12-2016

Advance Android PHAs/Malware Detection Techniques by Utilizing Signature Data, Behavioral Patterns and Machine Learning

Suyash Jadhav
ssj8127@rit.edu

R·I·T

Rochester Institute of Technology

Advance Android PHAs/Malware Detection Techniques by Utilizing Signature Data, Behavioral Patterns and Machine Learning

By
Suyash Jadhav

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computing Security

Department of Computing Security
B. Thomas Golisano College of Computing and Information Sciences
Rochester Institute of Technology Rochester
NY
December 2016

Advance Android PHAs/Malware Detection Techniques by Utilizing Signature Data, Behavioral Patterns and Machine Learning

By

Suyash Jadhav

December 2016

Department of Computing Security

Rochester Institute of Technology

Rochester, NY

Approved By:

_____**Date:**_____
Dr. Tae Oh
Primary Advisor – R.I.T. Dept. of Computing Security

_____**Date:**_____
Professor Bill Stackpole
CoChair – R.I.T. Dept. of Computing Security

_____**Date:**_____
Professor Chaim Sanders
Committee Member– R.I.T. Saunders College of Business

Acknowledgement

I wish to express my sincere gratitude towards my thesis committee. Their support and valuable inputs made this research work possible.

My sincere thanks to Dr. Tae Oh for encouraging to pursuing this research and providing with continuous feedback. He has been a source of motivation and encouragement throughout this research. His input through discussions and advices on designing research has been very helpful. His support motivated for putting rigorous efforts at each level in research work.

I would like to thank Professor Bill Stackpole for providing valuable and critical suggestions through out the research. His interest and creative thinking helped me a lot in designing this research. His continuous motivation helped doing out of box thinking, which made this research succeed truly.

I would like to extend my sincere thanks to Professor Chaim Sanders; his expertise in field of network security has helped me a lot in understanding the Android's network security aspects in depth. His expertise in the field has been a great source of knowledge for this research.

Thanks to Computing Security Department and Advisors, this thesis won't have been possible without the sincere efforts taken from you side.

Last but not the least most importantly, immense thanks to my parents for availing me an opportunity to study at RIT. Your faith and trust in me made all these accomplishments possible.

Abstract

During the last decade mobile phones and tablets evolved into smart devices with enormous computing power and storage capacity packed in a pocket size. People around the globe have quickly moved from laptops to smartphones for their daily computational needs. From web browsing, social networking, photography to critical bank payments and intellectual property every thing has got into smartphones; and undoubtedly Android has dominated the smartphone market. Android growth also attracted cyber criminals to focus on creating attacks and malwares to target Android users. Malwares in different category are seen in the Android ecosystem, including botnets, Ransomware, click Trojan, SMS frauds, banking Trojans.

Due to huge amount of application being developed and distributed every day, Android needs malware analysis techniques that are different than any other operating system. This research focuses on defining a process of finding Android malware in a given large number of new applications. Research utilizes machine learning techniques in predicting possible malware and further provide assistance in reverse engineering of malware. Under this thesis an assistive Android malware analysis system "AndroSandX" is proposed, researched and developed. AndroSandX allows researcher to quickly analyze potential Android malware and help perform manual analysis.

Key features of the system are strong assistive capabilities using machine learning, built in ticketing system, highly modular design, storage with non-relational databases, backup of analysis data for archival, assistance in manual analysis and threat intelligence. Research results shows that the system has a prediction accuracy of around 92%. Research has wide scope and lean towards providing industry oriented Android malware analysis assistive system/product.

Table of Content

Table of Contents

List of Figures

List of Tables

1 Introduction

Android application development and Android mobile market has expanded tremendously in last few year. One of the core success reasons behind Android OS is its easy application development and distribution. Google Play Store and many other Android application markets across the globe allow developer to upload and distribute their applications almost in runtime. Publishing and updates to application are very easy and goes through very little manual review. While the ease of distribution helped developers on one side, the adversity of system was it made distribution of malicious applications considerably easy. Due to tremendous volume and continuous increase in Android application uploads; it is highly difficult and costly to do manual review of applications. These Android market conditions resulted in fairly easy creation and distribution of Android malware.

Due to limitations on manual review of applications, Android market needed fast and efficient automated malware analysis system. Leading application stores like Google Play Store have developed their own automated malware analysis solutions and took down the existing malwares, but small size application stores still mostly depend on traditional hash based scanning techniques. This thesis study was carried out realizing a great need and scope for research in automated Android malware analysis.

Android malware analysis system "AndroSandX" is developed. AndroSandX is an assistive tool to reduce the set of possible malware in the given random set of Android application and provide ticket based malware analysis system for tracking and providing systematic approach for creation of IDS signature through manual analysis, storage of such signature and testing them.

Under this research 15000 Android applications were primarily used, which further narrowed down to 3083 malware and 1775 benign applications for machine learning purpose. Total of 153 unique features are extracted for each application from static and dynamic analysis data. A mixed application test set of 509 applications was used for testing prediction capability of AndroSandX. The system was successful with a predictions rate of around 92%, with a strong assistive capability in further performing reverse engineering, debugging and signature creation.

Research focuses way beyond just proving feasibility of machine learning based Android malware prediction. Research strongly aims towards giving a prototype of a system that defines standard process for Android malware analysis and provide assistance at every step in analysis. AndroSandX is equipped with Ticketing system, machine leaning, non-relational database, intrusion detection system integration, reverse engineering, debugging and signature deployment capability. Extensive threat intelligence is used.

Research implements experimentation based analysis approach. Experimentation required understanding in following domains: Android operating system, malware analysis, Python programming, MongoDB, Snort IDS/IPS implementations, Network security analysis, Machine learning, sandboxing implementations, network setup, server deployments and threat intelligence.

This research resulted into proving feasibility and implementation of machine learning based Android malware analysis approach. Research also helped in providing standard Android malware analysis approach, assistive tool and IDS signature creation assistance.

2 Related work:

2.1 Android Malware statistics, malware types and categorization

Almost 1.4 Billion mobile devices were sold in 2015, out of which 5/6 of the devices run Android OS [3]. The number of mobile devices running Android OS is on tremendous increase in recent years. Which is also the reason for increase in sophisticated attacks and exploit development on Android platform [2][3]. Cyber criminals are focusing and investing in development of considerably larger amount of malwares and exploits for Android platform [4]. Malware in Android are no more just SMS Trojans or Spyware, multiple categories of Android malware are on rise mainly including Ransomware, Hostile Downloaders, Botnet Trojan [1].

Google categorizes Android malware in to 17 different categories [1], as described in Table 1:

Table 1 : Google's Android Malware Categories

Backdoors	Commercial Spyware	Phishing Applications
Billing Fraud	Data Collection	Trojan
SMS Fraud	Denial of Service	Ransomware
Call Fraud	Hostile Downloaders	Rooting Apps
Spam	Spyware	Non-Android Threat
Wireless Access Protocol (WAP) Fraud	Privilege Escalation Apps	

2.2 Malware analysis Techniques

Android malware analysis involves multiple stages. Practical world detection techniques are limited to network-based detection and host-based signature detection. Network-based detection includes network data transfer level byte pattern match, network protocol, IP/Domain blacklisting and URL pattern match. Host-based detection most of the time involves hash and byte code pattern based detection. Android application development and application store have large diversity, which leads to rapid development of applications. Large number of applications gets uploaded every day across the globe, targeting multiple variants of Android platform. It's fairly difficult to perform manual analysis on all of the applications. Reusability

of code and easy distribution of application allows malware authors to easily avoid signature-based detection.

There are different approaches for malware detection suggested by Android security researchers around the world. Android malware can be detected through code level signatures and Android instruction code. Detection through certain pattern matching of Application package name and methods/functions is possible [11]. Some other interesting approaches include visual representation of binaries [12], hybrid analysis using an attack tree based approach [14]. Researchers have developed assistive malware prediction tools to allow malware researchers find possible unknown malware in given set of applications[19][20][24][25].

2.2.1 Android Malware Analysis Methods:

Analysis methods can be categorized based of the data collection by performing execution or without execution of application. Three main categories are defined as follows - static, dynamic and hybrid analysis.

2.2.1.1 Static Analysis

Static analysis mainly focuses on the data that can be collected from an Android application executable .apk file. Focus of the static analysis is bound around the code and possible strings and sub routines in the application code. Host intrusion detection system usually utilizes this approach. Static analysis most of the time focuses on creation of hash signatures and byte code pattern signature for detection of malicious applications. De-assembling and manual code reviews are core tasks in the static analysis. Hardcoded strings usually give idea about the possible malware domain and IP addresses. Also sometimes static strings include symmetric key used in encryption of data. In case of botnet malware static signatures reveal details about the command and control communication mechanism, capabilities and possible instruction set that a bot can perform. Even though a lot of information is revealed in the static analysis it is not the complete data that one might need to reveal the capabilities and features of any malware. Dynamic analysis on the other hand provides more of an execution time behavior of the application. Many time

malware author hides functionality by using runtime code download, which is not detected in the static analysis [11][12][15].

2.2.1.2 Dynamic Analysis

Dynamic analysis focuses on the runtime behavior of the application. Analysis involves execution and monitoring of the activity performed by the applications. Activity monitoring and data collection is carried out at system level, network level and at memory level. System level data includes system call and system functions invoked by the applications. Network level data included DNS request, payload downloaded, domain and IP addresses contacted by application. Network data further reveals details about protocol and http request made by the application. Memory level data usually is obtained during the debugging of application, which involves code level breakpoints and monitoring or registers, stack and heap data.

Dynamic analysis is sometimes way more powerful than the static analysis as it reveals actual behavior of malware and any intentionally hidden functionality[13][15]. Further analysis of dynamically downloaded payload or code modules is possible, most of the malwares now days download configuration files in runtime. For analyzing botnet and bot-network behavior dynamic analysis is very helpful. Network based intrusion detection/prevention system (NIDS/NIPS) signature required data is most of the time collected through dynamic analysis as post infection activity is very important for creation of network-based IDS/IPS.

2.2.1.3 Hybrid Analysis

The best way to gather all analysis relevant data is to follow hybrid analysis approach, meaning utilizing static and dynamic analysis together. This approach might be resource intensive and time consuming but gives most significant and accurate results due to completeness of data. Static analysis is usually time efficient and dynamic analysis is more depth oriented. Utilizing hybrid analysis allows creation of robust signatures and extraction of dynamic detection features. Research will be utilizing hybrid analysis. It allows

gathering maximum number of features allowing more accurate detection, also helping in-depth analysis and signature creation[13][14].

2.3 Detection Techniques:

Malware detection techniques are categorized in to two approaches, signature based and behavior based. Signature based approach is being very accurate in detecting known malwares on the other hand behavior based approach allows detection of potential unknown malware.

2.3.1 Signature Based Approach

Signature based detection works at two layers, first is network level and second is host level. Network level signatures are usually dependent on network traffic features and pattern in network packets to detect possible infection. Host level signatures focus more on hash and byte code pattern in malicious applications. Signature based detection is very accurate and dependable for detecting already analyzed malware or malware family [12][15].

2.3.2 Behavior Based Approach

Behavior based approach allows detection of possible unknown malware by utilizing knowledge of normal behavior on the network or any host. Behavior based analysis looks for possible indicator of compromise, instead of looking for any one specific feature. Different execution time behavioral pattern and series of activity from an application caused behavioral anomaly or possible indicator of compromise pattern match. This approach allows detecting suspicious behavior of application further suggesting manual analysis [13][15][24][25].

2.4 Malware detection evasion techniques

Symantec 2016 Internet Security Threat Report notifies a critical observation in their report stating that the Android malwares are using code obfuscation to avoid signature based detection, and additionally use multiple sandbox detection techniques [3]. These notes from leading vendors like Symantec raises an urge of focusing on

more than simple signature based malware detection system. Also robust sandbox level anti-evasion techniques need to be used while analyzing malware. Cuckoo Droid [5] focuses on providing an excellent Android sandboxing solution that considers the possible anti detection measures. Different techniques are implemented including falsifying the system properties, contacts and network properties on Android virtual device to overcome malware evasion techniques [5]. Use of anti evasion Android framework like Xposed [6] makes Cuckoo Droid a robust Android sandbox. Though there are techniques, which allow Android malware to detect emulator or virtual machine instance of Android based upon performance parameters, use of virtual machine for running Android is very much a strong indication of sandboxing environment [8].

2.5 Malware sandboxing and data collection

Different Android sandboxing and malware analysis solutions have been created and tested for their efficiency. Automation and extraction of data from different phases of Android application execution are the core requirements of sandbox [9][10]. Analysis data can be extracted from static code level data, dynamic execution time data, network data and Android OS memory related data [26]. Cuckoo Droid is one of the most excellent Android malware sandbox available, it provides data for static, dynamic and network level analysis, further it also provides Android system level function calls and application runtime data. Use of Cuckoo-Droid and its Python implementation allows easy integration and extension of functionality.

2.6 Android Malware analysis using classification or regression techniques

Google uses Bouncer a behavior based analysis system to detect malware in the Google Play Store [27]. Bouncer utilizes artificial intelligence to perform behavior-based analysis to predict potential known malware. Researchers utilize features that are extracted from static analysis or hybrid analysis data [16][17][18][21]. Machine learning algorithms have given very good results and are strong indicatives that more research needs to be preceded utilizing this approach. Machine learning algorithms used for detecting Android mainly includes support vector machine (svm), random forest, decision tree and neural networks [24][17][21][22][23]. Choice of machine

learning algorithm also depends on the data used for the feature extraction. Feature extraction input data can be strings, floating values, continuously varying values, binary values or Boolean. Classification algorithm efficiency also varies with type of analysis being used for extracting data. Static analysis, dynamic analysis or hybrid analysis needs different algorithms as efficiency of classification changes significantly.

2.7 Android malware analysis assistive systems

Android malware analysis due to its huge amount of applications needs a strong assistive malware analysis system. Such assistive system should allow researcher to automatically perform a classification and present with potential malwares for manual analysis. SherlockDroid is one of such system [19]. Assistive tools are also necessary to crawl larger amount of applications in any online Android application store and successfully predict benign and possible malicious applications [20]. Assistive analysis systems utilize machine learning algorithms, perform classification, data extraction, sandbox integration and provide manual analysis assistance.

3 Methodology

3.1 Problem Statement

Given a set of Android applications, researcher can perform automated analysis and collect data, allowing machine learning based classification. Classification results in malware/benign category. Further allowing creation of IDS signatures to detect flagged application. Achieve high efficiency of the malware analysis process by reducing manual intervention also making sure system is modular and resource rich.

3.2 Hypothesis

It is very much possible to extract data features of Android application that allows successful machine learning based classification of application in to malware/benign category. Perfectly sandboxed automated application testing environment provides static and dynamic analysis data that can be used for feature extraction. If a system is made with a combination of good sandboxing solution, quality threat intelligence data and machine learning algorithms then it is possible to find subset of Android application that can be classified as potential malware. Once a malware subset is found it can be further reverse engineered to create intrusion detection system signatures. This approach would help in detecting Android malware without needing extensive manual review, and manual analysis efforts will be focused on creating signatures for already predicted malware.

