BEHAVIOR-BASED SECURITY FOR MOBILE DEVICES USING MACHINE LEARNING TECHNIQUES

Sherif Rashad[1] and Jonathan Byrd[2,3]

[1]University of Southern Indiana
Evansville, IN, USA
[2]Colorado State University
Fort Collins, CO, USA
[3]Morehead State University
Morehead, KY, USA

ABSTRACT

The goal of this research project is to design and implement a mobile application and machine learning techniques to solve problems related to the security of mobile devices. We introduce in this paper a behavior-based approach that can be applied in a mobile environment to capture and learn the behavior of mobile users. The proposed system was tested using Android OS and the initial experimental results show that the proposed technique is promising, and it can be used effectively to solve the problem of anomaly detection in mobile devices.

KEYWORDS

Behavior-based Security, Anomaly Detection, Mobile Profiles, Mobile Computing.

1. INTRODUCTION AND RELATED WORK

In this research project we explore the rapidly evolving capabilities of modern smartphones with a goal to design an innovative approach to improve the security of smartphones and better understand the dynamic behavior of mobile users. With an increasing number of smartphone users, it becomes highly important to improve the efficiency and security of these smart phones. Individual computer users interact with machine interfaces in different ways, which includes analyzing keystroke patterns, CLI commands, or navigation pathways through GUIs. We can often differentiate between individual users based on these interactions.

In this project, we explore the problem of distinguishing Android phone users based on their behavior using machine learning techniques. We have developed an Android application that collects different features that represent user behavior information such as battery life, number of applications running, tilt of the device, location, etc, and use this data to learn the behavior of mobile users. This approach has a potential application in many areas including security. Anomaly detection techniques based on user behavior model could be used to detect malicious users who have gained access to a device. Additionally, it could be used to detect unusual low-level commands run by malware. In addition to providing interesting information, such as which features are most salient, classifying unique users has potential real-world applications such as:

DOI : 10.5121/ijaia.2018.9401

- Distinguishing between users on multi-user systems. If a single device is shared by multiple users, an application could prevent access of information from the incorrect user, or automatically log into accounts based on the user behavior.

- Service/product recommendations

- Building offline behavior models to find correlations between demographics (age, gender, location, etc.) and user behaviour

The problem of security of mobile devices is one of the most challenging problems because of the dynamic nature of the mobile computing environment and the rapid deployment of mobile devices. Machine learning can be used effectively to solve different types of security problems in mobile devices [1], [2], [3], [4], [5], [6]. These problems include malware and anomaly detection. Researchers have used several signature-based, behavior-based, and heuristic methods to solve the problem of malware detection for computer and network systems in general [7], [8]. The authors in [2] introduced an approach closes to the proposed approach with extracted features from smartphones and used observation time slices. They focused on using only the K-means clustering algorithm to build the user profiles. It will be helpful to consider the ranking of the extracted features and select a smaller set of features. The authors in [1] introduced Qualcomm's Snapdragon Smart Protect for malware protection for mobile devices. The proposed approach in [1] focused on analyzing the apps statically and analyzing their behavior at runtime with monitoring the way devices associated with Wi-Fi access points [1].

We focus on this paper on the security issues related to the problem of anomaly detection for mobile devices. The goal is to develop a smart approach that utilizes the data collected from different applications, computing resources, and sensors used by the mobile users to learn the dynamic behavior of these mobile users. Machine learning will play a crucial rule in learning the behavior patterns and identifying the anomaly behavior. We tried to collect different types of features and we used feature ranking techniques to select the minimum number of features that can be used to solve this problem. We focused also on applying different types of classification algorithms to predict the corresponding user.

The next section discusses the architecture of the proposed technique. The data collection and the feature selection process and its importance for this type of application are discussed in that section. Section three discusses the experimental results with a comparison between the results from three different classification algorithms using different features. Section four gives a conclusion and discusses directions for future research work related to this area.

