PREDICTING MALWARE APPS USING MACHINE LEARNING APPROACH

1Mr. A. Logeswaran*,

2B. Nivya Rathika, 3K. Priyanka, 4G. Surjith & 5M. R. Yogeshwaran.

1Assistant Professor & 2,3,4&5Final Year Students,

1,2,3,4&5Department of Information Technology

Velalar College of Engineering and Technology, Thindal,

Erode, Tamilnadu, India, Pin Code: 638012.

** Corresponding Author Email: logeswaran18@gmail.com*

Abstract:

Google Play (some time ago Android Market) is a computerized sharing assistance worked and created by Google. It fill in as the authority application store for the Android working framework, permitting clients to peruse and download applications created with the Android programming improvement pack (SDK) and distributed through Google. To distinguish malware, past work has been focused on the Android application advertise biological system of Google Play. The members, comprising of clients and designers, have Google accounts. Engineers make and transfer applications, which comprise of executables (i.e., "apks"), a lot of required consents, and a portrayal. The application advertise distributes this data. Right now, present FairPlay, a novel framework that finds and use follows left behind by fraudesters, to identify both malware applications and phony clients. FairPlay utilizes a social and conduct approach dependent on longitudinal application information. FairPlay distinguishes and misuses another connection among malware and search rank extortion.

Keywords: *Android Market, Fair Play, Fraudesters, Malware, Phony Clients, SDK – Software Development Kit.*

1. INTRODUCTION

Google Play fills in as an advanced media store, introducing music, magazines, books, motion pictures, and TV programs. It once in the past offered Google hardware device for purchase until the beginning of an alternate online gear retailer, Google Store, on March 11, 2015. Applications are open through Google Play for complimentary or at an expense. They can be

downloaded utilizing Android gadgets through the Play Store versatile application or by sending the application to a contraption from the Google Play site. Applications manhandling hardware capacities of a contraption can be engaged to customers of devices with explicit gear sections, for instance, a development sensor (for development subordinate games) or a forward looking camera (for online video calling).

There are 82 billion apps downloaded in the year of 2016 and has overcome 2.7 million apps established in 2017 in Google Play store. It has been the topic of numerous issues identifying with security, in which pernicious programming has been concurred and transferred to the store and downloaded by clients, with inconsistent degrees of seriousness. Google Play was published on March 6 2012, joining the Android Market, Google Music and the Google eBook store under one kind, denoting a move in Google's computerized dispersion system. The organization's working under the Google Play standard are: Google Play Books, Google Play Games, Google Play Movies and TV, Google Play Music, Google Play Newsstand. The business accomplishment of Android application markets, for example Google Play and the motivating force model they offer to famous applications, make them engaging focuses for fake and vindictive practices. Some false engineers misleadingly help the pursuit rank and fame of their applications (e.g., through phony surveys and sham establishment checks), while vindictive designers use application advertises as a platform for their malware.On everyday schedule, an application head board can be refreshed by application store which show graph rankings of most appreciated applications, additionally it is a moving thing to make supported the development of versatile applications.

Truth be told, for advancing cell phone Apps, pioneer leading group of applications is the for the most part significant method for up inclination in the market. An application ought to be positioned progressed relying on how its graph of development raise and continuously it can make number of downloads and eventually high pay. There were unique approaches to elevate Apps so as to get top situation in App pioneer sheets, the official one is white cap premise to elevate their App to get celebrated and on the other hand progressively number of downloads.

In the current framework, from the gathered authentic records, the top occasion and driving meeting of an application is perceived. There are two principle ventures for mining driving meetings. To begin with, need to see driving occasions from the App's chronicled positioning records. Second, need to join nearby driving occasions for developing driving meetings. Cautious examination shows that the portable Apps are not continually at top most places in pioneer board. Be that as it may, just in some timeframe called driving occasion which structure diverse driving meetings implies positioning extortion for the most part emerge right now.

In Review Based Evidences, notwithstanding appraisals, the greater part of the App stores additionally grant clients to keep in touch with some literary remarks as App audits. Along these lines, individuals may sure about downloading that specific application by perusing remarks determined in survey part and furthermore give their view about that application. Because of the tremendous number of applications, it is difficult to look positioning extortion

for each applications; in this way, it is principle to have an adaptable method to naturally see positioning misrepresentation without utilizing any benchmark data.