3.3 Experiment approach

The hypothesis of this research is proved using experimentation based proof. The research goes through development of a system, which allows sandboxing, automated malware analysis, feature extraction and machine learning. Results are collected and efficiency matrix is presented to prove the success in classification of Android applications as malware or benign category. Further the predicted malware are manually reverse engineered to create detection signatures and expected final state of experimentation is achieved.

3.4 Hypothesis proving strategy

Experimentation starts with application selection for machine learning phase. Collected set of malware and benign applications are sent through the AndroSandX

system to preform automated dynamic and static analysis. Once the data is collected machine-learning algorithm is used to build models, these machine-learning models are further used for predicting unknown applications. Tested unknown applications are checked against threat intelligence data to confirm the true positives and false positives. Accuracy rate is calculated and further manual analysis of predicted malware application is performed. During manual analysis intrusion detection signatures are created and tested against running instance of malicious application to prove successful detection using intrusion detection system. Machine learning based malware prediction accuracy and IDS alert generation using newly created signature successfully proves the proposed hypothesis.

3.5 Overview of the experimentation system

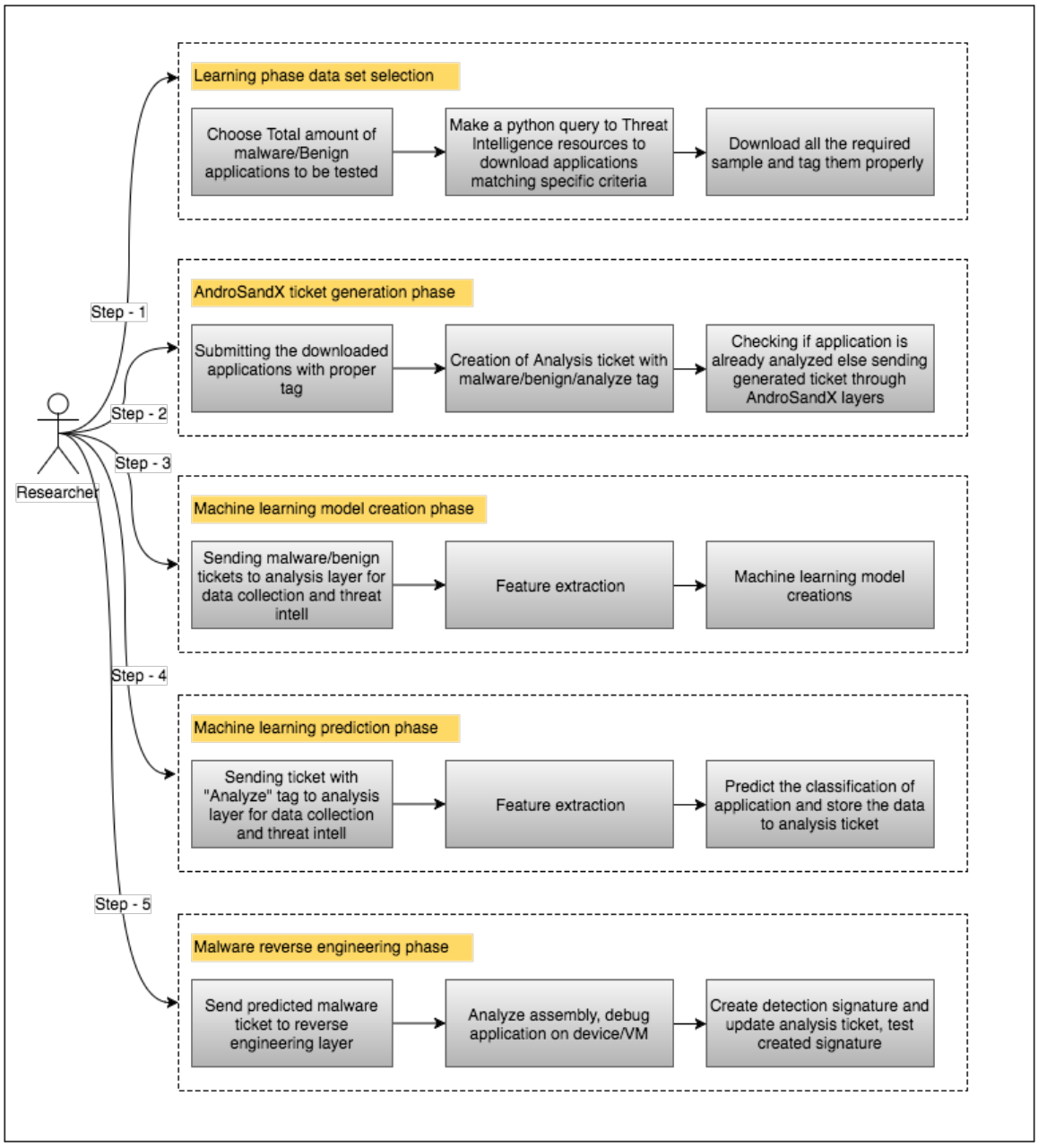

Figure 1 : AndroSandX Experimentation Process Overview

The experimentation process is divided in to five steps; these steps are logical abstraction of experimentation process.

- Learning phase data set selection,
- AndroSandX ticket generation phase,

- Machine learning model creation / Training phase,
- Machine learning prediction phase,
- Malware reverse engineering phase

3.5.1 Learning/Training phase data selection

For creation of a machine learning models to accurately perform prediction it is very important to choose the learning data set properly. The learning data set should be from proper trusted source and should be data rich, which allows accurate feature extraction. Under this research "VirusTotal.com"[40] is chosen as the source of collection of known categorized applications. VirusTotal python accessible APIs are used to download set of Android applications that are already detected by 10+ antivirus solutions as malware data set. For benign applications data set, VirusTotal Android applications with zero known detection from antivirus vendors are chosen. All the VirusTotal downloaded applications are not necessarily the right .apk or .zip formatted Android applications, and this is because of the files being uploaded all across the globe and not always are successfully installed on all Android devices. This non 100% clean data also helps to mimic real world data that any Android application store will come across. Android applications with compatibility issue, installation issue or running errors in sandbox are discarded in later phases to get a cleaner dataset. Once a malware/benign classified data set is obtained, then the data is properly tagged and forwarded to next layers.

3.5.2 AndroSandX ticket generation phase

Every application that needs to be tested requires a creation of AndroSandX analysis ticket. The ticketing system allows task separation and gives independent data modification at each layer in analysis. Every layer can check till what layer the ticket is currently processed. Layer with dependency on other layer can then check if they can pick up the ticket for their analysis phases if current phase of ticket is not amongst the acceptable phases of analysis then the layer can throw error indicating it cannot work on this ticket. The ticket creation phase requires proper tagging of applications. This phase also performs check to see if application is already tested.

3.5.3 Machine learning model creation/Training phase

Analysis tickets get tagged as malware/benign and are further passed to layer that perform static and dynamic analysis. During this phase data from static files, network traffic, runtime system calls, static functions and operating system level operations is collected. Further required threat intelligence is gathered. Once all the required data for generation of machine learning features is collected then 153 unique features are collected and stored in to the analysis ticket. When all the learning data set applications complete feature generation phase, feature data with known classification of application is passed through machine learning algorithms to create machine-learning models. These models are then tested to make sure they are not over fitted.

3.5.4 Machine learning prediction phase

This phase is the most interesting and crucial part in the experimentation. A certain set of mixed applications is chosen from VirusTotal. Actual VirusTotal given categorization of these applications is stored for calculating accuracy of the system. These applications are tagged as "Analyze" and sent for the static and dynamic analysis process. Once a feature generation phase is completed these applications are passed to machine learning based prediction. This process utilizes previously calculated machine learning models to classify each application into malware or benign category. Once all the required applications classes are predicted then they are compared with VirusTotal given signature based classification. This comparison gives out the efficiency matrix. Also the accuracy of prediction calculated by measuring precision and recall. Predicted values are stored in the analysis ticket. Prediction matrix is printed for user information, success of the AndroSandX and proof of research hypothesis depends on high prediction accuracy of this phase.

3.5.5 Malware reverse engineering phase

In a real world corporate network, most efficient way to detect malicious activity is to monitor network though one single point in network. All network traffic should pass through this point, so as to detect possible infection, malicious communications and web attacks. Other powerful way is to have host based scanners and intrusion detection system. In case of Android host intrusion detection is a little more difficult as devices are

not owned by enterprise and people don't like to have any monitoring application on their mobile as it might lead to privacy breach. One of the prime goal of AndroSandX system is once an Android application is classified as malware, it should be reverse engineered manually and intrusion detection signatures should be created. A good NIDS/HIDS signature helps in discovering active infection on the enterprise network, which is of the prime importance to any security researcher. Malware reverse engineering phase helps in decompiling, debugging the application to walk through assembly code and understand the capability of applications. Also extracting unique features and behavioral pattern of malware.

Manual analysis leads to gathering of lot of new data and signatures, this data is stored back into AndroSandX analysis ticket. And SNORT signatures are deployed and tested against running instance of Application to test their detection efficiency.

4 Experimentation Environment

The experimentation environment is designed in a way that the data and processing is always independent of each other; most of the data is at rest and not continuously in server memory. System is designed with thinking that it could be expanded to create a real world product. System architecture, data storage strategy and network architecture is built with segmented design and functionality. Ticketing system helps in separating different tasks in different layers. This section will explain the process details, hardware and physical network details of the developed experimentation system.

At the process level experimentation is separated in to machine learning training, prediction and reverse engineering process.

4.1 Machine learning training phase

Machine learning process is divided into 10 sub processes, given in the Table 2, Figure 2 shows a functional control flow of the machine learning process, these sub processes are tabularized with their computational complexity, network usage and data read/write cycles.

Table 2 : Machine Learning Training Process Overview

Sub-processes Name	Computational complexity	Network usage	Database, file read/write operations	Time consumption
Selecting machine learning data set	Manual process	High	High	High time consuming
Tagged data set submission to AndroSandX	Low computation	None	Moderate	Low time consumption
Creating AndroSandX analysis ticket	Moderate computation	Low	High	Moderate time consumption

Dynamic and static analysis	High utilization of processing power, extensive multithreading	High	High	High time consumption
Threat intelligence	Moderate computation	Moderate	High	High time consumption
Data abstraction and cleaning	Low computation	None	Moderate	Low time consumption
Feature extraction	Moderate Computation	Low	High	High time consumption
Creating machine learning model	High computation	None	Low	Low
Storing machine learning model	Low computation	None	Moderate	Low time consumption

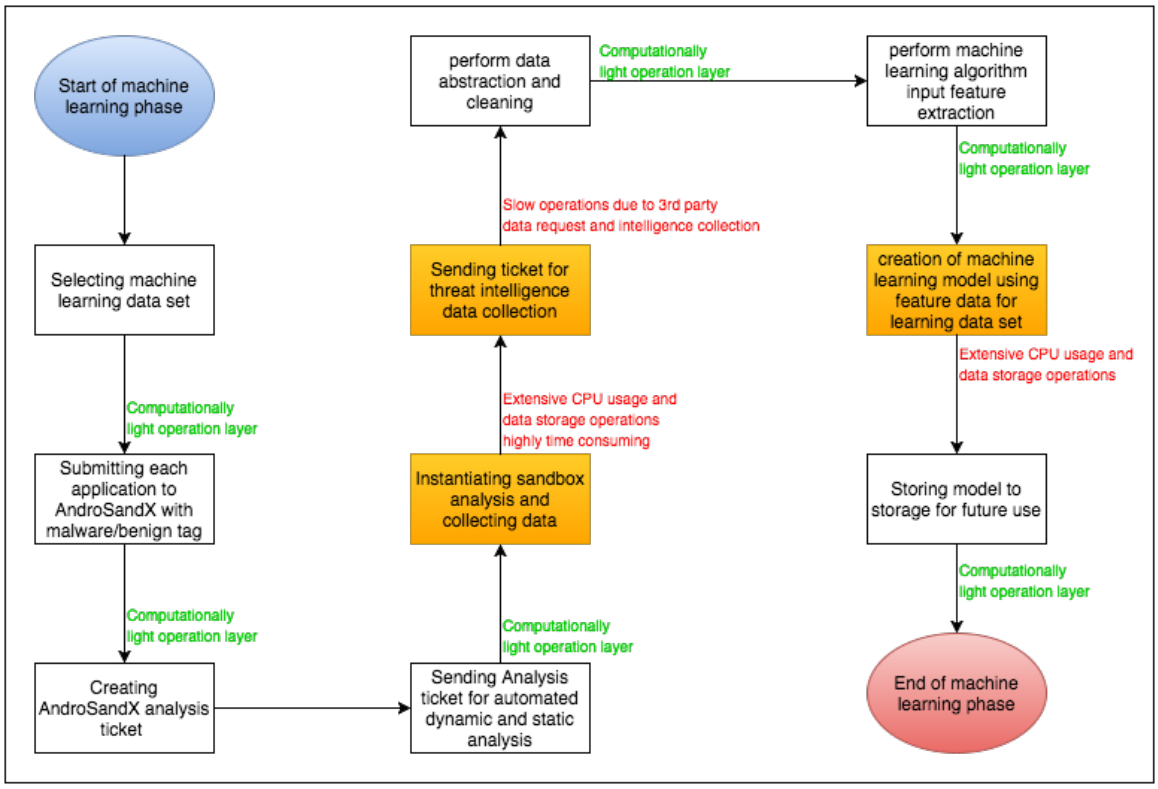

Figure 2 : Machine Learning Phase Process Graph

4.2 Machine learning prediction phase

Prediction phase is carried out after completion of machine learning training phase. Prediction uses already created model to perform classification on the given unclassified application. Following Table 3 and Figure 3 shows sub-processes in prediction phase.

