Universitat Politècnica de València
Departament de Matemàtica Aplicada

PhD. THESIS

Mathematical network models applied to the analysis of mobile applications behavior

Ph.D. CANDIDATE

Juan Alegre Sanahuja

ADVISORS

Dr. Juan Carlos Cortés López

Dr. Francisco José Santonja Gómez

Dr. Rafael Jacinto Villanueva Micó

Valencia - April 2016

Dr. Juan Carlos Cortés López, professor at the Universitat Politècnica de València, Francisco José Santonja Gómez, professor at the Universitat de València and Rafael Jacinto Villanueva Micó, professor at the Universitat Politècnica de València,

CERTIFY that the present thesis entitled *Mathematical network models applied to the analysis of mobile applications behavior* has been performed under our supervision in the Department of Applied Mathematics at the Universitat Politècnica de València by Juan Alegre Sanahuja. It constitutes his thesis dissertation to obtain the PhD degree in Mathematics.

In compliance with the current legislation, we authorize the presentation of this dissertation signing the present certificate.

Valencia, April 14, 2016

Juan Carlos	Francisco José	Rafael Jacinto
Cortés López	Santonja Gómez	Villanueva Micó

Resumen en Español

Las estructuras de redes están presentes en multitud de fenómenos sociales, políticos, económicos y tecnológicos. Estas estructuras permiten compartir información, constituir alianzas, influir en comportamientos, generar corrientes de opinión, y transmitir virus, entre otros aspectos.

Las redes on-line son un reflejo del mundo "analógico" y también presentan este tipo de estructura de red, de tal forma que permiten transmitir información, detectar comunidades, predecir afinidades entre individuos, generar recomendaciones, identificar individuos influyentes o producir fenómenos virales. Aunque todas estas redes son de naturaleza heterogénea, la estructura subyacente que presentan permiten su modelización para el estudio y análisis de los fenómenos indicados.

Actualmente, la línea que divide el mundo "analógico" y el mundo online es cada vez más difusa produciéndose estructuras de redes donde se entremezclan ambas naturalezas: Existen casi tantos teléfonos móviles como individuos y, en las sociedades desarrolladas, la omnipresencia de los smartphones en el día a día es incuestionable de tal forma que cualquier persona está conectada casi en todo momento y lugar. Esta conexión permanente conlleva que el individuo constituya simultáneamente y de un modo continuo un nodo de su estructura de red social y de su red social online.

Una parte fundamental de los smartphones son las aplicaciones que se pueden descargar en el dispositivo. Existen multitud de aplicaciones para infinidad de utilidades distintas y el comportamiento del usuario frente a esas aplicaciones es el que determina cómo se comportan dichas aplicaciones. Asimismo, las aplicaciones móviles son la principal fuente de contagio de virus en los smartphones y en este caso, también el comportamiento del usuario es el que determina la transmisión de esos virus. Es decir, el número de descargas de la aplicación, el tiempo de retención de la aplicación sin ser desinstalada, los minutos semanales de uso, la popularidad de la aplicación, la transmisión de virus en smartphones, etc., dependen del comportamiento del usuario y, puesto que el usuario forma parte de una red social "offline" y una red social online, en las cuales se comparte y transmite información, se consti-

tuyen comunidades, se influye en los comportamientos, se generan corrientes de opinión y se transmiten virus, podemos intuir que los comportamientos de las aplicaciones pueden ser modelizados considerando la estructura de red de la que el usuario forma parte, de tal forma que sea posible analizar y estudiar aspectos tales como predecir la descarga y retención de aplicaciones y/o la transmisión de virus entre smartphones.

El propósito de la presente tesis doctoral es modelizar y analizar el comportamiento de las aplicaciones móviles mediante estructuras de red. El comportamiento de las aplicaciones móviles vendrá definido por la red formada por los usuarios, teniendo en cuenta tanto parámetros de comportamiento de los usuarios como parámetros relacionados con aspectos técnicos de los dispositivos móviles, por lo que para la modelización de las redes se tendrán en cuenta ambos factores.

La estructura de esta memoria es la siguiente: En el capítulo 1 introduciremos el problema a estudiar.

En el capítulo 2 presentaremos un primer modelo de red. En este capítulo, consideraremos la influencia de la red social del usuario a la hora de descargarse una aplicación móvil y, puesto que la influencia y otros contagios sociales han sido modelizados con éxito mediante modelos epidemiológicos, proponemos un modelo de red epidemiológica aleatoria cuyas simulaciones permitirán predecir el comportamiento de una aplicación.

En el capítulo 3, presentaremos un segundo modelo de red. En este caso propondremos un modelo de agentes para cuantificar la transmisión de virus en smartphones considerando el comportamiento de los usuarios. Mediante simulaciones de este modelo, podremos predecir la propagación de virus en smartphones, el coste que conlleva para los usuarios, así como analizar la parte crítica en la transmisión de virus para smartphones: el comportamiento del usuario o cuestiones técnicas relacionadas con los dispositivos.

Finalmente, en el capítulo 4, se presentan las conclusiones de la presente tesis doctoral.

Resum en Valencià

Les estructures de xarxes estàn presents en multitud de fenòmens socials, polítics, econòmics i tecnològics. Estes estructures permeten compartir informació, constituir aliances, influir en comportaments, generar corrents d'opinió, i transmetre virus, entre altres aspectes.

Les xarxes online són un reflex del món analògic i també presenten este tipus d'estructura de xarxa, de tal forma que permet transmetre informació, detectar comunitats, predir afinitats entre individus, generar recomanacions, identificar individus influents o produir fenòmens virals. Encara que totes estes xarxes són de naturalesa heterogènia, l'estructura subjacent que presenten permeten la seua modelització per a l'estudi i anàlisi dels fenòmens indicats.

Actualment, la línia que dividix el món analògic i el món online és cada vegada més difusa produint-se estructures de xarxes on s'entremesclen ambdós naturaleses: Existixen quasi tants telèfons mòbils com individus i, en les societats desenvolupades, l'omnipresència dels smartphones en el dia a dia és inqüestionable de tal forma que qualsevol persona està connectada quasi en tot moment i lloc. Esta connexió permanent comporta que l'individu constituïsca simultàniament i d'una manera contínua un node de la seua estructura de xarxa social i de la seua xarxa social online.

Una part fonamental dels smartphones són les aplicacions que es poden descarregar en el dispositiu. Hi ha multitud d'aplicacions per a infinitat d'utilitats distintes i el comportament de l'usuari enfront d'eixes aplicacions és el que determina com es comporten aquestes aplicacions. Així mateix, les aplicacions mòbils són la principal font de contagi de virus en els smartphones i en este cas, també el comportament de l'usuari és el que determina la transmissió d'eixos virus. És a dir, el nombre de descàrregues de l'aplicació, el temps de retenció de l'aplicació sense ser esborrada, els minuts setmanals d'ús, la popularitat de l'aplicació, la transmissió de virus entre smartphones, etc., depenen del comportament de l'usuari i, ja que l'usuari forma part d'una xarxa social "offline" i una xarxa social online, en les quals es compartix i es transmet informació, es constituïxen comunitats, s'influïx en els compor-

taments, es generen corrents d'opinió i es transmeten virus, podem intuir que els comportaments de les aplicacions poden ser modelitzats considerant l'estructura de xarxa de què l'usuari forma part, de tal forma que siga possible analitzar i estudiar aspectes com ara predir la descàrrega i retenció d'aplicacions i/o la transmissió de virus entre smartphones.

El propòsit de la present tesi doctoral és modelitzar i analitzar el comportament de les aplicacions mòbils per mitjà d'estructures de xarxa. El comportament de les aplicacions mòbils vindrà definit per la xarxa formada pels usuaris, tenint en compte tant paràmetres de comportament dels usuaris com paràmetres relacionats amb aspectes tècnics dels dispositius mòbils, per la qual cosa per a la modelització de les xarxes es tindràn en compte ambdós factors.

L'estructura d'esta memòria és la següent. En el capítol 1 introduirem el problema a estudiar.

En el capítol 2 presentarem un primer model de xarxa. En este capítol, considerarem la influència de la xarxa social de l'usuari a l'hora de descarregarse una aplicació mòbil i, ja que la influència i altres contagis socials han sigut modelitzats amb èxit per mitjà de models epidemiològics, proposem un model de xarxa epidemiològica aleatòria la qual permetrà predir el comportament d'una aplicació.

En el capítol 3, presentarem un segón model de xarxa. En este cas proposarem un model d'agents per a quantificar la transmissió de virus en smartphones considerant el comportament dels usuaris. Per mig de simulacions d'este model, podrem predir la propagació de virus en smartphones, el cost que comporta per als usuaris, així com analitzar la part crítica en la transmissió de virus per a smartphones: el comportament de l'usuari o qüestions tècniques relacionades amb els dispositius.

Finalment, en el capítol 4 es presenten les conclusions de la present tesi doctoral.

Abstract in English

The network topologies are present in different social, political, economic and technological phenomena. These network structures allow to share information, alliances generation, behavior influence, opinion spread and virus transmission, among other aspects.

Online networks are a reflection of the offline world and they also show these kind of network structures, in such a way that they allow the information transmission, social circle or community detection, affinity prediction between individuals, generation of recommendations, detection of influence people and generation of viral phenomena. Although all of these networks exhibit heterogeneity, they have enough underlying structure to allow their modelization for the study and analysis of all the listed phenomena.

Nowadays, the line between the offline world and the online world is becoming more diffuse and there are network structures where both natures are mixed: There are almost as many mobile phones as individuals and in developed societies, the pervasiveness of smartphones on day-to-day is unquestionable in such a way that almost everybody is almost always connected everywhere. This permanent connection means that the individual, simultaneously and in a continuous mode, is a node belonging to its social network and its social network online.

A key aspect of smartphones are the mobile applications that can be downloaded to the device. There are many applications for a host of different uses and the user behavior with these applications is the factor that determines how these applications behave. Also, mobile applications are the main source of infection of viruses on smartphones and, in this case, also the user behavior is what determines the transmission of these viruses. That is, the number of downloads of the application, the retention time of the application without being uninstalled, weekly minutes of usage, the popularity of the application, the transmission of viruses between smartphones, etc., depend on user behavior and, since the user is part of a social "offline" network and a social online network, in which the information is shared, communities are generated, behavior is influenced, opinion is spread and viruses are

transmitted, we can intuit that the application behaviors can be modeled considering the network structure which user belongs to, so it is possible to analyze and study issues such as predicting the retention and download of applications and/or the transmission of viruses between smartphones.

The purpose of this thesis is to analyze the behavior of mobile applications through mathematical network models. The behavior of mobile applications will be defined by the network of the users, taking into account parameters such as user behavior and technical issues of the mobile devices, so for model the networks both factors will be taken into account.

The structure of this PhD thesis is as follows. In chapter 1, we introduce the problem to be studied.

In chapter 2, we present a first network model. In this chapter, we will consider the social influence when downloading a mobile application. The influence and other social contagions have been modeled successfully by epidemiological models so we propose an epidemiological random network model whose simulations allow us to predict the behavior of an application.

In chapter 3, we present a second network model. In this case, an agent-based model is proposed to quantify the spread and transmission of viruses on smartphones considering the behavior of the users. With the simulations of this model, we can predict the spread of viruses on smartphones, we will estimate the cost the users should afford when the malware is in their devices and we will analyze the most critical part in the transmission of viruses for smartphones: user behavior or technical issues related to the devices.

Finally, in chapter 4 the conclusions of this Ph.D. thesis are presented.

Contents

1 Introduction **17**
1.1 Overview of the dissertation 19

2 SAMOA I (Spread Analysis for MObile Apps): Epidemiological random network model to predict the spread of mobile apps **22**
2.1 Introduction 23
2.2 Material and Methods 25
2.2.1 Model 25
2.2.2 Parameters 25
2.2.3 Data 27
2.2.4 Simulations and selections 27
2.3 Results and discussion 29
2.3.1 Additional results 33
2.4 Conclusion 53
2.5 Appendix to Chapter 2 54
2.5.1 What was said recently by major actors in the mobile apps world 54
2.5.2 Web page SAMOA I model 56

3 SAMOA II (Spread Analysis for Malware On Android): Agent-based model to study and quantify the evolution dynamics of Android malware infection **57**
3.1 Introduction 58
3.1.1 State of the art 59
3.1.2 Proposed model 59
3.2 Material and methods 61
3.2.1 The Apps 62
3.2.2 Official market 63
3.2.3 Non–official market 65
3.2.4 Users 67

3.2.5 Methods 72
3.3 The App-Model evolution rules 73
3.4 Results and discussion 75
3.4.1 Model evolution depending on the number of users . . 76
3.4.2 Estimations 77
3.4.3 Model validation 79
3.5 Conclusion 81

4 Conclusion **84**

List of Figures

2.1 Flow chart of the SIR model applied. 25
2.2 App1. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data $d(4)$ and $d(11)$ corresponding to the 4-th and 11-th days, respectively. 29
2.3 App1. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data $d(4)$, $d(11)$ and $d(14)$ corresponding to the 4-th, 11-th and 14-th days, respectively. 30
2.4 App1. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(11)$, $d(14)$ and $d(21)$ corresponding to the 11-th, 14-th and 21-th days, respectively. 31
2.5 App2. Predicted behavior and 95% confidence interval of number of accumulated downloads vs. time, based on data from $d(6)$ and $d(8)$ corresponding to the 6-th and 8-th days, respectively. 32
2.6 App2. Predicted behavior and 95% confidence interval of number of accumulated downloads vs. time, based on data from $d(6)$, $d(8)$ and $d(13)$ corresponding to the 6-th, 8-th and 13-th days, respectively. 32
2.7 App3. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(5)$ and $d(7)$ corresponding to the 5-th and 7-th days, respectively. 34
2.8 App3. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(5)$, $d(7)$ and $d(11)$ corresponding to the 5-th, 7-th and 11-th days, respectively. 34

2.9 App3. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(5)$, $d(7)$, $d(11)$ and $d(14)$ corresponding to the 5-th, 7-th, 11-th and 14-th days, respectively. 35
2.10 App4. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(6)$ and $d(8)$ corresponding to the 6-th and 8-th days, respectively. 36
2.11 App4. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(6)$, $d(8)$ and $d(13)$ corresponding to the 6-th, 8-th and 13-th days, respectively. 37
2.12 App4. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(8)$ and $d(13)$ corresponding to the 8-th and 13-th days, respectively. 37
2.13 App5. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(5)$ and $d(7)$ corresponding to the 5-th and 7-th days, respectively. 38
2.14 App5. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(5)$, $d(7)$ and $d(12)$ corresponding to the 5-th, 7-th and 12-th days, respectively. 39
2.15 App6. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(3)$ and $d(10)$ corresponding to the 3-th and 10-th days, respectively. 40
2.16 App6. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(3)$, $d(10)$ and $d(13)$ corresponding to the 3-th, 10-th and 13-th days, respectively. 41
2.17 App7. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(4)$ and $d(6)$ corresponding to the 4-th and 6-th days, respectively. 42
2.18 App7. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(4)$, $d(6)$ and $d(9)$ corresponding to the 3-th, 10-th and 13-th days, respectively. 42

2.19 App8. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(4)$ and $d(6)$ corresponding to the 4-th and 6-th days, respectively. 43
2.20 App8. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(9)$, $d(10)$ and $d(13)$ corresponding to the 9-th, 10-th and 13-th days, respectively. 44
2.21 App9. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(5)$ and $d(7)$ corresponding to the 5-th and 7-th days, respectively. 45
2.22 App9. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(7)$, $d(10)$ and $d(11)$ corresponding to the 7-th, 10-th and 11-th days, respectively. 46
2.23 App9. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(7)$, $d(10)$, $d(11)$ and $d(14)$ corresponding to the 7-th, 10-th, 11-th and 14-th days, respectively. 46
2.24 App10. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(2)$ and $d(6)$ corresponding to the 2-th and 6-th days, respectively. 47
2.25 App10. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(6)$ and $d(8)$ corresponding to the 6-th and 8-th days, respectively. 48
2.26 App11. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(3)$ and $d(7)$ corresponding to the 3-th and 7-th days, respectively. 49
2.27 App11. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(7)$ and $d(9)$ corresponding to the 7-th and 9-th days, respectively. 50
2.28 App12. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(3)$ and $d(6)$ corresponding to the 3-th and 6-th days, respectively. 51

2.29 App12. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(6)$ and $d(20)$ corresponding to the 6-th and 20-th days, respectively. 51
2.30 App13. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(2)$ and $d(4)$ corresponding to the 2-th and 4-th days, respectively. 52
2.31 App13. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(2)$, $d(4)$ and $d(7)$ corresponding to the 2-th, 4-th and 7-th days, respectively. 53

3.1 General structure of the agent-based model. Issues we are going to take into account in the modelling process. 60
3.2 Agent attributes and functions. 62
3.3 Downloads per popularity. 69
3.4 App-Model flowchart. In this figure we describe the evolution process of the model from $t = 0$ (Start point) to $t = T$ (End point), showing, for every time instant, the creation of the agents, the assignation of attributes, the order of performance of the methods and their interaction. 76
3.5 Model evolution of the cumulative smartphone infections every month since Jul 2011 until Dic 2014. The line in the middle is the mean and those up and down correspond to 95% confidence interval. 78
3.6 Evolution of new smartphone infections every month since Jul 2011 until Dic 2014. The line in the middle is the mean and those up and down correspond to 95% confidence interval. Nowadays, there is a stabilization in the number of new infected smartphones. 80
3.7 Evolution of the cumulative smartphone infections due to Privilege Escalation (PE) on the left, and Financial Charge (FC) on the right, malware every month since Jul 2011 until Dic 2014. The line in the middle is the mean and those up and down correspond to the 95% confidence interval. 81

List of Tables

2.1 App1. Total number of accumulated downloads in eight different days. . . . 29
2.2 App2. Total number of accumulated downloads during six different days. . . . 31
2.3 App3. Total number of accumulated downloads during eight different days. . . . 33
2.4 App4. Total number of accumulated downloads during six different days. . . . 36
2.5 App5. Total number of accumulated downloads during six different days. . . . 38
2.6 App6. Total number of accumulated downloads during five different days. . . . 40
2.7 App7. Total number of accumulated downloads during seven different days. . . . 41
2.8 App8. Total number of accumulated downloads during nine different days. . . . 43
2.9 App9. Total number of accumulated downloads during nine different days. . . . 45
2.10 App10. Total number of accumulated downloads during four different days. . . . 47
2.11 App11. Total number of accumulated downloads during four different days. . . . 49
2.12 App12. Total number of accumulated downloads during four different days. . . . 50
2.13 App13. Total number of accumulated downloads during five different days. . . . 52
2.14 Sources of awareness of smartphone apps according to Google surveys. . . . 55
2.15 Reasons for downloading an app according to Google surveys. 56

3.1 Number of Apps in the official market. . . . 63
3.2 Number of malware Apps in the official market. . . . 64

3.3 Distribution of Apps by popularity in Jul 2011. 64
3.4 Distribution of malware Apps by popularity in July 2011, taking into account repackaging. 65
3.5 Distribution of malware Apps in the official by market according to their type. 65
3.6 Number of malware apps in the non-official market. 66
3.7 Distribution of Apps by popularity in in July 2011 in the non-official market. 66
3.8 Distribution of malware Apps by popularity in July 2011 in the non-official market. 67
3.9 Distribution of malware Apps in the non-official market according to their type. 67
3.10 Estimation of the percentage of download distribution of Android Apps per number of downloads. 68
3.11 Distribution of OS version in Android smart-phones from July 2011 until Feb 2013. 71
3.12 Percentage of devices that can be affected by the most common privilege escalation vulnerabilities, depending on the Android OS version. 71
3.13 Comparison of percentage of cumulative and residual infected users for month $t = 15$. The results are very similar. 77
3.14 Mean and CI95% of the accumulated infected smartphones in Jul 2013, Jul 2014 and Dic 2014 in the Community of Valencia and the corresponding percentages predicted by the model. . . 79
3.15 Mean and CI95% of the residual infected smartphones in Jul 2013, Jul 2014 and Dic 2014 in the Community of Valencia and the corresponding percentages predicted by the model. . . 79
3.16 Mean and CI95% of the accumulated infected smartphones by Privilege Escalation and Financial Chage in Jul 2013, Jul 2014 and Dic 2014 in the Community of Valencia and the corresponding percentages predicted by the model. These figures can give us an idea about the amount of money that the Financial Charge malware moves every month. 82

Chapter 1

Introduction

The ubiquity of smartphones for personal and business use is clear and growing every year. The worldwide smartphone market grew 13% year over year in the second quarter of 2015, with 341.5 million shipments and, in 2014, sales of smartphones worldwide topped 1.2 billion, which was up 28% from 2013 [50].

Furthermore, the combination of personal computer features with mobile and handheld features of the smartphones, produces the pervasiveness of smartphones on day-to-day in such a way that almost everybody is almost always connected everywhere. This permanent connection means that the individual, simultaneously and in a continuous mode, can be seen as a node belonging to its social network and its social network online.

The social network structures allow to share information, alliances generation, behavior influence, opinion spread and virus transmission, among other aspects. Online networks are a reflection of the offline world and they also show the effects of network structures, in such a way that they allow the information transmission, social circle or community detection, affinity prediction between individuals, generation of recommendations, detection of influence people and generation of viral phenomena.

