. The main contributions of this application are related to the creation of an information and communication technology platform, recommender systems, data integration systems, driver profile, and personalized range prediction. Thus, it is possible to deliver relevant information to the FEV drivers related to the electric mobility process, the electricity market, the public transportation, and the FEV performance.

III. CHALLENGES

The mobile app developing grows rapidly. The customer gives the reviews after using the app. The mobile app developers can understand the target audience easily.

- Meeting user requirements.
- Choosing the operating system.
- Choosing the development platform.
- Security.

After developing and using the app, customer rating is very useful to success the app. Therefore during the literature survey some of the observation is carried out, which is as followings-

- Low accuracy rate of true data prediction from given dataset.
- Using traditional System Analysis alone not sufficient for proper feature extraction.
- More classification error.
- No adaptive approach to prediction of true rating of apps.
- Sensitivity, Specificity, Precision, Recall and F measure values is not good optimized.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 11, Issue 07, July 2022)

IV. Proposed Strategy

- *Load the google play store dataset from the Kaggle*

In this step, the google play store dataset will be downloaded from kaggle source. It is a large dataset providing company. Then load this dataset into the python environment.

- *Visualizing the Dataset*

Now open the dataset files and view the various data in term of features like rating, size of appe etc.

- *Pre-process the Dataset*

Now the data preprocess step applied, here data is finalize for processing. Missing data is either removal or replace form constant one or zero in this step.

- *Splitting the Dataset into training and testing*

In this step, the final preprocessed of dataset is divided into the training and the testing dataset. In the machine learning, firstly the machine is trained through given dataset then it comes in tested period for remaining dataset.

- *Classification Using Machine Learning Algorithm*

Now apply the machine learning technique to find the performance parameters.

- *Performance Metrics*

(Accuracy, Precision, Recall,F1 - Score)

Now the performance parameters are calculated in terms of precision, recall, f-1 measure, accuracy etc by using the following formulas-

True Positive (TP): predicted true and event are positive.

True Negative (TN): Predicted true and event are negative.

False Positive (FP): predicted false and event are positive.

False Negative (FN): Predicted false and event are negative.

$$\begin{aligned} Precision &= \frac{|TP|}{|TP| + |FP|} \\ Recall &= \frac{|TP|}{|TP| + |FN|} \\ F1 &= 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \\ Accuracy &= \frac{|TP| + |TN|}{|TP| + |TN| + |FP| + |FN|} \end{aligned}$$

V. Conclusion

Developing apps for mobile devices requires considering the constraints and features of these devices.

Mobile devices run on battery and have less powerful processors than personal computers and also have more features such as location detection and cameras. An app with 1 star rating will still be available for download even after it gets poor ratings. However, there will be a huge impact on future downloads of the apps as most of it is dependent on reviews. This paper presents survey of machine learning classification approach for Google play store apps rating prediction. In future we implement the machine or deep learning based classification technique for the rating prediction of google play store app.

REFERENCES

[1] R. Gomes da Silva, J. de Oliveira Liberato Magalhães, I. R. Rodrigues Silva, R. Fagundes, E. Lima and A. Maciel, "Rating Prediction of Google Play Store apps with application of data mining techniques," in IEEE Latin America Transactions, vol. 19, no. 01, pp. 26-32, January 2021, doi: 10.1109/TLA.2021.9423823.

[2] S. Zhao et al., "Gender Profiling From a Single Snapshot of Apps Installed on a Smartphone: An Empirical Study," in IEEE Transactions on Industrial Informatics, vol. 16, no. 2, pp. 1330-1342, Feb. 2020, doi: 10.1109/TII.2019.2938248.

[3] C. Min et al., "Scalable Power Impact Prediction of Mobile Sensing Applications at Pre-Installation Time," in IEEE Transactions on Mobile Computing, vol. 19, no. 6, pp. 1448-1464, 1 June 2020, doi: 10.1109/TMC.2019.2909897.

[4] B. Guo, Y. Ouyang, T. Guo, L. Cao and Z. Yu, "Enhancing Mobile App User Understanding and Marketing With Heterogeneous Crowdsourced Data: A Review," in IEEE Access, vol. 7, pp. 68557-68571, 2019, doi: 10.1109/ACCESS.2019.2918325.

[5] G. Meng, M. Patrick, Y. Xue, Y. Liu and J. Zhang, "Securing Android App Markets via Modeling and Predicting Malware Spread Between Markets," in IEEE Transactions on Information Forensics and Security, vol. 14, no. 7, pp. 1944-1959, July 2019, doi: 10.1109/TIFS.2018.2889924.

[6] Y. Ouyang, B. Guo, X. Lu, Q. Han, T. Guo and Z. Yu, "CompetitiveBike: Competitive Analysis and Popularity Prediction of Bike-Sharing Apps Using Multi-Source Data," in IEEE Transactions on Mobile Computing, vol. 18, no. 8, pp. 1760-1773, 1 Aug. 2019, doi: 10.1109/TMC.2018.2868933.

[7] C. Fang, Y. Wang, D. Mu and Z. Wu, "Next-App Prediction by Fusing Semantic Information With Sequential Behavior," in IEEE Access, vol. 6, pp. 73489-73498, 2018, doi: 10.1109/ACCESS.2018.2883377.

[8] E. Liotou, K. Samdanis, E. Pateromichelakis, N. Passas and L. Merakos, "QoE-SDN APP: A Rate-guided QoE-aware SDN-APP for HTTP Adaptive Video Streaming," in IEEE Journal on Selected Areas in Communications, vol. 36, no. 3, pp. 598-615, March 2018, doi: 10.1109/JSAC.2018.2815421.

[9] Y. Kwon et al., "Mantis: Efficient Predictions of Execution Time, Energy Usage, Memory Usage and Network Usage on Smart Mobile Devices," in IEEE Transactions on Mobile Computing, vol. 14, no. 10, pp. 2059-2072, 1 Oct. 2015, doi: 10.1109/TMC.2014.2374153.

[10] J. C. Ferreira, V. Monteiro and J. L. Afonso, "Vehicle-to-Anything Application (V2Anything App) for Electric Vehicles," in IEEE Transactions on Industrial Informatics, vol. 10, no. 3, pp. 1927-1937, Aug. 2014, doi: 10.1109/TII.2013.2291321.