An ML and SMS remote access based model for Anti-theft protection of Android devices

Tanuja Sawant[1,*], *Dhrupal* Shah[2,**], *Smita* Sontakke[3,***], and *Prathmesh* Gunjgur[4,****]

[1]Department of Computer Engineering, Ramrao Adik Institute of Technology, Nerul, Navi-Mumbai, India.
[2]Department of Computer Engineering, Ramrao Adik Institute of Technology, Nerul, Navi-Mumbai, India.
[3]Department of Computer Engineering, Ramrao Adik Institute of Technology, Nerul, Navi-Mumbai, India.
[4]Department of Computer Engineering, Ramrao Adik Institute of Technology, Nerul, Navi-Mumbai, India.

Abstract. Android phones being stolen is a significant problem that causes concerns to intellectual privacy and property. Always protecting smartphones from being stolen is a problem that remains. The key findings of the survey of existing systems for theft protection are, they provide various efficient functionalities but fail when the internet is unavailable or require specialized equipment to detect thefts. Most of these solutions are not free of charge, inefficient, time-consuming, or/and inflexible. This paper puts forward a system that provides an ML-based real-time anti-theft and remote access system for android devices. It detects theft using SVM-RBF model trained on feature-set extracted from the inertial sensor's data with an accuracy of 0.76. Whereas remote access is provided using short message services (SMS). The salient feature of this system is minimal configuration without intruding human-assisted tasks. Moreover, it will be an excellent help for authentic smartphone users to realize the theft situation and utilize the remote access features.

1 Introduction

Android phones change people's way of living as they behave like a machine which is used to access data, documents, etc. and as a result, have become an essential part of their life. Android-based mobiles are ubiquitous these days as they provide a considerable number of facilities that function in a wallet as a computer. These Android phones are beneficial for doing business. It can be easily hacked since it is small in size, and any customer information or personal data stored on the phone could be revealed easily.

Theft is prohibited under normal circumstances by merely introducing application and socially embracing property law. Physical marks (license plates, name tags) also indicate ownership. When there is no clear identity of the owner and a lack of awareness, individuals may be interested in possessing the object to their benefit at the expense of the original owner (purpose and chance are elements that enable theft). While the causes for theft are diverse and intricate, and usually, victims can't control it, the majority of theft avoidance strategies focus on minimizing theft possibilities.

There is always a threat of losing data for all the users when the phone is stolen, and also forgetting your phone home can create a big hurdle. Many of the people are facing this problem, which may result in data leaks and unauthorized access to a device. If this data is confidential, it may lead to unexpected or illegal destruction, damage, modification, unlawful declaration of private data.

There are many solutions available to prevent this but failed without the internet, which led to the idea of making a system that will help users retrieve their phones and provide security even while the system has *no internet connection*. This paper presents a fast, user-friendly, android phone anti-theft system which can provide remote access to various functionalities of users phone and detect theft using ML and help the user to recover his/her device.

2 Background

Android is a collection of services and programs together to deliver results. Android phones utilize Linux kernel with few design modifications that Google makes and controls the hardware. It will serve as a middleman between the machinery and the application that handles files and passes information between both the file system and hardware. Android has given Android System Interfaces (APIs) so that the system hardware can communicate with the kernel. The Android SDK (Software Development Kit) offers resources and a lengthy collection of APIs for creating software using JAVA on an android platform. Essential components of android application [1].

- **Activity:** An activity gives a person the space to do anything and everything. For instance, any interaction like setting the pin, saving a contact, etc., is done by communicating with a screen offered by an operation.
- **Service:** Services offer no user interface and are meant to perform actions in the background regardless of what

*e-mail: sawanttanuja1998@gmail.com
**e-mail: 1289dhrupal@gmail.com
***e-mail: sontakke.smita19@gmail.com
****e-mail: prathmesh.gunjgur@rait.ac.in

the user is doing in the foreground. A good example would be short message services - the user receives SMS even if it is kept aside.

- **Content Providers:** Such modules introduce the object-oriented principles into the framework, such as data encapsulation. It provides another, serving as an interface with the content of one process. Content Provider, together with content Resolver, is used to retrieve names in a phone user's Contacts.
- **Android Intents and Broadcast Receivers:** Android Intents are the modes of communication, i.e., how components of apps interact with each other. It is used to communicate with server and trigger application on receiving SMS.