Although these networks exhibit heterogeneity, they have enough underlying structure to allow their modelization for the study and analysis of all the listed phenomena.

The traceability and availability of mobile phone datasets has opened the possibility to improve our understanding of large-scale social networks by investigating how people exchange information, build trust, create markets and develop social interactions. Mobile phone datasets can also be used to analyze mobility and better understand complex processes such as the spread of information and viruses or transportation and the use of urban infrastructures.

Many studies about phone datasets are available related with different fields as human mobility, economics, social sciences, demographic and urban studies, etc. [52] covering many different areas such as the study of spatio-temporal distribution of people, measure movements and migrations, trasmission of diseases, rural electrification planning in developing countries, energy consumption prediction using people dynamics or quantification of urban economic activity, between others [32, 33]. All of these studies are related with the analysis of phone record data collected by cell phone providers and considering the individual belonging to his/her "analog" social network.

In this dissertation, we focus our study in smartphones, and, specifically, in the mobile applications or apps that can be downloaded and installed on the device. For this work, we do not need phone record data from the cell phone providers companies as in most of the papers and works related with mobile phones and social networks but we will need other kind of parameters related with the user behavior, his/her mobile device and the apps installed on it.

An app is a computer program designed to run on mobile devices such as smartphones that are not preinstalled and are usually available through application distribution platforms. There are millions of mobile applications for a host of different uses available for downloading by millions of users at app markets [45] and the user behavior with these applications is the factor that determines how these applications behave. That is, the number of downloads, the retention time of the application without being uninstalled, weekly minutes of usage, the popularity of the application, etc., depend on user behavior. Since the user is part of a social "offline" network and a social online network, where the information is shared, communities are generated, behavior is influenced, opinion is spread and viruses are transmitted, we can intuit that the application behavior can be modeled considering the network structure which user belongs to, so it is possible to analyze and study issues such as predicting the retention and download of applications and/or the transmission of viruses between smartphones.

The behavior of mobile applications will be defined by the network of the users, taking into account parameters such as user behavior and technical issues of the mobile devices. Then, for model the networks both factors will be taken into account.

In the literature, there are not many contributions focused on the mobile applications and the influence of network effects on the users' adoption or download of the apps [28, 1, 23]. Furthermore, these contributions are based on a small group of participants. To the best of our knowledge, there is a lack of experiments carried out with big network models simulations, comparing the results with data from real app markets.

By the other hand, apps are the main contributors to the spread of mobile malware caused by malware applications

In the literature, there are several approaches to the mathematical modeling for the spread of viruses on mobile devices. In [11] the authors describe a framework and the main guidelines to design reliable agent-based malware models considering infections via SMS/MMS, Bluetooth RF, IM, P2P and email. In [19, 25, 29] the authors propose approaches based on mathematical epidemic techniques where the malware infection follows similar dynamics to the infectious diseases. Also, there are models based on the physical architecture of the mobile and wireless networks [14] or on the mobility of the users, but they do not consider the interconnectivity based on the exchange of applications [25]. To the best of our knowledge, none paper showing quantification, prediction and/or simulation about how the users install malware apps. Nevertheless, any of the above approaches do not take into account the infection model considering an app-market ecosystem, like smartphones environment is.

With the models presented in this work, and taking into account the mixed nature network structure, analog and online, which user belongs to, we will be able to analyze the main effects of a social network structure, i.e., information sharing, communities generation, behavior influence, opinion spreading and viruses transmission, related with the mobile applications.

For the simulation tasks of our models, we will use large networks running on large computational facilities that will allow us to execute many simulations with multiple parameter sets in order to compute reliable estimations based on 95% confidence intervals.

1.1 Overview of the dissertation

In this dissertation, our objective is to show the work and results obtained using mathematical network models to analyze the mobile applications behavior.

We start in Chapter 2 where our first modeling approach is presented. The name of this first model is SAMOA I, from Spread Analysis for MObile Apps. In this model, we will consider the social influence when downloading a mobile application. The influence and other social contagions have been modeled successfully by means of epidemiological models thus we propose an epidemiological random network model whose simulations allow us to predict the behavior of applications. For modeling simulations, we will use a 1,000,000 nodes network and a set of variable parameters that will be:

- The number of nodes having the application already installed on his/her

device.

- The user retention rate, that is, the time that the user has the app installed on his/her device.
- The "infection" rate of the application.

In order to compute reliable estimations based on 95% confidence intervals (CI 95%), the technique referred to as Latin Hypercube Sampling (LHS) will be used [15]. This technique will be applied to select sets of the variable parameters to be substituted into the model. LHS, a type of stratified Monte Carlo sampling, is an efficient method for achieving equitable samples of all input parameters simultaneously. In our problem, by LHS we obtain an equitable sample of 100,000 input parameters simultaneously. We substitute each set of the 100,000 parameters into the model and then we run a simulation. The set of results obtained from the simulations represent all the possible behavior of an app according to the considered parameters. After performing these simulations, a set of scenarios will be generated.

Based on the number of downloads over the time for a real app, then we should be able to select the behavior from our set of scenarios that best fit the real behavior of the app. In this manner, the evolution curve of that app will be estimated.

The main challenges of this model are, first, obtaining reliable parameters for the model simulations and, second, the computational effort involved in the simulation of such a big network model.

In Chapter 3 we present the second modeling approach. The name of this second model is SAMOA II, from Spread Analysis for Malware On Android. The proposed model will be an agent-based model to quantify the spread and transmission of viruses on smartphones considering the behavior of the users. An agent-based model is a computational model for simulating the actions and interactions between autonomous agents. In this case, we have selected this kind of model because we have the users, with their own behavior related with their devices, i.e.:

- The number of apps the user downloads per month.
- The Operative System version that the user has got in his/her device.
- The protection (antivirus or not) that the user has got installed on his/her device.
- The malware detection by the user.

And, on the other hand, we have the markets, with their own behavior related with the apps that are in the market, i.e.:

- The number of new apps entering every month to the market.
- The number of malware apps entering every month to the market.
- The distribution of apps according to their popularity.
- The distribution of malware apps according to their popularity.
- The malware detection by the market.
- The distribution of malware apps according to their type.
- The type of malware.

So, in this model, two domains with their own agents will be considered: the markets, where the agents are the apps that belong to different distribution platforms, and the users, where the agents are the mobile devices that belong to every user. Every agent will have its own behavior with different parameters.

With the simulations of this model, we will predict the spread of viruses on smartphones, then we will estimate the cost the users should afford when the malware is in their devices and, finally, we will analyze the most critical part in the transmission of viruses for smartphones, user behavior or technical issues related to the devices.

For modeling simulations, we will use a network made up of 50,000 nodes and an equitable sample of 100,000 input parameters, obtained with LHS technique, to run the model to compute estimations based on 95% confidence intervals.

The main challenges of this model are to obtain reliable parameters for model simulations and the computational effort for the simulation of such a big network model.

Finally, in Chapter 4 we enumerate the main goals achieved by this dissertation.

Chapter 2

SAMOA I (Spread Analysis for MObile Apps): Epidemiological random network model to predict the spread of mobile apps

In this chapter, our objective is to model the evolution of mobile apps spread. In app marketing, a key issue is to predict future app installations and the influence of the peers seems to be very relevant when downloading apps. Therefore, the study of the evolution of mobile apps spread may be approached using a proper network model that considers the influence of peers. Influence of peers and other social contagions have been successfully described using models of epidemiological type. Hence, in this chapter we propose an epidemiological random network model with realistic parameters to predict the evolution of downloads of apps. The name of the model, SAMOA, comes from Spread Analysis for MObile Apps, and, with this model, we are able to predict the behavior of an app in the market in the short-term looking at its evolution in the early days of its launch. The numerical results provided by the proposed network are compared with data from real apps. This comparison shows that predictions improve as the model is feedback. Marketing researchers and strategy business managers can benefit from the proposed model since it can be helpful to predict app behavior over the time anticipating the spread of an app.

2.1 Introduction

In 2014, more than one billion smartphones were shipped [50] and the sales of smartphones grew 20% in the third quarter of 2014 [47] being millions the number of applications (apps) available for downloading by millions of users at app markets [45].

The app business is a really big market growing constantly and, in app marketing, one key issue is to predict future app installations. In the literature, there are studies that examine how the information spreads in implicit networks [30] or related with the network effect on information dissemination on social network sites as explained in [18]. Specifically about mobile apps, there are contributions that examine how the adoption (downloads) of the apps is influenced by others in their social network [28] and several approaches to model the proliferation growth of apps over the users [1, 23], where the network effects in users' app downloads have been studied.

In [23], the authors use a composite network model, comprised by a call-log network, a Bluetooth proximity network, a friendship network, an affiliation network plus a network that takes into account the exogenous factors, like app popularity. The data used to validate the model came from a sample of 55 students.

In [28], a sample of 180 students was surveyed about their usage of apps and the results were analyzed to examine the influence of social contacts on the use of apps. The results show this influence, being the most significant app advisors friends and family members.

In [1], a sample of 200 participants was considered. The data was collected via a passive data collection software platform that registered Bluetooth proximity hits by closeness and via surveys. One of the conclusions of this work is that one should be cautious in using declared friendship networks to infer the spreading of smartphone apps and for applying viral marketing strategies, since the face-to-face interaction seems to have stronger correlation with app diffusion.

Thus, all these previous contributions claim for:

- Social networks play an important role in consumers' decisions to download and use mobile apps [28].
- The adoption of mobile apps appears to spread via social contagion [28].
- People who spend more time in face-to-face interaction are more likely to share common apps [1].

- Face-to-face interaction has a strong correlation with app dissemination [1].
- There are strong network effects in app installation patterns [23].

However, in [1] and [23], the experiments are based on a small group of participants. In [28], it is recognized that the sample size for the study is relatively small and the generalization of the results is limited. To the best of our knowledge, there is a lack of experiments carried out with big network simulations and with multiple repetitions, comparing the results with data from real app markets.

Taking into account the above comments and how social contagion has been successfully studied using models of epidemiological type [26, 12], in this chapter we propose an epidemiological random network model to estimate the evolution of downloads of the apps over a theoretical random network, analysing the potential spread of the apps and comparing the theoretical results with real data coming from real downloaded apps. One of the main contributions of this chapter is that the network simulation was ran over a big theoretical random network of 1,000,000 members and was repeated 100,000 times with different sets of realistic parameters via computational methods. The resulting set of simulations from running the model multiple times provided us a bank of possible behaviors. This bank of potential behaviors allows us to predict the future behavior of an app looking at its evolution in the early days of its launch.

To conduct our study and compare our results with real data, we have followed the evolution of apps in a real Android app marketplace, where the exact number of downloads was available [56]. The monitorized apps have been randomly chosen, among free apps, and the results have been scaled for comparison with available real data. The scaling has to be done because the potential public for each app is very different. For instance, game apps have usually more downloads than more specific ones.

Marketing researchers and strategy business managers can benefit from the model proposed in this chapter since it can be helpful to predict the app behavior over the time and then anticipating the spread of an app.

This chapter is organized as follows. In Section 2.2 we present the model building, model parameters, data used for modeling, simulations and comparison methods. Section 2.3 is devoted to present results and their discussion. Conclusions are drawn in Section 2.4.

2.2 Material and Methods

2.2.1 Model

Taking into account the network effects suggested by [28, 1, 23], we will build our model as a SIR-type epidemiological random network. The nodes will be the users and the edges will be the face-to-face relations between users. A user gets infected if he/she downloads the app; susceptible or not infected when he/she has never downloaded the app and, he/she will become recovered (and hence immune) when removes the app from his/her device. The number of initial infected nodes will be random; the infection rate will be defined based on the face-to-face relation between nodes and, the recovery rate will be based on the user app retention. A flow chart diagram for the SIR model applied in the chapter is shown in Figure 2.1.

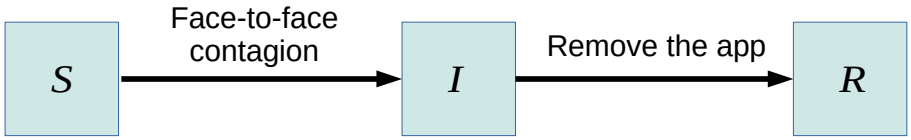

Figure 2.1: Flow chart of the SIR model applied.

Regarding our model, we will assume that other mechanisms for app adoption different from face-to-face relation as described in [23], i.e., exogenous factors due to app popularity and spontaneous app installation after browsing an app market by the user, are weaker and, hence, less significant than face-to-face relations. Comparing our results with real data, we will be able either to validate or reject this assumption.

2.2.2 Parameters

To build our theoretical random network, we consider a population of 1,000,000 users. For the number of edges (or users' friends), we will be based upon the results obtained in [21], regarding the face-to-face friends relations in Spanish population between 15 and 20 years old. According to this latter paper, we will consider a mean network degree of $k = 13.25$ friends with standard deviation of 8.27. We focus on teenage people because this group is considered as Mobile Addicts, i.e., a consumer that launches apps more than 60 times per day, and then, they are the most susceptible to app infection by others than any other age group [38]. The number of friends for every user will

be assigned randomly, generating for every user a random number of friends from the normal (Gaussian) distribution with mean $\mu = 13.25$ and standard deviation $\sigma = 8.27$, $N(\mu = 13.25; \sigma = 8.27)$.

To do this, we sample k from $N(\mu = 13.25; \sigma = 8.27)$. Then, the number of edges in the network is $e = [k \times 1,000,000)/2]$, where $[\,]$ denotes the integer part function. In order to assign the e edges, we select two nodes randomly. If there is not a previous edge between them, then we assign the edge to these two nodes. Otherwise, we select another couple of nodes. We repeat this process until the e edges have been assigned. Thus, our random network is an Erdös-Rényi random network, where all edges are independent [10].

To simulate the network evolution we will need to set the simulation time, the infection rate and the user retention rate parameters.

For the simulation time, we consider $t_s = 100$ days timeline. This decision is made because, as shown in [35], the Android OS app half-life is 3 months. In [35], the half-life of an app is defined as the time instant at which the number of users has declined 50% with respect to its maximum value throughout its lifetime. After this point, the *virality* or infectiousness of the app is weaker.

Retention rates of apps by users at 30, 60 and 90 days, are determined in reference [34]. We can express these rates as the probability that a user retains the app more than 30, 60 or 90 days, i.e., $P[X \geq 30]$, $P[X \geq 60]$ and $P[X \geq 90]$, respectively. Assuming that the retention time X has an exponential distribution of parameter $\lambda > 0$, and, since $P[X \leq x] = 1 - P[X \geq x]$, then we can calculate $P[X \leq 30]$, $P[X \leq 60]$ and $P[X \leq 90]$ as follows

$$f(x) = P[X \leq x] = 1 - e^{-\lambda x}, \qquad \lambda > 0\,, \tag{2.1}$$

being x the time the user has the app downloaded in his/her device, and $\lambda > 0$ the parameter needed to estimate the user retention days. Taking into account the values of x for different types of apps given in [34], we obtain the λ values satisfying the function $f(x)$ in Eq. (2.1) for 30, 60 and 90 days. Hence, we obtain an interval for λ values that will be between 0.008273 (for apps with high user retention rate) and 0.03539 (for apps with low user retention rate).

We assume that the infection rate parameter, β, will be a function of k/t_s,

$$\beta = \delta \frac{k}{t_s}\,, \tag{2.2}$$

where, as it has been previously defined, k and t_s, are the network mean

degree and the simulation time, respectively, and $\delta > 0$ is a tuning parameter. We will consider values for δ in the interval $[0, 0.65]$ in order to cover as many scenarios as possible.

2.2.3 Data

In order to compare our model simulations with real data, we have monitored several apps. They have been randomly chosen from [56]. These apps and their number of accumulated downloads in some dates are collected in Tables 2.1 and 2.2.

2.2.4 Simulations and selections

For modeling simulations, we use 1,000,000 of nodes and the variable parameters will be:

- The number of initial infected nodes: A random integer number generated uniformly in the interval $[1, 50]$. From an epidemiological point of view, a natural candidate for the number of initial infected nodes would be very small $(1 - 5)$, however considering a real context to our problem this number can be greater because the companies can use promotion campaigns where the app is offered for free use among some selected customers. Here, we will assume this number lying in the interval $[1, 50]$.

- The user retention rate: A random number, $\lambda > 0$, uniformly generated in the interval $[0.008273, 0.03539]$ that appears in Eq.(2.1).

- The infection rate: A random number generated uniformly in the interval $[0, 0.65]$ being δ the parameter that appears in Eq.(2.2).

In order to compute reliable estimations based on 95% confidence intervals (CI 95%), the technique referred to as Latin Hypercube Sampling (LHS) will be used, [15]. This technique will be applied to select sets of the variable parameters to be substituted into the model. In our problem, by LHS we obtain an equitable sample of 100,000 input parameters simultaneously. We substitute each set of the 100,000 parameters into the model and then we run a simulation. The set of results from the obtained simulations represent all the possible behavior of an app according to the considered parameters. After performing these simulations, a set of scenarios will be generated.

Based on the number of downloads over the time for a real app, then we should be able to select the behavior from our set of scenarios that best fit

the behavior of the real app. In this manner, the evolution curve of that app will be estimated. This curve will be built taking into account two issues: On the one hand, in a real scenario, we want to be able to know the expected behavior of an app based just on the early days of its launch. This means that in practice the number of downloads will be available only at some early dates. On the other hand, a set of 100,000 results from our simulations are available. Based on the two previous facts, we will select the simulations that best fit the real data. For that, we introduce the following notation:

- $d(i)$ denotes the total number of accumulated downloads at the i-th day. In practice, the values of $d(i)$ are only known for some specific days, say, $d(i_1), d(i_2), \ldots, d(i_p)$, $1 \leq i_1 < i_2 < \cdots < i_p \leq t_s = 100$.

- $s(i, j)$ denotes the total number of accumulated downloads at the i-th day ($1 \leq i \leq 100$) for simulation $j_x (1 \leq j \leq 10^5)$. To compare simulations with the available real data, $d(i_1), d(i_2), \ldots, d(i_p)$, just simulations $s(i_1, j), s(i_2, j), \ldots, s(i_p, j)$, $1 \leq j \leq 10^5$, will be required.

Taking into account that our network is comprised by 1,000,000 users and that the number of users in the real network is unknown, for each simulation j, a factor, $\alpha_j > 0$, will also be determined to scale the available real data $d(i_k)$, in such a way that the scaled real data $\alpha_j d(i_k)$ and the simulation $s(i_k, j)$ be close for all the days, $1 \leq k \leq p$. This approximation will be built using the Mean Square Error (MSE) as error measure. Thus, we calculate ϵ_j, the MSE of simulation $s(i, j)$, as

$$\epsilon_j = \sum_{k=1}^{p} \left(\alpha_j d(i_k) - s(i_k, j)\right)^2, \quad \alpha_j = \frac{\sum_{k=1}^{p} d(i_k) s(i_k, j)}{\sum_{k=1}^{p} (d(i_k))^2}, \quad 1 \leq j \leq 10^5. \tag{2.3}$$

This defines a set of mean square errors $\{\epsilon_j > 0 : \ 1 \leq j \leq 10^5\}$ associated to each simulation j. Notice that the best simulation $s(i, j^*)$, in the mean square sense, is given by the one where $\epsilon_{j^*} = \min\{\epsilon_j > 0 : \ 1 \leq j \leq 10^5\}$.

Now, we sort the simulations $s(i, j)$ by MSE in ascendant order. Thus, we search for the subset of simulations with smallest MSE such that, once it is calculated the 95% CI in each point (day), the available real data $d(i_1), d(i_2), \ldots, d(i_p)$ lie inside their corresponding 95% CI. Then, with this obtained subset of simulations we expect to estimate the behavior of the app downloads in the near future.

2.3 Results and discussion

This section shows the results obtained according to the method described in Section 3.4. We have monitorized fifteen apps from the real app market [56]. Hereinafter, we will show the results of our technique for two apps.

In Table 2.1, the total number of accumulated downloads during different days $d(i_k)$, for the first app (App1), is shown.

i_k-th day ($1 \leq k \leq 8 = p$)	4	11	14	21	28	34	84	95
# of accumulated downloads ($d(i_k)$)	281	707	873	1123	1284	1392	1886	1992

Table 2.1: App1. Total number of accumulated downloads in eight different days.

Considering a real scenario, we would only have the total number of downloads until the present, say 11-th day, given by $d(11)$. Therefore, the only available data is the total number of accumulated downloads corresponding to days, $d(4)$ and $d(11)$. By selecting the behaviors from our set that best fit the these two values and the 95% confidence interval, as explained in Section 3.4, we would obtain the results shown in Figure 2.2.

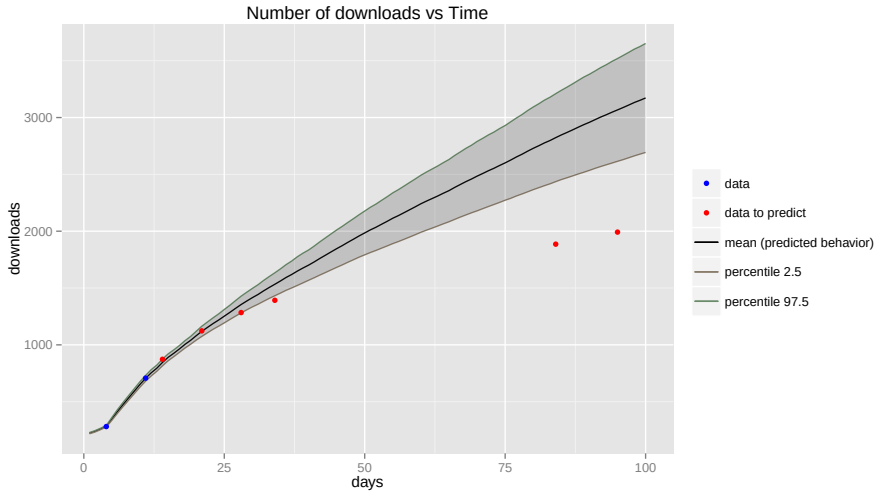

Figure 2.2: App1. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data $d(4)$ and $d(11)$ corresponding to the 4-th and 11-th days, respectively.