Table 3 : Machine Learning Prediction Phase Overview

Sub-processes Name	Computational complexity	Network usage	Database, file read/write operations	Time consumption
Submitting application to AndroSandX	Low computation	None	Moderate	Low time consumption
Creating	Moderate	Low	High	Moderate time

AndroSandX analysis ticket	computation			consumption
Dynamic and static analysis	High utilization of processing power, extensive multithreading	High	High	High time consumption
Threat intelligence	Moderate computation	Moderate	High	High time consumption
Data abstraction and cleaning	Low computation	None	Moderate	Low time consumption
Feature extraction	Moderate Computation	Low	High	High time consumption
Machine learning prediction of classification	Moderate	None	Low	Moderate time consumption
Storing predicted result analysis ticket	Low	Low	Low	Low time consumption

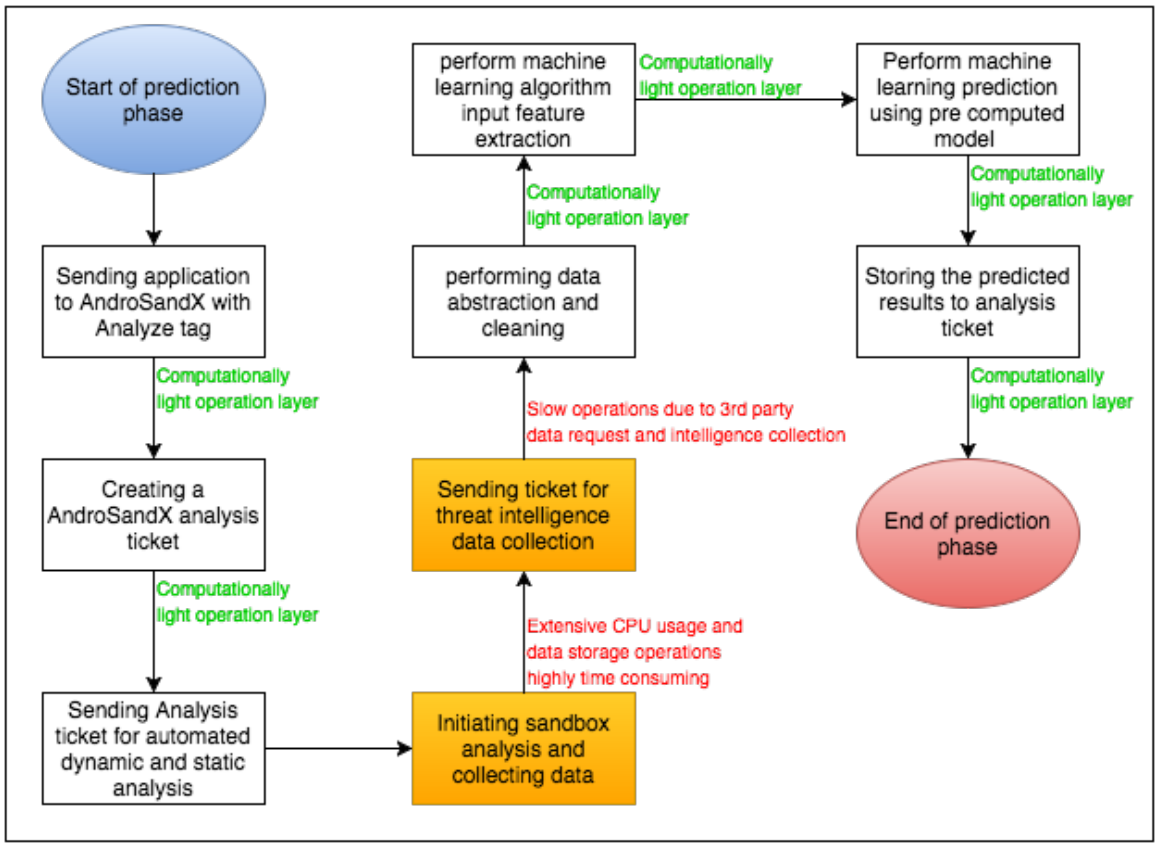

Figure 3 : Machine Learning Prediction Phase Process Graph

4.3 Malware reverse engineering/ manual analysis phase

Under this research a systematic stepwise Android malware analysis process is defined. Figure 4 gives flow of sub processes in manual analysis. Some part of this process is automated under AndroSandX. Few of the most important steps in Android malware analysis are finding important string, debugging application to find the code control flow, decompiling application of analyzing code, monitoring network to find malicious traffic, looking for background activity and analyzing any downloaded files. During manual analysis for debugging purpose malicious application is decompiled and recompiled in debug mode. This process required re-signing of Android apk file. Malware then can be installed on the Android virtual instance or real device. Main purpose of this process is to create signatures that will allow detection of malware using host and network intrusion detection systems.

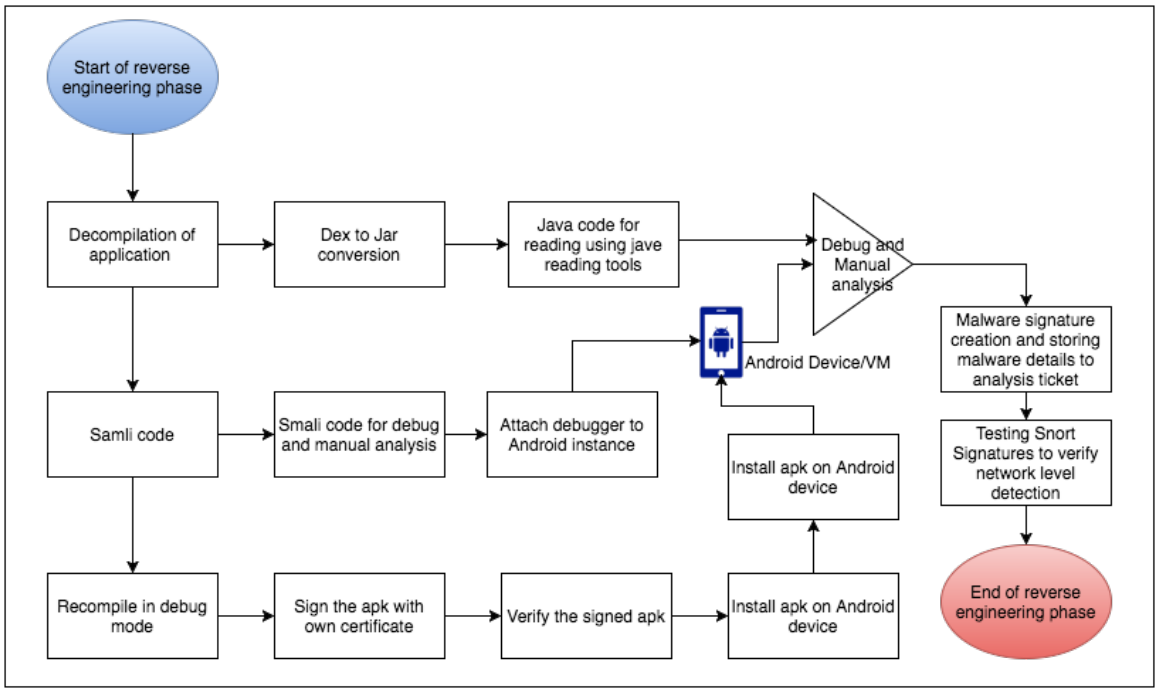

Figure 4 : Manual Malware Reverse Engineering Process Graph

4.4 Network Architecture

The AndroSandX is designed with an industry style product, which could be easily judged from its network architecture. AndroSandX network architecture is created with security consideration, easy extension, easy data sharing and functional independence capabilities. Main component of the network are as follows:

- Peripheral firewall
- WiFi router / Access point
- Gigabit switch
- IPtable based firewall
- AndroSandX main core processing server Dell R610
- Android virtual instances
- Android real devices
- Network Attached storage for data backup
- Hybrid drive mounted on drive mount
- Malware analysis laptops/desktops
- Cat5E cables

4.4.1 Functional explanation of each component in the network

4.4.1.1 Peripheral Firewall

Peripheral firewall protect the internal infrastructure from unauthorized access, also it allows only specific inbound open ports restricting exploitation of internal services like MongoDB. Firewall does not allow creation of new inbound connection to avoid Trojan backdoor infections. Peripheral rules for AndroSandX network are very restrictive and mostly will differ from a real world enterprise situation. In case of experimentation network peripheral firewall is running on TPLink TL-WR940N WiFi router.

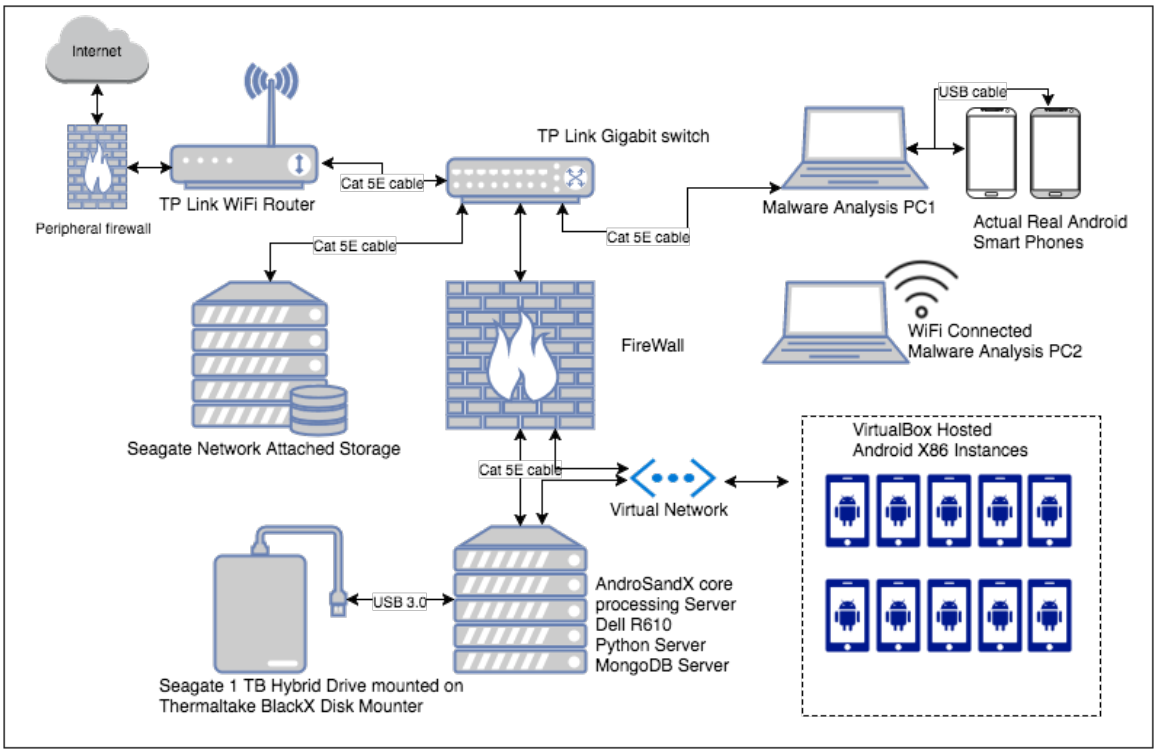

Figure 5 : Network Architecture Diagram

4.4.1.2 WiFi router / Access point

WiFi access point is used to proved wireless connectivity to experimentation environment, device connected to WiFi network can go out to internet as well as they have unrestricted access to Network Attached Storage and AndroSandX. This allows using actual Android device for testing malware and provides connectivity from laptops without necessarily requiring being on wired network.

4.4.1.3 Gigbit Switch

Switch provides network wired connection to all component in the network, allows traffic between server, router, WiFi access point, laptops and network attach storage. Helps in maintaining collision free layer 2 network data transfer, and keeps it fast.

4.4.1.4 IPtable firewall

Iptable firewall is running on the Dell R610 server. IPtables helps in allowing and denying certain traffic, it helps in restricting access to only certain services running on the server. IPtables also performs an essential function of providing same network path for infected virtual Android devices to go out to internet, and restrict inbound connections.

4.4.1.5 AndroSandX core processing server

AndroSandX is developed in Python, it uses MongoDB v3.2 as its database storage. Server runs on Ubuntu 14.04 server OS. In terms of hardware capability Dell R610 is having 24Gb RAM(6x4gb), 2x X5550 2.66 Ghz quad core processor capable of running 16 parallel threads, 4 Ethernet network interface cards [42].

4.4.1.6 Android virtual instances

AndroSandX server runs multiple Android virtual instances that are used for malware installation and monitoring. Android 4.4 r2 x86 image is used. VirtualBox is used as virtualization software. Configured base image contains necessarily applications to connect with Cuckoo sandbox, root applications and Xposed framework required applications to hide presence of virtualization. Every instance of Android is allocated with 4 GB disk space and 512 Mb RAM, network is configured to be isolated from server and IP forwarding is used to provide connectivity to isolated network.

4.4.1.7 Android real devices

AndroSandX also provide capability to connect and test applications on the real Android device. The devices can be directly attached to analysis PC over USB or they can be connected over WiFi. This provides more flexibility in manual analysis as any probable attempt from malware to detect virtualization can be detected.

4.4.1.8 Network attached storage for data backup

Due to experimentation size, large amount of data is generated this data is not always possible to be stored on the server hard drive due to disk size constraint. Also the expansion cost of enterprise grade disk is too high to be afforded for the experimentation environment. Solution to these problems was achieved by setting up a network attached storage, this allows easy access to analysis data from local network as well as Seagate NAS allows url based cloud sharing. Routine backup for the system data and of MongoDB databases is highly required to avoid data loss, this is achieved by having routine backups of all databases and files to Seagate NAS.

4.4.1.9 Hybrid drive mounted via USB3.0 HDD docking station

Core server is having an enterprise grade SATA hard drive of size 146 Gb, this drive is pretty much utilized by Ubuntu OS and other essential software installations. For storing MongoDB database and analysis files this core hard drive is not a good options as it limits easy backup and restore process. Having an externally mounted hybrid drives gives advantage of separating malware related data from core OS data, allows granular file/directory access control, easy backup/restore, easy hardware failure recovery, affordable high speed drive, larger disc space, easy mounting and OS changes. Considering all these advantages AndroSandX was designed to use external Seagate 1 TB SSHD hybrid drive to store MongoDB and analysis files.

4.4.1.10 Malware analysis laptops/desktops

AndroSandX manual analysis is carried out on the separate network attached analysis machines. This proved flexibility to perform malware analysis using operating system customized for Android forensics, for example Santoku-OS or MobiSec OS. This process isolates the possibility of directly infecting AndroSandX core server also provides flexibility to use virtual instance of analysis machine. Multiple analysis machines can be added to system allowing multiple researchers to work on AndroSandX tickets at the same time. Use of this architecture provides really a great flexibility to researchers and AndroSandX malware analysis environment.

4.4.1.11 Cat 5e cable

For networking purpose and faster local network complete networking is done using good quality cat5e cable.

4.4.1.12 Network device IP addresses and subnet details

Table 4 gives details bout the devices and their IP address, table also includes subnet and Internet connectivity details.

Table 4 : Experimentation Lab Network Subnet and Address Details

Device Name	IP	Subnet	Direct access to internet
WiFi router	192.168.0.1	192.168.0.0/24	Yes
AndroSandX core server	192.168.0.102, 192.168.0.103	192.168.0.0/24	Yes
Network Attached Storage	192.168.0.101	192.168.0.0/24	No (Only local access)
Virtual Android instances	192.168.56.10, 192.168.56.11, 192.168.56.12, 192.168.56.13, 192.168.56.14, 192.168.56.15, 192.168.56.16, 192.168.56.17, 192.168.56.18, 192.168.56.19,	192.168.56.0/24	Yes
Cuckoo service	192.168.56.1:2042	192.168.56.0/24	No (Only local access)
MongoDB service	192.168.0.103:27017	192.168.0.0/24	No (Only local access)
Analysis Laptop	DHCP	192.168.0.0/24	Yes

4.5 Firewall IP tables rules

IPtable based firewall is implemented at the AndroSandX server, this allows restricting access to unauthorized services. IPtables also restricts uncontrolled access to infected Android instance. Appendix B includes are the IPtable rules that are implemented.