3 Literature Survey

A mobile anti-theft program is a project that helps users monitor where smartphones are located. It consists of an Android program, which sends SMS automatically when the SIM card is updated. The location tracker is operating on both GPS and GPRS. On request with specific orders, latitude and longitude coordinates will be retrieved and sent as SMS, but this information is not sufficient to identify the thief using this information.

Tiwari Mohini *et al.* [2] created a smartphone application that permits users to remotely control their access to a phone using an internet network. This application checks internet access and then executes functions accordingly.

Sainath Vitthal Pawar *et al.* [3] implemented an android application retrieving the coordinates of the lost phone using SMS facilities. This application enables people to send SMS to the stolen phone and then get the location as a response from the stolen phone, which could then be viewed utilizing Google Maps.

Onkar Mule [4] implemented an android application retrieving the coordinates of the lost phone using SMS facilities. This application enables people to send SMS to the stolen phone and then get the location as a response from the stolen phone, which could then be viewed utilizing Google Maps.

Yongqing Goa *et al.* [5] suggested a workaround that would make the subscriber identity module (SIM) card interlock. This method scans the data on the new SIM card then contrasts that to the information on the cloned SIM card. When the findings of the test are accurate, the handset will start in a usual way; otherwise, the handset will immediately run anti-theft software, and does not need to be reported to the cell phone operator, and may avoid utilizing the subscriber identity module card, the lost handset cannot be utilized by another individual to prevent the leakage of the person's confidential data.

Azeem Ush Shan Khan *et al.* [6] presented a revolutionary android-based anti-theft program. The program deploys a security solution by supplying the thief's pictures and videos, making it simple for the operator to locate the stealer and arrest that person by providing the mobile location data with the aid of texts.

Iliyasu Yahaya Adam *et al.* [7] explored the literature on trend-setting techniques used during telephone monitoring to illustrate the issues and opportunities that may emerge from this and to instill trust. It explores the consequences of different methods used, and the actions they perform to detect fraud perpetrators, emphasizing the chances of technology misuse. It concludes with some suggestions that may serve as a guide for consumers to leverage these technologies and prevent their likely pitfalls.

Shirin Salim *et al.* [8] presented the idea of a device using the GPS and WI-FI to locate the site. When the SIM is substituted, the location will be identified, and the call information will be monitored for different operations. But those activities are unaware of the thief because they are not shown in the task manager as well; they are all working in the background.

Meng Jin *et al.* [9] described a smartphone device that detects theft in real-time using distinctive data. This data is from both the sequence of movements when the smartphone is taken-out action and the output signal of every specific movement, making it easy to identify theft activity sensitively and reliably using just the built-in smartphone sensors. iGuard is reliable and sturdy in different scenarios.

Xinyu Liu *et al.* [10] showed that inertial information is adequate to identify some common types of phone theft, such as pickpocket and pick-and-run, without compromising usability by flooding users with false alarms. It was successful and can detect 100% of simulated robberies. It's because the kinds of thefts they considered here included a short jerking movement followed by the thief trying to escape, which caused a distinct pattern in the sensor data.

Many software for anti-theft has been developed, which were real-time and provided excellent data recovery methods too. They all had one thing in common that is they required the internet the whole time. They used various methods to detect theft as well as identify the thief. After studying various existing approaches and methods, few drawbacks were observed. They are as follows :

- The majority of this software is not available for free of charge.
- Using this app, it's difficult to locate the thief, and in most cases, they fail due to lack of internet or for not having another device with the same app at hand.
- The current anti-theft systems aren't getting the thief's comprehensive information for which they are made, which in most cases results in innocents being accused.
- The current system is stated to be unreliable, time-consuming, poorly managed, and lacking flexibility.

4 Proposed Methodology

4.1 Proposed Work

Because of the limitations, as mentioned above, this paper puts forward a system that deals with theft and also provides primary remote access to the user. The system starts

the remote access services when it is intimated by a predefined template message and sends the data via SMS; while simultaneously working in the background to detect theft.

The whole system comprises of three major modules working hand-in-hand, as shown in Figure 1.