With our simulations, we are able to capture the real behavior of the App1 (red points in Figure 2.2) until the 28-th day after the app launch, since data $d(14)$, $d(21)$ and $d(28)$ is inside the 95% confidence interval generated by the proposed method. Notice that, although the value corresponding to 34-th day, $d(34)$, lies outside the confidence interval, it is not far from the 95% CI.

Following with the real scenario, if we reach the 14-th day, and we would dispose of the number of accumulated downloads $d(14)$, then we can feedback the proposed method with this new data. Then, using $d(4)$, $d(11)$ and $d(14)$, we would obtain the results shown in Figure 2.3. Now the real value $d(34)$ lies inside the new generated confidence interval. Our prediction also gives more downloads than the real future values.

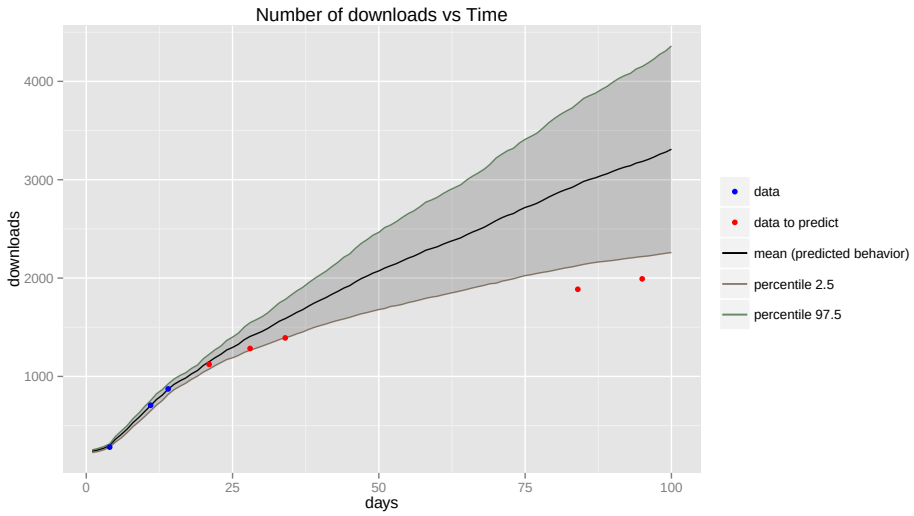

Figure 2.3: App1. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data $d(4)$, $d(11)$ and $d(14)$ corresponding to the 4-th, 11-th and 14-th days, respectively.

As showed in Figure 2.2 and Figure 2.3, the behavior far from the app launch, for example in 84-th and 95-th days, whose number of total downloads are given by $d(84)$ and $d(95)$, respectively, are not captured. However, if we again feedback the selection, from $d(11)$, $d(14)$ and $d(21)$, the predicted behavior by the mean fits the data in that days, as shown in Figure 2.4, but generating wide confidence intervals.

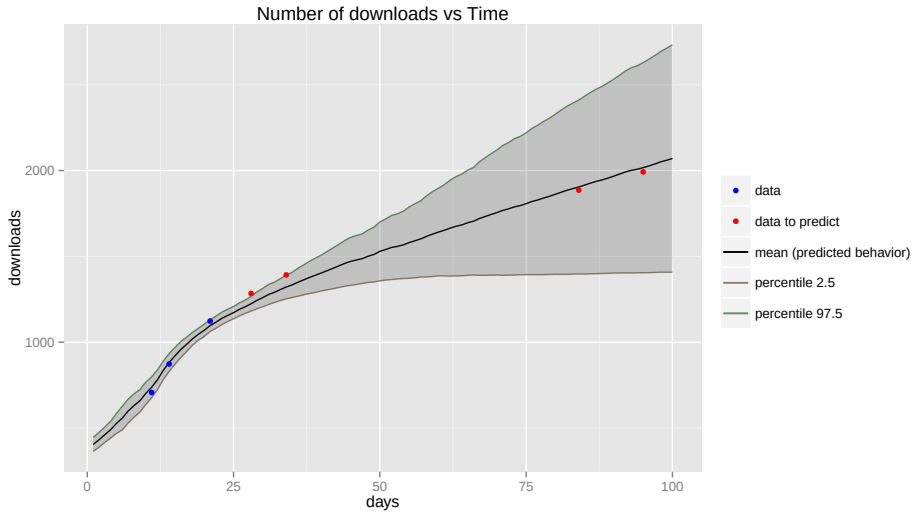

Figure 2.4: App1. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(11)$, $d(14)$ and $d(21)$ corresponding to the 11-th, 14-th and 21-th days, respectively.

Now, we consider an app with a low level of downloads, whose figures are listed in Table 2.2.

i_k-th day $(1 \leq k \leq 6 = p)$	6	8	13	20	28	70
# of accumulated downloads $(d(i_k))$	98	131	170	186	200	281

Table 2.2: App2. Total number of accumulated downloads during six different days.

In this case, using real data $d(6)$ and $d(8)$, the proposed model is able to predict the total number of accumulated of downloads in the 13-th day. This has been plotted in Figure 2.5.

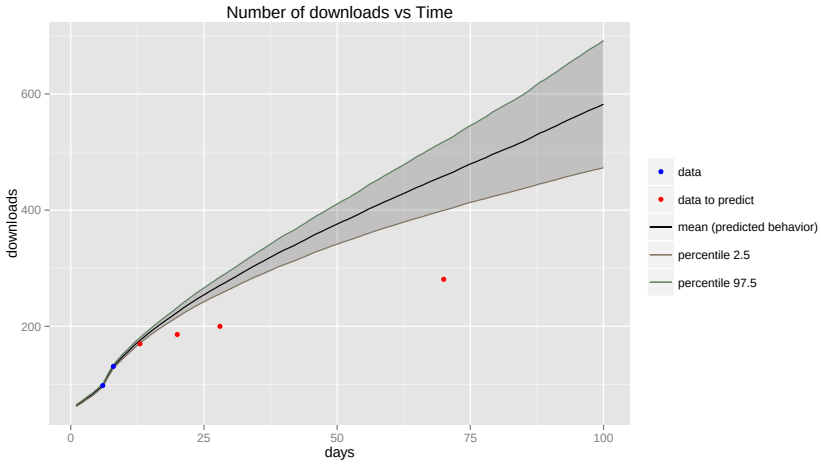

Figure 2.5: App2. Predicted behavior and 95% confidence interval of number of accumulated downloads vs. time, based on data from $d(6)$ and $d(8)$ corresponding to the 6-th and 8-th days, respectively.

If the model is feedback using data $d(13)$ corresponding to the 13-th day, then the model also captures the real data $d(20)$ and $d(70)$, being small the error corresponding the prediction at the 28-th day. The results can be seen in Figure 2.6.

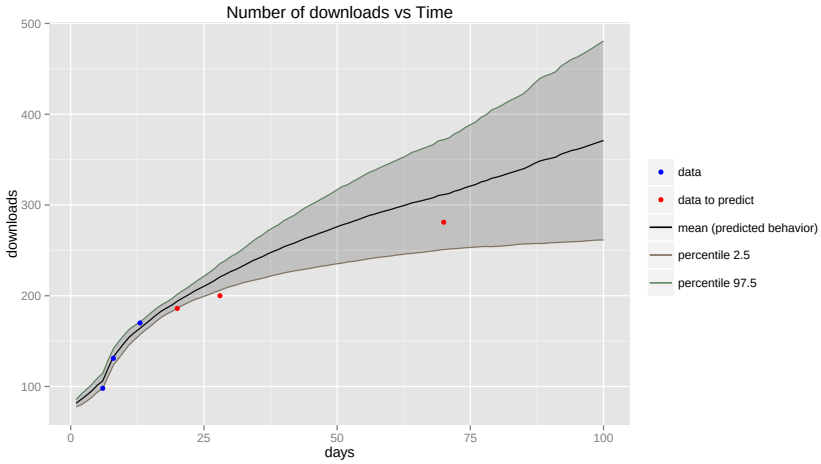

Figure 2.6: App2. Predicted behavior and 95% confidence interval of number of accumulated downloads vs. time, based on data from $d(6)$, $d(8)$ and $d(13)$ corresponding to the 6-th, 8-th and 13-th days, respectively.

Summarizing, in this Section we have shown the results provided by our method with two different types of apps. It has been shown that the proposed method improves the prediction when it is feedback. To show the robustness of the the proposed method, it has been tested with eleven additional apps from [56]. The results with these eleven additional apps can be found in section 2.3.1 and in [41].

In all the cases, good results have been obtained. On the one hand, we have shown the proposed model is able to predict the behaviour of apps with an average (standard) or low total number of downloads. On the other hand, the method does not provide correct results for the behavior of apps with a high number of downloads, i.e. high *virality*, at the first stages due to their fast growth. However, the predictions improve when the model is feedback providing correct results.

The proposed method shows that it is possible to provide an approximation for the behavior of the number of downloads using confidence intervals. The key for the prediction accuracy is to select the adequate parameters for the model building. Depending on the type of app that we want to anticipate its behavior, we should fix the set of parameters as retention time and infection rate according to its characteristics. For example, an app with a marketing campaign should increase its infection parameter according to the expected impacts of that campaign.

2.3.1 Additional results

App3

For application number 3, total number of accumulated downloads are shown in Table 2.3.

i_k-th day ($1 \leq k \leq 8 = p$)	5	7	11	14	22	28	43	51
# of accumulated downloads ($d(i_k)$)	709	897	1234	1360	1576	1686	1842	1895

Table 2.3: App3. Total number of accumulated downloads during eight different days.

In Figure 2.7, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(5)$ and $d(7)$ corresponding to the 5-th and 7-th days respectively, are shown.

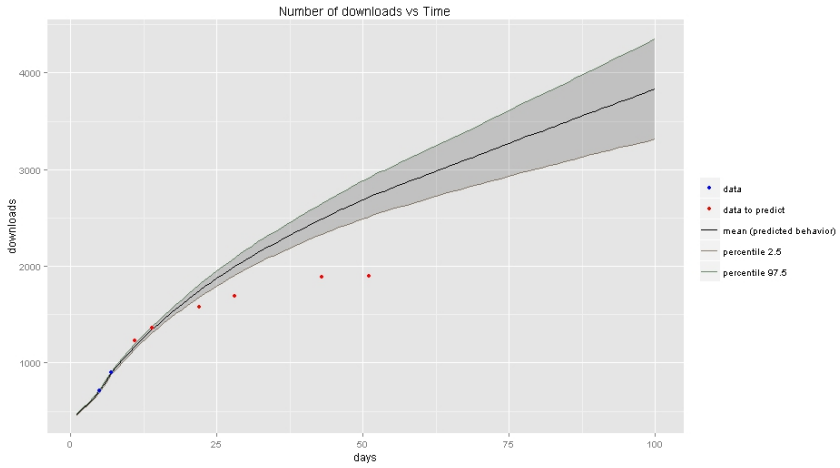

Figure 2.7: App3. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(5)$ and $d(7)$ corresponding to the 5-th and 7-th days, respectively.

In Figure 2.8, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(5)$, $d(7)$ and $d(11)$ corresponding to the 5-th, 7-th and 11-th days respectively, are shown.

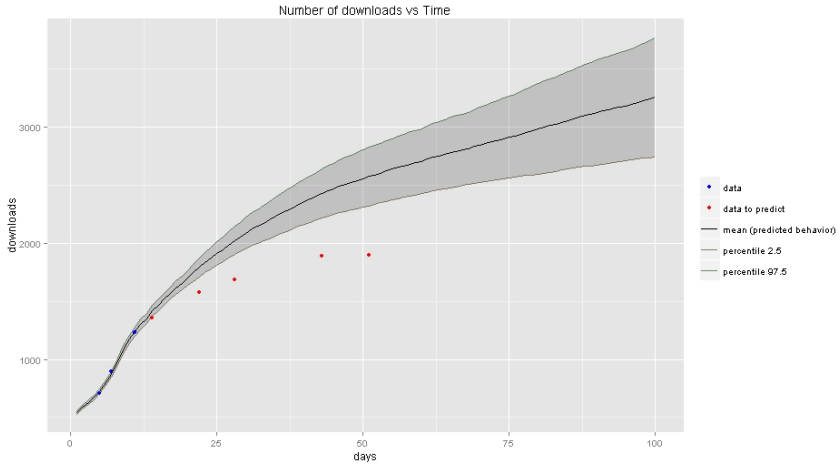

Figure 2.8: App3. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(5)$, $d(7)$ and $d(11)$ corresponding to the 5-th, 7-th and 11-th days, respectively.

In Figure 2.9, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(5)$, $d(7)$, $d(11)$ and $d(14)$ corresponding to the 5-th, 7-th, 11-th and 14-th days respectively, are shown.

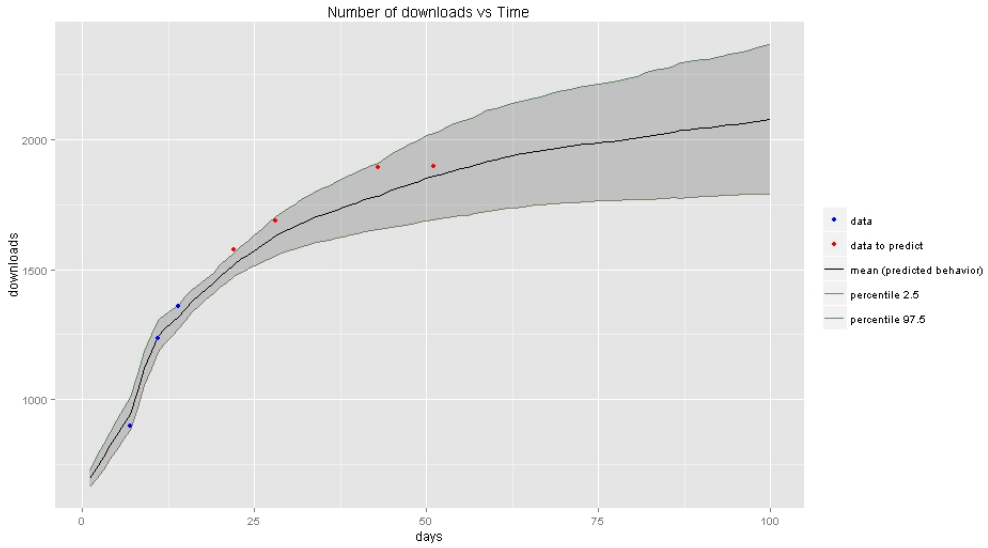

Figure 2.9: App3. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(5)$, $d(7)$, $d(11)$ and $d(14)$ corresponding to the 5-th, 7-th, 11-th and 14-th days, respectively.

With data from $d(5)$ and $d(7)$ we can predict correctly the behavior of the accumulated downloads until $d(14)$. The model is able to capture the real behavior for 11-th and 14-th days, whose total number of accumulated downloads are given by $d(11)$ and $d(14)$, respectively. If the model is feedback by including data $d(11)$ together with $d(5)$ and $d(7)$, then it is able to capture the real behavior including $d(14)$. However, notice that the rest of real data is outside the confidence interval. If the model is feedback again with data $d(7)$, $d(11)$ and $d(14)$, the rest of the real data are captured by the new 95% confidence interval. As we can see, when the model does not fit correctly in the first instance, if we feedback the model, the predictions are improved.

App4

For application number 4, total number of accumulated downloads are shown in Table 2.4.

i_k-th day $(1 \leq k \leq 6 = p)$	6	8	13	20	28	70
# of accumulated downloads $(d(i_k))$	2301	3493	6359	9261	11962	20307

Table 2.4: App4. Total number of accumulated downloads during six different days.

In Figure 2.10, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(6)$ and $d(8)$ corresponding to the 6-th and 8-th days respectively, are shown.

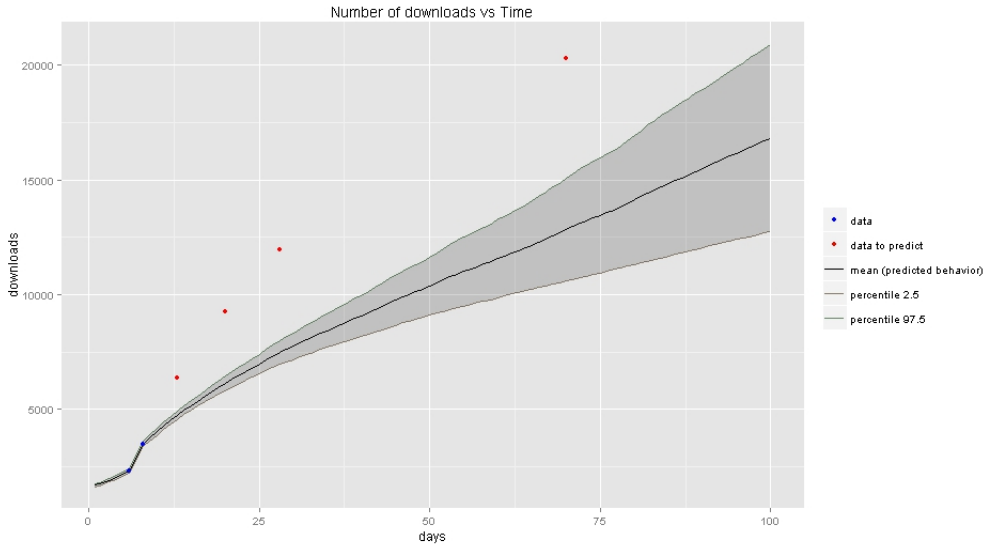

Figure 2.10: App4. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(6)$ and $d(8)$ corresponding to the 6-th and 8-th days, respectively.

In Figure 2.11, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(6)$, $d(8)$

and $d(13)$ corresponding to the 6-th, 8-th and 13-th days respectively, are shown.

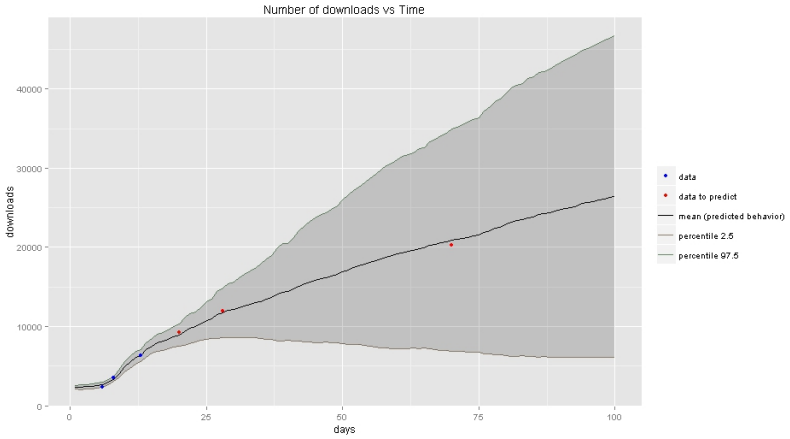

Figure 2.11: App4. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(6)$, $d(8)$ and $d(13)$ corresponding to the 6-th, 8-th and 13-th days, respectively.

In Figure 2.12, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(8)$ and $d(13)$ corresponding to the 8-th and 13-th days respectively, are shown.

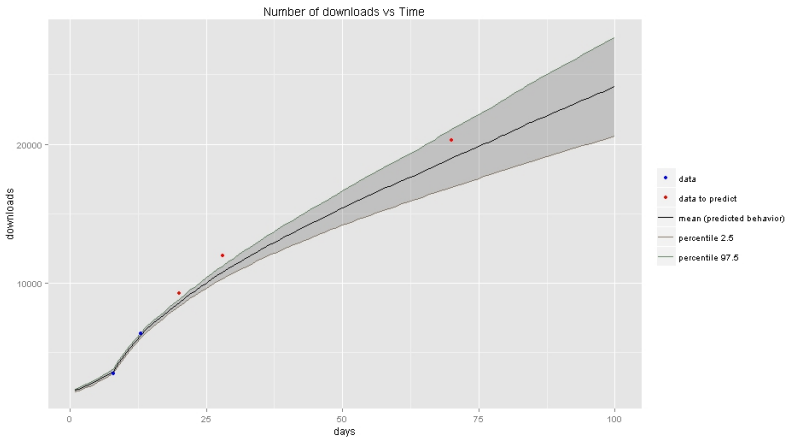

Figure 2.12: App4. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(8)$ and $d(13)$ corresponding to the 8-th and 13-th days, respectively.