4.6 Development Platform and database storage solution

Python is used as the programming language in the AndroSandX. For the efficient development PyCharm[45] is used as an integrated development environment. For machine learning Scikit-learn[43] and Numpy[44] libraries are used. Scikit-learn allows easy implementation of classification and regression algorithms, Numpy is used in creating Numpy array of inputs for Scikit-learn functions. Python version 2.7 is used for the development.

For database storage non-relational database MongoDB v3.2 [39] is chosen. MongoDB provides great ease of storing document format data that is generated in the AndroSandX analysis process. Non-relational database provides great flexibility in modifying/adding fields in the documents that are non consistent in all applications. Considering the type and requirements of data structures in AndroSandX MongoDB provides the best compatibility.

4.7 Application sample source and Threat Intel source details

All the applications used in the research are from VirusTotal.com [40], chosen applications are very well detected by VirusTotal as malware or benign. Malicious applications used have the detection ratio more than 10 and benign has detection ratio of 0. VirusTotal applications are downloaded using Python API. VirusTotal provides a very trusted source of categorized application for sample set, and seems to be the best choice among the possible available sources.

For collecting threat intelligence 2 main online sources are used, VirusTotal.com and IPvoid.com [47]. Virustotal gives great details about the URL, domain being involved in any malicious activity. IPvoid provides details about IP address being flagged as blacklisted. AndroSandX extensively depend on these sources for collecting lot of threat intelligence for different URL, domain, IP addresses that are found during

automated analysis process. Few of the features submitted to machine learning algorithms directly depend on the threat intelligence gathered from VirusTotal and IPvoid.

4.8 3rd party tools and platform in use

Table 5 : 3rd Party Tool Version and Use Details

Tool	Version Details	Details
Ubuntu 14.04 Server OS	14.04 Ubuntu LTS Server	Used for running AndroSandX application and hosting database server
MongoDB	MongoDB 3.2.9	Server as a database storage service
Android 4.4 r2 x86	Android Kitkat – x86	Used for creating virtual instances of Android for malware analysis
Santoku OS	Santoku 0.5 – Lubuntu 14.04	Provides tool rich environment for manual analysis on malicious applications
MobiSec	MobiSec – Ubuntu Linux LTS 10.04	Provides tool rich environment for manual analysis on malicious applications
Cuckoo Droid	Cuckoo 1.1	Provides malware sandboxing and analysis data collection tool
Snort	Snort 2.9	Intrusion detection system for testing signature performance
Xposed Framework	Xposed Version 2.7	Framework for hiding virtualization from running applications

Apktool	Apktool 2.0.0	De-compilation tool
JDGUI	JD-GUI 1.4.0	Graphical interface for reading .java extension files
Android Studio	Android Studio 2.1	Android application development and assistive tool, using mainly for debugging in this research
TCPDump	Tcpdump 4.7.3	Network packet capture tool
Wireshark	Wireshark 2.0.5	Network packet capture analyzer
Droidmon	Droidmon 2.0	Android Xposed framework based android logging and monitoring tool
jarsigner	--	Used in resigning modified Android application, used in recompiling and resigning application in debug mode
ADB	Android Build tool 23.0.0	Android command line access tool
Drozer	Drozer v2.3.4	Android penetration testing and exploitation framework

5 Programming details about AndroSandX

5.1 Overview of layered architecture and its importance

Figure 6 gives and overview of the steps and flow followed in experimentation. AndroSandX has layer-wise separation of different functions. Each layer works on AndroSandX ticket and can only work on certain tickets, as there is a dependency of each layer on other for completion of tasks and data collection. Fundamentally the control flow of the system takes two paths one reaches to end state by achieving creation of machine learning model (training phase) and other path reaches end state after creation of SNORT signatures for possible predicted malware applications. First logical control flow goes through dynamic analysis, feature extraction and machine learning model training. Second logical control flow goes through dynamic analysis, feature extraction, prediction using existing machine learning model, perform manual analysis on the predicted malware and end state is achieved by creation of SNORT signature and testing it to detect the malware under manual analysis. Details about each layer and its functions are explained in further sections.

5.2 MongoDB collection details

AndroSnadX data is stored using MongoDB, a non-relational database. The choice of such database storage is made as it allows runtime modification and addition of any new document element. Most of the data in analysis can be easily represented using key value pair with no pre defined data type, which can be very efficiently performed using MongoDB. AndroSandX has two different databases asx_core and analysis_core.

asx_core is used for storing all the AndroSandX ticket related date and other system required data. Following are the collections and data description under asx_core:

Table 6 : MongoDB Collection Details

Collection Name	Data description
vt_ipv_results	Stores the threat intelligence collected data for every ticket to avoid repetitive analysis, also separates threat intelligence data from AndrodSandX ticket to avoid unintended sharing of 3rd party

	data
analyzed_url_ip	Stores the already analyzed url and ip to avoid repetitive query for same resource to threat intelligence collection sources
static_var	Sores the current ticket id for keeping count of total ticket and new ticket id assignment
pcap_snort_log	Stores formatted SNORT logs for each ticket
new_tickets	This collection stored the new ticket while application submission phase
analysis_tickets	Stores the AndroSandX ticket, this collection have documents that are filled through out the execution of AndroSandX.

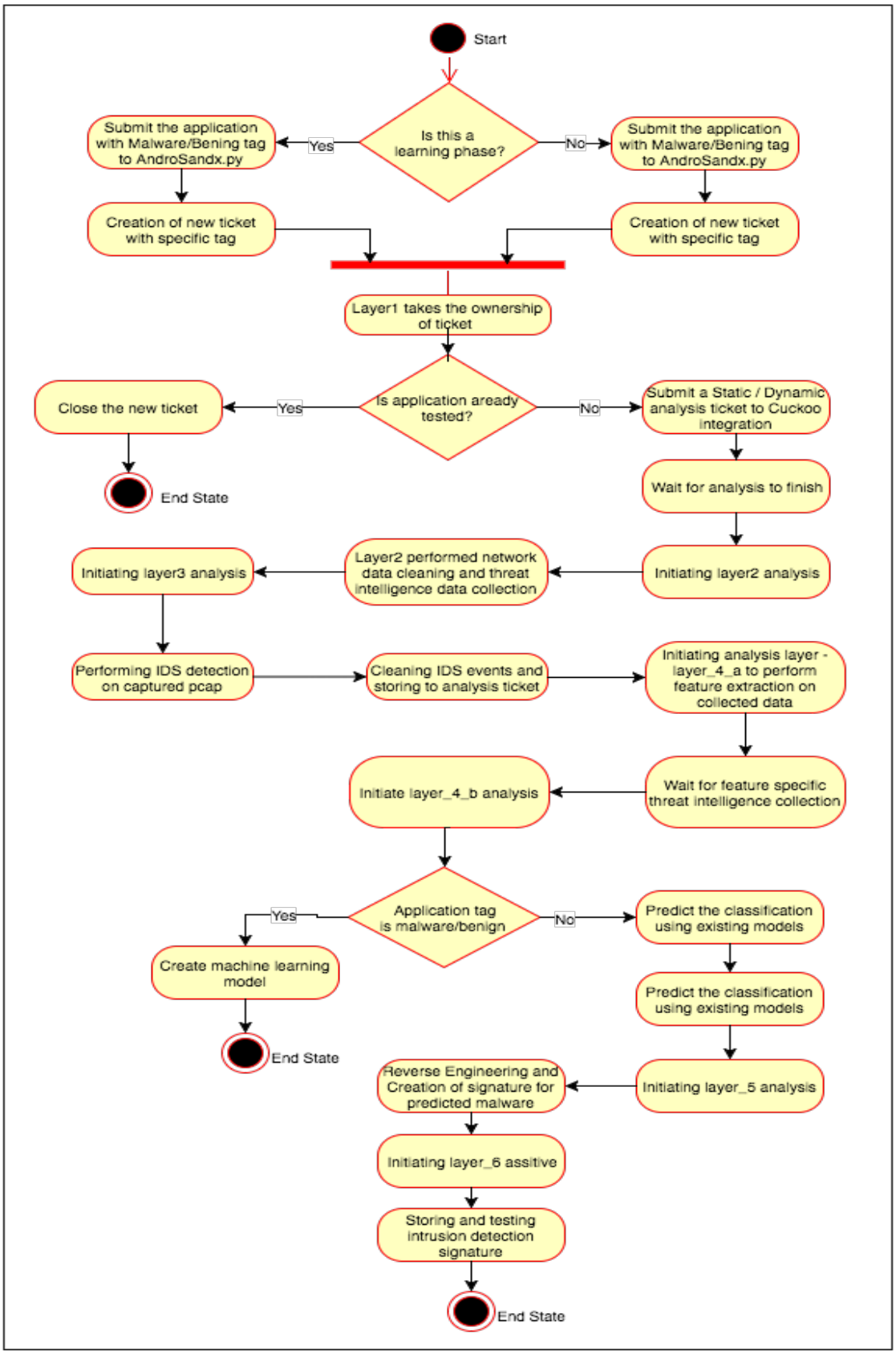

Figure 6 : AndroSandX Decision Flow Diagram

analysis_core data base is mainly used by Cuckoo Droid to add dynamic analysis related data. Some other data is also added using AndroSandX functions; the focus of this database is to store all the dynamic analysis extracted data. Name of the collection storing all dynamic analysis data is “analysis” some other Cuckoo Droid related collections are also part of this database.

5.3 AndroSandX ticket structure and File storage directory details

AndroSandX uses ticket-based process to interact between different layers. The ticket has unique structure and all submitted applications are represented as ticket in the system. Figure 7 shows the skeleton of the AndroSandX ticket, it gives details about the top level document elements and their data types. Table 7 gives details about the file storage structure for the Analysis data, some of the raw data is stored on the disk first and then used for data abstraction to retrieve relevant data in later phases of analysis.

```
{
"_id": <string>,
"analysis_start_time": <date>,
"ai_feature": <dictionary>{},
"ai_decision": {
"data": <list>[],
"analysis_conclusion": <string>
},
"cuckoo_id": <string>,
"report": <boolean>,
"analysis_end_time": <boolean>,
"snort_signatures": <boolean>,
"apktool_data_path": <boolean>,
"current_analysis_layer": <string>,
"cuckoo_folder_id": <integer>,
"analysis_status": <string>,
"time_submission": <boolean>,
"android_package_name": <boolean>,
"file_path": <string>,
"ticket_id": <integer>,
"hash_sha256": <string>,
"android_app_name": <boolean>,
"droidmon_exists": <integer>
}
```

Figure 7 : AndroSandX Ticket Object Variable Details

Table 7 : AndroSandX File Directory Structure Details

Directory/File name	Type(Directory/File)	Description
apktool_output	Directory	Used for storing output from Apktool,

		contains code for applications
binary	File	Binary file of the applications
dump.pcap	File	Pcap file captured during dynamic analysis
files	Directory	Analysis related extra files and intelligence used in manual analysis
reports	Directory	Cuckoo Droid reports
snort_logs	Directory	SNORT log directory for storing log and other SNORT related files while testing of network capture against SNORT rules
asx_ai.pcap	File	Sorted Pcap file after reducing unnecessary conversations from raw pcap
dex2jar_output	Directory	Java code for manual analysis extracted using dex2jar
dump_sorted.pcap	File	Sorted dump.pcap file
logs	Directory	Droidmon and AndroSandX related logs
shots	Directory	Screenshots for the applications during dynamic analysis

5.4 Layer Name: AndroSandX

5.4.1 Details:

AndroSandX module is functionally the client side module in the system. Its function is to initiate an analysis process.

5.4.2 Input:

Take file location / Folder location with tag malware/benign/analyze

5.4.3 Output:

Create a new ticket object and store in the collection, also put the new ticket into new ticket queue for further processing.

5.4.4 Function/Class details:

5.4.4.1 def start_asx(file_apk)

This function initiates the global variable and ticket creation process. Global variables hold details about current analysis ticket count and new ticket count.

5.4.4.2 def create_new_ticket(file_path)

Create AndroSandX new ticket and fill in initial details.

5.4.4.3 def hash_calculator(file_path)

Computes and return hash of submitted file.

5.4.4.4 def init_global_var()

Initiates global variable object, this objects holds details about system wide global variables. Ticket count and queue for new tickets is stored in this variable.

5.5 Layer Name: Layer_1

5.5.1 Details:

Layer 1 works on new tickets, checks if the analysis is already done or not and then creates an AndroSandX analysis ticket request. This request is different than new ticket request as this separates client given new tickets from actual AndroSandX ticket, which allows refining, and more controlled rejection of unnecessary analysis request. Also hides direct database interaction from client side. Cuckoo analysis for each ticket is also initiated at this layer.

5.5.2 Input:

Layer 1 takes new tickets as its input, it is designed to run in infinite looped thread. It continuously looks for new submitted requests and process them.

5.5.3 Output:

Output of this layer is to create AndroSandX analysis ticket with properly tagging current layer and completion status. Also deletes new ticket once a permanent AndroSandX ticket is created.

5.5.4 Function/Class details:

5.5.4.1 class cuckoo_analysis_req(object)

Cuckoo Droid accepts class cuckoo_analysis_req object for running a dynamic analysis on given application. This class just has initialization function to create object that need to be passed to Cuckoo Droid.

5.5.4.2 def layer_1_new_ticket_watcher()

This function continuously watch for new tickets submitted to AndroSandX and takes ownership of then and convert then in to AndroSandX tickets.

5.5.4.3 def layer_1_init()

This function initializes layer 1, it looks for any previously pending tickets that need to be worked on before taking any new tickets.

5.5.4.4 def layer_1_analysis()

Takes new pending tickets and check if they are already analyzed, if not submits to Cuckoo Droid. Once completed it proceeds with updating of AndroSandX ticket.

5.5.4.5 layer_1.__main__

Layer 1 main function runs in infinite loop to process any newly created analysis requests.

5.6 Layer Name: Layer_2

5.6.1 Details:

Layer 2 is responsible for collecting threat intelligence from different sources, it also perform some data abstraction to get relevant data for further analysis. Layer 2 takes ownership of ticket after Cuckoo Droid analysis is completed.

5.6.2 Input:

Accepts ticket that has dynamic analysis completed. Also the sub-layer details need to be given, this allows more granular control on threat intelligence collection process.

5.6.3 Output:

Layer 2 completes network data abstraction, apk details extraction and threat intelligence gathering from different source. Once all the data for a ticket under analysis is collected then it is stored in the AndroSandX ticket document. Also threat intelligence data is cached with indexing using hash.

5.6.4 Function/Class details:

5.6.4.1 def layer_2_A()

This function collects some primary details about the .apk file. Most of these are already been collected in the dynamic analysis. Some functioning is purposefully shifted to next layer and no more used at this level to increase efficiency.

5.6.4.2 def get_apk_info_thread_fn(temp_ticket)

This is thread function for performing apk information extraction, allows parallel processing on multiple tickets simultaneously.

5.6.4.3 def layer_2_B()

This function is designed to perform network data abstraction. It uses raw network captures to give out relevant network data abtraction.