The endeavors of Android markets to recognize and expel malware are not constantly effective. Google Play utilizes the Bouncer framework to evacuate malware. Be that as it may, out of the 7,756 Google Play applications which are broke down utilizing Virus Total, 12 percent (948) were flagged by in any event one enemy of infection instrument and 2 percent (150) were identified as malware by in any event 10 apparatuses. Past versatile malware, recognition work has concentrated on powerful examination of application executables just as static investigation of code and consents. In any case, ongoing Android malware examination uncovered that malware develops rapidly to sidestep hostile to infection apparatuses.

2. OBJECTIVE

The main objective of the project is to reduce the existence of unrealistic performance measures. To discover and leverage traces left behind by fraudsters, to detect both malware apps and fake users. The detection of the fraud use of the Mobile apps is found much faster. FairPlay is a system that provides relational, linguistic and behavioral approach based on longitudinal app data. FairPlay identifies and exploits a new relationship between malware and search rank fraud.

3. COMPLICATION STATEMENT

3.1 System Analysis

Malicious designers, who transfer malware, yet additionally judicious false engineers. Deceitful engineers endeavour to mess with the hunt rank of their applications, e.g., by enlisting misrepresentation specialists in publicly supporting locales to compose audits, post evaluations, and make sham introduces.

To survey or rate an application, a client needs to have a Google account, register a cell phone with that account, and introduce the application on the gadget. This procedure convolutes the activity of fraudsters, who are subsequently bound to reuse accounts across employments. Applications that rank higher in query items, will in general get more introduces. This is helpful both for false designers, who increment their income, and noxious engineers, who increment the effect of their malware.

1) ANDROID MALWARE DETECTION

As of late, advanced mobile phones have encountered precarious development. Assembled reports recommend that overall Smartphone deal in 1/3 quarter of 2011 arrived at 115 million units, an expansion of 42 percent from 0.33 quarter of past years. CNN comparatively shows that PDA shipments have significantly increased in the point of reference three years. As anyone might expect, numerous PDA stages are changing for expert on these cell phones. At present, Google's Android stage has been overwhelmed by Symbian and iOS to turn into the most acclaimed advanced cell stage, being set up on the greater part (52.5%) of every PDA sent.

Publicly supporting is utilized to gather framework call follows from genuine clients, at that point utilized a "partitional" bunching calculation to characterize kind and malevolent applications. Extricated highlights from observed applications (e.g., CPU utilization, bundles sent, and running procedures) and utilized AI to recognize malignant applications. Static examination, used to effectively recognize high and medium hazard applications. Utilized hazard signals removed from application consents, e.g., uncommon basic authorizations (RCP) and uncommon sets of basic consents (RPCP), to prepare SVM and advise clients regarding the dangers versus benefits tradeoffs of applications.

Google sent Bouncer, a structure that screens distributed applications to distinguish and evacuate malware. Analysts examined and uncovered subtleties of Bouncer (e.g., situated in QEMU, utilizing both static and dynamic examination). Bouncer isn't adequate outcomes show that 948 applications out of 7,756 applications that downloaded from Google Play are recognized as suspicious by at any rate 1 enemy of infection instrument.

2) GRAPH BASED OPINION SPAM DETECTION

Graph based methodologies have been proposed to handle assessment spam. They evaluate the opportunity of an item to be a spam crusade target and afterward group spammers on a 2-bounce sub diagram prompted by the items with the most elevated possibility esteems. At that point they outline misrepresentation discovery as a marked system arrangement issue and order clients and items that structure a bipartite system utilizing a spread based calculation. FairPlay's social methodology varies as it recognizes applications inspected in a bordering time interim, by gatherings of clients with a past filled with evaluating applications in like manner. FairPlay com-bines the consequences of this methodology with conduct and lin-guistic pieces of information, removed from longitudinal application information, to distinguish both hunt rank misrepresentation and malware applications. We accentuate that search rank misrepresentation goes past feeling spam, as it suggests manufacturing surveys, yet in addition client appl application introduce occasions and evaluations.

3.2 Problem Description

1) CROWDTURFING FOR FUN AND PROFIT

Publicly supporting frameworks make a genuine errand to existing security components conveyed to watch Internet administrations. A considerable lot of these security techniques depend on the presumption that malevolent action is produced mechanically by utilizing mechanized projects. Along these lines they would work severely or be handily avoided when assaults are produced by genuine clients working in a publicly supporting framework. Through estimations, it demonstrates stunning proof indicating that not exclusively do noxious publicly supporting frameworks exist, yet they are quickly developing in both client base and all out salary.