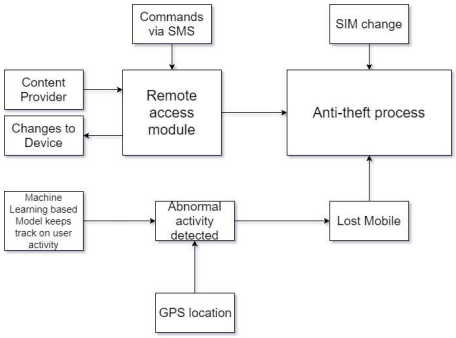

Figure 1. Architecture of proposed model

1. **Remote Access Module**

   The remote access is the module wherein the user can access remotely via other devices using SMS; the Authenticity of the user is checked based on various parameters, and if the user is authentic, it is allowed the remote access. As discussed in prerequisites, this module uses broadcasts and services, resulting in low battery consumption as the app features will be working as and when triggered.

   The SMS(s) meant for triggering the services have a predefined template, and some of its functionalities are as follows:

   - If SMS has “LOC < pin >", then the app sends location in the form of <lat, long>.
   - If SMS has “Retrieve #contact < pin >", then the app retrieves the contact and sends the number.
   - If SMS has “PROFILENAME <pin> ", then the app toggles the sound profile to the one specified in SMS.

Algorithm 1: Algorithm for location fetches

Result: Longitude and Latitude
Step 1. Start;
Step 2. Check for the SMS format;
Step 3. If format matched for location fetch;
Step 4. Turns GPS location based service is ON;
Step 5. Get the location from GPS_Provider;
Step 6. Send location information to remote user.
Create an intent for SMS,
Import SmsManager,
Use SendTextMessage() for sending SMS
Step 7. End;

The Algorithm 1 shows how the SMS triggers the process and sends the location. Similarly, it works for other functionalities.

2. **Anti-theft Module**

   As the name suggests, this module deals with the anti-theft process. This module has three sub-modules:

   - *Machine Learning* - The sub-module requires data-set as the input, which is then filtered and then is trained with the Support Vector Machine -Radial Bias Function kernel and deployed to detect theft. The system tackles false alarms by sending a notification when theft is detected. This notification would need to be declined by the user if it’s a false alarm.
   - *SIM Card Tracking* - It continuously keeps a check on SIM card bound with the phone. As soon As the SIM is changed without a proper procedure, the app triggers the anti-theft process.
   - *Remote access* - In case the application fails to recognize theft, the anti-theft process can also be triggered by sending an SMS Now, this makes the system real-time. If the theft is detected, the module runs the anti-theft process.

3. **Securing Power button options**

   Power Button is secured by depending on the external application, which protects the privacy and prevents theft by securing the power off and restart option with a password.

5 Algorithm

5.1 Support Vector Machine

Support Vector Machine (SVM) is simply scary, in any case. It turns out to be less startling once thought of it as a “road machine", which isolates the right, left-side buildings, vehicles, people on foot, and tries to make the amplest path as could reasonably be expected. What’s more, those cars, buildings, truly near the road is the support vectors. The following is the detailed clarification to see how this “road-machine" works.

Back to the discussion, consider all those objects, people as points now. The yellow points represent things on the right half of the road; red points represent the ones on the left. The road becomes dashed and solid vectors. SVM works to find the decision boundary to divide the points into different classes and amplify the margin [11].

5.1.1 SVM in linearly separable cases

Endless vectors will be able to divide the red and yellow points in the example 2. Now, this is where SVM comes into working and tries to locate the ideal vector with the requirement of accurately classifying either class:

- Follow the constraint: investigate all different hyperplanes, hyperplanes that effectively classify classes.
- Conduct optimization: select the hyperplane that amplifies the margin.

Figure 2 illustrates the concept of separate hyperplane and margin.

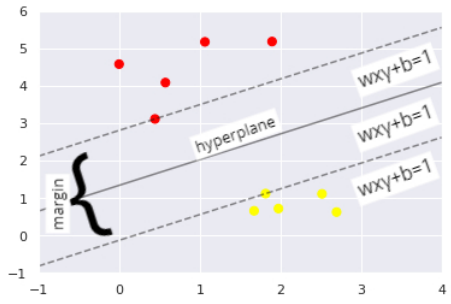

Figure 2. Linear SVM

5.1.2 SVM in linearly non-separable cases

In the cases that can be separable linearly, the goal of the SVM is to search for the hyperplane that amplifies the margin, provided that it can classify the classes effectively. But in an actual scenario, data-sets are not necessarily linearly separable, which results in not meeting the 100% effective classification criteria defined by a hyperplane. Whereas for cases that are not linearly separable, SVM deals it by presenting two ideas, which are discussed below, and flow of it is shown in Figure 3.