2. PROPOSED TECHNIQUE

2.1. Proposed Architecture

An overview of the architecture of the proposed approach is shown in Figure 1. The basic components of this architecture include the data collection app, the feature selection algorithms, and the machine learning algorithms. Once the data has been collected using different features we start to analyze the collected data to reduce the number of used feature and select only the most effective features. This can be accomplished by applying weight-based feature ranking algorithms and select the features with high ranks. This process of feature selections will reduce the dimensionality of the problem and the overall computational complexity. Machine learning algorithms will use the selected features to learn the behavior models and classify the users based on the learned behavior with a goal to detect any anomaly behavior.

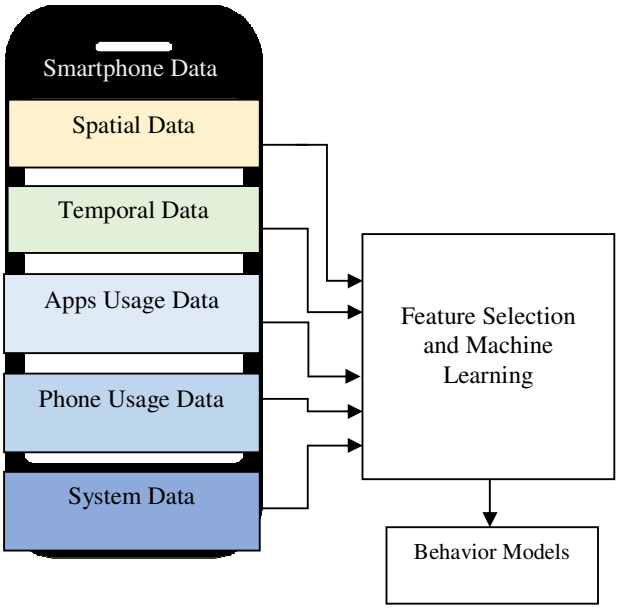

Figure. 1 Basic Architecture of the Proposed Approach

2.2. Data Collection

As shown in this Figure 1, there are different types of data that are collected from the smartphones:

- Spatial Data: this indicates data related to the location of the mobile user.
- Temporal Data: this indicates the time and the day information collected during the navigation of mobile users.
- Apps Usage Data: this includes data related to the foreground app and the running Apps.
- Phone Usage Data: this includes data related to using the phone such as the average SMS received/sent length.
- System Data: this includes data related to the system such available memory and battery life.

The data will be collected at regular time intervals when the device is powered on. Data includes different types of features as shown in Figure 1. A summary of the data features that are collated is shown in Table 1. The table shows a list of different features with their basic description.

Table 1. A list of Different Data Features for Smartphones

Feature No.	Feature Description
1	Date/Time
2	Forground App
3	Number of running Apps
4	List of Apps
5	Latitude
6	Longitude
7	Address
8	Tilt (x,y,z)
9	Available Memory
10	Battery Life
11	List of Available Networks
12	Bytes Received since Boot
13	Bytes Transmitted since Boot
14	Average SMS Received Length
15	Average SMS sent Length
16	Average Outgoing Call Length
17	Brightness
18	Charger Type
19	Orientation
20	Proximity

2.3. Feature Selection and Machine Learning

Feature reduction will play an important role in this problem to reduce the overall complexity in the mobile computing environment. Feature selection will help reduce the features by selecting the most relevant features and remove redundant and irrelevant features from the data [9].

One of the feature selection algorithms that can be used in this problem is the well-known Relieff algorithm [10], [11]. The Relieff algorithm is an extension of the well-known Relief algorithm that was introduced by Kira and Rendell [12], [13]. The Relieff algorithm can work with problems of multiple classes. The basic Relief algorithm is shown in Figure 2.

The Relief algorithm evaluates the different features based on the provided training data sets with the corresponding class labels and compute a ranking score based on ability of the feature to distinguish between the different classes in the training dataset [4]. This will be an essential process in the proposed approach to reduce the overall complexity related to process of the learning the behavior models from the collected data. This will also help to collect only the relevant data in the next phase of process.