Utilize whole creeps to remove realities about the size and operational structure of these crowdturfing frameworks. Inspect subtleties of battles offered and acted in these destinations, and evaluate their start to finish effectiveness by running dynamic, favorable crusades of claim.

At long last, examine and assess the wellspring of work force on crowdturfing sites in various nations. Results uncovers that crusades on these frameworks are compelling at arriving at clients, and their continuous development represents a solid risk to online networks. Unforeseen confirmations were resolved showing that now vindictive publicly supporting frameworks exist, yet they are rapidly expanding in both client base and benefits. In view of their closeness with both customary publicly supporting frameworks and as tormenting conduct, they are called as crowdturfing frameworks.

2) SCALABLE AND ACCURATE ZERO-DAY ANDROID MALWARE DETECTION

Malignant applications stow away with-in other typical applications, which makes their discovery troublesome. Existing versatile enemy of infection programming are not adequate in their responsive nature by depending on distinguished malware tests for signature mining. It depicts a proactive procedure to spot zero-day Android malware. Without depending on malware tests and their marks, this plan is invigorated to assess conceivable security dangers presented by methods for these untrusted applications. In particular, a robotized framework called Risk Ranker to scalably view a specific application that displays unsafe conduct (e.g., propelling a root adventure or sending foundation SMS messages). The yield is then used to make an organized rundown of decreased applications that legitimacy further examination.

3) PROBABILISTIC GENERATIVE MODELS FOR RANKING RISKS OF ANDROID APPS

One of Android's center security strategies against vindictive applications is a hazard specialized technique which, before a client introduces an application, cautions the client about the authorizations the application requires, believing that the client will make the right judgment. This methodology has been demonstrated to be pointless as it presents the hazard data of each application in an "independent" design and such that needs a lot of down to earth mindfulness and some an opportunity to blackmail significant data. Start the idea of hazard scoring and hazard positioning for Android applications, to develop chance correspondence for Android applications, and perceive three wanted information for a proficient hazard scoring plan.

Probabilistic generative models are utilized for chance scoring plans, and perceive a few models, going from the straightforward Naive Bayes, to cutting edge various leveled blend models. Exploratory outcomes completed utilizing genuine world datasets uncover that probabilistic general models significantly show improvement over other existing methodologies, and that Naive Bayes models give a promising danger scoring approach.

Probabilistic generative models have been utilized generally in a scope of utilizations in AI, PC vision, and computational science, to demonstrate complex realities. The preeminent quality is to demonstrate works in a huge amount of unlabeled data. Utilizing these models, it is accepted that some parameterized irregular methodology creates the application information and gain proficiency with the model parameter dependent on the data. At that point, figure the likelihood of each application created by the model. The hazard score can be any capacity that is conversely identified with the likelihood, so lower likelihood deciphers into a higher score.

4) MACHINE LEARNING APPROACHES

Android OS is one of the comprehensively utilized portable Operating Systems. The scope of malevolent applications and malwares are developing consistently with the wide assortment of cell phones. An incredible number of business signature based devices are introduced in the market which keeps away from the entrance and circulation of pernicious applications. Different looks into have been performed which asserts that conventional mark based identification framework function admirably up to certain degree and malware creators utilize various procedures to dodge these apparatuses. So given this situation, there is a developing requirement for another option, extremely intense malware location framework to supplement and resolve the mark based framework. Most recent significant research focused on AI calculations that inspect highlights from pernicious application and contract those highlights to arrange and spot unusual malignant applications. This investigation abridges the development of malware discovery methodologies dependent on AI calculations concentrated on the Android OS.

Malware creators use numerous methods to dodge the discovery, for example, (I) code muddling system, (ii) encryption, (iii) including authorizations which are not required by the application, (iv) mentioning for undesirable hardware's, (v) download or update assault in which a considerate application refreshes itself or update another application with noxious payload, which is difficult to recognize. This additionally energizes the requirement for new examinations on other identification strategies, including AI methods. Numerous investigations have demonstrated that AI calculations to distinguish the malevolent exercises are effective in identifying them with exceptionally high precision.

