Sri Lanka Institute of Information Technology

Evaluation of Trust in Android Devices: Deep Reinforcement Learning

Project ID: 16J – 008

Software Requirement Specification

Comprehensive Design & Analysis Project

June Intake – 2016

B.Sc. Special (Honors) Degree in Information Technology

Submitted on Thursday, 15th of September, 2016

Project ID: 16J – 008

Author:

Student ID	Name	Signature
IT13023256	Jayasekara R.A.N.T.	

Supervisor

…….

Mr. Lakmal Rupasinghe

Co - Supervisor

…….

Mr. Krishnadeva Kesawan

Declaration

"I declare that the project would involve material prepared by the group members and that it would not fully or partially incorporate any material prepared by other persons for a fee or free of charge or that it would include material previously submitted by a candidate for a Degree or Diploma in any other University or Institute of Higher Learning and that, to the best of our knowledge and belief, it would not incorporate any material previously published or written by another person in relation to another project except where the acknowledgement is made in the text."

……..

Jayasekara R.A.N.T.

Table of Contents

1. Introduction

This section of the Software Requirement Specification (SRS) provides an overall idea of the product. By reading this section any personal can get a thorough idea about the product. Includes purpose of the product, definitions, acronyms, abbreviations, intended audience, references and overview of the SRS.

1.1 Purpose

This document provides a detailed description about the project, “Evaluation of Trust in Android Devices using Deep Reinforcement Learning”. In the further chapters, the document will provide the sufficient details and descriptions of different interfaces, functions and security measures that are implemented within the system in order to achieve the supposed task.

The purpose of our system is to determine the trust of the devices involved in IoT by looking at various aspects that indicate the presence of malware in a device which will ultimately diminish trust. The features we collect are based on Network Information, Application information, Operating System Information and User Information. By considering many parameters from these features, the overall trust of the device is determined. Our ultimate goal is to build a trust network by connecting devices with higher trust values, so that sensitive information can be exchanged through IoT environment.

1.2 Intended Audience & Reading Suggestions

This document is directed towards a very limited target audience. They should read the intended chapters of the document according to their relevance.

Target Audience	**Suggested Chapters**
Supervisor / Co – Supervisor	Whole document
Developers	Whole document
CDAP Members	Chapters 1

1.3 Scope

This document will cover all aspects of the project, related to Network Information component as mentioned in the Project Proposal. Our system will extract features, such as, user information in the overlay network, network information, application information installed in the target device, android operating system information of the target device and will collect a dataset in order to build a machine learning algorithm. We will use Q – Learning in order to train the deep reinforcement learning model free system.

1.4 Definitions, Acronyms and Abbreviations

Definitions

- Android – A mobile operating system, based on JAVA and XML languages. [2]
- Machine Learning – Lets the machines learn, without being explicitly programmed. [3]
- Deep Learning – Creates architectures consisting of multiple layers of representations in order to learn high level abstractions [1]
- Reinforcement Learning – Allows a machine to learn from trial-and error interactions with its environment [1]
- Deep Reinforcement Learning – Is a combination of both deep learning and reinforcement learning.
- Q Learning – A model free reinforcement learning technique. [4]
- Theano – Theano is a Python library that allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently [10]

Acronyms

SRS	Software Requirements Specification
IoT	Internet of Things
MANETS	Mobile ad-hoc Networks
RAM	Random Access Memory
OS	Operating System
GB	Gigabyte
GPU	Graphics Processing Unit

1.5 Overview

This research is based on calculating the trust of a device based on multiple features by doing a Malware Analysis. When determining the trust of the device many aspects should be taken into consideration. But since machine learning algorithm is used to predict the overall trust based on features, the features we can use without degrading the performance is limited. Therefore we mainly focus on four main aspects which is User, Network, Applications installed in the device, Operating System. We take features from each of the four aspects and combine it to form a feature vector. This feature vector is used to predict the final trust of the device using a Reinforcement Learning Algorithm.

MANETS have various types of nodes with different characteristics such as malicious nodes, selfish nodes and etc. The behavior of a node and which type of node the device is examined using the collected features. We found that malware contribute a high percentage to degrade the overall trust of the device. Therefore by using applications installed we are looking for malicious applications in the device. By examining the user we are checking whether the user is malicious user or not and etc.