Figure 2. Basic Relief Algorithm [12], [13]

```
Input: Training Dataset S with size n samples and m
features

Initialize all weights for features to zero: W[A]=0
for i = 1 to n
      randomly select an instance X from the training
      dataset
      find nearest hit H and nearest miss M samples
      for A = 1 to m
```

We effectively applied different machine learning algorithms on the selected features. These algorithms include decision trees classifier, Naïve Bayes classifier, K-Nearest Neighbor (KNN) classifier. The decision trees and the Naïve Bayes classifiers will be more practical in the mobile computing environment. The KNN classifier is included for the comparison purposes.

3. EXPERIMENTAL RESULTS

3.1. Data Collection

We built an Android application to collect the data. The application collects the data at regular time intervals of approximately 5 minutes when the device is powered on. We ran our initial experiments and collected nearly 12,000 samples of different features from 4 classes (users: U1, U2, U3, and U4). This data was collected over a time frame of approximately two weeks. The interface of the Android Collector App is shown in Figure 3. We used Weka machine learning and data mining software [14] for implementations of feature selection and classification algorithms that can be used to predict the four classes simultaneously.

Figure 3. The Interface of the Developed Android Collector App

3.2. Feature Selection Results

We started by considering basic 13 features that are currently in a meaningful form. These

features are shown in Table 2. We applied the Relieff algorithm and the generated scores were used to the rank features. The corresponding ranks for the13 features are shown in Table 2. The results show that features related to address and foreground app are the most two important features. Also, the number of running apps and the available memory will play a major rule in the classification process. The results clearly show that we can effectively reduce the size of the collected data by removing many of the features with the low ranks at the end of the list. We still need to conduct additional experiments to determine the number of features that can be removed without affecting the process of learning the behavior of mobile users and use it effectively for the problem of anomaly detection. We conducted additional experiments with different number of features and compared between the results.

Table 2. Feature Ranking Results of the First Thirteen Features

Feature Description	Weight	Rank
Address	0.77866	1
Foreground App	0.76037	2
Number of running Apps	0.2932	3
Available Memory	0.19675	4
Bytes Transmitted since Boot	0.07945	5
Battery Life	0.06715	6
Latitude	0.05704	7
Bytes Received since Boot	0.05583	8
Proximity	0.02751	9
Longitude	0.02581	10
Charger Type	0.00584	11
Brightness	0.00128	12
Orientation	0.00124	13

3.3. Classification Results

Several experiments were conducted to study the performance of the proposed approach using different classifiers and different sets of features. We started by considering all the 13 features shown in Table 2 and we used with 10-fold cross-validation in our experiments because of the current data size. The well-known J48 decision tree algorithm was used, and we compared its results with the Naïve Bayes classifier and the KNN classifier. The accuracy and the confusion matrix results are shown in Figures 4 and 5 respectively for the three algorithms. The results show that the proposed technique can effectively distinguish between all users with high accuracy for all algorithms. This is a crucial for the process of anomaly detection. The decision tree classifier gives the highest accuracy (99.97%).

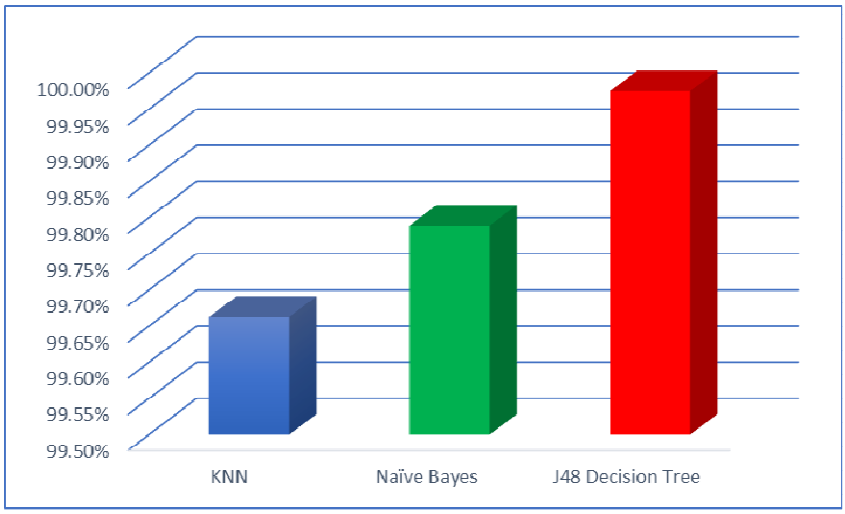

Figure 4. Accuracy results using 13 features & 10-fold cross-validation

	Predicted				
	U1	U2	U3	U4	
U1	1912	5	3	0	Actual
U2	4	3031	7	3	
U3	0	2	3441	0	
U4	1	3	12	3431	