Using real data $d(6)$ and $d(8)$, the estimation provided by the model does not capture the real behavior of the app, but if the model is feedback with data $d(13)$, then it is able to capture the future behavior, but providing non-informative confidence intervals due to their large amplitude. To overcome this drawback, we construct the prediction on account on $d(8)$ and $d(13)$, ignoring $d(6)$. In this manner, the obtained prediction curve and its confidence intervals are able to capture the 70-th day, although real data $d(20)$ and $d(28)$, corresponding to 20-th and 28-th days lie very close but outside the confidence intervals

App5

For application number 5, total number of accumulated downloads are shown in Table 2.5.

i_k-th day $(1 \leq k \leq 6 = p)$	5	7	12	19	27	69
# of accumulated downloads $(d(i_k))$	27	66	123	149	190	271

Table 2.5: App5. Total number of accumulated downloads during six different days.

In Figure 2.13, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(5)$ and $d(7)$ corresponding to the 5-th and 7-th days respectively, are shown.

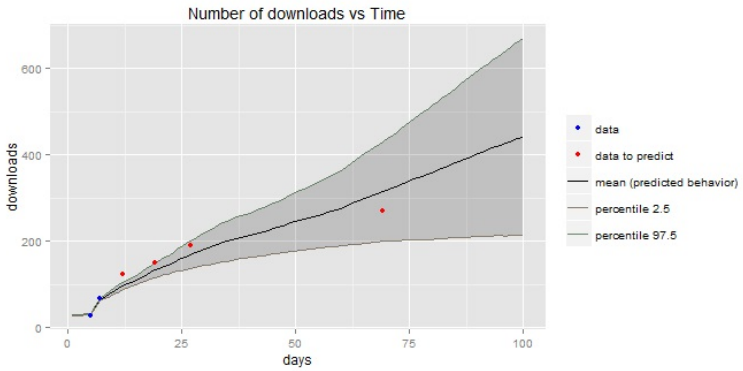

Figure 2.13: App5. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(5)$ and $d(7)$ corresponding to the 5-th and 7-th days, respectively.

In Figure 2.14, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(5)$, $d(7)$ and $d(12)$ corresponding to the 5-th, 7-th and 12-th days respectively, are shown.

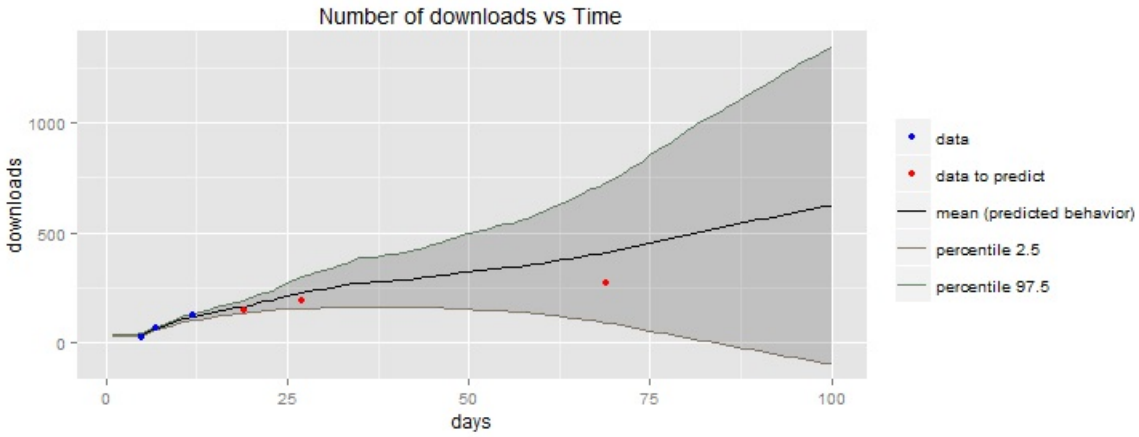

Figure 2.14: App5. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(5)$, $d(7)$ and $d(12)$ corresponding to the 5-th, 7-th and 12-th days, respectively.

With data from $d(5)$ and $d(7)$ we can predict correctly the behavior of the accumulated downloads until $d(69)$. If we feedback the model with data from $d(12)$, we predict again correctly the behavior of the accumulated number of downloads. All the data to be predicted is inside the 95% confidence interval generated by the proposed method. Notice that, although the value corresponding to 12-th day in the first simulation lies outside the confidence interval, its probabilistic error is very small, i.e., $d(12)$ is not far from the 95% CI.

App6

For application number 6, total number of accumulated downloads are shown in Table 2.6.

In Figure 2.15, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(3)$ and $d(10)$ corresponding to the 3-th and 10-th days respectively, are shown.

i_k-th day ($1 \leq k \leq 5 = p$)	3	10	13	27	33
# of accumulated downloads ($d(i_k)$)	1178	3143	4905	8635	9136

Table 2.6: App6. Total number of accumulated downloads during five different days.

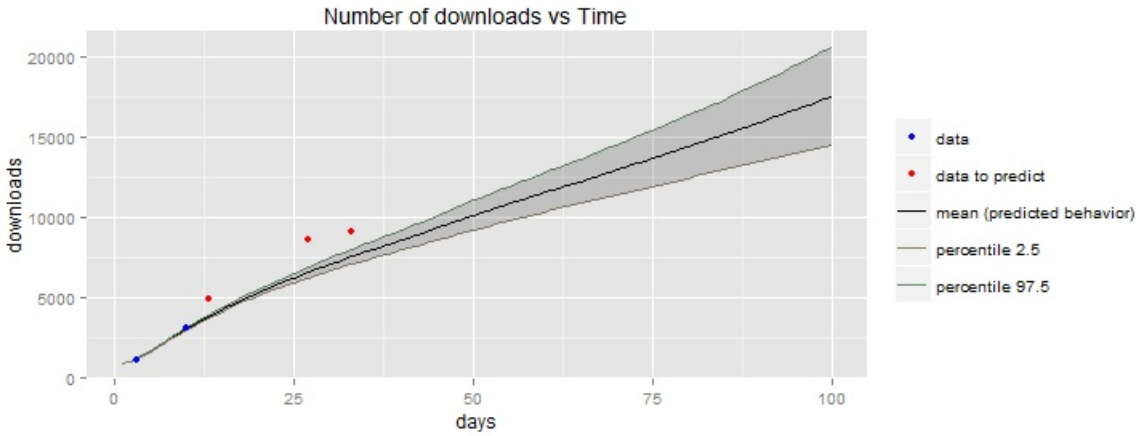

Figure 2.15: App6. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(3)$ and $d(10)$ corresponding to the 3-th and 10-th days, respectively.

In Figure 2.16, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(3)$, $d(10)$ and $d(13)$ corresponding to the 3-th, 10-th and 13-th days respectively, are shown.

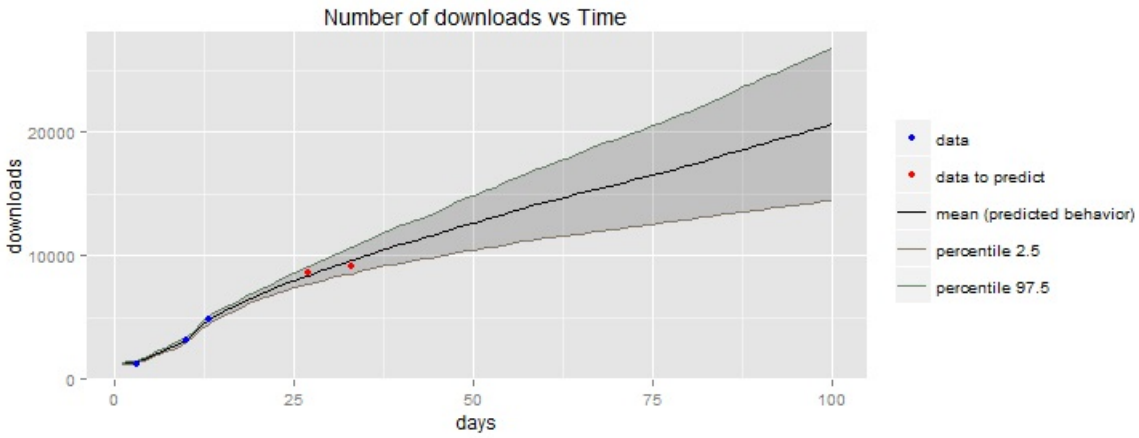

Figure 2.16: App6. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(3)$, $d(10)$ and $d(13)$ corresponding to the 3-th, 10-th and 13-th days, respectively.

Although with $d(3)$ and $d(10)$ we can not predict correctly the future behavior of the accumulated downloads, if we feedback the model with data from $d(13)$, we are able to predict correctly the future behavior. As we can see, when the model does not fit correctly in the first instance, if we feedback the model, the predictions are improved.

App7

For application number 7, total number of accumulated downloads are shown in Table 2.7.

i_k-th day $(1 \leq k \leq 7 = p)$	4	6	9	10	13	14	21
# of accumulated downloads $(d(i_k))$	285	630	889	949	1091	1119	1306

Table 2.7: App7. Total number of accumulated downloads during seven different days.

In Figure 2.17, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(4)$ and $d(6)$ corresponding to the 4-th and 6-th days respectively, are shown.

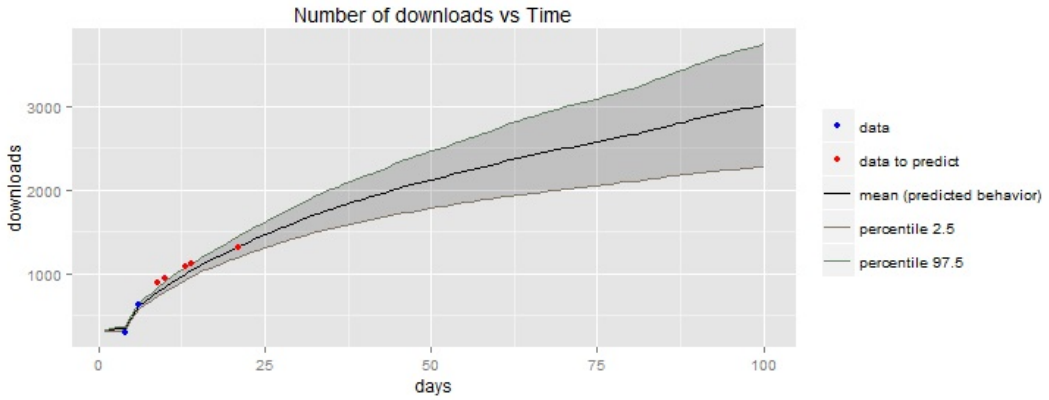

Figure 2.17: App7. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(4)$ and $d(6)$ corresponding to the 4-th and 6-th days, respectively.

In Figure 2.18, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(4)$, $d(6)$ and $d(9)$ corresponding to the 4-th, 6-th and 9-th days respectively, are shown.

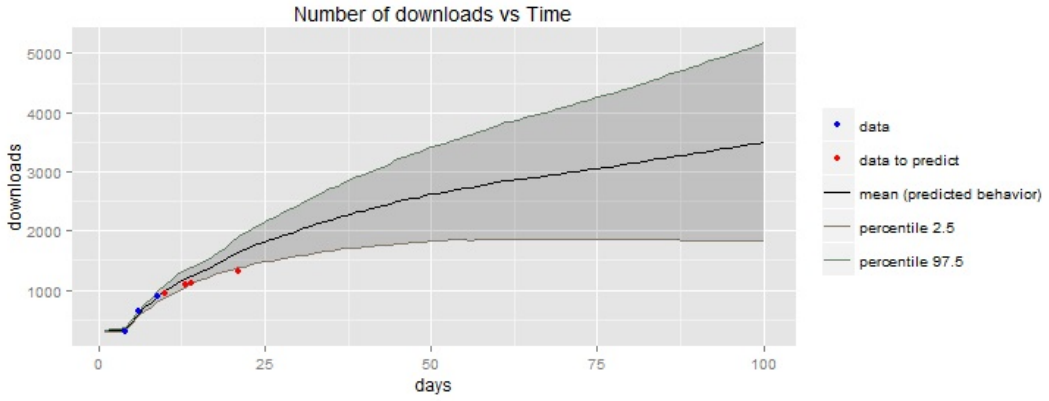

Figure 2.18: App7. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(4)$, $d(6)$ and $d(9)$ corresponding to the 3-th, 10-th and 13-th days, respectively.

With data from $d(4)$ and $d(6)$ we can predict correctly the behavior of the accumulated downloads. If we feedback the model with data from $d(9)$, we predict again correctly the behavior of the accumulated number of downloads. All the data to be predicted is inside the 95% confidence interval generated by the proposed method. Notice that, although the value corresponding to 9-th and 10-th day in the first simulation lies slightly outside the confidence interval, their probabilistic error is very small, i.e., is not far from the 95% CI.

App8

For application number 8, total number of accumulated downloads are shown in Table 2.8.

i_k-th day ($1 \leq k \leq 9 = p$)	4	6	9	10	13	14	21	27	42
# of accumulated downloads ($d(i_k)$)	85	198	256	263	286	290	296	297	300

Table 2.8: App8. Total number of accumulated downloads during nine different days.

In Figure 2.19, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(4)$ and $d(6)$ corresponding to the 4-th and 6-th days respectively, are shown.

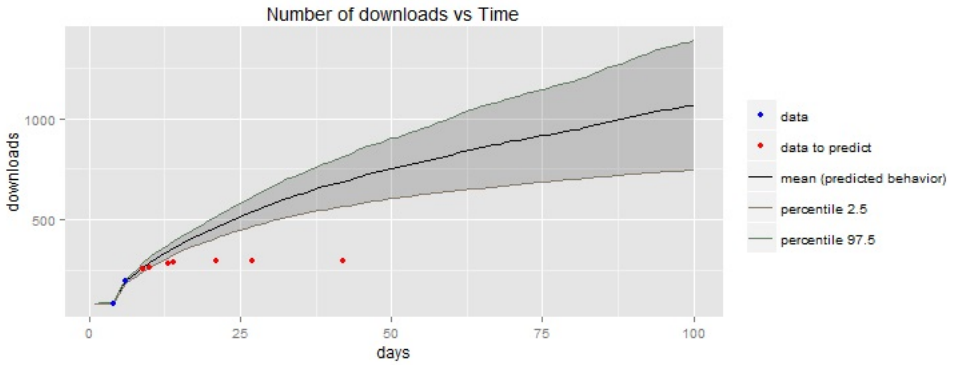

Figure 2.19: App8. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(4)$ and $d(6)$ corresponding to the 4-th and 6-th days, respectively.

In Figure 2.20, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(9)$, $d(10)$ and $d(13)$ corresponding to the 9-th, 10-th and 13-th days respectively, are shown.

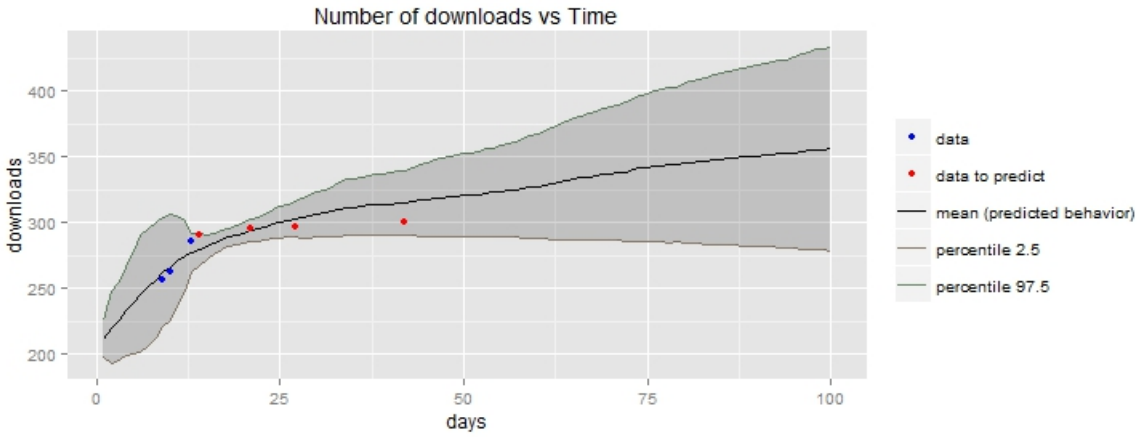

Figure 2.20: App8. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(9)$, $d(10)$ and $d(13)$ corresponding to the 9-th, 10-th and 13-th days, respectively.

With data from $d(4)$ and $d(6)$ we can predict correctly the behavior of the accumulated downloads until $d(10)$. If we feedback the model with data from $d(9)$, $d(10)$ and $d(13)$ we can predict correctly the behavior of the accumulated number of downloads at least until 42-th day, being all the data to be predicted inside the 95% confidence interval generated by the proposed method.

App9

For application number 9, total number of accumulated downloads are shown in Table 2.9.

In Figure 2.21, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(5)$ and $d(7)$ corresponding to the 5-th and 7-th days respectively, are shown.

i_k-th day ($1 \leq k \leq 9 = p$)	5	7	10	11	14	15	22	28	43
# of accumulated downloads ($d(i_k)$)	112	136	205	215	254	259	270	274	290

Table 2.9: App9. Total number of accumulated downloads during nine different days.

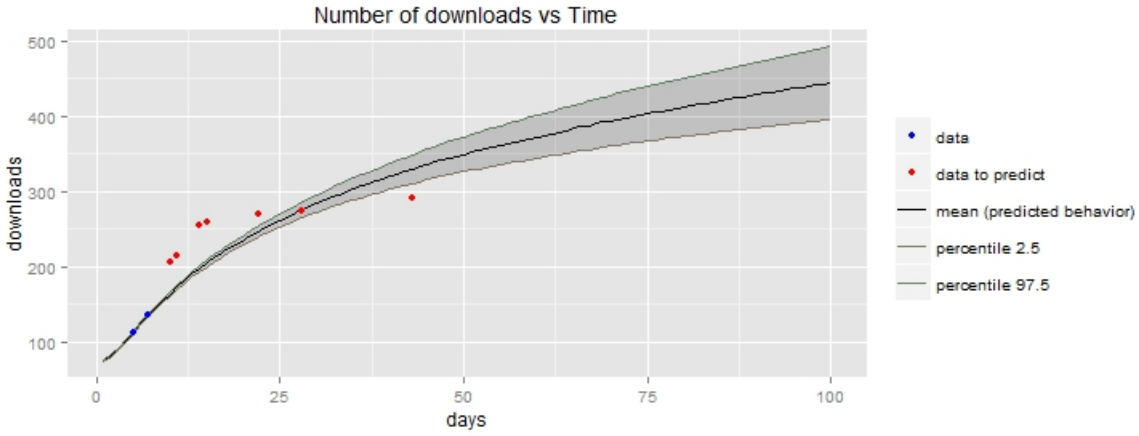

Figure 2.21: App9. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(5)$ and $d(7)$ corresponding to the 5-th and 7-th days, respectively.

In Figure 2.22, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(7)$, $d(10)$ and $d(11)$ corresponding to the 7-th, 10-th and 11-th days respectively, are shown.

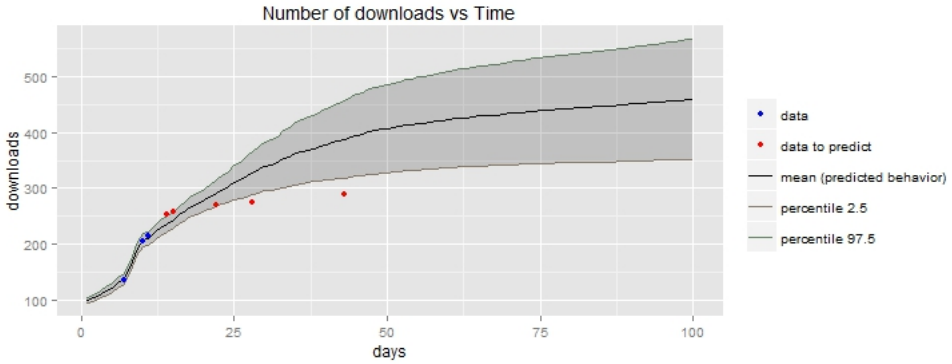

Figure 2.22: App9. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(7)$, $d(10)$ and $d(11)$ corresponding to the 7-th, 10-th and 11-th days, respectively.

In Figure 2.23, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(7)$, $d(10)$, $d(11)$ and $d(14)$ corresponding to the 7-th, 10-th, 11-th and 14-th days respectively, are shown.

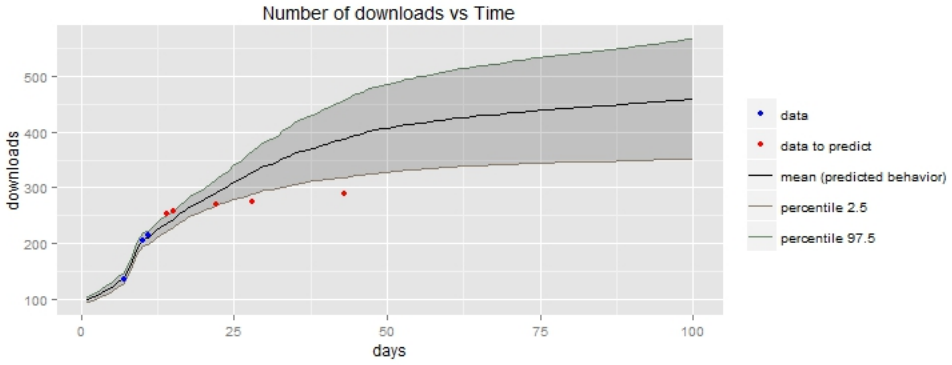

Figure 2.23: App9. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(7)$, $d(10)$, $d(11)$ and $d(14)$ corresponding to the 7-th, 10-th, 11-th and 14-th days, respectively.