The main goal of this research is to determine the overall trust of the devices in MANETS and make a secure network by connecting the devices with high trust values for effective and secure communication between two nodes. The ultimate aim is to provide a secure way to send sensitive data among devices in MANETS.

This document is divided into two sections. The main purpose of this document is about Information Extraction of the User or the owner of the device and how it contributes to the overall trust value. This document contains constraints, dependencies, assumptions and other important factors that affect the development and design of this project. The first section describes the functionalities and behavior of the system. The second section focuses on non-functional requirements and other requirements of the system. Use case diagrams and other UML diagrams are used to give a clearer idea about the system.

2. Overall Descriptions

2.1 Product Perspective

Numerous studies were carried out to determine the device trust of Android devices considering various aspects. Most of them are concerned on MANETS and peer-to-peer networks. And there are limited numbers of studies carried out on Device trust for IoT. Some of the studies are determine device trust based on Malware Analysis and some of them are based on other factors such as User's information.

A study carried out by Weiss and Reznik [5] on Trust Evaluation in Mobile Devices stated that Trust evaluation should integrate various metrics ranging from accuracy and reliability of the data sources to the security of the procedures and tools used. In order to fill these criteria they used application details, device feature security, sensor security, Battery Usage, CPU Usage, Network usage and level of privacy provided by the device as parameters. All these parameters are considered for the identification of malicious behavior. Since malicious program tend to use up more resources this method is effective. Yet they have failed to capture the user's details which represent trust just as much as other factors.

Zhao and Pan [6] introduced a Machine Learning based Trust Evaluation Framework for Online Social Networks. They have considered numerous factors about users such as the density of interactions between the two users, the similarities between the two users, the number of mutual friends between the two users and etc. Although this is a good approach they only give the final output as two values, i.e. whether we could trust or distrust a particular device. This approach isn't ideal to be used in an IoT environment or MANETS because we cannot completely disregard all the untrusted devices. It will degrade the performance. So rather than completely saying a particular user or device is untrusted, assigning a continuous value as the final output is more applicable cause then it allows us to compare.

Bao and Chen [7] introduced a Dynamic Trust Management for the IoT Applications. They provided a flexible and accurate trust assessment for IoT entities. Although their trust assessment is quite accurate and successful they have failed to detect the presence of malicious behavior and selfish nodes. They have mainly focused on user's relationships.

Trust or Reputation analysis is looked upon in various angles. Presence of malicious programs affects trust of the device most because most of the time the user is unaware about the presence of malware. We believe that presence of malware can be analyzed and confirmed in various aspects.

Various studies are based on malware detection and analysis. There are static analysis techniques, dynamic analysis techniques as well as hybrid approaches. There are multiple ways how malware get installed into devices and several different ways they get activated. They could exploit the vulnerabilities in the user level, network level, operating system and applications installed in the device. Therefore it is necessary to identify the weakness at each of these levels because all of these weaknesses will diminish the trust of the devices and provide many opportunities for malware to get installed in our devices.

Yerima and Sezer [8] introduced a Malware Detection Mechanism using Parallel Machine Learning classifiers. As the features for the feature vector they extracted API related features, App permissions and Standard OS and Android framework commands. Although they have considered both application and OS related information to detect the malware they have neglected the network and user related information which is just as important.

Mekouar, Iraqi and Boutaba [9] brought forward a method called Detecting Malicious Peers in a Reputation-Based Peer-to-Peer System. They allocate the reputation based on the feedback and Authenticity of the files given. Although this is a good approach to be used in a peer-to-peer system we cannot use this in an IoT environment because it's not practical to collect feedback.