(a) KNN Classifier

	Predicted				
	U1	U2	U3	U4	
U1	1919	0	0	1	Actual
U2	0	3044	0	1	
U3	1	0	3442	0	
U4	22	0	0	3425	

(b) Naïve Bayes Classifier

	Predicted				
	U1	U2	U3	U4	
U1	1919	0	0	1	Actual
U2	0	3044	0	1	
U3	1	0	3442	0	
U4	22	0	0	3425	

Figure 5. Confusion Matrix results using 13 features & 10-fold cross-validation

However, by using the Relieff algorithm for feature selection, we reduced the feature space to the first six features and then to the first four features and compared between the results for different algorithms. The accuracy and the confusion matrix results using only the first four features are shown in Figures 6 and 7 respectively for the three algorithms. The results show that there is a reduction in the accuracy of prediction for the J48 algorithm compared to the other algorithms, but it is still a high rang of accuracy. This result show that we can still classify the users with high accuracy with reducing the collected and processed dataset. This will be important for real-time process to speed up the process of learning and classification.

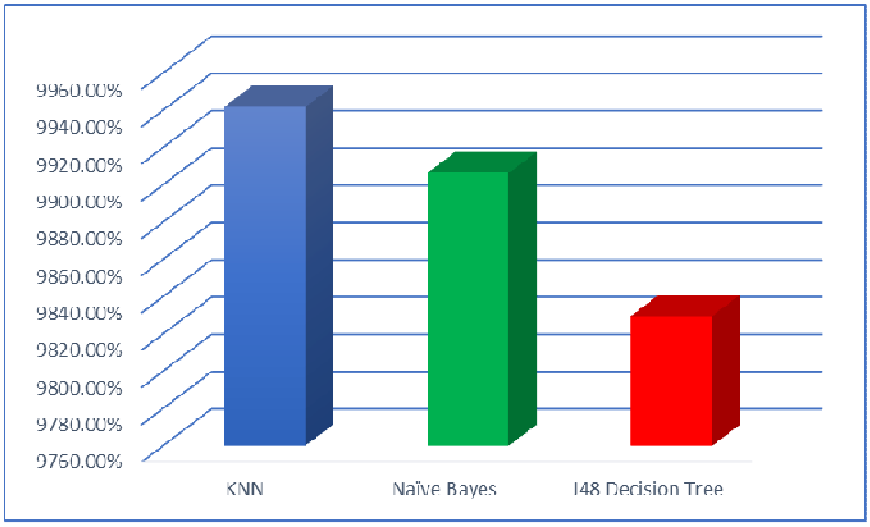

Figure 6. Classification accuracy results using the first four features & 10-fold cross-validation

	Predicted				
	U1	U2	U3	U4	
U1	1905	14	0	1	Actual
U2	24	3007	5	9	
U3	2	2	3437	2	
U4	0	2	8	3437	

(a) KNN Classifier

	Predicted				
	U1	U2	U3	U4	
U1	1910	7	0	3	Actual
U2	65	2980	0	0	
U3	0	0	3443	0	
U4	6	25	5	3411	

(b) Naïve Bayes Classifier

Predicted					
U1	U2	U3	U4		
1872	39	1	8	U1	Actual
26	2969	38	12	U2	
7	7	3418	11	U3	
5	46	2	3394	U4	

(c) J48 Decision Tree Classifier

Figure 7. Confusion Matrix results using the first four features & 10-fold cross-validation

4. CONCLUSION AND FUTURE WORK

The goal of this research project is to propose a smart approach for the problem of behavior-based security of mobile devices using machine learning techniques. We focused on the problem of anomaly detection, but the proposed approach has possible applications in many other areas to identify the mobile users by learning the behavior of using the mobile devices.