With data from $d(5)$ and $d(7)$ we can predict correctly the behavior of the accumulated downloads in day $d(28)$, but the rest of days are not predicted

correctly as they are outside the 95% confidence interval. If we feedback the model with data from $d(7)$, $d(10)$ and $d(11)$, the prediction is improved and we can predict correctly the behavior of the accumulated number of downloads in the short term (until 22-th day). If we feedback again the model with data from $d(14)$, we can predict correctly the behavior of the accumulated number of downloads in the long term (until 43-th day) being all the data to be predicted is inside the 95% confidence interval generated by the proposed method. Notice that the feedback of the model improves the predictions for the long term.

App10

For application number 10, total number of accumulated downloads are shown in Table 2.10.

i_k-th day ($1 \leq k \leq 4 = p$)	2	6	8	44
# of accumulated downloads ($d(i_k)$)	135	312	375	656

Table 2.10: App10. Total number of accumulated downloads during four different days.

In Figure 2.24, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(2)$ and $d(6)$ corresponding to the 2-th and 6-th days respectively, are shown.

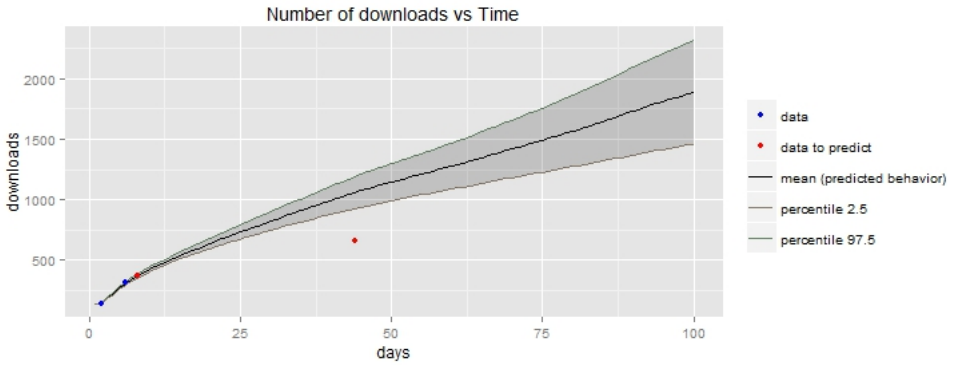

Figure 2.24: App10. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(2)$ and $d(6)$ corresponding to the 2-th and 6-th days, respectively.

In Figure 2.25, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(6)$ and $d(8)$ corresponding to the 6-th and 8-th days respectively, are shown.

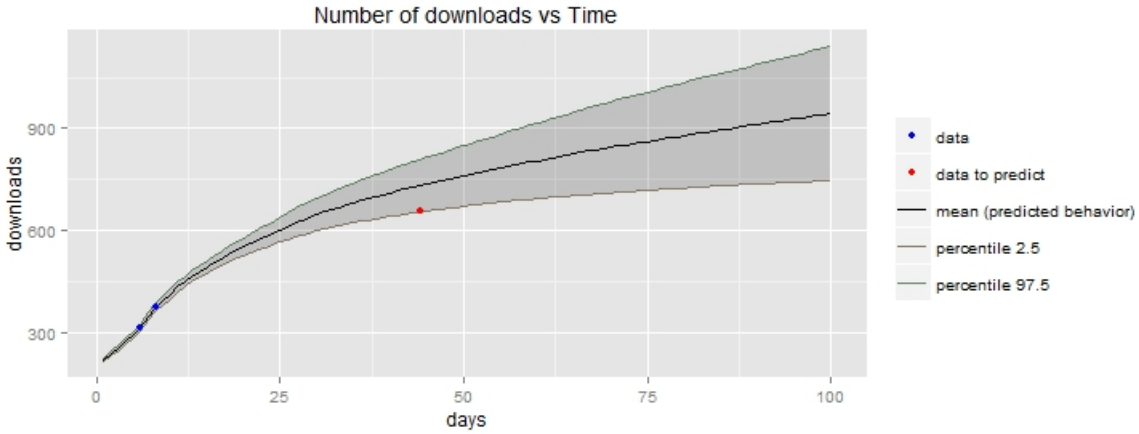

Figure 2.25: App10. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(6)$ and $d(8)$ corresponding to the 6-th and 8-th days, respectively.

With data from $d(2)$ and $d(6)$ we can predict correctly the behavior of the accumulated downloads in the short term but not in the long term (44-th day). If we feedback the model with data from $d(6)$ and $d(8)$, we can predict correctly the behavior of the accumulated number of downloads in the long term: The data to be predicted is inside the 95% confidence interval generated by the proposed method.

App11

For application number 11, total number of accumulated downloads are shown in Table 2.11.

In Figure 2.26, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(3)$ and $d(7)$ corresponding to the 3-th and 7-th days respectively, are shown.

i_k-th day ($1 \leq k \leq 4 = p$)	3	7	9	45
# of accumulated downloads ($d(i_k)$)	275	448	527	870

Table 2.11: App11. Total number of accumulated downloads during four different days.

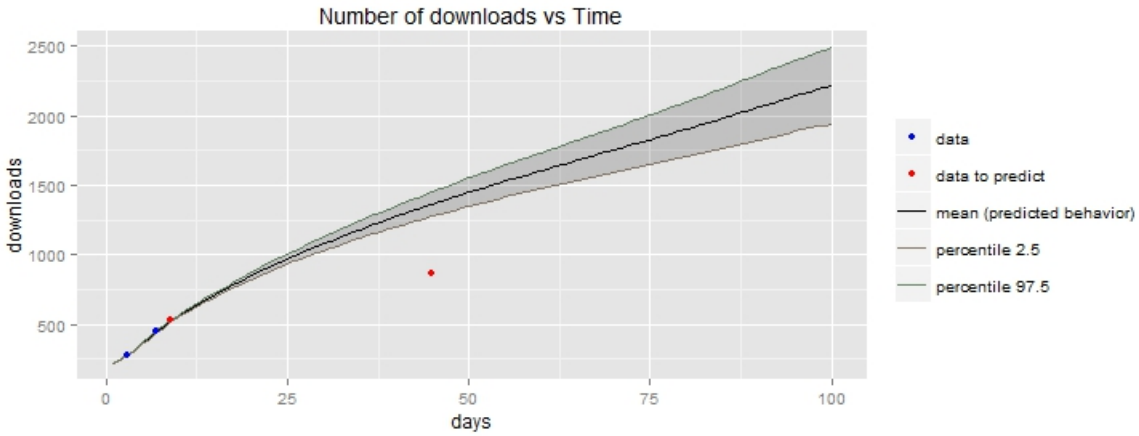

Figure 2.26: App11. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(3)$ and $d(7)$ corresponding to the 3-th and 7-th days, respectively.

In Figure 2.27, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(7)$ and $d(9)$ corresponding to the 7-th and 9-th days respectively, are shown.

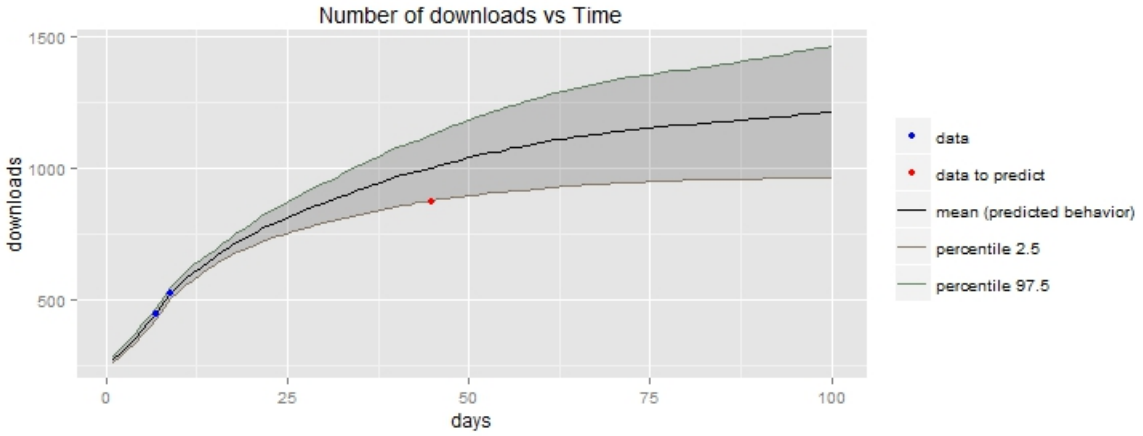

Figure 2.27: App11. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(7)$ and $d(9)$ corresponding to the 7-th and 9-th days, respectively.

With data from $d(3)$ and $d(7)$ we can predict correctly the behavior of the accumulated downloads in the short term but not in the long term (45-th day). If we feedback the model with data from $d(7)$ and $d(9)$, we can predict correctly the behavior of the accumulated number of downloads in the long term: The data to be predicted is inside the 95% confidence interval generated by the proposed method.

App12

For application number 12, total number of accumulated downloads are shown in Table 2.12.

i_k-th day ($1 \leq k \leq 4 = p$)	3	6	20	26
# of accumulated downloads ($d(i_k)$)	254	459	695	731

Table 2.12: App12. Total number of accumulated downloads during four different days.

In Figure 2.28, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(3)$ and $d(6)$ corresponding to the 3-th and 6-th days respectively, are shown.

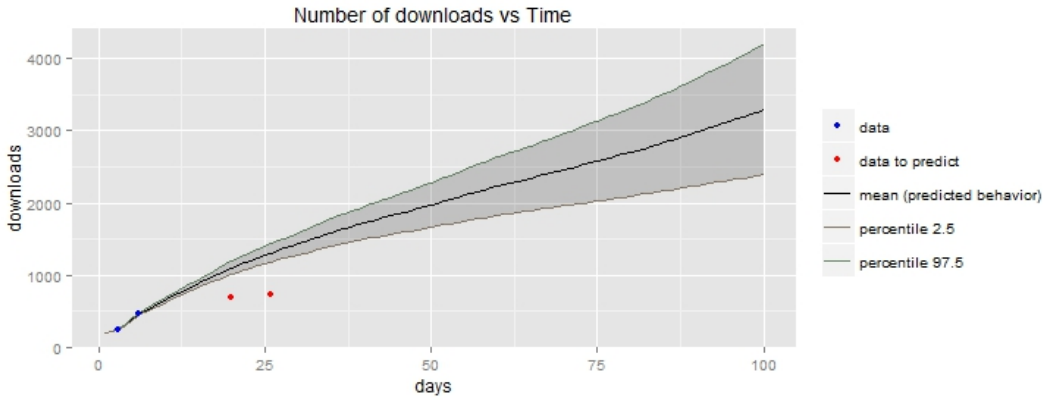

Figure 2.28: App12. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(3)$ and $d(6)$ corresponding to the 3-th and 6-th days, respectively.

In Figure 2.29, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(6)$ and $d(20)$ corresponding to the 6-th and 20-th days respectively, are shown.

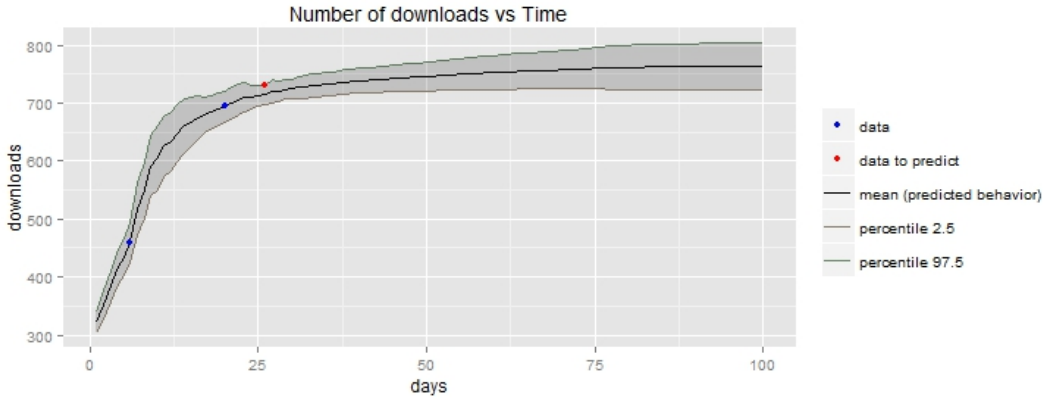

Figure 2.29: App12. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(6)$ and $d(20)$ corresponding to the 6-th and 20-th days, respectively.

Although with $d(3)$ and $d(6)$ we can not predict correctly the behavior of the accumulated downloads in $d(20)$, if we feedback the model with data from $d(20)$, we are able to predict correctly the future behavior in $d(26)$. As we can see, when the model does not fit correctly in the first instance, if we feedback the model, the predictions are improved.

App13

For application number 13, total number of accumulated downloads are shown in Table 2.13.

i_k-th day $(1 \leq k \leq 5 = p)$	2	4	7	8	11
# of accumulated downloads $(d(i_k))$	252	593	929	988	1099

Table 2.13: App13. Total number of accumulated downloads during five different days.

In Figure 2.30, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(2)$ and $d(4)$ corresponding to the 2-th and 4-th days respectively, are shown.

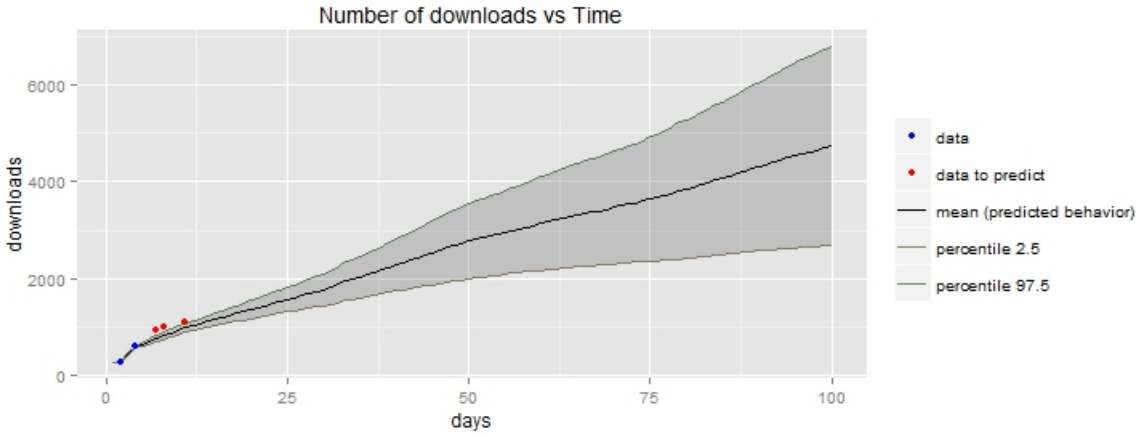

Figure 2.30: App13. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(2)$ and $d(4)$ corresponding to the 2-th and 4-th days, respectively.

In Figure 2.31, the predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(2)$, $d(4)$ and $d(7)$ corresponding to the 2-th, 4-th and 7-th days respectively, are shown.

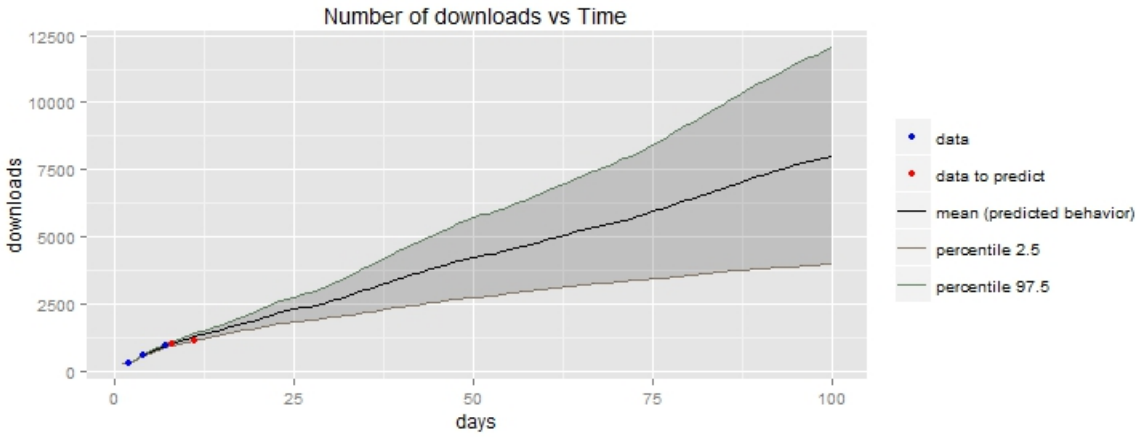

Figure 2.31: App13. Predicted behavior and 95% confidence interval of the number of accumulated downloads vs. time, based on data from $d(2)$, $d(4)$ and $d(7)$ corresponding to the 2-th, 4-th and 7-th days, respectively.

With data from $d(2)$ and $d(4)$, we can predict correctly the behavior of the accumulated downloads although the values lie slightly outside the confidence interval, being their probabilistic error very small. If we feedback the model with data from $d(7)$, we can predict again the behavior being all the data to be predicted inside the 95% confidence interval generated by the proposed method.

2.4 Conclusion

In this chapter, an epidemiological random network model to estimate the evolution of download of apps has been proposed. The model's goals have been to predict the total number of accumulated apps' download as well as to show the network effect on the apps' downloads validating our assumption that exogenous factors due to app popularity and spontaneous app installation after browse an app market by the user are weaker and less significant

than face-to-face relations. In addition, the proposed model generalizes the results obtained in other contributions [28, 1, 23] with very high networks.

The results show that the prediction of the evolution of the number of downloads of an app over the time is possible via computational methods whenever parameters are adequately chosen. The capability of the model to capture the behavior of the app by means of confidence intervals has been shown. Therefore the face-to-face relations are more important than other mechanisms for apps' adoption. Although, the proposed method does not consider exogenous factors, it is capable to forecast correctly, using confidence intervals, the evolution of the number of downloads for monitored apps.

The study has been based on 100,000 simulations. This permits to generalize the results obtained in other contributions about the face-to-face network effect in apps adoption, that were based on only one realization of the experiment.

Marketing researchers and strategy business managers can benefit from the proposed model since it can be helpful to predict app behavior over the time, anticipating the spread of an app as well as predicting its expected value.

Some of the results presented in this chapter have been summarized in a paper sent to *Simulation: Transactions of the Society for Modeling and Simulation International* pending to approval.

2.5 Appendix to Chapter 2

2.5.1 What was said recently by major actors in the mobile apps world

After developing and testing our model, obtaining the results shown in the previous sections, two major actors in the mobile applications world published, in the same date (May 2015), their own studies related with the spread of mobile apps:

- Facebook presented *The Lifecycles of Apps in a Social Ecosystem* [17] at the World Wide Web Conference 2015.
- Google published its *Mobile App Marketing Insights: How consumers really find and use your apps* [57].

Facebook, in [17], develops a novel framework for analyzing both temporal and social properties of a collection of apps on Facebook Login. At the temporal level, they develop a retention model that represents a user's

tendency to return to an app. At the social level, they organize the space of apps along two fundamental axes, popularity and sociality.

They show that a user's probability of adopting an app depends both on properties of the local network structure and on the match between the user's attributes, his or her friends' attributes, and the dominant attributes within the app's user population. Also, they claim that even the most asocial apps exhibit some social clustering.

One of the methods that they use to predict the spread or success of an app is a SIR model. They fitted the model using a Monte Carlo process using time series from June 2, 2012 to May 25, 2013 and used the fitted model to generate predictions between May 26, 2013 and May 15, 2014. With their SIR model, they were able to fit over two-thirds of the followed apps, which were 2,319 apps. However, they claim that some underlying assumptions in the SIRS model, such as the constant rate of user adoption through advertisement or word-of-mouth process, may not hold in reality. As a result, the model would not converge for certain apps, especially the ones that experienced large fluctuations in their lifecycles. With our model, this problem is overcome because we consider a random and not constant rate for user contagion, as showed in Eq.(2.2).

On the other hand, the method followed by Google to obtaining the results of its report [57] were online surveys. Based on the obtained results Google claims that:

- Apps are often discovered outside the app store.
- Recommendations and interest/fun level are top reasons to download apps.

Extracted from this Google report [57], the sources of awareness of smartphone apps, in percentage, are shown in Table 2.14 and the reasons for downloading an app, also in percentage, are shown in Table 2.15:

Source of awareness	**Percentage**
Friends, family, and colleagues	52%
Browse the app store	40%
Search engines	27%
Company website	24%
TV	22%

Table 2.14: Sources of awareness of smartphone apps according to Google surveys.

Reason	Percentage
Recommended by others	33%
Sounded interesting/fun	31%
Familiarity with company/brand	24%
Access exclusive discounts/rewards	18%

Table 2.15: Reasons for downloading an app according to Google surveys.

The resuls by Facebook and Google are in line with the assumptions, method and results of our model.

2.5.2 Web page SAMOA I model

A web page has been developed with the aim of offering the results of our model as a software as a service (SaaS) [54] under the name of SAMOA (Spread Analysis for MObile Apps). In the web page there is a demo wizard showing how the model works at the Wizard section [55].

Chapter 3

SAMOA II (Spread Analysis for Malware On Android): Agent-based model to study and quantify the evolution dynamics of Android malware infection

In Chapter 2 we presented a first model to model and analyze the applications behavior related with their spread and, in this chapter, we propose a second model to study and analyze another applications behavior, related with the spread of malware.

