ACSIJ Advances in Computer Science: an International Journal, Vol. 4, Issue 6, No.18 , November 2015
ISSN : 2322-5157
www.ACSIJ.org

ACSIJ
WWW.ACSIJ.ORG

Social Impact on Android Applications using Decision Tree

Waseem Iqbal [1], Mohammad Irfan[2] and Muhammad Asif[3]

[1] Department of IT & CS , University of Sargodha, Gujranwala Campus
Gujranwala, Punjab, Pakistan
waseem@canwascomputers.com

[2] Department of IT & CS , University of Sargodha, Gujranwala Campus
Gujranwala, Punjab, Pakistan
arfan.uosgrw@gmail.com

[3] Department of IT & CS , University of Sargodha, Gujranwala Campus
Gujranwala, Punjab, Pakistan
masif.uosgrq@gmail.com

Abstract

Mobile phones have evolved very rapidly from black and white to smart phones. Google has launched Android operating system (OS), based on Linux targeting the smart phones. After this, people became addicted to these smart phones due to the facilities provided by these phones. But the security leaks possess in Android are the big hurdle to use it in a secured way. The Android operating system is mostly used because it is an Open Source/freeware and most of its applications are also freely available on different online applications stores. To install any application, we must accept the terms and conditions regarding the access to multiple part of device and personal information, otherwise unable to install these free or paid applications. The main problem is that when we allow the access to multiple parts of our device and our personal information, the inherited security leaks become more vulnerable to threat. A very simple and handy solution is that we only install the applications that are positively reviewed by other users who already installed and are still using these applications. We implement the Decision Tree, a machine learning technique, to analyze these positively reviewed application and make a recommendation whether to install them in the device or not.

Keywords: *Android, Decision tree, Machine Learning Technique, Social Impact, Entropy*.

1. Introduction

Mobile devices have become essential part of our life. These mobile devices, especially mobile phones have evolved very rapidly [1] from a simple mobile phone with a black and white display to smart phones with color display. These smartphones not only make usual phone calls and SMS's but also read documents, create presentation, enjoy audios and videos, play games, and surfing the internet [2]. The Google introduced "Android" as its first open platform operating system for mobile devices, in2007 [3]. Now android has become the largest operating system for mobile devices and on every passing day, more than one million devices worldwide are being activated on android [4]. Android is extensively used open source operating system for mobile devices under the Open Handset Alliance [5] which makes it compatible with several hardware (devices and architectures) and software (applications). The popularity of Android has also been increased due to availability of source code with no cost [6]. Now android has become a standard for hardware manufacturer as well as software developers. Android provides you a world-class platform for creating apps and games for android users everywhere, as well as an open marketplace for distributing to them instantly [7].

The Android operating system (OS) was built on Linux Kernel [6] and specially designed to run on mobile devices. In spite of gaining popularity with every passing day [8], these mobile devices are also facing numerous security threats [9] [10].

In August 2010, a report from Essential Security against Evolving Threats (ESET) security systems shows the past five months description in which 65% of the threats were reported and if we categorize these threats, 30% are available for download in different markets, 37% spread through SMS and 60% threats are transferred through one device to another [11].

In year 2012, two departments of USA, the department of justice and the homeland security also issued a report in which these departments have shown that 79 percent of mobile OS malware threats over the whole world took

place over the android platform while the 0.7 percent threats have been detected over iPhone Operating System (IOS) platform [12]. The public sector organizations suggested their workers not to use the android due to its security threats. The report indicated that android is the main target for such attacks due its enormous market shares, free of cost behavior and open-source architecture [12].

In 2014, another report from Kaspersky labs stated that 14900 new malicious programs have been added in database of Kaspersky in 1st quarter of 2012 while this figure increased by three times in 2nd quarter of the same year [13]. The reason behind this huge number of attacks is that the android devices are based on Linux which is an open source and can easily be exploited to create different malicious applications. These applications can easily bypass the android permissions system to complete the installation process.