5.6.4.4 def get_network_conversations(temp_ticket_cuckoo_folder_id)

Extracts network end-to-end conversation details. It focuses on protocol and conversation streams.

5.6.4.5 def layer_2_C()

This function controls the threat intelligence collection from different sources. IPVoid and Virustotal are used to collect most of the data. Also this function makes sure that cache is maintained to avoid request for already analyzed resource. URL, domain, IP related information is collected.

5.6.4.6 def trusted_ip(ip)

Function returns true if the submitted IP is in the trusted list. This helps in avoiding collecting unnecessary information for Android services related IPs or any trusted websites.

5.6.4.7 class virustotal(object)

Virustotal class provides required functions and variables for collecting threat intelligence from the Virustotal.com. Required access key and other hardcoded information is kept in the class global variable.

5.6.4.7.1 virustotal.__init__(self,url_or_ip,parameter)

Takes threat intelligence collection resource and performs resource analysis request to Virustotal, if resource is already analyzed then this step is skipped. Going ahead the results are collected from the Virustotal and stored in the AndroSandX ticket.

5.6.4.8 class ipvoid(object)

IPvoid class provides functionality to query IPvoid service to gather blacklist status for given IP. Extensive html parsing is performed in this class.

5.6.4.8.1 ipvoid.__init__(self, ip_address)

Initialized required class variables and requests for IP blacklist status to IPvoid.com

5.6.4.8.2 ipvoid.get_ipVoid_results(self)

Fetches and returns the IPvoid results for the ipvoid object stored IP.

5.6.4.9 class ipvoid_parser(HTMLParser)

This is a derived class for providing HTML parsing functionality.

5.6.4.10 layer_2.__main__

Controls the flow of sub sections of layer 2 for granular control. Provides option to run sub functions or run complete layer 2 analysis.

5.7 Layer Name: Layer_3

5.7.1 Details:

This layer performs execution of SNORT against the captured network traffic for an application during dynamic analysis. The purpose of this layer to check if the existing SNORT rules could possibly detect the Android malware or generate malicious activity related alerts.

5.7.2 Input:

Takes each ticket completing the layer 3 analysis as the input.

5.7.3 Output:

Any SNORT triggered alerts are parsed and stored in the AndroSandX ticket.

5.7.4 Function/Class details:

5.7.4.1 def execute_snort()

Creates a queue of tickets are that ready for layer 3 analysis and passes SNORT execution. Ticket are processed with 10 parallel thread each initiating different instance of SNORT.

5.7.4.2 def snort_exec_thread(ticket)

Instantiates the SNORT instance and passed the PCAP for detection. Any alerts that are generated during the process are stored and parsed to store in AndroSandX ticket. Parsed field includes signature id, group id, summary of alert, classification of attack vector, priority of alert, time stamp, source IP destination IP.

5.7.4.3 layer_3.__main__

Main function of layer 3 simply initiates the layer process.

5.8 Layer Name: Layer_4_a

5.8.1 Details

'Layer_4_a' is responsible for machine learning feature extraction; this layer is one of the most crucial in terms of functioning of AndroSandX. This layers does a lot of data abstraction and rule based feature value calculations. This layer also performs some threat intelligence gathering based on the feature requirements. Also utilizes cache to avoid repetitive request for same resource.

5.8.2 Input

This layer takes tickets that has already completed layer 3 analysis and have all the SNORT logs parsed. This layer requires ticket to have all the necessary data stored in defined key value pair in ticket and analysis data document in database. This layer makes

sure that it will only work on ticket that have Droidmon logs captured, this is required to have a valid set of feature data for machine learning.

5.8.3 Output

This layer computes all the 153 features for each AndroSandX ticket submitted to it, and it stores it in ticket document.

5.8.4 Function/Class details:

5.8.4.1 class ai_features(object)

Provides a 153 machine learning feature extraction functions. Defines a data structure to hold features for each ticket. Each ticket needs different instance of this object to extract features. It has total of 153 distinct features and also requires threat intelligence collection from IPvoid and Virustotal.

5.8.4.2 layer_4_a.__main__

Simply reads all the AndroSandX tickets that are ready for layer_4_a analysis and submits to ai_feature class object initialization to extract feature set.

5.9 Layer Name: Layer_4_b

5.9.1 Details:

Layer_4_b focuses on machine learning and prediction functions. Total of 153 features are extracted, all feature are stored as float data type. Most of the features are having 0 or 1 value. Each AndroSandX ticket is worked on to generate and store the required features. Any required threat intelligence collection is also carried out during feature extraction. There are two sub tasks at this layer, first is training and creation of machine learning models and second is prediction phase. Machine learning classification and regression algorithms are used, layer provides storing and restoring of machine learning models.

5.9.2 Input:

This layer works on tickets that have completed layer_4_a analysis. Utilizes data collected by previous layers.

5.9.3 Output:

Multiple outputs are achieved at this layer, main goal is to predict the classification of a submitted application. While achieving that machine learning models and feature set creation also takes place.

5.9.4 Function/Class details:

5.9.4.1 def create_feature_input()

Works on each of the AndroSandX ticket; collects data stored in analysis ticket and Cuckoo Droid generated data. Total of 153 features are extracted, and stored in the AndroSandX ticket. This function also collects any additional threat intelligence related data that might be required by for extracting feature.

5.9.4.2 def save_classifier()

Once all the required features are extracted and stored for all the learning phase applications then machine learning algorithm Random Forest is used to generate an AI model. save_classifier functions save this model to disk space for later use.

5.9.4.3 def restore_classifier()

This function reads back previously stored machine learning model. Restored models are further used for performing prediction.

5.9.4.4 def save_csv()

This functions save the comma separated values for prediction data and other data like feature importance.

5.9.4.5 def create_ai_model()

Creates machine-learning model using Random Forest algorithm. Model over-fitting is checked at this function. Created model is generated under a global object of sklearn, which is used by other functions to perform classification of applications and storing of model.

5.9.4.6 def predict()

predict function performs classification and regression using the extracted features of a given applications.

5.9.4.7 def score()

Computes accuracy and different efficiency matix of machine learning model created at this layer.

5.9.4.8 layer_4_b.__main__

Mainly orchestrates the different functions performed at this layer.

5.10 Layer Name: Layer_5

5.10.1 Details:

Layer 5 is designed for assisting in manual analysis and reverse engineering of malicious android applications. This layer provides utilities and tool integrations such as repacking of apk, signing of apk, getting JAVA code for applications. Provides easy and quick pre manual analysis assistance.

5.10.2 Input:

Takes in the AndroSandX ticket to be used for performing manual analysis, no other input is required.

5.10.3 Output:

There is no defined output for this layer, multiple sub functions do achieve output like giving out signed apk, application code extraction. Ultimate goal at this layer is to assist in manual analysis, increase efficiency and store user created SNORT signatures to AndroSandX ticket.

5.10.4 Function/Class details:

5.10.4.1 def get_apktool_info(temp_ticket)

Function utilizes Apktool to extract code, repack apk in debug mode. Also dex2jar and apk signer applications are also done.

5.10.4.2 def get_app_name(aoktool_output_folder)

Extracts the app name from the application code and store it in AndroSandX ticket.

5.10.4.3 layer_5.__main__

Simply orchestrates the different functions at this layer.

5.11 Utility function: mongodb_ASX

5.11.1 Details:

mongodb_ASX is an utility module providing different database access class and functions to delete/reinitialize MongoDB collections.

5.11.2 Function/Class details:

5.11.2.1 class my_mdb(object)

my_mdb class object initiates connection to multiple MongoDB collections.

5.11.2.2 mongodb_ASX.__main__

Reinitialize all MongoDB collections by invoking drop_all, create_all sub functions.

5.12 Utility functions: object_ASX

5.12.1 Details:

Provides AndroSandX ticket object class and global variable class. Allows taking document from database and convert it into object and work on it, makes the functioning and modifications very easy.

5.12.2 Function/Class details:

5.12.2.1 class ticket(object)

Defines the AndroSandX ticket. Class functions allow printing class or getting a dictionary object for the AndroSandX ticket.

5.12.2.2 class global_var(object)

Global variable used to share data between multiple layers in the AndroSandX system. This class object allows access to global variable stored in database and allows updating the variable.

6 Results

6.1 Choice of machine learning algorithm

Over all machine learning algorithms can be categorized into 3 main category, supervised learning, unsupervised learning and reinforcement learning. Choosing one of these completely determined based on what sort of data is available for training phase. If the output state and distinct input features are available then the supervised learning might be best fit. In other case if the learning data do not have any certain output state defined already then unsupervised learning would be a good approach. In a situation where learning data set is not available and machine learning models are required to be built based on ongoing data input then reinforcement learning is best fit. Reinforcement learning is trial and error based approach, and system learns from past decision gaining knowledge every time it makes any decision.

One more important decision to be made during choice of algorithm is either use regression to get a continuous output value or have a classification-based approach. In regression, the output of system is a continuous value. Depending on the parameters output can be some what clustered in specific regions on output ranges. Classification on the other hand has certain defined classes, output of machine learning model need to predict one of the predefined class as output. In case of this research, classification based algorithms are more relevant as the results are expected to be of only two types/class. In this research application can be member of either “Benign” class or “Malware” class. Though to prove the justification of use of classification algorithms certain regression algorithm results are also presented.

Some of the well-known machine learning algorithms and the reason for choosing or rejecting them for this research is stated bellow [48].

6.1.1 Linear Regression

Linear regression is trying to find a linear relation which best fits all the given data to given output in minimum error in predicting output. An attempt is made to find a polynomial function, which suitably represents the input to output relation. Linear regression is not a good approach in this research as the input data feature is mixture of binary and continuous data and resultant state is expected to

be just either “Malware” or “Benign”. The type of the data generated in research and expected output is not linear or not expecting a linearly distributed output so this approach is discarded for research.

6.1.2 Decision Tree

Decision tree is mostly used for performing classification-based problems. Feature set is grouped in to multiple subsets to predict the output based on learning data set. Decision tree works very well in case of mixed data set having continuous and binary feature data. Decision tree is well suited for this research, though a better accuracy implementation “Random Forest” is used which at its core use decision tree.

6.1.3 SVM (Support Vector Machine)

SVM is a graph plotting based algorithm, each of the feature is plotted as a dimension in n-dimensional space where n is the number of features. Plotting allows distribution of sample data in n dimensional space and next step would be to find a unique plane that separates the samples in a most distinct and efficient manner. The plane becomes the equation that predicts the class of the input sample. SVM is a supervised learning algorithm. In this research SVM do not give good result because the features have binary value.

6.1.4 Naïve Bayes

Naïve Bayes machine learning algorithm uses Bayes theorem. Works very well with supervised learning and classification requiring problems. In security industry Naïve Bayes and Random Forest are the two competing algorithms, based on the data set each gives better accuracy. For the research decision was made to use Random Forest over Naïve Bayes algorithm.

6.1.5 KNN (K-Nearest Neighbors)

KNN is a regression and classification algorithm, works on finding votes of the K neighbors. In this research KNN is not that suitable as the data is binary resulting into too much of clustering in one region. Further the combination of features is more important than a value of an individual feature.

6.1.6 Random Forest

Random Forest is an ensemble of decision trees. Multiple decision tree votes for the final classification. A small sub set of features is chosen at random to create a decision trees, all these decision tree then vote for the outcome of total forest. Random Forest works very well with mixed data, in this research features are mixture of binary and float data type. Random Forest is the most suited algorithm for this research.

6.2 Regression and classification

Both regression and classification based statistics are presented here for the data set used under this research. One of the most important factors in building machine learning model is to avoid over-fitting and obtain a good cross-validation score [49]. Cross validation score indicates how generalized the modes are for any unknown data set. This research was able to achieve a cross valuation score of 0.98 for Random Forest based classification which a very much an indication of a successful model building. Total of 500 trees are used in the forest, which seems to create fairly accurate models.

```
============== Random_forest Model building statistics ==============
Classification :
RandomForestClassifier(bootstrap=True, compute_importances=None,
        criterion=gini, max_depth=None, max_features=auto,
        min_density=None, min_samples_leaf=1, min_samples_split=2,
        n_estimators=500, n_jobs=1, oob_score=False, random_state=None,
        verbose=0)
cross_val_score
0.975503727597
Regression :
RandomForestRegressor(bootstrap=True, compute_importances=None,
criterion=mse,
        max_depth=None, max_features=auto, min_density=None,
        min_samples_leaf=1, min_samples_split=2, n_estimators=500,
        n_jobs=1, oob_score=False, random_state=None, verbose=0)
cross_val_score
0.829872505601
```

Figure 8 : Machine Learning Model Creation Statistics

6.3 Details about total sample set – Collection, Threat intelligence, confidence and cleaning

Choosing the right sample set for research is very important. It is important to use correctly classified learning data set. In this research sample set of Android application is used which is pre classified as malware and benign. Such sample set need to be obtained from a trusted source, in this case it VirusTotal. VirusTotal accumulates analysis from multiple well know industry leading anti-virus products. For this research total of 15000 applications were downloaded. Out of which after running through AndroSandX and cleaning based of availability of completeness of behavioral data subset of 4858 application is choose for building machine learning models. This 4858 application includes 3083 malicious applications and 1775 benign applications. All the chosen applications have all the possible analysis data available allowing accurate feature extraction.

Collection of all samples is done using VirusTotal which is very trusted and accurate source of malware sample collection. Other important external data collection involves threat intelligence collection. During the feature extraction and multiple analysis stages threat intelligence data collection is carried out. For collection of threat intelligence two resources VirusTotal and IPVoid are used. These two websites are highly reputed and hold open source threat intelligence data collected by researchers all across the world. IPVoid is used for collection the IP address blacklist status. VirusTotal is used for collection information about hashes of files, IP, URL and domain related malicious activity.

Collected data from the threat intelligence source is either HTML formatted webpage or JSON objects. Collected data need parsing or cleaning to gather required data. Data cleaning and storing is done using Python classes designed under AndroSandX for threat intelligence collection.

6.4 Positive sample, Negative samples and analysis

During the collection of samples from VirusTotal, a decision was made that the malware samples chosen for research should have 10+ hit on VirusTotal. That indicates the sample considered as malware in research is detected by at least 10 anti-

virus detection engines. And the benign applications will be the one with 0 match on the VirusTotal. These conditions are fairly logical and accurate considering an assumption that the uploaded sample is not very new, allowing anti-virus vendors to perform research and classify as malware or benign application.

6.5 AI feature collection details

This section explains detail and logic behind collection of each feature used for building machine learning models. These features and their correct extraction is the heart of the research, complete success of the research is dependent on choosing right features.

6.5.1 'know_adware'

This features detects the already know Android malware based of the signature and key strings present in the apk file. Mainly detects “Umeng” and “AirPush” adware. This list of adware is extensible and can be made more through by adding more string match.

6.5.2 'dangerous_permissions':

This feature is based on the Cuckoo Droid defined dangerous permissions. Triggers on use of combination of highly suspicious permission in manifest file.