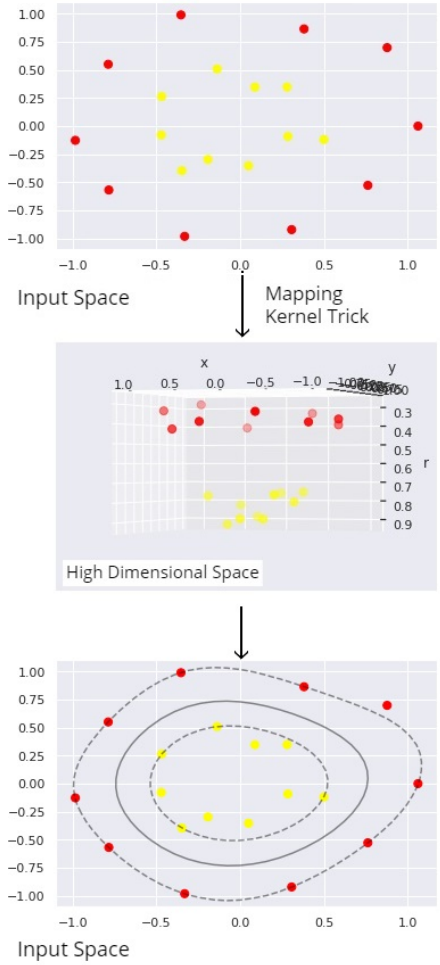

Figure 3. SVM in non-linearly separable cases

- **Soft Margin:** Using Soft Margin, SVM tries to tolerate some points to get wrongly classified and attempts to adjust balance by looking for a vector that amplifies the margin and limits the wrong classification. SVM tolerates two kinds of misclassifications under soft margin refer Figure 4 :

  - The points on the correct side of the margin but wrong side of the decision boundary (shown in first).

  - The points on the wrong side of the margin and the wrong side of the decision boundary (shown in second).

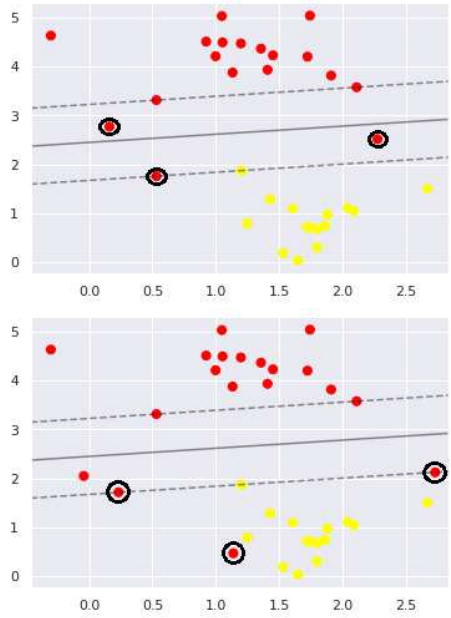

Figure 4. Soft Margin

- **Kernel Trick:** Using the existing features, Kernel Trick creates new features by applying some transformations. These transformed features are the important factors for SVM to discover the non-linear decision boundaries; refer Figure 5 for this. There are many kernel/transformations that one can choose from, for example 'RBF', 'sigmoid', 'poly', 'linear', 'precomputed'. As this system uses the Radial Basis Function (RBF), it is explained with an example below.

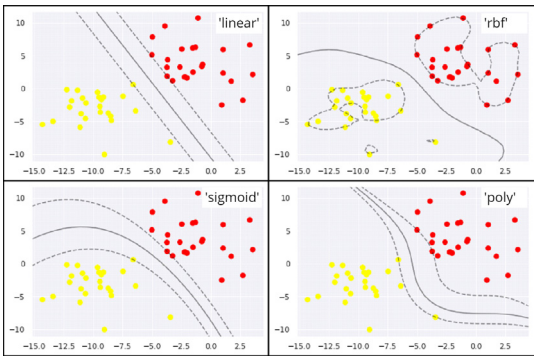

Figure 5. Kernel Trick

5.2 Radial Basis Function kernel

Radial Basis function (RBF) kernel can be imagined as a transformer/processor to produce modified features by estimating the distance between every single other point to a particular point/point-centers. The Gaussian Radial Basis Function is the most famous and simple RBF kernel. Equation 1 is the mathematical representation of RBF:

$$\phi(x, center) = exp(-\gamma \|x - centre\|^2) \quad (1)$$

Gamma(γ) decides the impact of transformed features — ϕ(x, center) on the decision boundary. As the gamma increases the impact of the features on the decision boundary also increases. To get more wiggled boundary greater gamma is required.