	Product/Paper				
Feature	Andromaly	[9]	[8]	Trust Evaluation in mobile devices	A Machine Learning based Trust evaluation
Ideal for IoT environment.	**N**	**N**	**Y**	**N**	**N**
Parameters for the computations can be easily obtained.	**Y**	**N**	**Y**	**N**	**Y**
User level parameters considered.	**N**	**N**	**Y**	**N**	**Y**
Presence of malware is considered.	**Y**	**N**	**N**	**Y**	**N**

Selfish nodes are detected.	N	Y	N	N	N
Malicious nodes are detected.	N	Y	N	N	N
Network parameters (Packet Size, No. of Packets, Duration of a Transaction, No of Transactions Completed) are considered.	N	N	N	Y	N
Vulnerabilities in Android OS is considered.	Y	N	N	Y	N
Vulnerabilities in the apps installed are considered	Y	N	N	Y	N
Overall trust value for the device is given	N	Y	Y	N	N
Creates a secure network	N	Y	N	N	Y

There's a need for evaluating device trust in the IoT domain considering factors, such as;

1. Network Information

   Extract Network Information by hop count between the two devices, number of packets transferred between 2 nodes, number of packets received, size of a packet, number of completed transactions, duration of a transaction, presence of Selfish nodes, etc.

2. Application Information

   Identify vulnerabilities in the installed applications by considering the factors such as number of times a particular application has been installed, number of reviews for a particular application, is the application up-to date, user rating a particular application has received, is the Developer recognized

3. Operating System Information

   Identify the vulnerabilities in the Android OS by considering the factors such as, is the device updated, appropriate security tools are utilized, mobile Data usage in the background processes, mobile data usage in the foreground processes, number of System Calls made by a particular app.

4. User Information

   Extract user information in social media sites. For e.g. if there are two users called x and y, we need to find whether x is a friend of y, the designation of x, the similarities between the two users, the density of interactions of x and y, the distance between x and y, ratio of mutual friends to total number of friends

2.1.1 System Interfaces

- Theano

  Theano is a Python library that allows define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently.

- Android Studio

2.1.2 User Interfaces

As of yet, no user interfaces have been identified. The document will be updated, if any found in the future.

2.1.3 Hardware Interfaces

- Android Smartphone

  The main hardware component for the system. The device will check, if the selected device could be trusted using the deep reinforcement model built.

- NVIDIA Graphic Card

  When building the deep reinforcement model, the computations takes time. For parallel processing we need an NVIDIA GPU.

2.1.4 Software Interfaces

- Android Operating System: The system is mainly built for evaluating trust on Android Devices, thus having Android as the operating system is mandatory.

- Cuda: As stated in section 2.1.3, this will be used to configure the NVIDIA GPU.
- Web Server: A server is needed to save and retrieve data gathered

2.1.5 Communication Interfaces

Data transmission among devices and the server will be done using 3G or 4G network. Since the system is run real time, having a high bandwidth is better.

2.1.6 Memory Constraints

Device memory will be used to save certain data obtained through the analysis. Having at least 1GB of free space is compulsory.

2.1.7 Operations

The system user, i.e. the mobile device, can perform the following operations.