6.5.3 'dynamic_code':

Looks for Cuckoo Droid detection of dynamic code in the Android application, dynamic code can be in the form of external download or pre packed jar files.

6.5.4 'known_malware_sig_string':

Feature is based on the Cuckoo Droid malware signature detection, mainly looks for “sandrorat”, “androrat”, iBanking” malware signatures.

6.5.5 'use_of_native_code':

Detects use of native code, this mainly indicates use of C++ code.

6.5.6 'hiden_payload':

Detects the hidden payload in the apk file. Applications hide the payload by different methods, payload my inclide image files, jar, apk files and other native precompiled files.

6.5.7 'use_of_android_packer':

Detects use of packer to obfuscate the code and avoid malware analysis. Packer makes the manual and automated analysis difficult. String match is completely evaded by use of packer.

6.5.8 'use_of_reflection_code':

Java reflection[50] gives an ability to determine the environment though checking for available APIs. This technique is very well used by Android malware to detect the sandbox environment and the possible updates on the device. This technique is very well used to hide the true functionality of malware from automated malware analysis system, which do not have all the API calls available as that of actual Android devices.

6.5.9 'aborted_broadcast':

This feature looks for use of Android java functions "void abortBroadcast ()". This function allows application to terminate the further broadcast receiving by other applications. This could allow receiving of SMS and other broadcast messages silently and hindering user awareness of any such activity.

6.5.10 'contains_another_apk':

Detects the presence of another executable apk file inside the application apk. Usually application can use "DexClassLoader" built in java class to load the class files from non compiled apk and jar files. This allows runtime dynamic code loading.

6.5.11 'contains_arm_binary':

Detects presence of ARM compiled binary, this can be ran on the ARM devices. This is very much using a kernel module and native applications, a good example would be use of TCPDUMP binary to capture all network traffic on Android network interface.

6.5.12 'contains_dex':

Detects the presence of Dalvik executable, dex file can be executed by the Android application as the external code.

6.5.13 'contains_jar':

Detects the presence of jar file. Jar files contains the compiled java code in terms of .class files. Malicious application can download or package malicious classes and functions as a .jar files.

6.5.14 'contains_so':

.so file stands for shared object files. These are dynamically linked code files. It allows modification or download of external code at runtime. Malware authors can use this technique to avoid detection.

6.5.15 'device_admin_granted':

Detects the use of admin access on the Android device. Malware authors can use the admin level execution to steal data from other applications and perform system level actions.

6.5.16 'execution_of_shell_command':

Detect the execution of Linux shell commands.

6.5.17 'device_fingerprinting':

This feature detects the applications attempt to gather device specific data and network service provider related data. Mostly of the time device fingerprinting involves gathering information about SIM card, country locations, IMEI number, mobile network service provider details. These techniques allow malware authors to determine possible exploit vectors to be used.

6.5.18 'installed_new_application':

Detects if the applications installed another application.

6.5.19 'queried_account_info':

Detects if the application queried for user accounts, which mainly includes users Google account details.

6.5.20 'queried_installed_apps':

Detects applications querying for installed applications on the device.

6.5.21 'accessed_private_information':

Private information includes contact information, files on storage, photos and document files.

6.5.22 'audio_recording':

Detects audio recording by application, this technique is used by Android malware to eavesdrop on victim.

6.5.23 'registered_new_receiver_at_runtime':

Android receivers are used to receive system level broadcasts. Runtime registering of receiver indicates application attempting to intercept activities or system actions that it did not declared in manifest file.

6.5.24 'sent_sms':

Detect the application sending sms.

6.5.25 'stoped_processes':

Detects if application stopped any other processes.

6.5.26 'uses_location':

Detects the applications attempting to get current physical location of the device.

6.5.27 'using_camera':

Detects the use of camera by application.

6.5.28 'creates_exe':

Detects if the application creates an .exe file on the Android device. This technique is seen from APT (advance persistent threat) adversaries. Malware authors try to compromise other Windows machines on the corporate network though compromised Android devices, this is very sophisticated technique of lateral movement in the network.

6.5.29 'use_of_ICMP_request':

Detects the application sending ICMP requests, this is useful in performing network scan for detection devices and open services.

6.5.30 'use_of_IRC':

Detect the use of network communication protocol IRC (internet relay chat). IRC is very well used in the malware command and control mechanism.

6.5.31 'use_of_SMTP':

Detects use of SMTP (Simple Mail transfer Protocol), though used by normal email applications SMTP is widely used to perform data exfiltration.

6.5.32 'cryptography_used':

Detects the use of cryptographic functions in the application.

6.5.33 'crypto_algo_DES':

Detects the use of cryptographic algorithm DES. DES is a block cipher.

6.5.34 'crypto_algo_AES':

Detects the use of cryptographic algorithm AES. AES is a block cipher.

6.5.35 'crypto_algo_RC4':

Detects the use of cryptographic algorithm RC4. RC4 is a stream cipher.

6.5.36 'Use_of_Shared_preference':

Shared preferences allows android application to store data persistently even after the closing of applications. This feature detects use of shared preferences by application.

6.5.37 'Shared_preference_contain_URL':

Detects the presence of URL string in the shared preference, this technique can be used by malware authors to download malicious code or file containing commands and configurations.

6.5.38 'shared_preference_URL_flaged_by_virustotal':

Detects if VirusTotal flags URL stored in shared preferences. Most of the time these URLs lead to download of malicious applications.

6.5.39 'apk_accessed_from_storage':

Detects the accessing of APK file from storage. Malicious applications most of the times downloads malicious application apk file to storage and then install it.

6.5.40 'zip_accessed_from_storage':

Detects accessing of zip file form storage.

6.5.41 'jar_accessed_from_storage':

Detects accessing of jar file from storage.

6.5.42 'accessed_so_file':

Detects the access of .so files from the storage.

6.5.43 'shared_preference_contains_URL_to_apk_file':

Detects if the URL is pointing to a apk file.

6.5.44 'internet_permission':

Applications asked for internet permissions.

6.5.45 'write_external_storage':

Applications asked for access to writing to external storage.

6.5.46 'change_wifi_state':

Application can change the WiFi state. Can start and stop WiFi, and detect presence of connections.

6.5.47 'access_fine_location':

Can query for accurate location of the device. This allows access to GPS coordinate.

6.5.48 'access_coarse_location':

Can access network based location, this location coordinate are not highly accurate but still accurate enough to point device to a 500 meters area.

6.5.49 'wake_lock':

Allows application to keep the device awake and keep running processes.

6.5.50 'read_history_bookmark':

Allow permission to read users browser bookmark history.

6.5.51 'write_history_bookmark':

Allow permission to write user bookmark history.

6.5.52 'read_phone_state':

Read the phone device IMEI and other manufacturer details.

6.5.53 'access_camera':

Allow permission to access camera.

6.5.54 'read_external_storage':

Allow permission to read external storage.

6.5.55 'read_logs':

Allow application to read system logs.

6.5.56 'read_calendar':

Allow application to read user calendar giving access to user schedules and meeting details.

6.5.57 'write_calendar':

Allow permission to write to user calendar.

6.5.58 'mount_unmount_filesystem':

Allow permission to mount and unmounts file system.

6.5.59 'install_uninstall_shortcut':

Allow application to install and uninstall shortcut on main screens. Allow hiding malicious installs from user.

6.5.60 'download_without_notification':

Allows applications to download other files in background and without notification of consent from user.

6.5.61 'modify_secure_system_setting':

Allow application to modify system's security related settings.

6.5.62 'directly_install_application':

Allow application to install other applications without user consent. Allows malware author to download and install malicious applications.

6.5.63 'modify_phone_state':

Allow modifying options like silent, network connection strength, GSM operator details, call details and other telephony service details.

6.5.64 'write_accesspoint_proxy_setting':

Allow changing accesspoint proxy settings.

6.5.65 'turn_phone_on_off':

Allows application to turn the phone on and off.

6.5.66 'create_bluetooth_connection':

Allows application to create new Bluetooth connection, can be used by malware to intercept Bluetooth devices around the compromised phone.

6.5.67 'intercept_outgoing_call':

Allows intercepting outbound calls from devices.

6.5.68 'directly_call_phone_numbers':

Allow application to directly make phone calls without user consent.

6.5.69 'access_superuser':

Allow application to access superuser permission.

6.5.70 'kill_background_process':

Allow application to kill background process.

6.5.71 'read_contact':

Allow application to read contacts on device.

6.5.72 'receive_WAP_push_message':

Application can receive WAP push message received by device.

6.5.73 'get_account':

Allow application to access account details on the device.

6.5.74 'manage_account_and_passwords':

Allow access to user accounts and passwords.

6.5.75 'disable_key_lock':

Application can disable key lock on mobile device which allows user to avoid unintended access to device.

6.5.76 'access_download_manager':

Allow application to access download manager, this allows scheduling download if other downloads are in progress.

6.5.77 'control_vibrator':

Controls device vibrator.

6.5.78 'sms_send_permission':

Allow application to send SMS.

6.5.79 'restart_packages':

Application can restart loaded packages.

6.5.80 'write_sms_mms':

Allows application to create or modify SMS, MMS.

6.5.81 'read_sms_mms':

Allow application to read SMS and MMS received by device.

6.5.82 'receive_sms':

Allows application to receive SMS receiving broadcasts.

6.5.83 'change_network_state':

Allow application to change state of network, this allows getting details about network details and perform connection modification.

6.5.84 'interact_accross_user_full':

Allows applications to use all resources available to user.

6.5.85 'broadcast_sticky':

Allows to receive the intent for broadcast activity even after completion of broadcast.

6.5.86 'write_setting':

Allows modification of system settings.

6.5.87 'get_running_task':

Access all the reunning tasks on device.

6.5.88 'system_level_alert':

Allows applications to receive and read system level alrts.

6.5.89 'receive_boot_completed':

Application can detect boot completion message.

6.5.90 'MAC_address':

Application has permission to access device MAC address.

6.5.91 'Network_Operator':

Application has permission to access network operator.

6.5.92 'Device_ID':

Application has permission to get device ID.

6.5.93 'Sim_serial_Number':

Application has permission to access SIM serial number.

6.5.94 'Sim_Operator_Name':

Application has permission to access SIM operator name.

6.5.95 'get_network_country':

Application has permission to get network country code.

6.5.96 'sim_iso_country':

Application has permission to get SIM ISO country code.

6.5.97 'get_line_number':

Application has permission to get the telephone number of the device.

6.5.98 'screen_On':

Application has permission to turn screen on.

6.5.99 'screen_off':

Application has permission to turn device screen off.

6.5.100'package_added':

Application has permission to add new package.

6.5.101'package_removed':

Application has permission to removed installed package.

6.5.102'SMS_received':

Application has permission to check received SMS broadcast.

6.5.103'SMS_delivered':

Application has permission to check if SMS was delivered.

6.5.104'device_admin_enabled':

Application has permission to check admin access on device.

6.5.105'boot_completed':

Application has permission to check boot completed status of device.

6.5.106'action_power_connected':

Application has permission to check if the charging chord is connected.

6.5.107'action_power_disconnected':

Application has permission to check if the charging chard was disconnected.

6.5.108'battery':

Application has permission to check state of battery.

6.5.109'user_present':

Application has permission to check if certain user is present on the device.

6.5.110'media_changes':

Application has permission to check media changes on device.

6.5.111'head_set_detect':

Application has permission to check headset detected status.

6.5.112'proxy_change':

Application has permission to check proxy changes.

6.5.113'connectivity_change':

Application has permission to check connectivity changes.

6.5.114'baidu_silent_install':

Application have custom permission biadu silent install

6.5.115'baidu_detect_root':

Application have custom application permission to detect if device is rooted.

6.5.116'cypto_function':

Application makes function call or have a static method to perform cryptographic functions.

6.5.117'contact_read':

Application makes function call or have a static method for reading contact on device.

6.5.118'camera':

Application makes function call or have a static method to access camera.

6.5.119'bookmark':

Application makes function call or have a static method to access or modify bookmarks.

6.5.120'location':

Application makes function call or have a static method to access location of device.

6.5.121'send_sms':

Application makes function call or have a static method to send SMS.

6.5.122'location_wallpaper_readlogs':

Application have functional calls which indicates it call access location, set wallpaper and read system calls, such combination of calls are unexpected unless performing malicious activity.

6.5.123'credential_killProcess_audioRecord_camera_location_internet':

Application have function calls that indicates potential malicious behavior, application access credentials, kill process, record audio, use camera, get location and have access to internet.

6.5.124# Network forensics

6.5.125'IP_address_blacklisted':

Detects if the IP address related to application's network communication are blacklisted.

6.5.126'IP_address_belongs_high_severity_country':

Detects if the IP communication is destined to IP address in high severity countries.

6.5.127'IP_address_hosting_flaged_url_on_Virustotal':

Detects if IP address involved in communication have malicious URL flagged in VirusTotal.

6.5.128'IP_address_hosting_malicous_files_on_Virustotal':

Detects if the IP address involved in communication have malicious file hosted on as per VirusTotal reports.

6.5.129'URL_flaged_on_virustotal':

Detects if the URL accessed from application is flagged on VirusTotal.

6.5.130'ethernet_frames':

This has a floating data type, it simple counts the number of Ethernet frames transferred by application during dynamic analysis.

6.5.131'ethernet_bytes':

Number of Ethernet data bytes transferred during dynamic analysis.

6.5.132'arp_frames':

Number of ARP frames during the dynamic analysis.

6.5.133'arp_bytes':

Number of ARP data bytes transferred during dynamic analysis.

6.5.134'ip_frames':

Number of IP protocol frames during dynamic analysis.

6.5.135'ip_bytes':

Number of IP protocol data transfer bytes during dynamic analysis.

6.5.136'udp_frames':

Number of UDP frames during dynamic analysis.

6.5.137'udp_bytes':

Number of UDP protocol data transfer bytes during dynamic analysis.

6.5.138'tcp_frames':

Number of TCP frames transferred during the dynamic analysis.

6.5.139'tcp_bytes':

Number of TCP bytes transferred during the dynamic analysis.

6.5.140'dns_frames':

Number of DNS frames transferred during dynamic analysis.

6.5.141'dns_bytes':

Number of DNS data transfer bytes during dynamic analysis.

6.5.142'irc_frames':

Number of IRC frames transferred during the dynamic analysis.

6.5.143'irc_bytes':

Number of IRC data transfer bytes during dynamic analysis.

6.5.144'smtp_frames':

Number of SMTP frame transferred during the dynamic analysis.

6.5.145'smtp_bytes':

Number of SMTP data transfer bytes during the dynamic analysis.

6.5.146'udp_uploaded_data':

Total amount of UDP data uploaded.

6.5.147'udp_downloaded_data':

Total amount of UDP data downloaded during dynamic analysis.

6.5.148'tcp_uploaded_data':

Total amount of TCP data uploaded during the dynamic analysis.

6.5.149'tcp_downloaded_data':

Total amount of TCP data downloaded during the dynamic analysis.

6.5.150'udp_coversation_average_data_transfer_rate':

Data transfer rate for UDP conversation during the dynamic analysis.