An Android application was used to collect location and behavior data of mobile users to build a behavior-based security system. The Relieff feature ranking algorithm was used effectively to reduce the dimension and the volume of the collected data. Different algorithms were used to analyze the behavior data of mobile users. Our initial experimental results show that the proposed techniques can be used to model the behavior of mobile users and identify the mobile users based on behavior data with a high accuracy and a reduced set of collected data.

Future work includes improving the overall performance and considering the most relevant features based on the mobile user. We need to test multiple users on the same device and we expect that would provide useful data. We also need to consider more users and other possible applications of the proposed techniques to provide new identity-based applications and services.

REFERENCES

[1] Nayeem Islam, Saumitra Das, and Yin Chen, “On-Device Mobile Phone Security Exploits Machine Learning”, IEEE Pervasive Computing, vol. 16, Issue 2, pp. 92-96, April-June 2017.

[2] Khurram Majeed, Yanguo Jing, Dusica Novakovic, and Karim Ouazzane, "Behaviour Based Anomaly Detection for Smartphones Using Machine Learning Algorithm", Proceedings of the International conference on Computer Science and Information Systems (ICSIS’2014), pp. 67-73, 2014.

[3] Ashkan Shamili, Christian Bauckhage, and Tansu Alpcan, “Malware Detection on Mobile Devices Using Distributed Machine Learning”, Proceedings of the 20th International Conference on Pattern Recognition, pp. 4348 – 4351, 2010.

[4] M. Waqar Afridi, Toqeer Ali, Turki Alghamdi, Tamleek Ali, and Muhammad Yasar, “Android Application Behavioral Analysis Through Intent Monitoring”, Proceedings of the 6th International Symposium on Digital Forensic and Security (ISDFS), pp. 1-8, 2018.

[5] Saba Arshad, Munam Shah, Abdul Wahid, Amjad Mehmood, Houbing Song, and Hongnian Yu, "SAMADroid: A Novel 3-Level Hybrid Malware Detection Model for Android Operating System", IEEE Access, vol. 6, pp. 4321-4339, January 2018.

[6] A. Saracino, D. Sgandurra, G. Dini, and F. Martinelli, ''MADAM: Effective and efficient behavior-based android malware detection and prevention,'' IEEE Transactions on Dependable and Secure Computing., vol. 15, no. 1, pp. 83-97, Jan-Feb. 2018.

[7] Huaqing Lin, Zheng Yan, Yu Chen, and Lifang Zhang, "A Survey on Network Security-Related Data Collection Technologies", Huaqing Lin, Zheng Yan, Yu Chen, and Lifang Zhang, IEEE Access, vol. 6, pp. 18345-18365, January 2018.

[8] Zahra Bazrafshan, Hashem Hashemi, Seyed Mehdi Hazrati Fard, and Ali Hamzeh, "A survey on heuristic malware detection techniques", Proceedings of the 5th Conference on Information and Knowledge Technology (IKT), pp.113-120, 2013.

[9] Kantardzic, M. "Data-Mining Concepts, in Data Mining: Concepts, Models, Methods, and Algorithms", Second Edition, John Wiley & Sons, Inc., 2011.

[10] Kononenko, I., E. Simec, and M. Robnik- Sikonja, "Overcoming the Myopia of Inductive Learning Algorithms with RELIEFF", Applied Intelligence, vol 7, pp. 39–55, 1997.

[11] M. Robnik-Sikonja, and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF. Machine Learning", vol. 53, pp. 23-69, 2003.

[12] K. Kira and L. Rendell, "A practical approach to feature selection", Proceedings of the 9th International Conference on Machine Learning, Morgan Kaufmann, pp.249-256, 1992.

[13] K. Kira and L. Rendell, "The feature selection problem: traditional methods and new algorithm", Proceedings of the 10th National Conference on Artificial Intelligence (AAAI 92), pp. 129-134, 1992.

[14] Weka 3 - Data Mining with Open Source Machine Learning Software in Java. [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/