In the last years the number of malware Apps the users download to their devices have risen. To study and analyze the spread of malware, the model we propose is an agent-based model to quantify the Android malware infection evolution, modeling the behavior of the users and the different markets where the users may download Apps. The name of the model, SAMOA, comes from Spread Analysis for Malware On Android, and the model is able to predict the number of infected smartphones over the time depending on the type of malware. Additionally, we will estimate the economic cost the users should afford when the malware is in their devices. We will be able to analyze which part is more critical: the users, giving indiscriminate permissions to the Apps or not protecting their devices with antivirus software, or the Android platform, due to the vulnerabilities of the Android devices that permit their rooted. We focus on the Community of Valencia, Spain,

although the obtained results can be extrapolated to other places where the number of Android smartphones remains fairly stable.

3.1 Introduction

The security in devices connected to the Internet is an issue that has long been concerned, from governments and companies to individual users. However, this threat seems not being perceived by the smartphone users taking into account the potential risky behavior of them and the sensitive data and pictures the users store in their devices. Moreover, the risk increases with the new companies policies that permit the employees the use of their own smartphones in the work accessing to company sensitive data and applications (Bring Your Own Device BYOD).

Different types of malware have already been documented [51, 31] and it may be a threat that must be studied to quantify the users' potential risk. Here, we will focus on Android platform because most of the smartphones use Android OS [37].

The architecture of the Android system is based on Linux, and as a result of that, the security model is based on three milestones:

- Sandboxing: The Android platform uses a technique called "sandboxing" to put virtual walls between applications and other software on the device. So, if you download a malicious application, it cannot access data on other parts of your phone and its potential harm is drastically limited.

- Permissions: Android provides a permission system to help you understand the capabilities of the apps you install, and manage your own preferences. That way, if you see a game unnecessarily requests permission to send SMS, for example, you do not need to install it.

- Malware removal: The official Android market has a service named Bouncer, which provides automated scanning of apps uploaded to android market before being available for the users that detects potentially malicious software.

However, despite this security model, multiple types of malware embedded in apps released in the apps stores have been found. As Google says "No security approach is foolproof" [49].

During the year 2011, appeared the first study on the characterization of viruses on mobile OS Android [51, 31]. This study categorizes the types and

families of viruses found, depending on the type of installation, activation, effects on the infected device, the user management of the permissions, etc., showing the diversity of different virus families and the ineffectiveness of the traditional antivirus methods on mobile devices.

Also, there are several works that have approached the analysis and detection of malware on the Android platform [13, 16, 27]. The common objective of these works is to propose new methods of virus detection on mobile devices from a dynamic point of view, that is, to detect at runtime anomalous or unwanted behavior of the device (system calls, network access, memory or file modifications). In contrast, static and classic antivirus methods are based on repositories of previously known viruses that do not protect the user in case of the spread of an unknown new virus type. However, dynamic detection of viruses are unsuitable for mobile devices for their CPU and memory consumption. The two approaches, static and dynamic methods, have their own advantages and disadvantages, and both may be bypassed and unable to avoid the spread of new viruses.

3.1.1 State of the art

In the literature, there are several approaches to the mathematical modeling for the spread of viruses on mobile devices. In [11] the authors describe a framework and the main guidelines to design reliable agent-based malware models considering infections via SMS/MMS, Bluetooth RF, IM, P2P and email. In [19, 25, 29] the authors propose approaches based on mathematical epidemic techniques where the malware infection follows similar dynamics to the infectious diseases.

Also, there are models based on the physical architecture of the mobile and wireless networks [14] or based on the mobility of the users, but they do not consider the interconnectivity based on the exchange of applications [25].

To the best of our knowledge, there is no paper showing quantification, prediction and/or simulation about how the users install malware Apps. However, literature about the application of machine learning techniques to detect malware Apps in the markets can be found [27]. Nevertheless, any of the above approaches do not take into account the infection model based on an App-market ecosystem, like smartphones environment is.

3.1.2 Proposed model

Likely, the model guidelines suggested in [11] are the most suited to the current scenario. In that contribution, an agent-based model of malware

dynamics covering all the possible infection models except the App-market ecosystem model is proposed. The integration of the App-market ecosystem is the key aspect that we will consider in this chapter.

As was indicated previously, researchers and companies characterized mobile malware and proposed alternative methods to prevent, detect and avoid mobile malware. Also, different companies publish periodically mobile malware reports with estimations and statistics. However, in the literature there is a lack of studies that quantify the effects of the malware infection in the Android platform in order to show realistic data to know the extent of the threat as our model does [24]. Our model (*App-Model*) complements the agent-based malware modeling suggested in [11] introducing a new infection process based on applications downloaded from the App-market. In Figure 3.1 we can see a rough description of the items we deal with to build the App-Model.

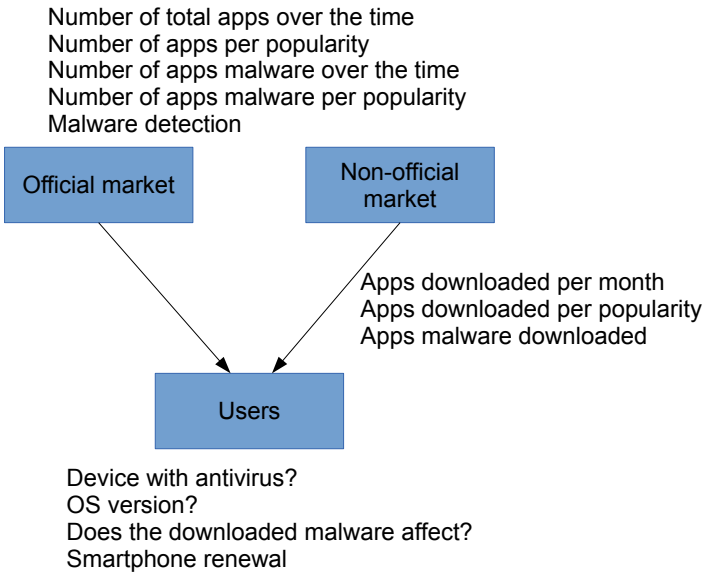

Figure 3.1: General structure of the agent-based model. Issues we are going to take into account in the modelling process.

The App-model will quantify the Android malware infection evolution (to know the real threat for the users), the number of potential infected smartphones (to estimate the population of smartphones affected by malware) and the type of malware that affects these infected smartphones in the Community of Valencia, Spain [46]. The results can be exported to other regions where the number of Android smartphone users is fairly stable.

We must say that other approaches as machine learning or data mining techniques could be used to study the evolution of the malware infection, however these techniques do not take into account the behavior of the actors (Markets, Apps and Clients). The knowledge of their behavior and how they interact allows us to simulate new scenarios where the behavior may be different and predict the evolution of the malware infection considering these changes.

Additionally, note that with the results of the model we will be able to analyse the critical part of the smartphones business model related to malware, i.e., we will find out which part is more critical: the users, giving indiscriminate permissions to the Apps or not protecting their devices with mobile antivirus software, or the Android platform, due to the vulnerabilities of the Android devices that permits their rooted. Furthermore, we will be able to estimate the cost the users should afford in case that they have in their devices malware that causes financial charges.

The chapter is organized as follows. In Section 3.2 we present the agents of the model: Apps, markets, users, habits, etc. In Section 3.3 we describe how the agent-based model evolves over the time. Section 3.4 is devoted to carry out simulations, present results and discuss them. Conclusions are drawn in Section 3.5.

3.2 Material and methods

To conduct our study, we set the time period in a month. The starting time-point ($t = 0$) is Jul 2011. This has been chosen because in Jul 2011 none or only very few smartphones could have been infected.

The agent-based approach allows the analysis of service interactions among the agents and fits perfectly the relation between mobile device users and App markets.

Then, in this model, two domains including their agents will be considered: the markets, where the agents are the Apps that belong to different markets; the users, where the agents are the mobile devices (or clients) that belong to every user. The study of the behavior of the agents is studied in this section.

The users, with their own characteristics in their devices, access the markets and download applications (Apps) with also different characteristics. Thus, we consider two domains interacting between them:

- The markets environment.
- The users environment.

Figure 3.2 shows an UML (Unified Modelling Language) representation of the Apps and Clients (users). *App*, that represents the application in a given market, and *Client*, that represents the device of every user.

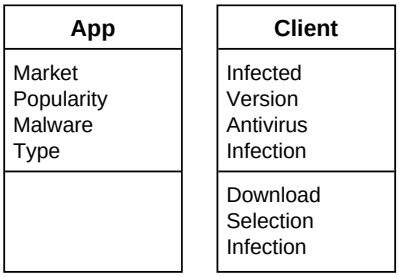

Figure 3.2: Agent attributes and functions.

3.2.1 The Apps

The mobile malware spread through the Apps that are in the markets and the users download to their device. The Apps are stored in the markets and these markets can be official, as Google Play, or alternative or non-official markets. This is determined by the attribute *Market*. Every App has its own popularity that determines the probability to be downloaded and that it is stored at the attribute *Popularity*. Furthermore, malware can be classified depending on the effect they produce over the *Client* [31]:

- *Privilege Escalation*: The App gets the root privileges of the device. Depending on the Client's OS version, this kind of malware affects or not.
- *Remote Control*: Remote servers take the control of the device.
- *Financial Charge*: The App sends messages to premium accounts from the device and the money these messages cost has to be paid by the user of the smart-phone.
- *Information Collection*: The App takes private information of the device, like the contacts, agenda, SMS messages, user accounts, etc., and upload the information to a remote server.

If an App is malware and what kind of malware is, it is established by attributes Malware and Type respectively.

3.2.2 Official market

The official market, also known as Google Play [53], is a repository of Apps where the users of Android smart-phones can download freely or under payment Apps, music, movies or books. We are going to focus only in Apps because they are the ones responsible of malware in the cell phones.

New Apps entering every month in the official market

In Jul 2011, the number of applications in Google Play were 221 875, [42]. Now, we want to describe the behavior of the official market to estimate the number of new apps every month. Some values taken in different dates from July 2011 can be seen in Table 3.1 [42].

Date	**#Apps**
July 1, 2011	221 875
Sept 1, 2011	271 875
Nov 1, 2011	309 375
Jan 1, 2012	343 750
Mar 1, 2012	400 000
May 1, 2012	440 645
May 20, 2012	443 920
Feb 12, 2013	626 865

Table 3.1: Number of Apps in the official market.

Taking into account that the evolution is practically a linear function, we can fit $b + a\,t$ with data of Table 3.1, obtaining the function

$$f_{OM}(t) = 225\,970 + 20\,740.1\,t, \tag{3.1}$$

where t is the number of months since July 2011. Function (3.1) allows us to estimate the number of Apps in the official market over the next months.

New malware Apps entering every month in the official market

Data about malware is very difficult to find and they may not be reliable because the sources use to be antivirus developer companies. In spite of this, in order to conduct the study, we have had to trust in the few available data published in [31] appearing in Table 3.2.

In this case, we have less data as before and an appropriate fitting is not as good as we did above. Nevertheless, we are going to assume that the

Date	#Apps
July 1, 2011	86
Aug 1, 2011	86
Sept 1, 2011	103
Oct 1, 2011	200

Table 3.2: Number of malware Apps in the official market.

growing of malware Apps also has a linear increasing and the line that best fit the data in Table 3.2 is the function

$$f_{OMm}(t) = 64.9 + 35.9\,t, \tag{3.2}$$

where t is the number of months since July 2011.

Distribution of Apps according their popularity

In the Android markets, Apps are classified according to their popularity as: none; less than 2.5 stars; $2.5 - 3$ stars; $3 - 3.5$ stars; $3.5 - 4$ stars; $4 - 4.5$ stars; greater than 4.5 stars. After some accesses to the distribution of the Apps by popularity website in different dates [43], we noted that there were minor changes and consequently we assume that the distribution of Apps by popularity is constant over the time. The distribution for Jul 2011 is given in Table 3.3.

Popularity	None	2.5	$2.5 - 3.0$	$3.0 - 3.5$	$3.5 - 4.0$	$4.0 - 4.5$	> 4.5
#Apps	114 789	5 985	6 053	13 512	21 599	31 493	28 444
%Apps	51.74%	2.70%	2.73%	6.09%	9.73%	14.19%	12.82%

Table 3.3: Distribution of Apps by popularity in Jul 2011.

Distribution of malware Apps according their popularity

The distribution of malware Apps is not uniform among popularity ratings. There is a way to create malware Apps called *repackaging* [31]. Repackaging consists of taking a popular App, introducing some malware code and upload it again. 86% of malware is repackaging [31] and we are going to assume that these malware Apps have popularity $4.0 - 4.5$ or > 4.5 distributed uniformly. Thus, in 3.4, we can see the distribution of the malware Apps in Jul 2011 distributed by popularity.

Popularity	None	2.5	$2.5-3.0$	$3.0-3.5$	$3.5-4.0$	$4.0-4.5$	>4.5
#Apps	9	0	0	1	2	39	35

Table 3.4: Distribution of malware Apps by popularity in July 2011, taking into account repackaging.

Malware detection

The official Android market has a service named *Bouncer* which provides automated scanning of Apps uploaded to Android market before being available for the users that detects potentially malicious software. The admitted effectiveness of this service is around 40% [49]. This parameter will be considered in order to know the probability that the official market detects a malware App and withdraw it.

Distribution of malware Apps according to their type

The distribution of malware Apps, according to [31], in the official market by their type is shown in Table 3.5.

Type	**%**	**Type**	**%**
Financial Charge	14.22%	Remote Control	43.49%
Privilege Escalation	14.22%	Information Collection	28.07%

Table 3.5: Distribution of malware Apps in the official by market according to their type.

3.2.3 Non–official market

Non-official markets are markets other than Google Play where the users can also download Android Apps. The behavior of these markets are similar to the official market, however, some differences should be taken into account because their relevance on the malware infection. First of all, we are going to assume that all the non-official markets are gathered in only one with more than 2 600 000 000 of downloads [48].

New Apps entering every month in the non-official market

Non-official market [48] had 568 661 available Apps in Jan 2012 whereas GooglePlay had, in the same month, 343 750. Taking into account that

no much more data about the number of Apps in non-official market are available, we are going to assume that the ratio 1.65 (568 661/343 750) is a constant relation of the number of Apps in the official and non-official markets over the time. Therefore

$$f_{NOM}(t) = 1.65\, f_{OM}(t), \tag{3.3}$$

describes the evolution of Apps in the non-official market, where t is the number of months since July 2011.

Malware Apps entering every month in the non-official market

Data about malware in the non-official market can be found in [31] and can be seen in Table 3.6.

Date	**#malware Apps**
July 1, 2011	485
Aug 1, 2011	810
Sept 1, 2011	1008
Oct 1, 2011	1172

Table 3.6: Number of malware apps in the non-official market.

The above data can be fitted accurately using a linear function, obtaining

$$f_{NOMm}(t) = 529.9 + 225.9\, t, \tag{3.4}$$

where t is the number of months since July 2011.

Distribution of Apps according their popularity

Using the same criteria as in the official market, we classify the Apps depending on their popularity in the non-official market as given in Table 3.7. We also consider this distribution constant over the time.

Popularity	None	2.5	$2.5 - 3.0$	$3.0 - 3.5$	$3.5 - 4.0$	$4.0 - 4.5$	> 4.5
#Apps	189 894	9 900	10 014	22 353	35 730	52 098	47 055
%Apps	51.74%	2.70%	2.73%	6.09%	9.73%	14.19%	12.82%

Table 3.7: Distribution of Apps by popularity in in July 2011 in the non-official market.

Distribution of malware Apps according their popularity

Using the same criteria as in the official market, we classify the Apps depending on their popularity in the non-official market as given in Table 3.8. Repackaging is also considered.

Popularity	None	2.5	$2.5-3.0$	$3.0-3.5$	$3.5-4.0$	$4.0-4.5$	>4.5
#Apps	48	3	3	6	9	219	198

Table 3.8: Distribution of malware Apps by popularity in July 2011 in the non-official market.

Malware detection

We do not have any information about the existence of an antivirus checking if the new Apps contain malware code in the non-official market. Therefore, we are going to assume that the non-official market does not have any control about the malware Apps.

Distribution of malware Apps according to their type

The distribution of malware Apps, according to [31], in the non-official market by their type is shown in Table 3.9.

Type	**%**	**Type**	**%**
Financial Charge	50.10%	Remote Control	10.06%
Privilege Escalation	35.58%	Information Collection	4.26%

Table 3.9: Distribution of malware Apps in the non-official market according to their type.

3.2.4 Users

The *Client* attributes determine if it is infected or not (attribute Infected), the OS version of the client's device (attribute *Version*), if the device has or not software protection (attribute *Antivirus*) and the kind of infection in case of an infected client (attribute *Infection*). The *Version* attribute is used in order to know if a *Privilege Escalation* malware affects or not the *Client*.

In 2011 there were in Spain a population of 47 190 493 people [39]. 46% of them had a smartphone [2] and 50% of them had an Android terminal

[44], that is, 10 853 813. The population in the Community of Valencia in 2011 was 5 117 190 inhabitants [39]. Applying the same rule as above we have that, in the Community of Valencia there were 1 176 954 Android smartphones.

Number of Apps downloaded per month

The number of Apps downloaded by a user in a month follows a Poisson distribution:

$$f(k, \lambda) = \frac{e^{-\lambda}\, \lambda^k}{k!}, \tag{3.5}$$

where k is the number of downloaded Apps and λ is the average number of Apps downloaded, both every month in every smart-phone. Taking into account that, in Spain, 67 293 800 Apps were downloaded in Spain in Apr 2012 [58] by 10 853 813 smart-phones, every user downloads an average of 6.2 Apps per month. Therefore, $\lambda = 6.2$.

App downloads by popularity

In Chapter 2, we have shown that top reason for downloading an app is recommendation by others. When a user downloads an App, unless he/she wants to download a specific App, the probability of download a popular App will be higher, because the more recommended Apps are the more popular, or, if the user is searching on the market, because he/she will have a look among the most popular Apps. Therefore, the Apps are not downloaded following a uniform distribution. In order to approach this behavior, let us consider the Figure 3.3, [43].

As we mentioned before, we assume that the distribution of the Apps per popularity is constant over the time. However, we do not have data about the values where the colors change in Figure 3.3, and we did an estimation gathered in the Table 3.10.

Popularity/ #downloads	< 500	$500 - 5\,000$	$5\,000 - 50\,000$	$> 50\,000$
None	92%	8%	0%	0%
< 2.5	28%	51%	19%	2%
$2.5 - 3.0$	17%	45%	32%	6%
$3.0 - 3.5$	16%	44%	31%	9%
$3.5 - 4.0$	14%	37%	35%	14%
$4.0 - 4.5$	14%	37%	32%	17%
> 4.5	37%	42%	16%	5%

Table 3.10: Estimation of the percentage of download distribution of Android Apps per number of downloads.

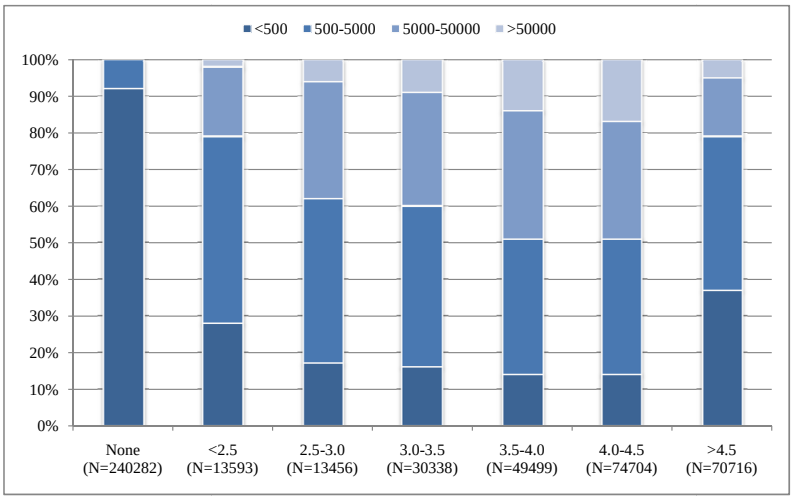

Figure 3.3: Downloads per popularity.