A very large number of applications (freely or for some cost) are available on different online applications stores like Google Play Store, 1 Mobile Market and many others. These applications have become the vital part of mobile devices and without these applications these devices lose their importance. The applications, for example, Facebook, Skype, WhatsApp, Viber, MS Office 365 and many others are the beauty of smartphones and due to these applications the usage of smartphones has increased. These and many other available applications are very user friendly and make the user more productive and connected with others at any-time, anywhere. Users just only view the application and try to install the application without knowing the security threats about the application. Whenever a user wants to install an application, this process requires an access to the multiple parts of device and personal information and if user denies granting the access, he or she will not be able to install the application [14].

Many other system [15] [16] [17] [18] have been devised that work on intrusion detection systems but they do not analyze the social impact on the application which is a very easy way to differentiate a trusted application from a non-trusted one. However, some research works [19] [20] have also analyzed the social impact in another way. This paper mainly describes a very handy and easy to access information for checking the social impact of any software with the help of Decision Tree. Every application, whenever anyone tries to install, gives information of total downloads, user reviews, ranking given by the users and many other information.

As we know that the bulk of data or applications are available on the internet but due to the enormous number of users, all the data or applications are reviewed or commented by the users [19] [20]. These reviews/comments are equipped with the rating in term of five stars or numbers from 0 to 5. These rating are very helpful in a way that the concerned data/application has an overall behavior among the current users.

In this paper, we have inspected the usefulness to classify the applications on the basis of available information by applying machine learning technique like decision tree [21]. This inspection is based on certain aspects. When anyone tries to download an application, the system checks available application's information and decides whether to install this application or not.

As part of the effort to prevent the threat happening, we are trying to help the users install these applications in a way by categorizing these applications either positive or negative on the basis of information available on application stores from current users of these applications. We are introducing a proactive approach to install the applications on the basis of available information which can reduce the chances of threat.

2. Proposed Solution

In this paper, we have worked on available information on application stores especially the total number of downloads, ranking in the form of stars or in the form of numbers from 1 to 5 and the recent feedbacks or reviews of the users. It is very convenient to use this information because there are huge numbers of reviews available even for a recently launched application.

The reviewers of any application often recapitulate their overall opinion about that application in the form of ratings and this information will be used for Decision Tree. So, we do not need any kind of manual data for application evaluation purposes. Same type of research work has been found in [22].

As discussed earlier, there are several online application stores but the data source, selected in this paper, is the Google Android store which is a very huge database for this purpose and contains thousands of applications with hundreds and thousands of reviews even for a very fresh application available in app store. Another reason is that the Google app store ratings or reviews are in the form of stars and numbers from 0 to 5 which can be converted or used in numerical values for

analysis purpose. The analysis result is in the form of positive or negative which recommends whether an application should be installed or not. In this paper, Decision Tree has been used for analysis purpose.

2.1 Decision Tree

Decision Tree is a best choice to apply in this scenario because other algorithms do not have ability to generate proper results accordingly because of some imperfections inherited in them. The other algorithms are List-Then-Eliminate, Find-S and Candidate Elimination methods are not suitable in this scenario due to some of these listed limitations [23]:

2.1.1 List-Then-Eliminate Algorithm:

- This algorithm can only be applied when we have finite Hypothesis Space (H).
- This algorithm requires to find all hypothesis in H, which is an improbable approach.

2.1.2 Find-S Algorithm:

- This algorithm cannot tell whether it has learned concept or not.
- This algorithm cannot give information about any inconsistency in the training data.
- This algorithm only selects the most specific hypothesis h, that's why is called Find-S.
- A complete hypothesis space may contain more than one consistent hypothesis but this algorithm only selects the most specific one.

2.1.3 Candidate Elimination Algorithm:

- The Candidate-Elimination algorithm covers many limitations of the List-Then- Eliminate and Find-S algorithms but still it has some shortcomings in it:
  - This we want to classify a new instance, it can only be classified if all the selected consistent hypotheses (Version Space) agree on the classification.
  - This target concept (say c) should exist within the hypothesis space H.
  - This algorithm does not have disjunctive learning ability.
  - This algorithm can only identify noisy data.