When using SVM with kernels, (γ), that can be tuned plays a significant role as shown in Figure 6, and its effect is summarized in Table 1.

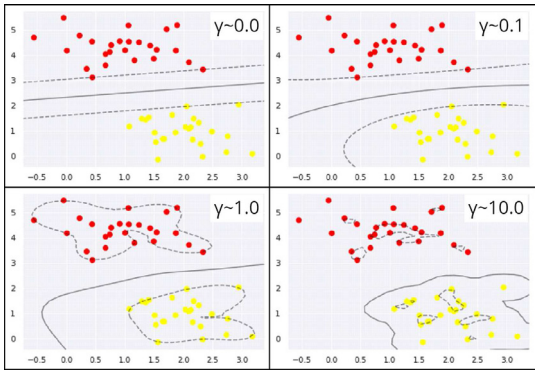

Figure 6. Illustration of how Gamma affects the Model

Table 1. Gamma results

Sr	value	impact	decision boundary
1	1E3	null	almost straight
2	0.1	small	appropriate
3	1	large	wiggled
4	10	very large	too wiggled

How applying a Gaussian rbf ((γ) = 0.1) can benefit, is illustrated in the example below:

Feature:
X = [-2,-1,0, 1,2]

Label:
Y =[0,1,1,0,0]

New Features:
X_1=[1.01, 1.00, 1.01, 1.04, 1.09]
X_2=[1.09, 1.04, 1.01, 1.00, 1.01]

ϕ**(x1,center1)** = 1.01

#x1=[-2,0]; #center1=[-1,0]; #gamma=0.1

As shown in Figure 7 in the first image, it isn't easy to discover a line separating the points. Nevertheless, if Gaussian RBF kernel is applied using the centers [(2,0), (-1,0)] to get transformed features, a line can be drawn that will be able to separate the red-yellow points as shown in the right.

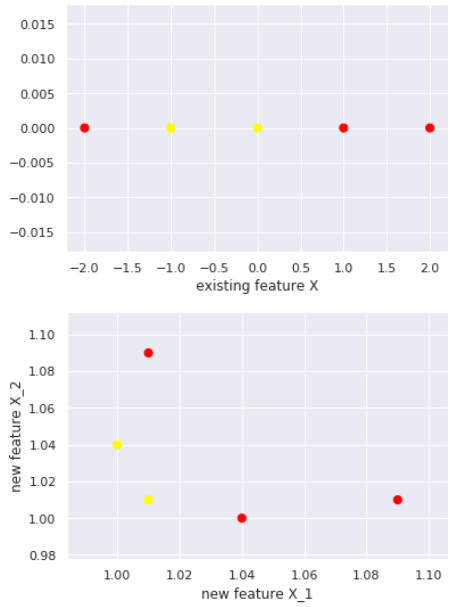

Figure 7. Transformation of Feature Set

5.3 Data-set Description

The data-set is formed from the inertial sensor readings. Classification results are much dependent on the performance of feature selection. Right now, the feature vector is extracted and chose from the time-domain velocity signal. The extraction procedure is generally straightforward. The average uniaxial velocities are investigated as the three features for classification.

Algorithm 2: How to get sensor reading

```
Result: Get Accelometer Reading
float x, y, z;
if mySensor.getType() ==
 Sensor.TYPE_ACCELEROMETER then
    x = sensorEvent.values[0];
    y = sensorEvent.values[1];
    z = sensorEvent.values[2];
end
```

The data set that is formed of inertial reading has five columns x, y, z velocities, sample number, and class. A feature-set is then formed of the above data-set that has four columns. The first three columns contain average velocities in the x-direction, y-direction, and z-direction, and 4th column is the class label for each sample. The final data-set has 350 samples.

6 Results and Discussion

For the training - testing the data-set is divided into a 70-30 ratio. The results and analysis below are based on the same.

1. **Confusion Matrix:** Confusion matrix, as shown in Table 2, is also known as an error matrix. It describes the performance of a classifier on a test dataset with known actual values and helps to visualize the performance of an algorithm.