5) PERMISSION USAGE TO DETECT MALWARE IN ANDROID

Android gadgets have showed up nowadays and the quantity of projects accessible for this working framework has progressed exponentially. Google as of now has its Android Marketplace where applications are organized and it is far in danger to abuse. Indeed, malware journalists put in pernicious applications into this market, yet moreover among other various markets. Hence, PUMA, another strategy for identifying noxious Android applications through AI techniques by breaking down the separated consents from the application itself. In the most recent decade, clients of these gadgets have encountered issues when introducing versatile applications. There was not a brought together spot where clients could acquire applications, and they needed to peruse the Internet looking for them. At the point when they found the application they needed to introduce, the issues start. So as to monitor the gadget and keep away from robbery, various working frameworks, for example, Symbian, utilized a confirmation framework dependent on declarations that brought a few bothers for the clients (e.g., they couldn't introduce applications paying little heed to having purchased them).The stages have utilized unique ways to deal with secure against this sort of programming. AI systems were significantly done for characterizing applications that are explicitly focused on nonexclusive malware discovery. In addition, a few methodologies have been proposed to classify applications indicating the malware class; e.g., Trojan, worms, infection; and, even the malware family.

4. MODEL OF ANDROID MALWARES

In this section, we proposed four models to predict malware propagation and spread between markets.FairPlay utilizes a social, etymological and conduct approach dependent on longitudinal application information. FairPlay distinguishes and abuses another connection among malware and search rank extortion. FairPlay additionally utilizes general highlights, for example, the application's normal rating, absolute number of surveys, evaluations and introduces, for an aggregate of 28 highlights.

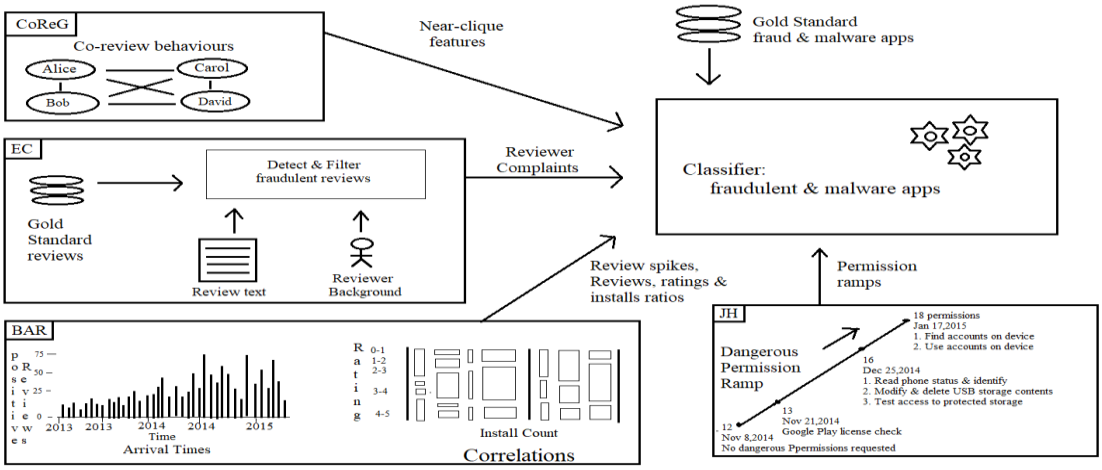

Fig.1.System Architecture

4.1 Co-Review Graph Model

This module exploits the observation that fraudsters who control many accounts will re-use them across multiple jobs. Its objective is then to recognize sub-sets of an application's commentators that have performed critical normal survey exercises before. Let the co-audit chart of an application, be where hubs compare to client accounts who explored the application, and undirected edges have a weight that shows the quantity of applications inspected in like manner by the edge's endpoint clients.

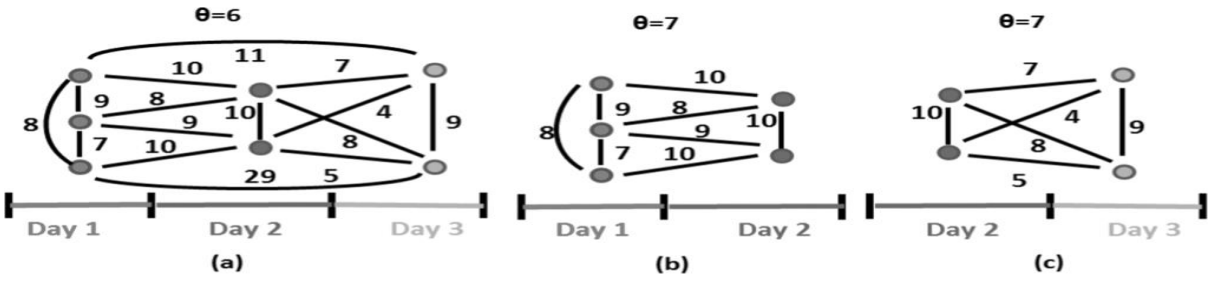

Fig.2.Co-Review Graph of an App

In a Weighted Pseudo-Clique Enumeration Problem for a graph $G = V, E$ and a threshold value θ, say that a vertex sub-set U (and its induced sub-graph $G|U|$) is a pseudo-clique of G if its weighted density $\rho = \frac{\sum_{e \epsilon \mathrm{E}} w(e)}{\binom{n}{2}}$ exceeds θ; $n = |v|$.[1] U is a maximal pseudo-clique if in addition,