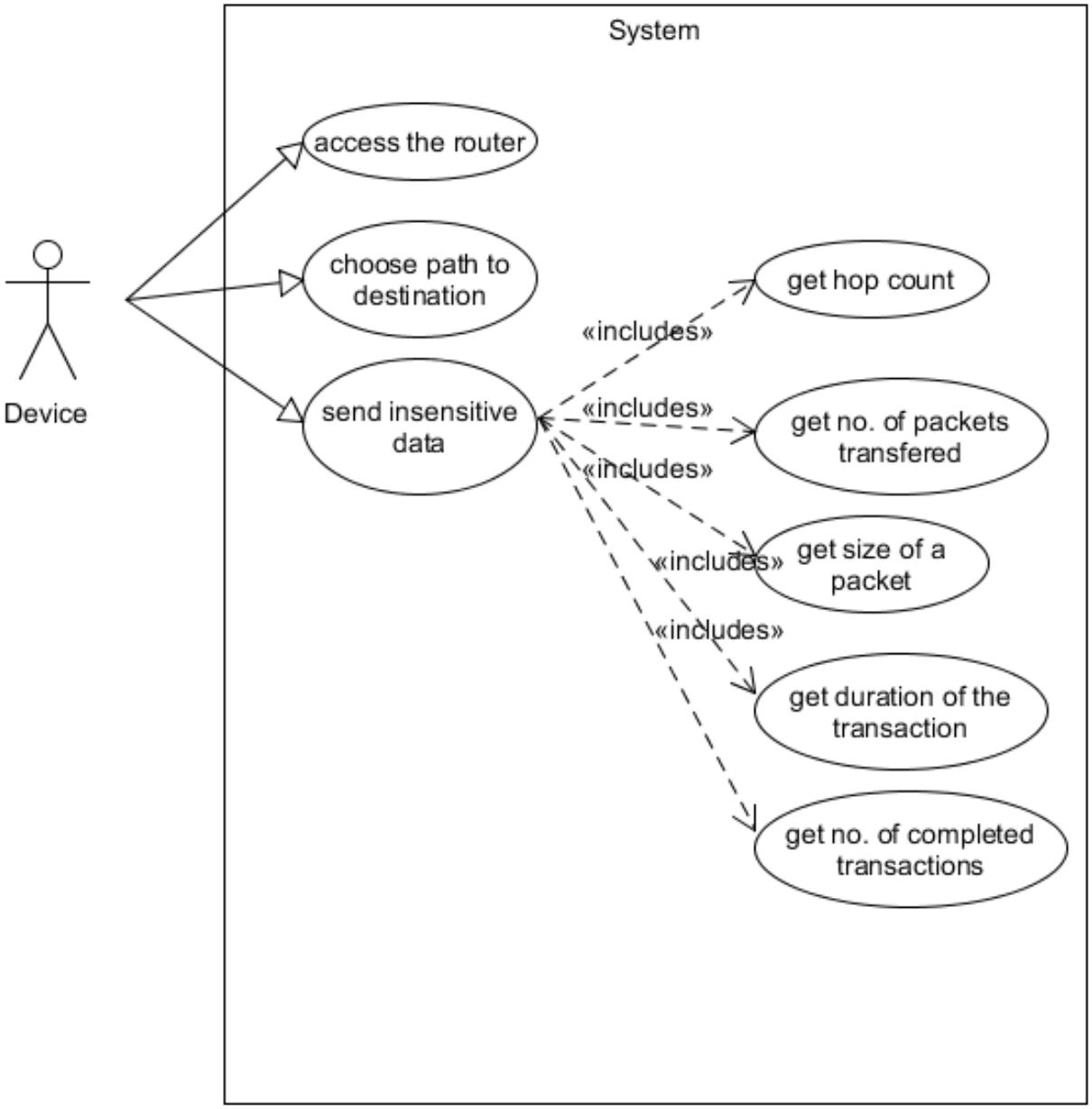
- Request details from a remote/immediate neighboring device.
- Save data gathered in the device for future usage.
- Warn other neighboring nodes, if a malicious node is found.
- Send and retrieve data from the server.
- Alert the user about the analysis.

2.2 Product Functions

Extract Network Information

- Hop count between the two devices
- Number of packets transferred
- Number of packets received
- Size of a packet
- Number of completed transactions
- Duration of a transaction
- Presence of Selfish nodes

2.2.1 Use Case Diagram



2.2.2 Use Case Scenarios

Use Case Name	Access the router
Preconditions	Device should be connected by the network
Success Condition	Interface requesting username and password
Actor	Android Device
Main Success Scenario	1. Have connectivity to the network 2. Display next interface
Extensions	If the request is not authentic, it will get ignored

Use Case Name	Choose path to destination
Preconditions	Device should be connected to the network by router
Success Condition	Interface showing neighboring nodes in the network
Actor	Android Device
Main Success Scenario	1. Received target node's destination 2. Display next interface
Extensions	If the request is not authentic, it will get ignored

Use Case Name	Send insensitive data
Preconditions	Permission to access the device should be granted
Success Condition	Interface requesting what features to extract
Actor	Android Device
Main Success Scenario	1. Have required information about the features 2. Display next interface
Extensions	If the request is not authentic, it will get ignored

2.3 User Characteristics

This product/system is mainly targeted towards, the users who needs a secure environment in their cyber space.

2.4 Constraints

Hardware Constraints

The product will be highly resource hungry, thus in order to get the optimum usage of the product, the android device needs;

- Quad – core processor
- 2GB of RAM

Software Constraints

The product will run on Android Jellybean (4.2) and higher.