Table 2. Confusion Matrix

	Actually Positive (1)	**Actually Negative (0)**
Predicted Positive (1)	33 True Positive (TP)	16 False Positive(FP)
Predicted Negative(0)	9 False Negative (FN)	47 True Negatives(TN)

2. **Accuracy:** Accuracy is the most common measure of performance and is a ratio of correctly predicted observations to total observations, refer Equation 2.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2}$$

3. **Precision :** Precision is the ratio of correctly predicted positive observations to the total predicted positive observations, refer to Equation 3.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

4. **Recall:** The Recall is the ratio of correctly predicted positive observations to all observations in an actual class, refer to Equation 4.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

Table 3. Results of various SVM Kernel

Parameter	**SVM kernels**		
	Linear	**RBF**	Sigmod
Accuracy	0.68	**0.76**	0.74
Precision	0.33	**0.67**	0.63
Recall	**0.94**	0.79	0.74

Since all performance measures (as seen in Table 3) of Sigmoid kernel are low, it is not considered. The linear kernel has a high Recall but a very low Precision, which is not suitable as it would result in a low theft detection rate. Now coming to RBF even though Recall is comparatively less than that of Linear, it has a pretty high Precision. Hence RBF was selected.

7 Conclusion and Future Work

The primary goal of the implemented solution is that when an android based phone of the user is stolen, the mobile phone will automatically detect theft and start the anti-theft process if the internet is available. In case the app fails to recognize the robbery, the process can be triggered by merely sending an SMS. Since the power off and restart options are locked by the external app, mobile phones if lost/stolen, can be misused by another person and also cannot be formatted until the battery goes down, which might give users a chance to retrieve their smartphones.

The implementation of the ML model for motion detection is still on the primary stage of research and implementation. The project being on the primary stage, the training efficiency factor for algorithms is considered as their accuracy values obtained during the training phase. In the future, once the research is done, the training efficiency factor can be generated using elaborate methods to get more efficient values. For making the application more powerful, the external dependency for securing power-off and restart options would be developed. As the android only provides the image and voice-based ML models to be deployed as of now, so as further developments are made, the whole system would be able to work offline.

References

[1] Android Architechure Description [Online]. Available: https://tinyurl.com/w8b34g4

[2] Tiwari Mohini, Srivastava Kumar and Gupta Nitesh; “Review on Android and Smartphone Security" NRI Institute of Information Science and Technology, Bhopal, Madhya Pradesh, INDIA. Vol. 1(6), 12-19, November (2013).

[3] Sainath Pawar, Saiprasad Pore, Suprita Tendulkar, Vinayak Malavade “Android Application for Antitheft Security through SMS" IJSRD Vol. 4, Issue 02, 2016.

[4] Onkar Mule, Nihal Shaikh, Pratik Shinde, Amit Wagaskar, Prof. Sneha Ramtek. “Remote Access of Android Smart Phone" Volume 7, Issue 4, April 2017.

[5] Yongqing Gao, Chunlai Zhou, Dan Shang, “A Smart Phone Anti-theft Solution Based on Locking Card of Mobile Phone", International Conference on Computational and Information Sciences, 2011.

[6] Azeem Ush Shan Khan, Mohammad Naved Qureshi and Mohammed Abdul Qadeer, “Anti-Theft Application for Android-Based Devices", IEEE International Advance Computing Conference (IACC), 2014.

[7] Iliyasu Adam, Cihan Varo and Asaf Varol, “Problems and Prospects of Anti-Theft and Mobile Phone Tracking: A case in Nigeria", 7th International Symposium on Digital Forensics and Security (ISDFS), 2019.

[8] Shirin Salim “Monitoring System for detecting mobile theft" International Journal of Computational Science and Information Technology Vol.4, No.2, May 2016.

[9] Meng Jin, Yuan He, Dingyi Fang, Xiaojiang Chen, Xin Meng & Tianzhang Xing, “iGuard: A Real-Time Anti-Theft System for Smartphones", IEEE Transactions on Mobile Computing, Vol:17 Issue:10 Oct 2018.

[10] Xinyu Liu, David Wagner, Serge Egelman, “Detecting Phone Theft Using Machine Learning", 14^{th} ICISS, 2018.

[11] Support Vector Machines Explanation [Online]. Available: https://tinyurl.com/uaouav7