no other pseudo-clique of G contains U. The weighted pseudo-clique enumeration problem outputs all the vertex sets of V whose induced subgraphs are weighted pseudo-cliques of$G$.

Algorithm: AC Algorithm Pseudo-Code

Input: days, an array of daily variety of every day audits, andθ, the weighted threshold density

Output: allCliques, set of all identified pseudo-cliques

```
1.   for d := 0 d < days.size(); d++
2.   Graph AC := new Graph();
3.   bestNearClique(AC, days[d]);
4.   c := 1; n := AC.size();
5.   for nd := d + 1; d < days.size() & c = 1; d++
6.   bestNearClique(AC, days[nd]);
7.   c := (AC.size() > n);
8.   endfor
9.   if (AC.size() > 2)
10.  allCliques := allCliques.add(AC); fi
11.  endfor
12.  return
13.  function bestNearClique(Graph AC, Set revs)
14.  if (AC.size() = 0)
15.  for root := 0; root < revs.size(); root++
16.  Graph candClique := new Graph ();
17.  candClique.addNode (revs[root].getUser());
18.  do candNode := getMaxDensityGain(revs);
19.  if (density(cand          Clique U {candNode}) ≥ 0))
20.  candClique.addNode(candNode); fi
21.  while (candNode != null)
22.  if (candClique.density() > maxRho)
23.  maxRho := candClique.density();
24.  AC := candClique; fi endfor
25.  else if (AC.size() > 0)
26.  do candNode := getMaxDensityGain(revs);
27.  if (density(candClique U candNode) ≥ 0))
28.  PC.addNode(candNode); fi
29.  while (candNode != null);
30.  return
```

CoReG separates the accompanying highlights from the yield of PCF:

(i) The number of factions whose thickness rises to or surpasses ,

(ii)The greatest, middle and standard deviation of the densities of recognized pseudo-coteries,

(iii)The greatest, middle and standard deviation of the hub check of recognized pseudo-inner circles, standardized by (the application's audit tally),

(iv)The all out number of hubs of the co-survey chart that have a place with at any rate one pseudo-faction, standardized by.

4.2 Evaluation Comment Module

Surveys composed by certifiable clients of malware and false applications may depict negative encounters. The EC module misuses this perception through a two-stage approach:

(i) Detect and channel out deceitful audits,

(ii) Identify malware and extortion characteristic criticism from the rest of the surveys.

Step EC.1: Fraudulent Review Filter.

Certain highlights can precisely pinpoint certifiable and counterfeit audits.

- Reviewer based highlights: The ability of for application A, characterized as the quantity of surveys composed for applications that are "comparative" to An, as recorded by Google Play. The inclination of towards A: the quantity of audits composed by for different applications created by A's designer. What's more, separate the absolute cash paid by on applications it has checked on and the quantity of applications that has enjoyed, and the quantity of Google+ devotees of.

- Text based highlights: The NLTK library and the Naive Bayes classifier, prepared on two datasets: (I) 1,041 sentences extricated from haphazardly chosen 350 positive and 410 negative Google Play audits, and (ii) 10,663 sentences removed from 700 positive and 700 negative IMDB film surveys. 10-overlay cross approval of the Naive Bayes classifier over these datasets uncovers a bogus negative pace of 16.1 percent and a bogus positive pace of 19.65 percent, for a general exactness of 81.74 percent.

The prepared Naive Bayes classifier is utilized to decide the announcements of that encode positive and negative opinions. Concentrate the accompanying highlights: (I) the level of proclamations in that encode positive and negative conclusions separately, and (ii) the rating of and its percentile among the surveys composed by.

Step EC.2: Analyst Feedback Extraction.