2.5 Assumptions & Dependencies

- All the nodes/devices in the network are Android Smartphones.
- Users are aware of the system
- High speed internet connection in devices.
- Bandwidth of the internet connection will not affect the data transfer.
- Users have a non – rooted smart phone.

2.6 Apportioning of Requirements

In sections 1 and 2 of this document the primary specification of the system has been stated i.e. detailed descriptions of exactly what is required; and those in section 3 are functional requirements of the system. The two levels of requirements are intended to be consistent. Inconsistencies are to be logged as defects. In the event that a requirement is stated within both primary and functional specifications, the application will be built from functional specification since it is more detailed. The functional requirements in the section 3 are planned to be implemented to get the system up and running in the first release. Desirable requirements are to be implemented in this release if possible, but are not committed to by the developers. It is anticipated that they will be part of future release. Optional requirements will be implemented at the discretion of developers.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The User Interface section will be added in a future release of the Software Requirement Specification (SRS).

3.1.2 Hardware Interfaces

- Android Smartphone
  The main hardware component for the system. The device will check, if the selected device could be trusted using the deep reinforcement model built.
- NVIDIA Graphic Card
  When building the deep reinforcement model, the computations takes time. For parallel processing we need an NVIDIA GPU

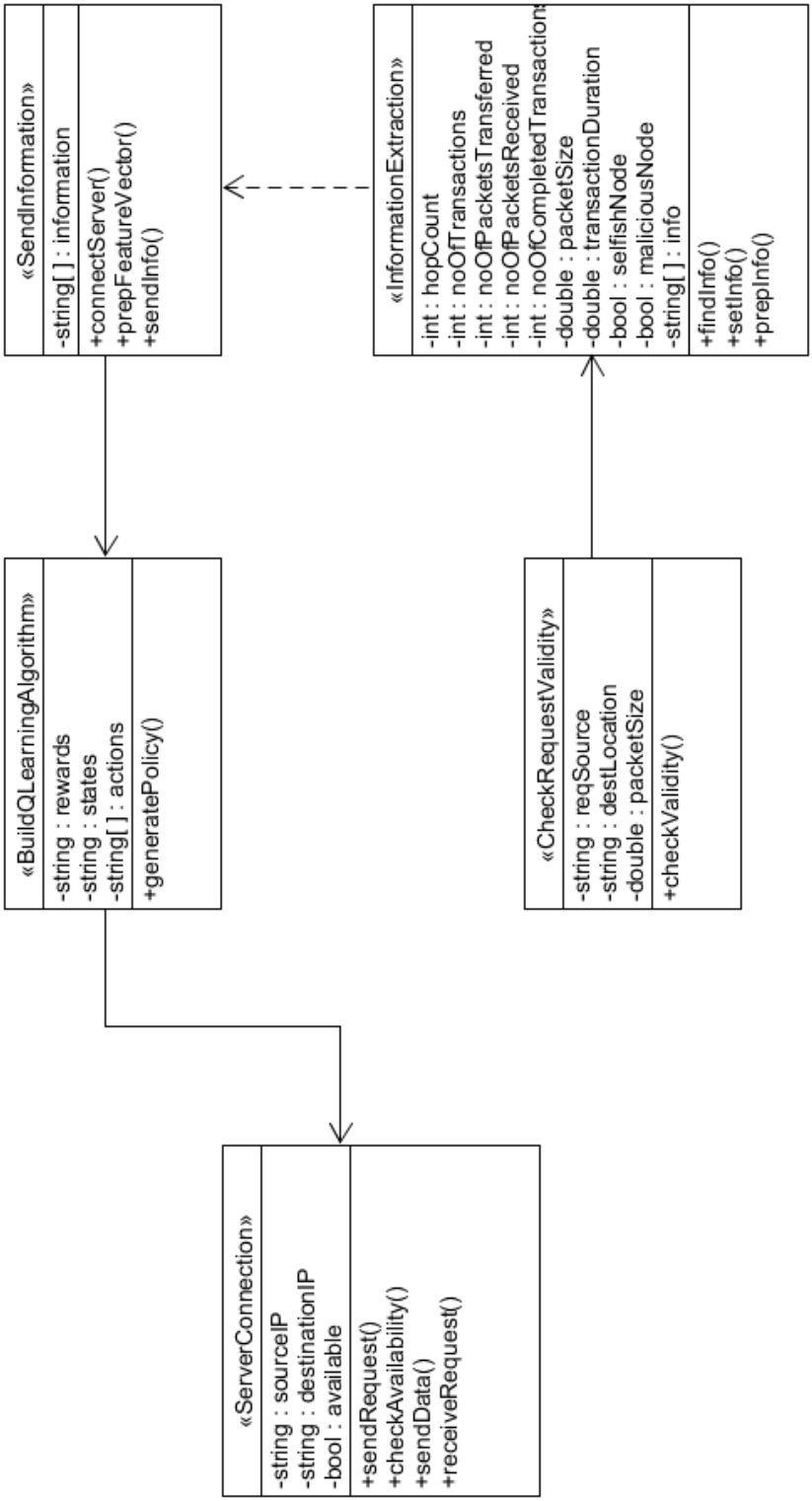
3.1.3 Software Interfaces