Let us denote by $p(i,j)$, $i = 1,\dots,7$, $j = 1,2,3,4$ the entries in Table 3.10. For instance, $p(2,3) = 19\%$. Also, we call c_1, c_2, c_3 and c_4, $0 \leq c_1 < c_2 < c_3 < c_4$, the average number of downloads of Apps with less than 500 downloads, between $500 - 5\,000$ downloads, between $5\,000 - 50\,000$ downloads and more than 50 000 downloads, respectively. Then, taking into account that the number of Apps per popularity (h_j) are 240 282, 13 593, 13 456, 30 338, 49 499, 74 704 and 70 716 (see Figure 3.3), the number of total downloads will be

$$\sum_{j=1}^{7} \sum_{i=1}^{4} c_i \, p(j,i) \, h_j \tag{3.6}$$

where

$$\begin{gathered} h_1 = 240\,282, \quad h_2 = 13\,593, \quad h_3 = 13\,456, \quad h_4 = 30\,338, \\ h_5 = 49\,499, \quad h_6 = 74\,704, \quad h_7 = 70\,716. \end{gathered} \tag{3.7}$$

Substituting and simplifying the expression (3.6) we have

$$\frac{1}{50} \left(13\,778\,021\, c_1 + 6\,060\,737\, c_2 + 3\,441\,893\, c_3 + 1\,348\,749\, c_4\right).$$

If we were trying to find out the average number of downloads (c_i, $i = 1,2,3,4$) for all the people over the world, we would have to assume that $c_1 < 500$, $500 \leq c_2 < 5\,000$, $5\,000 \leq c_3 < 50\,000$ and $c_4 > 50\,000$. However,

we are going to restrict the downloads to the Community of Valencia and consequently, c_i, $i = 1, 2, 3, 4$ do not have to satisfy the above restrictions. In fact, they will be much lower. Thus, taking into account that 67 293 800 Apps were downloaded in Spain in Apr 2012 [58] (closest data available to Jul 2012), the population in Spain in Apr 2012 was 46 185 697 inhabitants and in the Community of Valencia is 5 009 635 [39], we are going to assume that the number of Apps downloaded in the Community of Valencia in Apr 2012 was

$$67\,293\,800\,\frac{5\,009\,635}{46\,185\,697} = 7\,299\,173 \text{ Apps}. \tag{3.8}$$

Consequently, for the Community of Valencia we have that the following equality should be satisfied

$$\begin{aligned}\frac{1}{50}\,(13\,778\,021\,c_1 + 6\,060\,737\,c_2 + 3\,441\,893\,c_3 + 1\,348\,779\,c_4) \\ = 7\,299\,173.\end{aligned} \tag{3.9}$$

Isolating c_1, we have

$$\begin{aligned}c_1 \;=\; & \frac{887\,150\,988\,850\,000}{33\,491\,973\,850\,823} - \frac{6\,060\,737}{13\,778\,021}\,c_2 \\ & - \frac{3\,441\,893}{13\,778\,021}\,c_3 - \frac{1\,348\,7497}{13\,778\,021}\,c_4.\end{aligned} \tag{3.10}$$

Taking into account that $0 \leq c_1 < c_2 < c_3 < c_4$, c_4 will take its maximum value when $c_1 = c_2 = c_3 = 0$ and, in this case, we have that

$$\frac{887\,150\,988\,850\,000}{33\,491\,973\,850\,823} - \frac{1\,348\,7497}{13\,778\,021}\,c_4 = 0, \tag{3.11}$$

and the maximum value that c_4 can reach is 270.59. Summarizing the above reasoning, if we call d_1, d_2, d_3 and d_4 the probabilities a user in the Community of Valencia downloads an App which number of downloads in all the world are less than 500, between $500 - 5\,000$, between $5\,000 - 50\,000$ or more than 50 000 downloads, respectively, is

$$d_i = \frac{c_i}{C}, \quad i = 1, 2, 3, 4, \tag{3.12}$$

where $C = c_1 + c_2 + c_3 + c_4$ and c_i, $i = 1, 2, 3, 4$ should satisfy that

$$\begin{aligned}c_1 \;=\; & \frac{887\,150\,988\,850\,000}{33\,491\,973\,850\,823} - \frac{6\,060\,737}{13\,778\,021}\,c_2 \\ & - \frac{3\,441\,893}{13\,778\,021}\,c_3 - \frac{1\,348\,7497}{13\,778\,021}\,c_4,\end{aligned} \tag{3.13}$$

where

$$0 \leq c_1 < c_2 < c_3 < c_4 < 270.59. \tag{3.14}$$

OS version evolution and infection by *Privilege Escalation* malware

The OS version is an important parameter in order to estimate the infection by *Privilege Escalation* malware. We assume the evolution of the OS version installed on the smart-phones as given in Table 3.11 [44].

Version	Affected	July 2011	Oct 2011	Feb 2012	June 2012	Oct 2012	Feb 2013
1.5	Cupcake	1.40%	0.90%	0.40%	0.20%	0.10%	0.00%
1.6	Donut	2.20%	1.40%	0.80%	0.50%	0.30%	0.20%
2.1	Eclair	17.50%	10.70%	6.60%	4.70%	3.10%	1.90%
2.2	Froyo	59.40%	40.70%	25.30%	17.30%	12.00%	7.50%
2.3	Gingerbread	18.60%	44.40%	62.00%	64.00%	54.20%	44.10%
3.0	Honeycomb	0.90%	1.90%	3.30%	2.40%	1.80%	1.20%
4.0	Icecream	0.00%	0.00%	1.60%	10.90%	25.80%	28.60%
4.1	Jelly	0.00%	0.00%	0.00%	0.00%	2.70%	16.50%

Table 3.11: Distribution of OS version in Android smart-phones from July 2011 until Feb 2013.

The percentage of devices that can be affected by the most common Android privilege escalation vulnerabilities is given in Table 3.12 [40].

Version	**Name**	**Affected**
1.5	Cupcake	100%
1.6	Donut	100%
2.1	Eclair	96.70%
2.2	Froyo	98.80%
2.3	Gingerbread	100%
3.0	Honeycomb	0.00%
4.0	Icecream	31.00%
4.1	Jelly	0.00%

Table 3.12: Percentage of devices that can be affected by the most common privilege escalation vulnerabilities, depending on the Android OS version.

Users with antivirus installed in their devices

The number of users with antivirus installed in their devices is 33% [20]. The admitted effectiveness of these antivirus software is between 20.2% and

79.6% [31].

Conditions for a user to be infected by malware App

We will know if a downloaded malware App infects the client if one of the following conditions are met:

- Malware App (Privilege escalation) + Vulnerable OS + None antivirus installed.
- Malware App (Remote Control, Financial Charge or Information Collection) + not antivirus installed.
- Malware App (Privilege escalation) + Vulnerable OS + Probability of no detection by antivirus installed.
- Malware App (Remote Control, Financial Charge or Information Collection) + Probability of no detection by antivirus installed.

Probability a user detects his/her smart-phone is infected and repair it

We assume that a user only detects and repairs infections caused by Financial Charge malware. The detection is made monthly when the user receives the mobile bill. Other cases are difficult to estimate but we consider also that a smart-phone user changes his/her smart-phone, as average, every 11.5 months.

3.2.5 Methods

The App has the attributes *Malware* and *Type* that indicate whether an App is malware and its type, respectively. Given that the effect over the client produced by a malware App can be one or more of the malicious payload described, we consider that, if a malware App carries more than one payload, the type of the malware App belongs to the most upper level payload, according to Financial charge; Privilege escalation; Remote control; Information collection.

Whether the *Client* is infected or not, the OS version, if the device has or not software protection and the kind of infection, are the attributes of the client. The *Privilege Escalation* malware affects the client depending on the OS version as we have seen before.

Additionally, we consider that *Clients* download a certain number of Apps every month, determined by *Download* method, selects the downloaded App

by the method *Selection* and determines if the downloaded App infects the client or not with the *Infection* method. More details related to download process are:

- *Download Method*: We admit that the number of Apps downloaded by a user in a month follows a Poisson distribution:

$$f(k, \lambda) = \frac{e^{-\lambda}\, \lambda^k}{k!},\ k = 0, 1, 2, \ldots, \tag{3.15}$$

  where k is the number of downloaded Apps and $\lambda > 0$ is the average number of Apps downloaded every month in every smartphone ($\lambda = 6.2$).

- *Selection Method*: Knowing k from *Download Method*, this method selects randomly k Apps from the markets. The selection will depend on the popularity and the number of downloads.

- *Infection Method*: With the k selected Apps, we take the ones that are malware, and this method determines if the App affects the *Client* or not, depending on the App attributes (*Malware* and *Type*) and the *Client* attributes (OS version and Antivirus).

3.3 The App-Model evolution rules

The users and the markets have their own rules that define the initialization point and the evolution for the agents sets. The evolution rules for the client agents simulate the behavior of the users, establishing how many Apps are downloaded monthly by a client, how the App selection method by the client based on the App's popularity is, if the downloaded App infects the device and how long a user changes his/her device.

The evolution rules for the App markets establish the number of new Apps in every market each month, how the markets control the new submitted Apps (Google Play uses *Bouncer* which scans submitted Apps looking for malware), how the markets distribute the Apps by popularity, etc.

Then, using the considerations introduced so far, we are going to describe the evolution rules of the model. Recall that the time period is a month and the starting point of the model $t = 0$ corresponds to Jul 2011.

First, we sample percentages d_1, d_2, d_3 and d_4 as described in Equations (3.12), (3.13) and (3.14). Then for every month t:

- State the official market:

- Determine the number of Apps in this market in month t according to Equation (3.1).
- Distribute them according to their popularity following the percentage values in Table 3.3.
- Determine the number of malware Apps in this market in month t according to Equation (3.2).
- Distribute them according to their popularity following the percentage values in Table 3.4.
- Malware detection: 40% of malware is detected and removed.

- State the non-official market:
  - Determine the number of Apps in this market in month t according to Equation (3.3).
  - Distribute them according to their popularity following the percentage values in Table 3.7.
  - Determine the number of malware Apps in this market in month t according to Equation (3.4).
  - Distribute them according their popularity following the percentage values in Table 3.8.

- User behavior. For every user:
  - Download method: Take a random value u between 0 and 1 and obtain the maximum value of k such that $\sum_{j=1}^{k} f(j, \lambda) \leq u$ (see expression (3.15)).
  - Selection method. Select k Apps from each market with a probability of 50%, in such a way that their popularity is rated according to the probabilities d_1, d_2, d_3 and d_4, and malware or not with probability $\frac{f_{OMm}(t)}{f_{OM}(t)}$ for the official market and $\frac{f_{NOMm}(t)}{f_{NOM}(t)}$ for the non-official market.
  - Infection method: If some of the downloaded Apps is malware, for each malware App:
    1. If it has been downloaded from the official market, determine its type with probabilities given in Table 3.5. Then, it infects the smartphone depending on the OS installed (Table 3.11), if there is antivirus and its effectiveness.

2. If it has been downloaded from the non-official market, determine its type with probabilities given in Table 3.9. Then, it infects the smartphone depending on the OS installed (Table 3.11), if there is antivirus and its effectiveness.

– Check if the user detects if the smartphone is infected and fix it. This happens only in case the malware is Financial Charge and the repair is done at the end of the month.

– Check if the user changes his/her smartphone (every 11.5 months in average).

The algorithmic evolution of the App-Model described above, is drawn as the flowchart shown in Figure 3.4. The left side of the figure represents the evolution of the clients and the right side, the evolution of the apps, that evolve in parallel. The start point represents the initial month of the model ($t = 0$), where the model creates the clients and set their attributes. After this, and for every step ($t = i$), the models begins its evolution and all the clients (left side of the figure), run their methods in the showed order and change, if needed, their attributes. Also, for every step ($t = i$), the model establishes the markets, that are changing every month, set the apps attributes and group them depending on the number of downloads (right side of the figure). After this, and for every step, the number of apps of the markets are recalculated according their evolution curve. All these processes run in parallel, but on every step, the selection method of the clients can be executed only after the apps are grouped.

3.4 Results and discussion

Once the model has been built and the evolution rules stated, there are some model parameters unknown but satisfying some restrictions:

- Apps download percentages per popularity d_1, d_2, d_3, d_4 and d_5, satisfying Equations (3.12), (3.13) and (3.14),
- the percentage of smartphones with antivirus, denoted by A, is in $[0, 0.66]$ [20],
- the effectiveness of the antivirus protection, denoted by E, is in $[0.202, 0.796]$ [31].

Now, in first place, we are going to see if the model output depends on the number of smartphone users. If it is, we will have to simulate the behavior

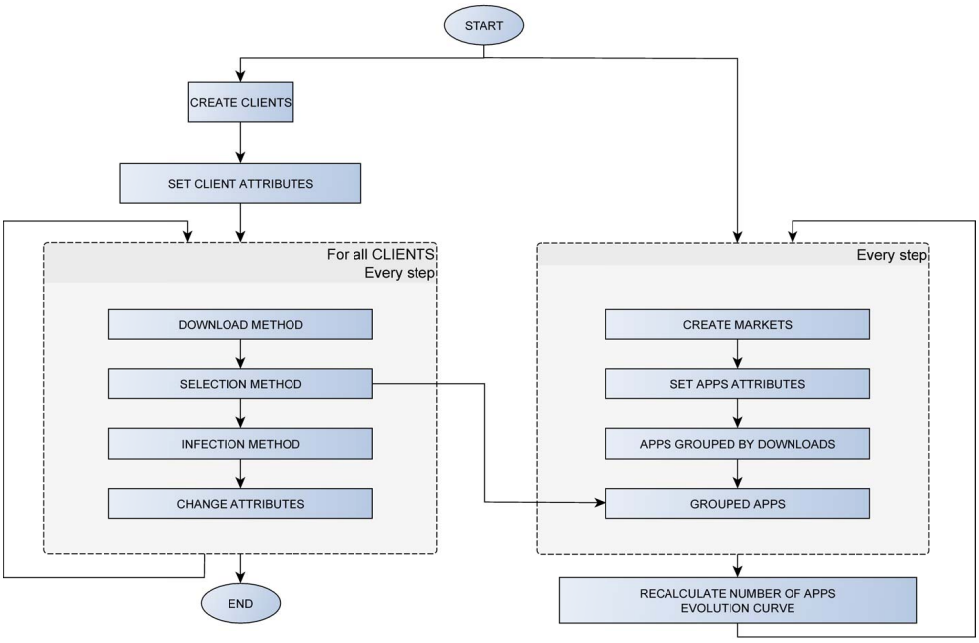

Figure 3.4: App-Model flowchart. In this figure we describe the evolution process of the model from $t = 0$ (Start point) to $t = T$ (End point), showing, for every time instant, the creation of the agents, the assignation of attributes, the order of performance of the methods and their interaction.

of 1 176 954 users. Otherwise, we will be able to reduce the number of users in order to run the simulation very much quicker.

Secondly, we will simulate a large amount of runs in order to estimate the number of the monthly infections by malware Apps.

3.4.1 Model evolution depending on the number of users

In this first experiment, we take fixed values of d_1, d_2, d_3, d_4, d_5, A and E and we run simulations for 1000, 5000, 7000, 10000, 15000, 20000, 30000, 40000, 50000, 65000, 80000, 100000, 120000 and 150000 users during $t = 1, \ldots, 15$ months. Then, in Table 3.13 we can see the comparison of percentage of cumulative (aggregated) and residual (new ones) infected users for month $t = 15$. Few differences can be noted. Therefore, we do not need to simulate the 1 176 954 Android smartphones in the Community of Valencia to obtain

reliable and accurate results. After some tests, we decided to consider 50 000 users.

No. of users	% of accumulated infected	% of residual infected
1000	5.10%	0.70%
5000	5.52%	0.68%
7000	5.29%	0.59%
10000	5.23%	0.63%
15000	4.77%	0.53%
20000	4.95%	0.48%
30000	4.88%	0.51%
40000	5.19%	0.52%
50000	5.13%	0.55%
65000	5.12%	0.53%
80000	5.06%	0.55%
100000	5.15%	0.56%
120000	4.89%	0.56%
150000	5.01%	0.55%

Table 3.13: Comparison of percentage of cumulative and residual infected users for month $t = 15$. The results are very similar.

3.4.2 Estimations

Thus, in order to compute reliable estimations based on 95% confidence intervals (CI95%), we use the technique called Latin Hypercube Sampling (LHS) [15] to select sets of parameters to be substituted into the model. Latin Hypercube Sampling (a type of stratified Monte Carlo sampling) is an efficient method for achieving equitable samples of all input parameters simultaneously. Moreover, the random selection of the sets of parameters done by LHS, will allow us to study the model sensitivity by the CI95%.

In our case, taking 50 000 smartphone users, starting in Jul 2011 and finishing in Dic 2014 ($t = 0, 1, 2, \ldots, 41$ months), and following the evolution rules, LHS was used to generate 100 000 different values of each input parameter d_1, d_2, d_3, d_4, d_5, A and E sampled as follows:

1. Sample values $0 \leq c_1 < c_2 < c_3 < c_4 < 270.59$ such that $c_1 = 26.4885 - 0.439884c_2 - 0.24981c_3 - 0.0978913c_4$, and calculate $d_i = \frac{c_i}{C}$, $i = 1, 2, 3, 4$.

2. Sample a value of A uniformly in the interval $[0, 0.66]$.

3. Sample a value of E uniformly in the interval $[0.202, 0.796]$.

We used these samples to run 100 000 evaluations of the model obtaining 100 000 model outputs (infected smartphones) for each month $t = 0, 1, 2, \ldots, 41$. Then, for each month we take the 100 000 model outputs and calculate the mean and the 95% confidence intervals taking into account the empirical 2.5% and 97.5% percentiles.

In Figure 3.5 we can see the evolution of the cumulative infections since Jul 2011 until Dic 2014 with a 95% confidence interval. In Table 3.14 we can see the numerical values of the mean and CI95% of the cumulative infections in the Community of Valencia in Jul 2013, Jul 2014 and Dic 2014.

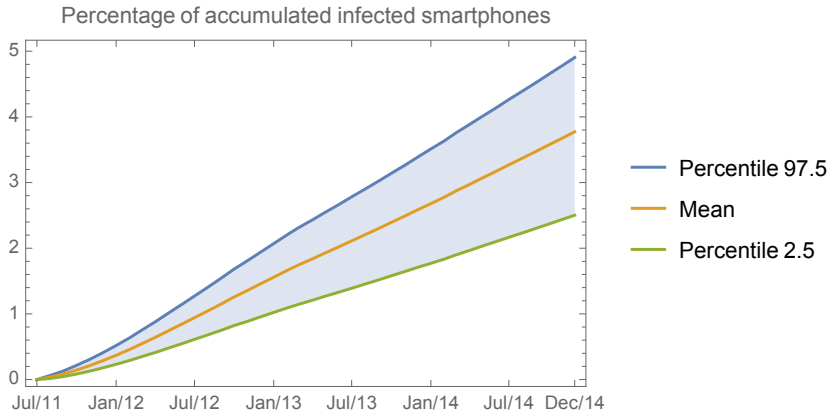

Figure 3.5: Model evolution of the cumulative smartphone infections every month since Jul 2011 until Dic 2014. The line in the middle is the mean and those up and down correspond to 95% confidence interval.

In Figure 3.6 we can see the evolution of the new (residual) infected smartphones every month with a 95% confidence interval since Jul 2011 until Dic 2014. It can be seen that, since Oct 2012, there is a certain stabilization in the number of new infected smartphones. In Table 3.15 we can see the numerical values of the mean and CI95% of the residual infections in the Community of Valencia in Jul 2013, Jul 2014 and Dic 2014.

Finally, in Figure 3.7, we show the mean and the 95% confidence interval of cumulative infected smartphones by Privilege Escalation (PE) and Financial Charge (FC) malware. Comparing Figure 3.7 to Figure 3.5 we can see that Financial Charge malware infects a half of the smartphones according to [36, 31]. In Table 3.16 we can see the numerical values of the mean and CI95% of the cumulative infections in the Community of Valencia in Jul 2013, Jul 2014 and Dic 2014.

	Mean	**CI**95%
Jul	86163	[57388, 110540]
2013	7.32%	[4.88%, 9.39%]
Jul	139623	[93191, 178450]
2014	11.86%	[7.92%, 15.16%]
Dic	162788	[108680, 207756]
2014	13.83%	[9.23%, 17.65%]

Table 3.14: Mean and CI95% of the accumulated infected smartphones in Jul 2013, Jul 2014 and Dic 2014 in the Community of Valencia and the corresponding percentages predicted by the model.

	Mean	**CI**95%
Jul	3818	[2448, 5108]
2013	0.32%	[0.21%, 0.43%]
Jul	4037	[2613, 5367]
2014	0.34%	[0.22%, 0.46%]
Dic	4105	[2660, 5461]
2014	0.35%	[0.23%, 0.46%]

Table 3.15: Mean and CI95% of the residual infected smartphones in Jul 2013, Jul 2014 and Dic 2014 in the Community of Valencia and the corresponding percentages predicted by the model.

3.4.3 Model validation

In [24], S.M. Patterson talks about Google's Android Security chief Adrian Ludwig who gave a talk at the Virus Bulletin conference in Berlin. In this talk, Ludwig said that the problem Google wants to solve is that most independent security researchers do not have access to a platform such as Google's to measure how many times a malware App has been installed. Also, he mentioned that security researchers are very good at finding and fixing malware, but in the absence of reliable data that indicate how frequently a malware App has been installed, the threat level can become exaggerated. Reports that reach publication are often extremely exaggerated. To emphasize this point, Ludwig revealed in his analysis that some of the most publicized recent malware discoveries are installed in less than one per million installations. Additionally, he reported that based on the data from tracking over one and a half billion App installs, Google obtained convincing evidence that the rate of *potentially harmful Apps* installed is stable at about 1 200 per million App

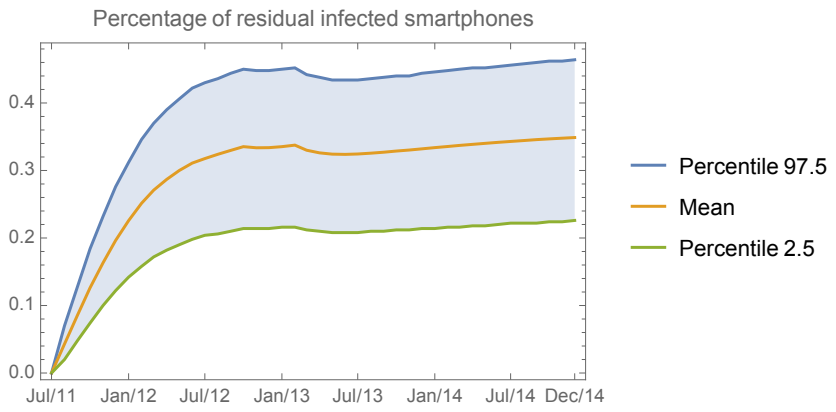

Figure 3.6: Evolution of new smartphone infections every month since Jul 2011 until Dic 2014. The line in the middle is the mean and those up and down correspond to 95% confidence interval. Nowadays, there is a stabilization in the number of new infected smartphones.

installs, or about 0.12%.