Guess that (I) since no application is great, an "adjusted" survey that contains both application positive and negative estimations is bound to be authentic, and (ii) there should exist a connection between the audit's ruling assessment and its rating. In this way, subsequent to sifting through deceitful surveys, extricate criticism from the rest of the audits. For this, utilization NLTK to extricate 5,106 action words, 7,260 things and 13,128 descriptors from the 97,071 surveys gathered from the 613 highest quality level applications. Evacuated non ascii characters and stop words, at that point applied lemmatization and disposed of words that show up all things considered once. Endeavored to utilize stemming, removing the underlying foundations of words, be that as it may, it performed ineffectively. This is because of the way that audits frequently contain (I) shorthands, e.g., "promotions", "seeya", "gotcha", "inapp", (ii) incorrectly spelled words, e.g., "pathytic", "folish", "gredy", "dispear" and even (iii) underlined incorrect spellings. In this manner, disregarded stemming.

Utilize the subsequent words to physically recognize arrangements of words characteristic of malware, fake and considerate practices. Malware pointer word list contains 31 words (e.g., chance, hack, degenerate, spam, malware, counterfeit, misrepresentation, boycott,

promotions). The extortion pointer word list contains 112 words (e.g., cheat, revolting, gripe, squandered, crash) and the benevolent marker word list contains 105 words.

4.3 Bury Appraisal Relative Module

This module use transient relations between surveys, just as relations between the audit, rating and introduce checks of applications, to distinguish suspicious practices. So as to make up for a negative audit, an assailant needs to post a critical number of positive surveys.

Guarantee 1. Let indicate the normal rating of an application not long before accepting a 1 star audit. So as to make up for the 1 star audit, an assailant needs to post at any rate positive surveys.

Confirmation let be the entirety of the considerable number of surveys got by before time . At that point, . Let be the quantity of deceitful surveys got by . To make up for the 1 star audit posted at time, is limited when each one of those surveys are 5 stars. At that point have that:The numerator of the last part signifies the aggregate of the considerable number of appraisals got by after time and the denominator is the all-out number of audits. Modifying the last correspondence, get that. The last equity follows by separating both the numerator and denominator by.

Foes who need to build the rating of an application, i.e., counteract recently got negative audits, should post an expanding, huge number of positive surveys. Such a "compensatory" conduct is probably going to prompt suspiciously high quantities of positive audits. Distinguish such practices by recognizing exceptions in the quantity of every day positive surveys got by an application. The following diagram shows the courses of events and suspicious spikes of positive surveys for 2 applications from the fake application dataset. Distinguish days with spikes of positive surveys as those whose number of positive audits surpasses the upper external fence of the case and-hair plot worked over the application's quantities of every day positive surveys.

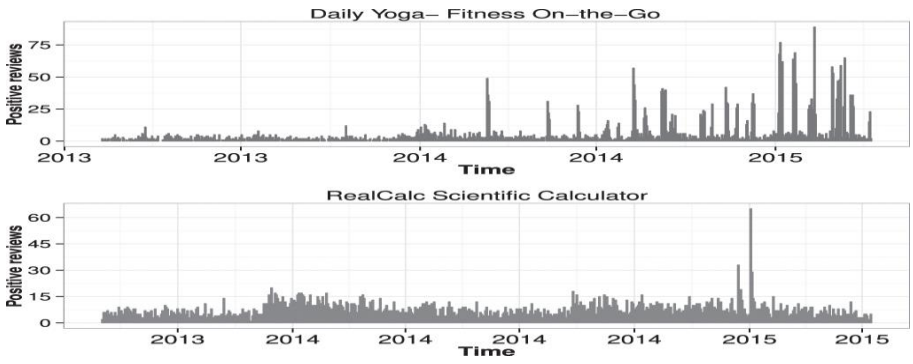

Fig.3.Timelines of reviews for two apps.

Utilize the Pearson's test to examine connections between the introduce tally and the rating tally, just as between the introduce check and the normal application rating of the 87 K new applications, toward the finish of the assortment interim. At that point bunch the rating include in containers of a similar size as Google Play's introduce check cans. The mosaic plot of the connections among appraisals and introduce checks, in this way finish up reliance between the

rating and introduce tallies. The institutionalized residuals identifythe cells (square shapes) that contribute the most to the test. The most critical rating: introduce proportion is 1:100.

Fig.4.Mosaic plot of introduce as opposed to rating check.

In the mosaic plot of the app install count versus the average app rating, rectangular cells correspond to apps that have a certain install count range (x axis) and average rating range (y axis) It shows that few popular apps, i.e., with more than 1,000 installs, have below 3 stars, or above 4.5 stars.