Decision Tree approximately resolves all these stated problems, however it has its own limitations. But in so far our scenario is concerned, it is the best one because we have discrete value attributes on which we decide whether to install the application or not. In decision tree, each inner node evaluates the attribute while each branch corresponds to the attribute value and at the end leaf nodes represent the classification for new instances. This representation has built in conjunction and disjunction properties, for example each path has conjunctions of attribute evaluation while Decision Tree itself is disjunctions of these paths (conjunctions). The Decision Tree is Preference Biased which means that the hypothesis space is complete (target function must be there) but the search is incomplete (missing some attributes while constructing the Decision tree). The basic Decision Tree has been implemented by the algorithm ID3 which has a drawback that we cannot back track it. The same algorithm has been adopted in many other research studies like [24]. The proposed model used for our research analysis is:

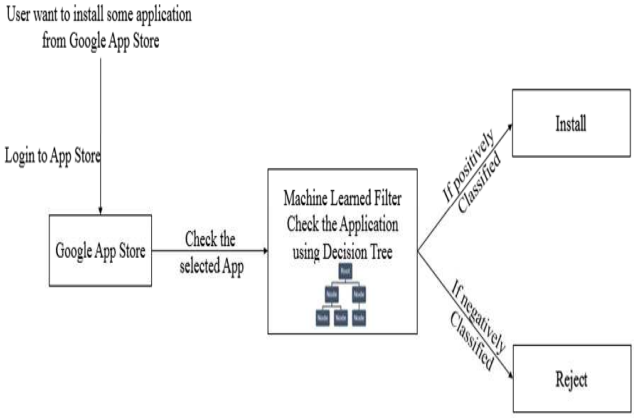

Fig. 1. The Abstract Model

Following 15 training examples have been taken to construct the decision tree:

Table 1. Real time Training Data for 15 applications with their classifications

S. No.	App Name	Google App Store	Rating	Rating in latest 30 comments	Application Downloads	Classification
1	Face Book Messenger	Yes	4.1	122	1000000000	Positive
2	Latest Android	No	4.2	104	200000	Negative
3	Whats App Messenger	Yes	4.4	110	5000000000	Positive
4	Subway Train Rush	Yes	3.7	111	1000000	Positive
5	360- Antivirus Security Free	Yes	4.5	117	5000000000	Positive
6	Free Password	No	3.5	68	250000	Negative
7	Simulator Laser Game	Yes	3	42	1000000	Negative
8	Blurb Check Out	Yes	2.1	41	100000	Negative
9	Bubble Blast	No	4.8	106	150000	Negative
10	Rabbit Rush	No	3.5	53	10000000	Negative
11	Temple Run 2	Yes	4.3	107	1000000000	Negative
12	Candy Crush Saga	Yes	4.4	111	1000000000	Positive
13	Electric Screen WallPaper	Yes	3.4	110	5000000	Negative
14	Toilet & Bath Room Rush	Yes	3.6	101	1000000	Negative
15	Mobile Hacker	No	4.8	123	20000000	Negative

2.1.4 Selection of Attributes as nodes in Decision Tree:

In this paper, the ID3 algorithm has been used for constructing the Decision Tree. The ID3 algorithm tests attributes to place the nodes at each level of Decision Tree. The finest way to find the best suitable attribute for a node or root node is Information Gain, a statistical quantitative measure, which defines how well a selected attribute splits the training examples according to the target function. The calculation of Information Gain (IG) assists in selecting the attribute to classify the examples at each level, the root node has been selected on the basis of maximum information gain from all attributes.