Furthermore, the official reports as F-Secure Report (Mobile Threat Report Sep 2013), Trend Micro Report (Trend Labs Securtity Report 3Q 2013) or Secure List Report (Mobile Malware Evolution Feb 2014) do not show the number of devices affected by installed malware Apps, but also the number of Apps detected as malware.

As a consequence, to compare the figures given by the proposed model to the real ones is not going to be an easy task because of lack of real data. In fact, to our knowledge, the only data about potentially harmful Apps installed is the one mentioned above: stable and about 0.12%.

Then, taking into account that the conference was held in Oct 3rd, 2013 [24], we may compare this data with prediction of the model for new smartphone infections in Sep 2013: stable and mean 0.33% with CI95% $[0.21\%, 0.44\%]$.

Hence, our model predicts a stable situation of harmful Apps installed, as Google says, and a little bit higher number of infected smartphones than Google. This slight difference may be due to the development of the techniques for detecting malware during the period of time considered in our simulation, resulting in increased effectiveness of antivirus software than that used in the initial parameters of our simulation in terms of the effectiveness of antivirus software and, therefore reducing the number of malware installed in the Google analysis. Taking into account this regard, we consider that our model provides valid results in terms of estimation of number of infected

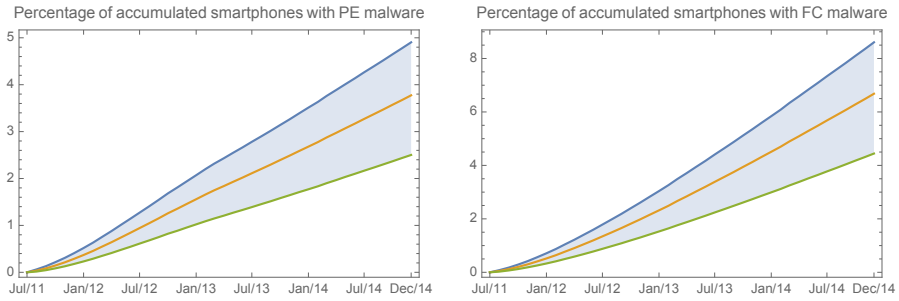

Figure 3.7: Evolution of the cumulative smartphone infections due to Privilege Escalation (PE) on the left, and Financial Charge (FC) on the right, malware every month since Jul 2011 until Dic 2014. The line in the middle is the mean and those up and down correspond to the 95% confidence interval.

smartphones and in terms of stable evolution of the infections.

3.5 Conclusion

In this chapter we present an agent-based model to quantify the Android malware infection evolution. Some model outputs are compared to data given by Google and the results are fairly similar, stable and a little bit higher for the model predictions.

Considering the parameters of our model and our simulations, the obtained results show that, given a specific population of devices with Android OS:

- A mean of 0.3% of devices are infected every month by some kind of malware. This number is stable over the time from Oct 2012 onwards, considering the growing curve for the total Apps and malware Apps.
- Taking into account cumulative values from Jul 2011 to Dic 2014, we predict that the infections will be around a mean of 13.83% over the total number of devices considered.
- From this 13.83%, around the half of the total (48%) will be infections by Financial Charge malware type, and around a third (27%) will be infections by Privilege Escalation malware type. The remainder (25%) will be infections by Remote Control and Information Collection malware type.

	Privilege Escalation			Financial Charge	
	Mean	**CI**95%		**Mean**	**CI**95%
Jul 2013	24857 2.11%	[16360, 32743] [1.39%, 2.78%]		39824 3.38%	[26340, 51857] [2.24%, 4.41%]
Jul 2014	38481 3.27%	[25493, 50185] [2.17%, 4.26%]		66861 5.68%	[44418, 86341] [3.77%, 7.34%]
Dic 2014	44414 3.77%	[29447, 57718] [2.50%, 4.90%]		78643 6.68%	[52304, 101289] [4.44%, 8.61%]

Table 3.16: Mean and CI95% of the accumulated infected smartphones by Privilege Escalation and Financial Chage in Jul 2013, Jul 2014 and Dic 2014 in the Community of Valencia and the corresponding percentages predicted by the model. These figures can give us an idea about the amount of money that the Financial Charge malware moves every month.

- Thus, the infections by Financial Charge, Remote Control and Information Collection malware type are due to the users because they give indiscriminate permissions to the Apps and do not protect properly their the mobile with antivirus software. Therefore, we show that two thirds of the infections are caused by these two factors, showing that the most critical part for the malware infections at smartphones are the users habits and the ineffectiveness of the traditional antivirus software, not due to the OS vulnerabilities.

- Quantifying and monetizing the Financial Charge malware incidence, we can consider that, from the 0.3% new infected devices during a month, the half part are infected by Financial Charge and that every infection causes a monthly overrun of 30 euros[1] in every device. Considering that the total population of Android devices in Spain is 10 853 813, the number of infected devices by Financial Charge malware type during a month are 16 280 (i.e. 0.15%) and the financial charge caused by this kind of malware during a month will be 488 400 euros.

With our model, we show realistic data that can be considered in order to quantify the real threat for the users and the number of potential infected smartphones. With these results, we consider that preventive strategies against mobile malware should be developed mainly focusing on new mal-

[1]We have some examples of mobile bills such that their owners suffered an infection of Financial Charge malware and the amount of these bills are around 30 euros.

ware detection approaches before being downloaded by the users, because, as we shown, the users decisions and the ineffectiveness of the traditional antivirus software approach are the critical part for the infections.

Moreover, with the presented model, despite the increasing of Apps, we could see that the number of new infected smartphones achieved stable figures, and then, it is not expected a significant change in the current stable trend.

One of the most interesting features of the presented model is that if some of the parameters vary because of changes in the behavior of the actors (Markets, Apps and Clients) we only have to tune the corresponding model parameters and perform the simulations to predict the evolution of the infected smartphones for the new scenario.

Finally, we want to point out that this model and simulations can be extrapolated to other regions where the number of Android smartphones is fairly stable over the time.

Some of the results presented in this chapter have been published in [7, 4].

Chapter 4

Conclusion

In this dissertation, we focus our study in the network effects related with the mobile applications of the smartphones.

To do that, we propose mathematical network models to analyse the dynamics of the user behavior and the mobile applications, considering the networks ("offline" and online) at which the users belong to. These networks determine how the information and viruses are shared and transmitted, thus, the mobile applications' spread and behavior, and the spread of malware through mobile apps, can be modeled taking into account parameters such as users behavior, on their "offline" and online social network, and technical issues of the mobile devices, thus, to model the networks, both factors have been taken into account.

As a result of the work done, in the following, we point out the main contributions and general conclusions of this dissertation:

1. Under the mobile applications social behavior point of view:
    - We have shown and validate that network effects are present in the spread of mobile apps.
    - We have shown that face to face relations are the most important factor for apps' adoption.
    - We have shown that apps are often discovered outside the app store and sources of awareness of smartphone apps are friends, family, and colleagues, as Google claims.
    - We have shown that top reason for downloading an app is recommendation by others, as Google claims.
    - We have shown that the user's probability of adopting an app depends on properties of the local network structure, as Facebook claims.

- We have shown and validate that network effects are present in the spread of malware on smartphones.
- Our model predicts a stable situation of harmful apps installed, as Google says.
- The malware infection on smartphone are due to the users because they give indiscriminate permissions to the apps and do not protect properly their the mobile with antivirus software. Therefore, we have shown that two thirds of the infections are caused by these two factors, showing that the most critical part for the malware infections at smartphones are the users habits and the ineffectiveness of the traditional antivirus software rather than the Operative System vulnerabilities.
- The main type of malware infection is related with *Financial Charge* malware, i.e., malware that implies an economical cost to the user.
- We consider that preventive strategies against mobile malware should be developed mainly focusing on new malware detection approaches before being downloaded by the users, because, as we have shown, the users decisions and the ineffectiveness of the traditional antivirus software approach are the critical part for the infections.

2. Under the mathematical point of view:

   - We have developed a network model that can estimate the number of downloads of an app over time and the retention time of the application without being uninstalled in such a way that let us estimate the evolution of apps over time.
   - We have developed a network model that can estimate the spread of malware on smartphones over time.
   - We have shown that the prediction of the evolution of the spread of an app over the time and the spread of malware on smartphones is possible via computational methods whenever proper parameters are adequately chosen. The capability of the model to capture the behavior of the app and the malware by means of confidence intervals has been shown.
   - For both models, we have used very large network models running on large computational facilities that have allowed us to execute many simulations with multiple parameter sets in order to compute reliable estimations. In the literature there are no works of

this type, carried out with big network simulations and comparing the results with data from real apps.

Additionally, with this work we have extended the traditional field of work related with networks and mobile phones datasets from a social point of view, where phone record data collected by cell phone providers are used considering the individual belonging to his/her "analog" social network. Also we have extended the field considering the mobile applications as a new agent online whose behavior, or the user behavior related with the apps installed on his/her device, can be studied and analyzed using mathematical networks models.

Considering the emerging technology of the internet of things (IoT), which is a network of physical objects embedded with electronics, software, sensors and network connectivity which enables these objects to collect and exchange data and considering that mobile applications will allow us to monitorize all of our activities, the possibilities of study social aspects through network models applied to mobile applications or connected objects, mixing online and offline features of the connected device or agent and the user behavior, as we have presented in this work, will be a very interesting field in the near future.

Bibliography

[1] N. Aharony, W. Pan, C. Ip, A. Pentland, *Tracing Mobile Phone App Installations in the Friends and Family Study*, Proceedings of the 2010 Workshop on Information in Networks (WIN'10) (2010). http://web.media.mit.edu/~panwei/pub/funf_highlevel_n_apps_win_final.pdf, accessed March 10, 2015.

[2] T.T. Ahonen, A. Moore, *Smartphone Penetration Rates by Country! We Have Good Data (finally)* Consulting Analysis December 2011, based on raw data from Google/Ipsos, the Netsize Guide/Informa and TomiAhonen Almanac 2011 reported data (2011). http://communities-dominate.blogs.com/brands/2011/12/smartphone-penetration-rates-by-country-we-have-good-data-finally.html, accessed July 20, 2014.

[3] J. Alegre, *Is it possible to predict the mobile apps' lifecycle?* App Trade Centre, III Congreso Nacional del Sector de las Apps, Museo Príncipe Felipe de la Ciudad de las Artes y las Ciencias de Valencia, Spain, 23-24 September (2015)

[4] J. Alegre, J. Camacho, J.C. Cortés, F.J. Santonja, R.J. Villanueva, *Agent-Based Model to Study and Quantify the Evolution Dynamics of Android Malware Infection*, Abstract and Applied Analysis, Volume 2014, Article ID 623436, Hindawi Publishing Corporation (2014). http://dx.doi.org/10.1155/2014/623436.

[5] J. Alegre, J.C. Cortés, F.J. Santonja, R.J. Villanueva, *Analysis of mobile Apps lifecycle. Epidemiological modelling approach by random networks* Abstracts of Modelling for Engineering & Human Behavior 2014, p.151-160, Instituto Universitario de Matemática Multidisciplinar, Universitat Politècnica de València, Spain (2014)

[6] J. Alegre, J.C. Cortés, F.J. Santonja, R.J. Villanueva, *Predicting mobile apps spread: An epidemiological random network modeling approach*,

submitted to Simulation: Transactions of the Society for Modeling & Simulation International (2015)

[7] J. Alegre, J.C. Cortés, F.J. Santonja, R.J. Villanueva, *Quantifying the behaviour of the actors in the spread of Android malware infection*, Mathematical Modeling in Social Sciences and Engineering, Ch.10, p. 101–112, Nova Science Publishers Inc., New York (2013).

[8] J. Alegre, J.C. Cortés, F.J. Santonja, R.J. Villanueva, *The evolution of Android malware infection: An agent-based modelling approach* Abstracts of Modelling for Engineering and Human Behavior 2013, p.151-155, Instituto Universitario de Matemática Multidisciplinar, Universitat Politècnica de València, Spain (2013)

[9] J. Alegre, W. Lu, A. Takasu, *Wavelet Transform Based Vehicle Detection from Sensors for Bridge Weigh-in-Motion*, SAC 2016 31st ACM Symposium on Applied Computing Pisa, Italy, 4-8 April, 2016, In press.

[10] B. Bollobás, *Random Graphs*, 2nd ed. Cambridge University Press, 2001.

[11] A. Bose, K.G. Shin *Agent-based modeling of malware dynamics in heterogeneous environments*, Security Communications Networks 6(12), p. 1576–1589 (2013), doi:10.1002/sec.298.

[12] J.C. Cortés, F. Sánchez, F.J. Santonja, R.J. Villanueva, *A probabilistic analysis to quantify the effect of March 11th 2004 attacks in Madrid on the March 14th elections in Spain. A dynamic modelling approach*, Abstracts and Applied Analysis, ID–387839 (2015), doi:10.1155/2015/387839

[13] F. Di Cerbo, A. Girardello, F. Michahelles, S. Voronkova, *Computational Forensics: Detection of malicious applications on Android OS.*, Lecture Notes in Computer Science (LNCS) 6540 IWCF 2010, p. 138–149, Springer-Verlag, Berlin (2011), doi:10.1007/978-3-642-19376-7_12.

[14] C. Fleizach, M. Liljenstam, P. Johansson, G.M. Voelker, A. Méhesz, *Can You Infect Me Now? Malware Propagation in Mobile Phone Networks.*, WORM'07, Alexandria VA, USA (2007), doi:10.1145/1314389.1314402.

[15] A. Hoare, D.G. Regan, D.P. Wilson, *Sampling and sensitivity analyses tools (SaSAT) for computational modelling*, Theoretical Biology and Medical Modelling 5, article 4 (2008), doi:10.1186/1742-4682-5-4.

[16] T. Isohara, K. Takemori, A. Kubota , KDDI R&D Labs Saitama, *Kernel-based Behavior Analysis for Android Malware Detection*, 7th International Conference on Computational Intelligence and Security, Japan (2011), doi:10.1109/CIS.2011.226.

[17] I. Kloumann, L. Adamic, J. Kleinberg, S. Wu, *The Lifecycles of Apps in a Social Ecosystem*, WWW 2015 May 18–22, Florence, Italy (2015), doi: http://dx.doi.org/10.1145/2736277.2741684

[18] P. Luarn, J.C. Yang, Y.P. Chiu, *The network effect on information dissemination on social network sites*, Computers in Human Behavior 37, p. 1–8 (2014), doi: 10.1016/j.chb.2014.04.019

[19] J.W. Mickens, B.D. Noble : *Modeling Epidemic Spreading in Mobile Environments*, WiSE 2005, Cologne, Germany (2005), doi:10.1145/1080793.1080806.

[20] A. Mylonas, A. Kastania, D. Gritzalis, *Delegate the smartphone user? Security awareness in smartphone platforms*, Computers & Security 34, p. 47–66 (2013), doi:10.1016/j.cose.2012.11.004.

[21] E. Navarro-Pertusa, A. Reig-Ferrer, E. Barber Heredia, R.I. Ferrer Cascales, *Grupo de iguales e iniciación sexual adolescente: Diferencias de género (Groups of pairs and adolescent sexual initiation)*, International Journal of Clinical and Health Psychology 6, no. 1, p. 79–96 (2006).

[22] A. Olsson, G. Sandberg, O. Dahlblom, *On Latin hypercube sampling for structural reliability analysis*, Structural Safety 25, p. 47-68 (2003), doi:10.1016/S0167-4730(02)00039-5.

[23] W. Pan, N. Aharony, A. Pentland, *Composite Social Network for Predicting Mobile Apps Installation*, Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, p. 821–827 (2011). http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3729, accessed March 10, 2015.

[24] S.M. Patterson, *Contrary to what you've heard, Android is almost impenetrable to malware* (2013), http://qz.com/131436/contrary-to-what-youve-heard-android-is-almost-impenetrable-to-malware/, accessed July 20, 2014.

[25] K. Ramachandran, B. Sikdar, R.P.I. Troy, *Modeling Malware Propagation in Networks of Smart Cell Phones with Spatial Dynamics* INFOCOM 2007, Anchorage AK, USA (2007), Doi:10.1109/INFCOM.2007.312.

[26] F.J. Santonja, A. Morales, R.J. Villanueva, J.C. Cortés, *Analysing the effect of public health campaigns on reducing excess weight: a modelling approach for the Spanish Autonomous Region of the Community of Valencia*, Evaluation and Program Planning 35, no. 1, p. 34–39 (2012), doi:10.1016/j.evalprogplan.2011.06.004.

[27] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer, Y. Weiss, *Andromaly: A Behavioral Malware Detection Framework for Android Devices*, J. Intelligence Information Systems, Springer Science-Business Media, New York (2011), doi:10.1007/s10844-010-0148-x.

[28] D.G. Taylor, T.A. Voelker, I. Pentina, *Mobile application adoption by young adults: A social network perspective*, International Journal of Mobile Marketing, 6, no. 2, p. 60–70 (2011).

[29] P. Wang, M.C. González, C.A. Hidalgo, A.L. Barabási, *Understanding the Spreading Patterns of Mobile Phone Viruses* Science 324(5930), p. 1071–1076 (2009), doi:10.1126/science.1167053.

[30] J. Yang, J. Leskovec, *Modeling Information Diffusion in Implicit Networks*, http://cs.stanford.edu/people/jure/pubs/lim-icdm10.pdf, (2010), accessed March 10, 2015.

[31] Y. Zhou, X. Jiang, *Dissecting Android Malware: Characterization and Evolution*, Proceedings of the 33rd IEEE Symposium on Security and Privacy, Oakland, San Francisco, CA. (2012), doi:10.1109/SP.2012.16.

[32] Book of abstracts. Conference on the scientific analysis of mobile phone datasets 7-10 April 2015, MIT Media Lab.

[33] Book of abstracts. Conference on the scientific analysis of mobile phone datasets. Data for Development Challenge Senegal 7-10 April 2015, MIT Media Lab.

[34] App Engagement: The Matrix Reloaded. Flurry insights. http://www.flurry.com/bid/90743/App-Engagement-The-MatrixReloaded#.VH2tJNKG9e8, accessed March 10, 2015.

[35] Benchmarking the Half-Life and Decay of Mobile Apps. Flurry insights. http://www.flurry.com/bid/108904/Benchmarking-the-Half-Life-and-Decay-of-Mobile-Apps#.VH3dwtKG9e8, accessed March 10, 2015.

[36] Department of Homeland Security, Federal Bureau of Investigation. (U//FOUO) DHS-FBI Bulletin: Threats to Mobile Devices Using the Android Operating System, August 2013. http://publicintelligence.net/dhs-fbi-android-threats/, accessed July 20, 2014.

[37] Gartner Smart Phone Marketshare 2013 Q2, http://www.gartner.com/newsroom/id/2573415 accessed July 20, 2014.

[38] The Rise of the Mobile Addict. Flurry insights. http://www.flurry.com/blog/flurry-insights/rise-mobile-addict#.VH20UNKG9e8, accessed March 10, 2015.

[39] Spanish National Institute of Statistics. http://www.ine.es, accessed July 20, 2014.

[40] Webinar: X-Ray Results- Mobile Device Vulnerabilities, Duo Security, www.duosecurity.com, October 2012. Accessed July 20, 2014.

[41] http://app.imm.upv.es, accessed October 21, 2015.

[42] http://www.appbrain.com/stats/number-of-android-apps, accessed July 20, 2014.

[43] http://www.appbrain.com/stats/android-app-ratings, accessed July 20, 2014.

[44] http://en.wikipedia.org/wiki/Android_(operating_system), accessed July 20, 2014.

[45] http://en.wikipedia.org/wiki/List_of_mobile_software_distribution_platforms, accessed March 10, 2015.

[46] http://en.wikipedia.org/wiki/Valencian_Community, accessed July 20, 2014.

[47] https://www.gartner.com/newsroom/id/2944819, accessed March 10, 2015.

[48] http://www.getjar.com, accessed July 20, 2014.

[49] http://googlemobile.blogspot.com.es/2012/02/android-and-security.html, accessed July 20, 2014.

[50] http://www.idc.com/prodserv/smartphone-market-share.jsp, accessed March 10, 2015.

[51] http://www.malgenomeproject.org, accessed July 20, 2014.

[52] http://www.netmob.org/, accessed October 21, 2015.

[53] https://play.google.com/store, accessed July 20, 2014.

[54] http://www.samoamodel.com, accessed October 21, 2015.

[55] http://www.samoamodel.com/demo/samoa_demo.html, accessed October 21, 2015.

[56] http://www.slideme.org, accessed March 10, 2015.

[57] https://think.storage.googleapis.com/docs/mobile-app-marketing-insights.pdf, accessed October 21, 2015.

[58] http://xyologic.com, accessed July 20, 2014.