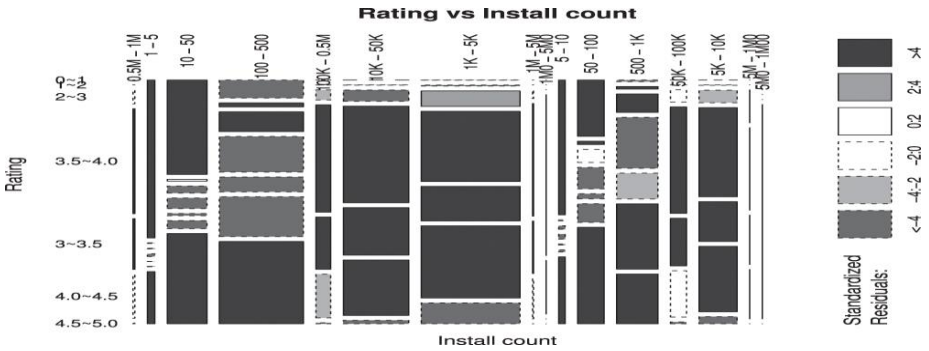

Fig.5.Mosaic plots between introduce tally and application rating.

Extract temporal features: the number of days with detected spikes and the maximum amplitude of a spike. Also extract the following:

i) The ratio of installs to ratings as two features, I_1/Rt_1 and I_2/Rt_2

ii) The ratio of installs to reviews, as I_1/Rv_1 and I_2/Rv_2. (I_1, I_2) and denotes the install count interval of an app, (Rt_1, Rt_2) its rating interval and (Rv_1, Rv_2) its (genuine) review interval.

4.4 Jekyll-Hyde App Detection Module

The following shows the distribution of the total number of permissions requested by malware, fraudulent and legitimate apps. Surprisingly, not only malware and fraudulent apps but also legitimate apps request large numbers of permissions. Likewise, Android's API level 22 marks 47 consents as "hazardous". It compares the distributions of the number of dangerous permissions requested by the gold standard malware, fraudulent and benign apps. The most well-known risky authorizations among these applications are "adjust or erase the substance of the USB stockpiling", "read telephone status and personality", "discover accounts on the

gadget", and "access exact". Perhaps surprisingly, most legitimate (69 percent), malware (76 percent) and fraudulent apps (61 percent) request between 1 and 5 dangerous permissions.

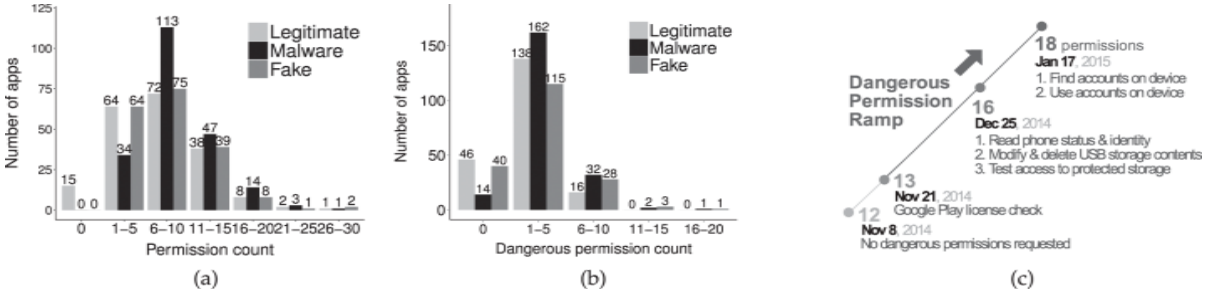

Fig.6.(a) Distribution of total permissions (b) Compares the number of dangerous permissions. (c) Dangerous permissions ramp.

After an ongoing Google Play arrangement change, Google Play composes application authorizations into gatherings of related consents. Applications can demand a gathering of consents and addition verifiable access additionally to risky authorizations. Upon manual assessment of a few applications, there recognized another sort of malevolent aim conceivably executed by beguiling application designers: applications that look to draw in clients with insignificant authorizations, however later solicitation hazardous consents. The client might be reluctant to uninstall the application "just" to dismiss a couple of new consents. Here call these Jekyll-Hyde applications. What's more it shows the hazardous consents included during various adaptation updates of one highest quality level malware application.