- Android Operating System: The system is mainly built for evaluating trust on Android Devices, thus having Android as the operating system is mandatory.
- Cuda: As stated in section 2.1.3, this will be used to configure the NVIDIA GPU.
- Web Server: A server is needed to save and retrieve data gathered

3.1.4 Communication Interfaces

Data transmission among devices and the server will be done using 3G or 4G network. Since the system is run real time, having a high bandwidth is better.

3.2 Classes/Objects



3.3 Performance Requirements

- Supporting Instances

  System will allow only one instance of this service to run on the mobile device at a given time.
- Simultaneous Users

  The application will allow multiple devices to use this service but not from the same mobile device. With several devices, many users can access the server and obtain services.

3.4 Design Constraints

System will use standard principles of designing and it will maintain the consistency throughout. There will be no specific design constraints to the project.

3.5 Software System Attributes

3.5.1 Reliability

This service has a chance of failing due to OS failures such as low battery or system crashes. But since the server is handling most of the computation part results will be accurate almost all the time.

3.5.2 Availability

As long as the server is not down or there is some sort of network problem which will cause devices to reach the server difficult and vice versa the availability of the system will be high.

3.5.3 Security

The messages that will be sent to and from server to the devices will be encrypted, since we deal with very sensitive data that if fallen into wrong hands it will be catastrophic. And information of the mobile devices and their owners will not be obtained without their consent.