6.5.151'tcp_coversation_average_data_transfer_rate':

Data transfer rate for TCP conversation during the dynamic analysis.

6.5.152'snort_alert_for_malaware_related_activity':

Detection of malware related activity using latest SNORT malware signatures.

6.5.153'snort_alert_for_exploit_kit_activity':

Detection of exploit kit related activity using latest SNORT signatures.

6.5.154'snort_high_severity_alert':

Triggering of high severity signature in by SNORT on traffic captured during dynamic analysis.

6.6 Learning data

A learning data set is a set of samples that have been already analyzed and classified. Sample set contains trusted well-analyzed and cleaned feature set samples. These applications are passed through analysis process to obtain the required feature set. After cleaning samples for availability of all the features, there were 3083 malicious applications and 1775 benign applications left.

After the learning phase and machine learning model creation, a test sample set of 509 applications is submitted for analysis. These 509 applications have 156 malware and 353 benign applications. This sample set is submitted to evaluate the prediction and classification efficiency of trained models.

6.7 Prediction sample and prediction results

Following data set array shows the predicted output for the classification and regression of sample set. Support Vector Machine (SVM) and Random Forest Algorithm prediction data set is given bellow. Total sample set length is 509 and 1 in output represent "Malware" class and 0 represent "Benign" class.

Support Vector Machine Classification output:

```
[1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 0 0 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
```

Figure 9 : Support Vector Machine Classification output

Random Forest Classification output:

```
[0 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
1 1 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 1 1 1
1 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1
1 1 1 1 0 1 1 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1
1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
```

Figure 10 : Random Forest Classification Output

Random Forest Regression output:

```
[ 0.03   1.         0.966  0.98   0.97   0.982  0.974  0.998  0.546  0.84
 0.588  0.994  0.858  0.89   0.738  1.       0.988  1.         0.942  0.404
 0.426  0.736  0.996  0.  0.874  0.958  0.636  0.426  0.982  0.958
 0.982  0.998  0.836  0.704  0.996  0.3    0.992  0.998  0.992  0.746
 0.968  0.816  0.998  0.158  0.    0.412  0.83   1.  0.954  0.846
 0.99   0.854  0.842  0.858  0.962  0.94   0.986  0.984  0.788  0.556  1.
 0.888  0.824  1.         0.504  0.832  0.066  0.064  0.858  0.996  0.        0.972
 0.968  0.676  0.996  1.  0.422  0.638  0.176  0.79   0.424  0.992
 0.99   0.402  0.998  1.   0.544  0.         0.972  0.822  0.062  0.824
 0.59   0.648  0.396  0.97   0.996  0.418  0.996  0.61   0.756  0.962  1.
 0.996  0.998  0.938  0.768  1.    0.998  0.982  0.506  0.972  0.4    0.782
 0.928  0.596  0.86   0.988  0.96   0.908  0.916  0.006  0.994  0.98
 0.536  1.         1.         0.564  0.768  0.984  0.43   0.952  0.056  0.634  1.
 0.992  0.434  0.582  0.714  0.88   0.998  0.506  0.988  0.908  0.962
 0.962  0.06   0.96   0.98   0.956  0.972  0.988  0.172  0.97   1.     0.478
 0.        0.038  0.008  0.25   0.01   0.014  0.44   0.         0.004  0.006
 0.056  0.374  0.672  0.058  0.018  0.002  0.       0.004  0.         0.062
 0.014  0.072  0.         0.57   0.         0.       0.       0.         0.        0.002  0.
 0.008  0.102  0.188  0.052  0.    0.008  0.012  0.01   0.104  0.     0.922
 0.        0.068  0.002  0.022  0.   0.         0.108  0.07   0.  0.146
 0.314  0.         0.24   0.         0.238  0.         0.022  0.01   0.124  0.016
 0.002  0.         0.       0.6      0.       0.       0.05   0.         0.        0.028
 0.002  0.         0.222  0.         0.008  0.006  0. 0.082  0.026  0.348  0.
 0.        0.         0.002  0.612  0. 0.048  0.318  0. 0.002  0.         0.
 0.        0.304  0.946  0.114  0.05   0.056  0.       0.       0.188  0.046  0.
 0.004  0.014  0.004  0.  0.         0.004  0.114  0.146  0.018  0.002
 0.004  0.006  0.         0.         0.       0.       0.       0.        0.072  0.022
 0.002  0.         0.02   0.34   0.404  0.    0.       0.72   0.         0.034
 0.004  0.         0.         0.002  0.036  0.392  0.232  0.916  0.036  0.012
 0.374  0.01   0.126  0.   0.         0.       0.002  0.126  0. 0.        0.026
 0.        0.01   0.026  0.  0.004  0.002  0.494  0.   0.048  0.162
 0.144  0.076  0.004  0.004  0.    0.       0.       0.012  0.032  0.514
 0.062  0.         0.         0.382  0.         0.008  0.004  0.07   0.092  0.     0.002
 0.08   0.         0.         0.006  0.014  0.01   0.006  0.     0.002  0.182
 0.864  0.022  0.082  0.  0.024  0.062  0. 0.216  0.416  0.038  0.
 0.006  0.         0.         0.         0.356  0.02   0.10.002  0.         0.        0.004
 0.014  0.96   0. 0.         0.03   0.63   0.002  0.628  0.     0.42   0.
 0.208  0.012  0.006  0.056  0.176  0.     0.       0.07   0.         0.006  0.
 0.        0.112  0.222  0.016  0.   0.03   0.042  0.032  0.    0.072
 0.026  0.234  0.002  0.028  0.    0.006  0.008  0. 0.        0.002
 0.03   0.006  0. 0.654  0.014  0.044  0.062  0.    0.01   0.012
 0.724  0.         0.002  0.954  0.028  0.06   0.068  0.       0.        0.04   0.65
 0.168  0.04   0.114  0.132  0.016  0.     0.       0.       0.        0.        0.
 0.02   0.         0.006  0.046  0.002  0.128  0.     0.       0.026  0.008
 0.102  0.006  0.978  0.024  0.    0.084  0.         0.044  0.136  0.162
 0.032  0.222  0.014  0.522  0.004  0.006  0.008  0.06   0.242  0.066  0.2
 0.14   0.         0.268  0.228  0.036  0.03   0.052  0.002  0.004  0.032
 0.066  0.006  0.         0.06   0.03   0.016  0.142  0.006  0.042  0.322
 0.014  0.278  0.026  0.022  0.304  0.054  0.214  0.264  0.258]
```

Figure 11 : Random Forest Classification output

Actual Sample Set classification:
Following data array shows the actual classification of the applications. Out of 509 applications submitted for analysis 156 were classified as "Malware" and 353 were "Benign" applications.

```
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
```

Figure 12 : Actual Sample Set Classification

6.8 Feature Importance:

In total 153 features were used as input for machine learning algorithm. After the prediction to validate that few features are not over weighted cross validation score was calculated. Cross validation score indicates successful creation of AI models and to what extent models can accurately predict unknown samples. Also cross validation score indicate if the model is over fitted. Over fitting of model is an indication of some feature set values always gets priority an predication is always result into same classification.

Feature importance is way to check if certain features are not over weighted. Table shows importance of each feature out of total 1.000. Feature importance value indicates how much each feature contributes during the prediction using created model.

Table 8 : Feature Importance Values

Feature Name	Feature	Feature

	Importance [M: 2000 B: 1000]	Importance [M: 3083 B: 1775]
SMS_delivered	0.000225	0.000274
modify_secure_system_setting	0.000191	0.000234
device_fingerprinting	0.007239	0.006053
tcp_coversation_average_data_transfer_rate	0.006215	0.006865
screen_off	0.003100	0.002495
Use_of_Shared_preference	0.005814	0.006033
receive_sms	0.004008	0.003553
broadcast_sticky	0.000913	0.000926
install_uninstall_shortcut	0.010185	0.009681
head_set_detect	0.000247	0.000224
get_line_number	0.003943	0.003280
disable_key_lock	0.001576	0.001953
battery	0.006140	0.006452
use_of_ICMP_request	0.000166	0.000187
location	0.006495	0.006517
wake_lock	0.002782	0.002799
smtp_frames	0.000281	0.000284
contact_read	0.080827	0.082306
accessed_private_information	0.001367	0.001497
tcp_bytes	0.008204	0.008201
tcp_frames	0.006546	0.007287
Sim_serial_Number	0.007664	0.005687
shared_preference_contains_URL_to_apk_file	0.000032	0.000036
registered_new_receiver_at_runtime	0.008681	0.012259
contains_dex	0.000000	0.000000
read_sms_mms	0.004153	0.004716
IP_address_belongs_high_severity_country	0.053409	0.059927

contains_jar	0.000000	0.000000
manage_account_and_passwords	0.000790	0.000925
receive_WAP_push_message	0.000599	0.000451
directly_call_phone_numbers	0.001595	0.001538
screen_On	0.001001	0.001076
tcp_downloaded_data	0.008375	0.008732
write_external_storage	0.005832	0.005646
internet_permission	0.010057	0.012358
arp_bytes	0.004225	0.003951
ethernet_bytes	0.010868	0.010567
smtp_bytes	0.000207	0.000209
uses_location	0.000237	0.000171
snort_alert_for_malaware_related_activity	0.000000	0.000000
cypto_function	0.015592	0.011215
action_power_connected	0.003530	0.002688
audio_recording	0.000040	0.000022
use_of_native_code	0.003228	0.003046
get_running_task	0.035906	0.030702
dns_frames	0.005726	0.005673
access_camera	0.002402	0.002125
Shared_preference_contain_URL	0.000941	0.001013
snort_high_severity_alert	0.000000	0.000000
snort_alert_for_exploit_kit_activity	0.000000	0.000000
stoped_processes	0.000059	0.000078
read_phone_state	0.024298	0.024702
read_history_bookmark	0.005333	0.005165
crypto_algo_RC4	0.000000	0.000003
ip_frames	0.007820	0.007470
arp_frames	0.003449	0.003500
udp_uploaded_data	0.000654	0.000510

accessed_so_file	0.020489	0.020991
connectivity_change	0.002889	0.003166
IP_address_hosting_flaged_url_on_Virustotal	0.003509	0.004307
crypto_algo_DES	0.013869	0.014671
udp_bytes	0.008282	0.008403
action_power_disconnected	0.002597	0.003175
Device_ID	0.053690	0.055602
apk_accessed_from_storage	0.038469	0.033273
user_present	0.002118	0.002894
write_sms_mms	0.002359	0.001736
installed_new_application	0.000000	0.000000
crypto_algo_AES	0.003038	0.003217
device_admin_enabled	0.000000	0.000000
dns_bytes	0.008345	0.009235
irc_bytes	0.000000	0.000000
control_vibrator	0.002338	0.002246
tcp_uploaded_data	0.007722	0.007902
dynamic_code	0.005111	0.006226
using_camera	0.000000	0.000000
get_account	0.003942	0.004455
receive_boot_completed	0.007926	0.008128
Network_Operator	0.003012	0.003347
access_fine_location	0.004098	0.003793
queried_account_info	0.001300	0.001308
change_wifi_state	0.010473	0.010435
change_network_state	0.002014	0.002272
udp_downloaded_data	0.000553	0.000690
contains_another_apk	0.009256	0.009598
dangerous_permissions	0.011083	0.015858

baidu_silent_install	0.000003	0.000004
Sim_Operator_Name	0.001029	0.001081
device_admin_granted	0.000000	0.000000
write_calendar	0.001628	0.001757
location_wallpaper_readlogs	0.001451	0.001463
known_malware_sig_string	0.000000	0.000000
bookmark	0.003837	0.003963
jar_accessed_from_storage	0.002845	0.002553
restart_packages	0.001943	0.001888
irc_frames	0.000000	0.000000
camera	0.003308	0.003318
use_of_SMTP	0.000263	0.000178
use_of_android_packer	0.020429	0.021273
ethernet_frames	0.007564	0.008015
zip_accessed_from_storage	0.001509	0.002002
write_history_bookmark	0.001260	0.001121
udp_frames	0.006532	0.005916
ip_bytes	0.010512	0.011033
sim_iso_country	0.002002	0.001797
system_level_alert	0.037465	0.035081
write_setting	0.010177	0.012814
credential_killProcess_audioRecord_camera _location_internet	0.000000	0.000000
read_calendar	0.001730	0.002114
creates_exe	0.000000	0.000000
boot_completed	0.000027	0.000035
baidu_detect_root	0.000003	0.000002
send_sms	0.006520	0.006374
access_download_manager	0.000360	0.000347
sms_send_permission	0.010916	0.009283

write_accesspoint_proxy_setting	0.000338	0.000324
download_without_notification	0.000334	0.000431
know_adware	0.007757	0.008787
aborted_broadcast	0.000000	0.000000
directly_install_application	0.002567	0.002455
modify_phone_state	0.000778	0.000792
read_logs	0.004988	0.002659
access_superuser	0.000055	0.000053
access_coarse_location	0.010026	0.010192
udp_coversation_average_data_transfer_rate	0.041368	0.048152
contains_so	0.003535	0.004523
get_network_country	0.003323	0.003731
IP_address_hosting_malicous_files_on_Virustotal	0.000809	0.000710
package_removed	0.000881	0.000858
URL_flaged_on_virustotal	0.001156	0.000934
turn_phone_on_off	0.000708	0.000767
mount_unmount_filesystem	0.008521	0.006661
use_of_IRC	0.000152	0.000199
package_added	0.049717	0.049025
shared_preference_URL_flaged_by_virustotal	0.000000	0.000000
create_bluetooth_connection	0.000959	0.001011
cryptography_used	0.012460	0.012507
IP_address_blacklisted	0.000000	0.000000
contains_arm_binary	0.000000	0.000000
MAC_address	0.027863	0.025944
use_of_reflection_code	0.052308	0.043432
sent_sms	0.000753	0.001029

queried_installed_apps	0.007316	0.005174
read_external_storage	0.001863	0.001680
read_contact	0.001625	0.001434
SMS_received	0.000275	0.000296
kill_background_process	0.001527	0.002007
proxy_change	0.005473	0.005407
execution_of_shell_command	0.001169	0.001084
hiden_payload	0.002470	0.002244
intercept_outgoing_call	0.001262	0.001287
interact_accross_user_full	0.000312	0.000266
media_changes	0.000391	0.000344

6.9 Efficiency matrix:

Following matrix indicate the efficiency of the prediction using created models:

Score indicates the predication accuracy of specific model, higher the score better is the success of guessing correct classification of submitted application.

Support Vector Machine Model Accuracy Score:
0.320235756385

Random Forest Classification Model Accuracy Score:
0.92141453831

Random Forest Regression Accuracy Score:
0.661604214571

Precision Score:

Precision indicates the accuracy from false positive, false negative, true positive and true negative score prospective:

Precision :

Precision = True Positive / (True Positive + False Positive)
For class malware and class benign respectively: [0.92876712, 0.90277778]
Average precision: 0.896551724138

Recall :

Recall = True Positive / (True Positive + False Negative)
For class malware and class benign respectively: [0.96033994, 0.83333333]
Average recall: 0.83333333

6.10 Proposed process for Android signature creations and manual analysis

After an application is predicted as malware then analyst can carry out manual analysis. Following standard process is defined:

Step 1: Pulling malware and source code from AndroSandX repository, download binary and network traffic.