Table 2. Training examples distribution

Positive Examples	Negative Examples	Total Training Examples
10	05	15

Now calculate the entropy, which describes the impurity of an arbitrary collection of examples S, in our scenario the collection comprises fifteen training examples. For the collection S that contains positive and negative examples of target concept, the entropy relative to this Boolean classification is 0.9182 using the following formula:

$$\text{Entropy}(S) \equiv -p^{\delta} \log_2 p^{\delta} - p_{\varphi} \log_2 p_{\varphi}$$

Eq. (1)

The calculation for Information Gain, from collection of training examples S and attribute A, has been done using the formula:

$$\text{Gain}(S,A) \equiv \text{Entropy}(S) - \sum_{V \in \text{Values}(A)} \frac{|Sv|}{|S|} \text{Entropy}(Sv)$$

Eq. (2)

Table 3. Overall Information Gain of all attributes

Attributes	Entropy	Information Gain (IG)	Ranking based on IG
Google app store	0.9182	**0.2515**	1st
Ratings	0.9182	0.00	4th
Latest 30 comments	0.9182	0.1893	2nd or 3rd
Downloads	0.9182	0.1893	2nd or 3rd

From the above table, it is clear that the Google app store is the best classifier and selected as the root node on the basis of maximum Information Gain. After selecting the root node, now move further to select the other attributes as sub-node(s) of root nodes on the bases of IG. The entropy of Google app store is 1 based on training examples. The IG values of other attributes under the Google app store are given as:

Table 4. Information Gain of attributes based on Google app store

Attributes	Entropy	Information Gain (IG)	Ranking based on IG
Ratings	1	0.1081	2nd or 3rd
Latest 30 comments	1	**0.2365**	1st
Downloads	1	0.1081	2nd or 3rd

Now the level 2 node has been selected based on maximum IG calculated from the remaining attributes.

Table 5. Information Gain of attribute based on Latest 30 comments

Attributes	Entropy	Information Gain (IG)	Ranking based on IG
Ratings	0.994	0.1104	2nd
Downloads	0.994	**0.1832**	1st

The attribute under the Latest 30 comments attribute has also been selected based on maximum IG calculated from the remaining attributes and placed at level 3. The only remaining attribute is rating, which is placed at level 4. The following decision tree has been constructed according to the above information:

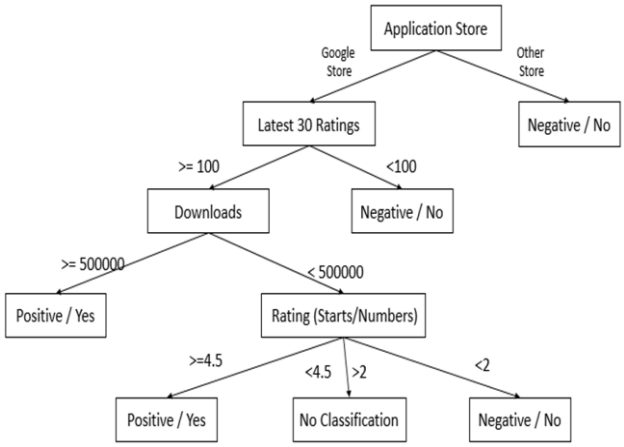

Fig. 2. Decision Tree before pruning

But this decision tree has an issue that is, the examples which have Google app store is yes, Latest 30 ratings >= 100, Downloads < 500000 and Rating is between 4.5 and 2 (terminal values are excluded) have not been classified. For decision making process, it is necessary to apply the post pruning technique of decision tree so that all examples are classified as positive or negative. As there are more negative examples for rating between 4.5 to 2, so, all the examples are negatively classified. The final decision tree after post pruning is as under:

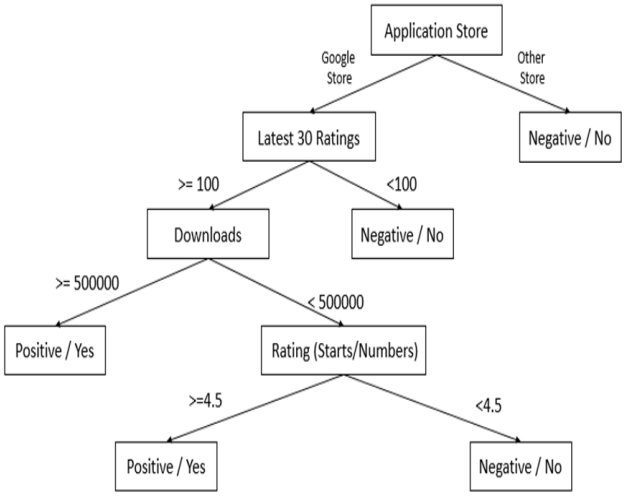

Fig. 3. Decision Tree after pruning

The Pseudo code for the final decision tree is:

Level 1. Check the availability from Google App Store

Application would be further checked at Level 2
Application would not be installed

Level 2. Check the latest 30 comments
Aggregate rating >=100 (out of 150)

Application would be further checked at Level 3

Application would not be installed

Level3. Check the numbers of downloads >=500000

Application would be further checked at Level 4
Application would not be installed

Level 4. Check the overall application rating >= 4.5
Application would be classified positively and installed
Application would not be installed if the overall rating < 4.5

After checking the given testing examples on the final constructed decision tree, following results have been found:

Table 6. Real time Testing Data for 15 applications with their classifications

No.	App Name	Google App Store	Rating in latest 30 comments	Application Downloads	Classification
1	PTCL TV	Yes	122	1000000	Positive
2	Film Actors HD Pics	No	72	10000000	Negative
3	Chrome Browser Google	Yes	96	5000000000	Negative
4	Pepi Skate 3D	Yes	102	1000000	Positive
5	Fingerprint Thermometer	Yes	107	10000000	Positive
6	Wifi Password Hacker	Yes	51	1000000	Negative
7	Speed VPN	Yes	105	1000000	Positive
8	Go Weather Forecast	Yes	104	5000000000	Positive
9	Subway Rusher 3D	No	111	10000000	Negative
10	Top Racing Car Game	Yes	108	5000000	Positive
11	Nimbuzz Messenger	Yes	96	10000000	Negative
12	Thumb Password	No	99	250000	Negative
13	Geo TV	Yes	102	10000000	Positive
14	NOAA Weather Radio	Yes	63	10000	Negative
15	Free TV	No	132	10000000	Negative

All the testing examples are classified correctly. These values can be adjusted to discrete values if there is any restriction for example the 4.5 can be rounded to 5 etc. Also application markets other than Google app store are not reliable and most likely to be vulnerable to the threats that are why we reject all other types of markets in our example.

3. Conclusion

The proactive approach, described in this paper, cannot prevent all the threats but can reduce the probability of threat happening. There are always back doors or security holes even in a very secure OS and we are always trying to mitigate them but cannot stop them all. In the same sense, we apply a machine learning technique that filters the malicious program on the basis of social impact (previous and current comments or feedback from users) and recommends whether to install the application or not.

4. Future Work

Until now, we have applied the Machine Learning Technique (Decision Tree) which decides on the basis of given stars or numbers rating by the users. In future, our aim is to apply an extra layer to filter the applications by analyzing the written comments through Natural Language Processing (NLP). The users not only give stars or numbers rating but they also give description which contains more precise knowledge about the behavior of applications.