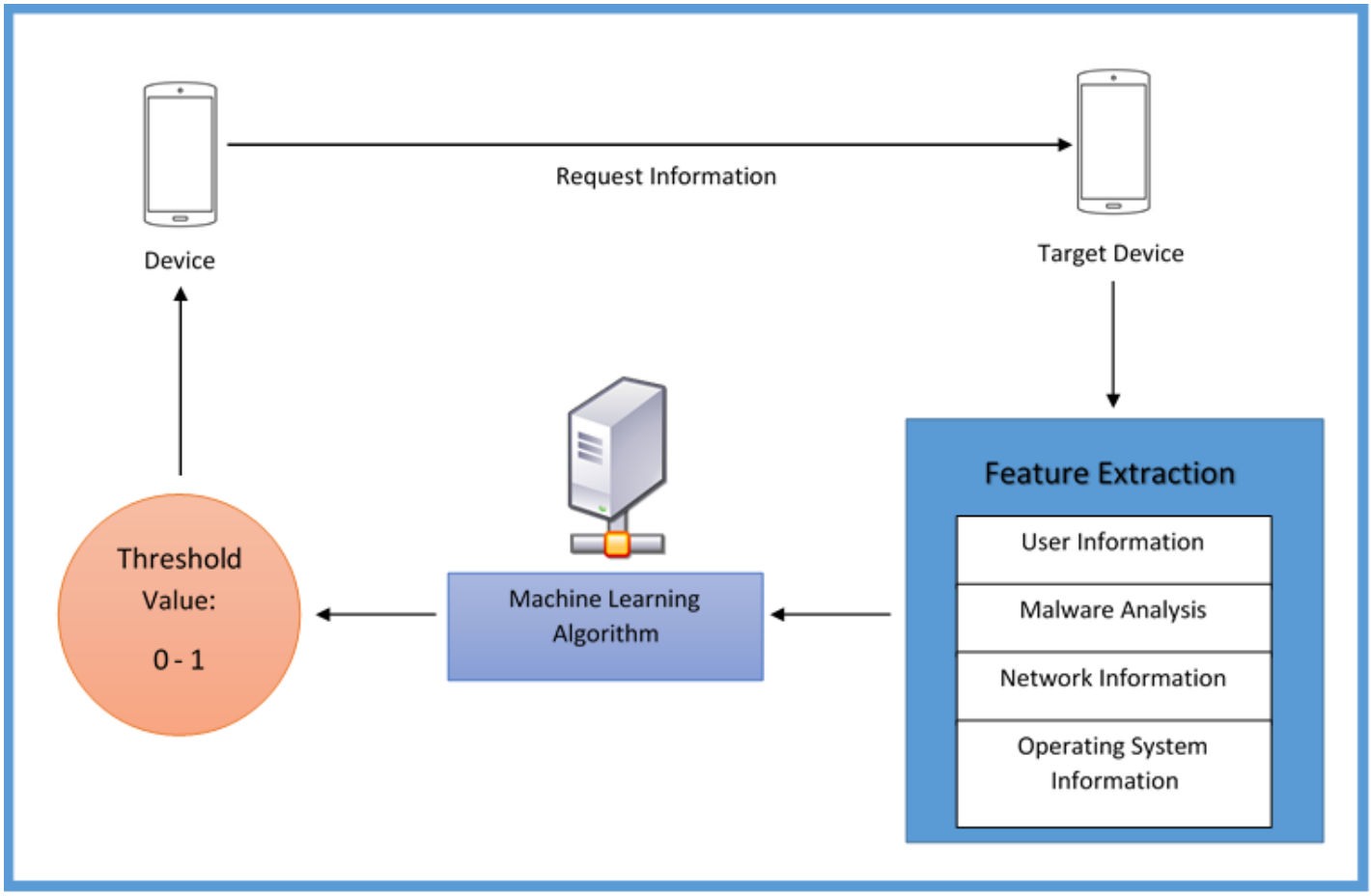
3.5.4 Maintainability

The service will be installed to android devices as an Android Service. As most of the logic and computation is in the server, Server crashes and failures should be fixed.

4. Supporting Information

4.1 Appendices

4.1.1 System Diagram



5. References

[1] "Deep Reinforcement Learning". VUB Artificial Intelligence Lab. N.p., 2016. Web. 15 Sept. 2016.

[2] "Android (Operating System)". Wikipedia. N.p., 2016. Web. 15 Sept. 2016.

[3] "Machine Learning". Wikipedia. N.p., 2016. Web. 15 Sept. 2016.

[4] "Q-Learning". Wikipedia. N.p., 2016. Web. 15 Sept. 2016.

[5] R. Weiss, L. Reznik, Y. Zhuang, A. Hoffman, D. Pollard, A. Rafetseder, T. Li and J. Cappos, "Trust Evaluation in Mobile Devices: An Empirical Study", 2015 IEEE Trustcom/BigDataSE/ISPA, 2015.

[6] K. Zhao and L. Pan, "A Machine Learning Based Trust Evaluation Framework for Online Social Networks", 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, 2014.

[7] F. Bao and I. Chen, "Dynamic Trust Management for Internet of Things Applications", Proceedings of the 2012 international workshop on Self-aware internet of things - Self-IoT '12, 2012.

[8] S. Yerima, S. Sezer and I. Muttik, "Android Malware Detection Using Parallel Machine Learning Classifiers", 2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies, 2014.

[9] Y. Loubna Mekouar, "Detecting malicious peers in a reputation-based peer-to-peer system", Second IEEE Consumer Communications and Networking Conference, 2005. CCNC. 2005.

[10] "Welcome — Theano 0.8.2 Documentation". Deeplearning.net. N.p., 2016. Web. 15 Sept. 2016.