Concentrate the accompanying highlights:

i) The all out number of consents mentioned by the application,

ii) Its number of risky consents,

iii) The application's number of risky consent inclines

iv) Its all out number of risky authorizations included over all the inclines.

5. CONCLUSION

The proposed strategy Fairplay is a framework to distinguish both deceitful and malware Google Play applications. Trials on a recently contributed longitudinal application dataset have indicated that a high level of malware is associated with search rank extortion; both are precisely recognized by FairPlay. Also, it demonstrated FairPlay's capacity to find many applications that avoid Google Play's discovery innovation, including another sort of coercive extortion assault.

What's more, with the ongoing development of portable stages equipped for executing progressively complex programming and the rising universality of utilizing versatile stages in delicate applications, for example, banking, there is a rising threat related with malware focused at cell phones. The issue of identifying such malware presents novel difficulties because of the restricted assets accessible and constrained benefits conceded to the client, yet in addition presents special open door in the necessary metadata appended to every application.

REFERENCES

[1] Guozhu Meng,Matthew Patrick,Yinxing Xue ,Yang Liu , Jie Zhang "Securing Android App Markets via Modeling and Predicting Malware Spread Between Markets," IEEE Transaction on Information Forensics and Security, VOL. 14, NO. 7, JULY 2019.

[2] B. Liu, W. Zhou, L. Gao, H. Zhou, T. H. Luan, and S. Wen, "Malware propagations in wireless ad hoc networks," IEEE Trans. Dependable Secure Comput., vol. 15, no. 6, pp. 1016–1026, Nov./Dec. 2018.

[3] C. Yang, J. Zhang, and G. Gu, "Understanding the market-level and network-level behaviors of the Android malware ecosystem," in Proc. 37th Int. Conf. Distrib. Comput. Syst., Jun. 2017, pp. 2452–2457.

[4] L. Li et al., "Understanding Android app piggybacking: A systematic study of malicious code grafting," IEEE Trans. Inf. Forensics Security, vol. 12, no. 6, pp. 1269–1284, Jun. 2017.

[5] G. Meng, Y. Xue, Z. Xu, Y. Liu, J. Zhang, and A. Narayanan, "Semantic modelling of Android malware for effective malware comprehension, detection, and classification," in Proc. ISSTA, 2016, pp. 306–317.

[6] C. Kang, N. Park, B. A. Prakash, E. Serra, and V. S. Subrahmanian, "Ensemble models for data-driven prediction of malware infections," in Proc. Int. Conf. Web Search Data Mining, 2016, pp. 583–592.

[7] M. Isaac. (2017). Android OS Now World's Leading Smartphone Platform.[Online].

[8] K. Chandrasekar et al., "Internet security threat report," Symantec,Mountain View, CA, USA, Tech. Rep., Apr. 2017.

[9] CheckPoint. (2017). The Judy Malware: Possibly the largest malware campaign found on Google Play. [Online].

[10] J. Kirk. (2016). Android Root Malware Widespread in Third-Party AppStores.[Online].

[11] D. Steele. (2016). Third Party App Stores Blamed for Malware Infections.[Online].

[12] C. Nowzari, V. M. Preciado, and G. J. Pappas, "Analysis and control of epidemics: A survey of spreading processes on complex networks,"IEEE Control Syst. Mag., vol. 36, no. 1, pp. 26–46, Feb. 2016.

[13] S. Yu, G. Gu, A. Barnawi, S. Guo, and I. Stojmenovic, "Malware propagation in large-scale networks," IEEE Trans. Knowl. Data Eng., vol. 27, no. 1, pp. 170–179, Jan. 2015.

[14] M. Lindorfer et al., "AndRadar: Fast discovery of Android applications in alternative markets," in Proc. DIMVA, 2014, pp. 51–71.

[15] L. H. Newman. (2017). kHow Malware Keeps Sneaking Past Google Play's Defenses. [Online].

[16] M. R. Faghani and U. T. Nguyen. (2017). "Modeling the propagation of trojan malware in online social networks." [Online].

[17] G. Meng, Y. Xue, J. K. Siow, T. Su, A. Narayanan, and Y. Liu. (2017). "AndroVault: Constructing knowledge graph from millions ofAndroid apps for automated analysis." [Online].

[18] M. Sebastián, R. Rivera, P. Kotzias, and J. Caballero, "AVclass: A tool for massive malware labeling," in Proc. Int. Symp. Res. Attacks, Intrusions,Defenses (RAID), Paris, France, Sep. 2016, pp. 230–253.