References

[1] A. AH and R. M, "Device-aware desktop web page transformation for rendering on handhelds,"Personal and Ubiquitous Computing, vol. 9, no. 6, pp. 368-380, 2005.
[2] J. A. Chow GW, "A Framework for Anomaly Detection in OKL4-Linux Based Smartphones," in *6th Australian Information Security Management Conference*, 2008.
[3] J. DiMarzio, Android A Programmers Guide, McGraw-Hill Osborne Media, 2008.
[4] http://developer.android.com/about/index.html, http://developer.android.com. [Online]. [Accessed December 2014].
[5] W. Enck , O. Machigar and D. Patrick, "Understanding Android Security," *IEEE security & privacy*, vol. 7, no. 1, pp. 50-57, 2009.
[6] A. Shanker and L. Somya , "Android porting concepts," in *IEEE International Conference on Electronics Computer Technology (ICECT)*, 2011.
[7] "http://developer.android.com/guide/basics/whatis-android.html," Google, [Online]. Available: http://developer.android.com. [Accessed 8 June 2014].
[8] Gartner, "Gartner Says Worldwide Mobile Phone Sales Declined 8.6 Per Cent and Smartphones Grew 12.7 Per Cent in First Quarter of 2009," Gartner, Egham, UK, 2014.
[9] B. Sun, Z. Chen, R. Wang, F. Yu and V. Leung, "Towards adaptive anomaly detection in cellular mobile networks," in *IEEE Consumer Communications and Networking Conference*, 2006.
[10] B. Sun, Y. Xiao and K. Wu, "Intrusion Detection in Cellular Mobile Networks," in *Wireless Mobile Network Security*, Springer, 2007, pp. 183-210.
[11] BORTNIK and SEBASTIÁN, http://www.welivesecurity.com/2011/12/20/2012-predictions-more-mobile-malware-and-localizedattacks/, ESET, 20 December 2011. [Online]. [Accessed 14 October 2014].
[12] A. Majumdar, "http://tech.firstpost.com/newsanalysis/ us-warns-government-workers-aboutandroid- malware-threats-104558.html," 27 August 2013. [Online]. [Accessed 11 December 2014].
[13] M. La Polla, F. Martinelli and D. Sgandurra, "A Survey on Security for Mobile Devices," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, vol. 15, no. 1, pp. 446-471, 2013.
[14] A. P. Felt, E. Chin, S. Hanna, D. Song and D. Wagner, "Android permissions demystified," in *ACM conference on Computer and communications security*, 2011.

[15] D. Damopoulos, S. A. Menesidou, G. Kambourakis, M. Papadaki, N. Clarke and S. Gritzalis, "Evaluation of anomaly-based IDS for mobile devices using machine learning classifiers," *Security and Communication Networks*, vol. 5, no. 1, pp. 3-14, 2012.
[16] Y. Zhang, W. Lee and Y. A. Huang, "Intrusion detection techniques for mobile wireless networks," *Wireless Networks*, vol. 9, no. 5, pp. 545-556, 2003.
[17] M. Miettinen, P. Halonen and K. Hatonen, "Hostbased intrusion detection for advanced mobile devices," in *IEEE conference on Advanced Information Networking and Applications*, 2006.
[18] A. Shabtai, U. Kanonov and Y. Elovici, "Intrusion detection for mobile devices using the knowledgebased, temporal abstraction method," *Journal of Systems and Software*, vol. 83, no. 8, pp. 1524- 1537, 2010.
[19] A. Girardello and F. Michahelles, "Explicit and Implicit Ratings for Mobile Applications," *GI Jahrestagung*, vol. 1, pp. 606-612, 2010.
[20] A. Girardello and F. Michahelles, "AppAware: Which mobile applications are hot?," in *ACMIinternational conference on Human computer interaction with mobile devices and services*, 2010.
[21] K. N. and H. C. C. , "Input feature selection for classification problem," *IEEE Trans on Neural Networks*, vol. 13, no. 01, pp. 143-159, 2002.
[22] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in *Association for Computational Linguistics*, 2002.
[23] T. M. Mitchell, Machine Learning, New York: McGraw-Hill Comp., Inc., 1997.
[24] A. Kalpesh, G. Aditya, D. Amiraj, J. Rohit and H. Vipul, "Predicting Students Performance Using ID3 and C4.5 classification Algorithms," *International journal Data mining and knowledge management process*, vol. 3, no. 5, 2013.

Waseem Iqbal is currently working as IT Manager at Canwas Computers, Gujranwala. He is MSCS scholar at University of Sargodha, Gujranwala Campus. His main interests are Machine Learning and Artificial Intelligence.

Mohammad Arfan is currently working as Network Administrator University of Sargodha, Gujranwala Campus. He is MSCS scholar at same University. His main interests are Wireless Network, Mesh Networks and Artificial Intelligence.

Muhammad Asif is currently working as a Lecturer at University of Sargodha, Gujranwala Campus. He is MSCS scholar at same University. His main interests are Machine Learning and Operating System.