Step 2: Start debugging in IDA pro

Step 3: Use Linux for providing fake services like HTTP server, DNS server, Windows for running IDA pro, Android VM for execution of malware.

Step 3: Following tools are used Wireshark, IDA pro, Logcat, FakeDNS, INetSim, Honeynet.

Step 4: Track down the behavioral and network unique identifier by analyzing outbound HTTP requests and network traffic. Analyzing Logcat data create system level identifiers.

Step 5: Utilizing the unique identifiers create a snort signature and Logcat based IOCs

Step 6: Store the signature in the AndroSandX ticket for further use and documentation.

7 Conclusion

The hypothesis for the research that "It is feasible to predict a sub set of potentially malicious Android application using machine learning from a given larger set of unclassified Android applications. Such system will need a good sandboxing solution and will utilize static and dynamic behavioral data." The hypothesis for the research is proved using a successful experimentation based approach. Research work went through rigors experimentation system design and development. The outcome of the research is very positive and successful proof of the hypothesis is presented. Large data set results are made available to prove the efficient working and accuracy of the designed system.

8 References:

[1] "Google Android Security PHA classification", *source.android.com*, 2016. [Online]. Available: https://static.googleusercontent.com/media/source.android.com/en//security/reports/Google_Android_Security_PHA_classifications.pdf. [Accessed: 09- Jul- 2016].

[2] "Google Android Security Report: Malware Down, Harmful Apps Still the Platform's Biggest Threat - Security News - Trend Micro USA", *Trendmicro.com*, 2016. [Online]. Available: http://www.trendmicro.com/vinfo/us/security/news/mobile-safety/google-android-security-report-harmful-apps-the-biggest-threat. [Accessed: 09- Jul- 2016].

[3] "Internet Security Threat Report 2016 | Symantec", *Symantec.com*, 2016. [Online]. Available: https://www.symantec.com/security-center/threat-report. [Accessed: 09- Jul- 2016].

[4] "Android-Malware-Threat-Report", *www.bitdefender.com*, 2016. [Online]. Available: http://download.bitdefender.com/resources/files/News/CaseStudies/study/85/Android-Malware-Threat-Report-H2-2015.pdf. [Accessed: 09- Jul- 2016].

[5] "CuckooDroid Book — CuckooDroid v1.0 Book", Cuckoo-droid.readthedocs.io, 2016. [Online]. Available: http://cuckoo-droid.readthedocs.io/en/latest/. [Accessed: 09- Jul- 2016].

[6] "Xposed Installer | Xposed Module Repository", Repo.xposed.info, 2016. [Online]. Available: http://repo.xposed.info/module/de.robv.android.xposed.installer. [Accessed: 09- Jul- 2016].

[7] J. Gajrani, J. Sarswat, M. Tripathi, V. Laxmi, M. Gaur and M. Conti, "A robust dynamic analysis system preventing SandBox detection by Android malware", *Proceedings of the 8th International Conference on Security of Information and Networks - SIN '15*, 2015.

[8] T. Vidas and N. Christin, "Evading android runtime analysis via sandbox detection", *Proceedings of the 9th ACM symposium on Information, computer and communications security - ASIA CCS '14*, 2014.

[9] M. Spreitzenbarth, F. Freiling, F. Echtler, T. Schreck and J. Hoffmann, "Mobile-sandbox", *Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC '13*, 2013.

[10] S. Bhandari, R. Gupta, V. Laxmi, M. Gaur, A. Zemmari and M. Anikeev, "DRACO", *Proceedings of the 8th International Conference on Security of Information and Networks - SIN '15*, 2015.

[11] M. Zheng, M. Sun and J. Lui, "Droid Analytics: A Signature Based Analytic System to Collect, Extract, Analyze and Associate Android Malware", *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2013.

[12] A. Jain, H. Gonzalez and N. Stakhanova, "Enriching reverse engineering through visual exploration of Android binaries", *Proceedings of the 5th Program Protection and Reverse Engineering Workshop on - PPREW-5*, 2015.

[13] A. Damodaran, F. Troia, C. Visaggio, T. Austin and M. Stamp, "A comparison of static, dynamic, and hybrid analysis for malware detection", *Journal of Computer Virology and Hacking Techniques*, 2015.

[14] S. Zhao, X. Li, G. Xu, L. Zhang and Z. Feng, "Attack Tree Based Android Malware Detection with Hybrid Analysis", *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, 2014.

[15] Z. Qin, Y. Xu, Y. Di, Q. Zhang and J. Huang, "Android Malware Detection Based on Permission and Behavior Analysis", *Proceedings of the 2013 Fifth International Conference on Multimedia Information Networking and Security*, pp. 915-918, 2013.

[16] S. Yerima, G. McWilliams and S. Sezer, "Analysis of Bayesian classification-based approaches for Android malware detection", *IET Information Security*, vol. 8, no. 1, pp. 25-36, 2014.

[17] Hyo-Sik Ham and Mi-Jung Choi, "Analysis of Android malware detection performance using machine learning classifiers", *2013 International Conference on ICT Convergence (ICTC)*, 2013.

[18] R. Raphael, Vinod P. and B. Omman, "X-ANOVA ranked features for Android malware analysis", *2014 Annual IEEE India Conference (INDICON)*, 2014.

[19] A. Apvrille and L. Apvrille, "SherlockDroid: a research assistant to spot unknown malware in Android marketplaces", *Journal of Computer Virology and Hacking Techniques*, vol. 11, no. 4, pp. 235-245, 2015.

[20] S. Chakradeo, B. Reaves, P. Traynor and W. Enck, "MAST", *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks - WiSec '13*, 2013.

[21] H. Chuang and S. Wang, "Machine Learning Based Hybrid Behavior Models for Android Malware Analysis", *2015 IEEE International Conference on Software Quality, Reliability and Security*, 2015.

[22] S. Pai, F. Troia, C. Visaggio, T. Austin and M. Stamp, "Clustering for malware classification", *Journal of Computer Virology and Hacking Techniques*, 2016.

[23] L. Apvrille and A. Apvrille, "Identifying Unknown Android Malware with Feature Extractions and Classification Techniques", *2015 IEEE Trustcom/BigDataSE/ISPA*, 2015.

[24] S. Jadhav, S. Dutia, K. Calangutkar, T. Oh, Y. Kim and J. Kim, "Cloud-based Android botnet malware detection system", *2015 17th International Conference on Advanced Communication Technology (ICACT)*, 2015.

[25] T. Oh, S. Jadhav and Y. Kim, "Android botnet categorization and family detection based on behavioural and signature data", *2015 International Conference on Information and Communication Technology Convergence (ICTC)*, 2015.

[26] S. Jadhav, T. Oh, Y. Kim and J. Kim, "Mobile device penetration testing framework and platform for the mobile device security course", *2015 17th International Conference on Advanced Communication Technology (ICACT)*, 2015.

[27] D. Kaplan and D. Kaplan, "Google employs Bouncer to cleanse Android malware", iTnews, 2016. [Online]. Available: http://www.itnews.com.au/news/google-employs-bouncer-to-cleanse-android-malware-289242. [Accessed: 30- Jul- 2016].

[28] "SNORT IDS/IPS", SNORT ORG, 2016. [Online]. Available: https://www.snort.org/. [Accessed: 04- Aug- 2016].

[29] "Android-x86 - Porting Android to x86", Android-x86.org, 2016. [Online]. Available: http://www.android-x86.org/. [Accessed: 04- Aug- 2016].

[30] "About Santoku · Santoku Linux", Santoku-linux.com, 2016. [Online]. Available: https://santoku-linux.com/about-santoku/. [Accessed: 04- Aug- 2016].

[31] "Ubuntu Server - for scale out workloads | Ubuntu", Ubuntu.com, 2016. [Online]. Available: http://www.ubuntu.com/server. [Accessed: 04- Aug- 2016].

[32] "Apktool - A tool for reverse engineering Android apk files", Ibotpeaches.github.io, 2016. [Online]. Available: https://ibotpeaches.github.io/Apktool/. [Accessed: 04- Aug- 2016].

[33] "pxb1988/dex2jar", GitHub / Dex2Jar, 2016. [Online]. Available: https://github.com/pxb1988/dex2jar/wiki. [Accessed: 04- Aug- 2016].

[34] "java-decompiler/jd-gui", GitHub : jd-gui, 2016. [Online]. Available: https://github.com/java-decompiler/jd-gui/releases. [Accessed: 04- Aug- 2016].

[35] "Download Android Studio and SDK Tools | Android Studio", Developer.android.com, 2016. [Online]. Available: https://developer.android.com/studio/index.html. [Accessed: 04- Aug- 2016].

[36] "TCPdump documentation", tcpdump org, 2016. [Online]. Available: http://www.tcpdump.org/#documentation. [Accessed: 04- Aug- 2016].

[37] "idanr1986/droidmon", GitHub: DroidMon, 2016. [Online]. Available: https://github.com/idanr1986/droidmon. [Accessed: 04- Aug- 2016].

[38] "Wireshark · Go Deep.", Wireshark.org, 2016. [Online]. Available: https://www.wireshark.org/. [Accessed: 04- Aug- 2016].

[39] "The MongoDB 3.2 Manual — MongoDB Manual 3.2", Docs.mongodb.com, 2016. [Online]. Available: https://docs.mongodb.com/manual/. [Accessed: 04- Aug- 2016].

[40] "About - VirusTotal", Virustotal.com, 2016. [Online]. Available: https://www.virustotal.com/en/about/. [Accessed: 14- Aug- 2016].

[41] Z. Ashraf, "DIY: Android Malware Analysis - Taking apart OBAD (part 1)", Security Intelligence, 2013. [Online]. Available: https://securityintelligence.com/diy-android-malware-analysis-taking-apart-obad-part-1/. [Accessed: 16- Aug- 2016].

[42] "Dell PowerEdge R610 Specification", www.dell.com, 2016. [Online]. Available: http://www.dell.com/downloads/global/products/pedge/en/server-poweredge-r610-specs-en.pdf. [Accessed: 19- Aug- 2016].

[43] "scikit-learn: machine learning in Python — scikit-learn 0.17.1 documentation", Scikit-learn.org, 2016. [Online]. Available: http://scikit-learn.org/stable/#. [Accessed: 21- Aug- 2016].

[44] "NumPy — Numpy", Numpy.org, 2016. [Online]. Available: http://www.numpy.org/. [Accessed: 21- Aug- 2016].

[45] "PyCharm :: Docs & Demos", Jetbrains.com, 2016. [Online]. Available: https://www.jetbrains.com/pycharm/documentation/. [Accessed: 21- Aug- 2016].

[46] S. Jadhav, "suyashsjadhav/AndroSandX", GitHub, 2016. [Online]. Available: https://github.com/suyashsjadhav/AndroSandX. [Accessed: 21- Aug- 2016].

[47] "About IPVoid", Ipvoid.com, 2016. [Online]. Available: http://www.ipvoid.com/about-us/. [Accessed: 21- Aug- 2016].

[48] "Essentials of Machine Learning Algorithms (with Python and R Codes)", Analytics Vidhya, 2015. [Online]. Available: https://www.analyticsvidhya.com/blog/2015/08/common-machine-learning-algorithms/. [Accessed: 05- Sep- 2016].

[49] "Cross-validation (statistics)", En.wikipedia.org, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Cross-validation_(statistics). [Accessed: 21- Oct- 2016].

[50] S. Darcey, "Learn Java for Android Development: Reflection Basics", Code Envato Tuts+, 2010. [Online]. Available: https://code.tutsplus.com/tutorials/learn-java-for-android-development-reflection-basics--mobile-3203. [Accessed: 22- Oct- 2016].

9 Appendix A

Test sample set result data:

Ticket_id :: 13002| Hash :: fc48ce80abdab553399621e94bed7664a79fe51500bdb6e75b1923a8bd854c97 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13004| Hash :: cc460a4ec9ecdebef2f16810f775d508eaaab7ac496db154a5b470b1fe909bc6 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13005| Hash :: 30fe9bea4fd5063ddce7ec6b5632aed113cbfe7903234ade1db06c1eebde3447 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13007| Hash :: bdba03cb55ad08d998b4b00bb5df7b6254ce28b6a0fbfbdc0e7ea20b0a05ec53 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13008| Hash :: 5f0bbd247d78eed18333b83828b2cf6e840480857422ec933ffbaa5db4676004 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13009| Hash :: c9b2e5ee8ef9a9a3c9521286406579f25a985ef8113b54310a07c07055ce683c |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13014| Hash :: 4fdca175d101ef41a94412bf582ed2117d4b939de39a491971db278c8c1b63e8 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13015| Hash :: 1146032aa497a96f6d98106fd36114229d563ca8de6d38d8189c9ea6f57c0e9f |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13018| Hash :: c7e918ccc41b9a244e6d46f8b384744797bb38d15b5f4cbf4bb431b9573e7d86 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13019| Hash :: 136443d3b65200a32f62cd496dab11ff7f84cc9d94c08fb4bce1bcfb421ad275 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13021| Hash :: c6f2ad2b248bf928037a9f8fbd5f38680ce482ad55a954c7fbf2b98a627f8759 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13023| Hash :: 62c41f31ec7473cb2dd6310e68636c7f673ad6118093318bd45ea7289294bed8 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13024| Hash :: c21537da1fef6702bbf61b51d8b5bc2cfe223b58926b1d2ef426a6172296ae57 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13026| Hash :: c9eb4f7267409ca252296aabf04e0ecdf03f2beb7ceee2801ddc6546e295b6bc |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13027| Hash :: 087201acfeb561b8cb4e14c7ec6ae847c962aba45e44f15b567ed43c1fdb2af1 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13028| Hash :: ca5632a3a313875af76761843e72ab744888c0d1c4ae91f88ef1b854567bfd8d |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13031| Hash :: 64f2a03f57d33341833aec900d3c2b258acdc47edda177a9b82be66429b97d5a |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13032| Hash :: b9b7961e4fa943795eeae65173493f9b62eff26394ab1d76f24f5706da821857 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13033| Hash :: ccb910c869dd5f595d8aabfd2b44ed0f9788d6797a13af97d9d0f83ecbe9eff6 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13034| Hash :: 22df0b592f8cecee2f3690c11403bd9304d961882e0fcdbda3ec2ee69451987b |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13035| Hash :: 014065f379f1d47dd920ab331113aa90b2068d123f5b47cd8c89eac61bc47b62 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13038| Hash :: ccc3c7fbdf3a215a5e12dbd380becdae0af77714ca229396213ee0faf5311e54 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13039| Hash :: c77b0da14f75481c1602182d5348baca3392c50d22d9b14bdb491926b61c0abf |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13043| Hash :: 59cc054fd154285d476c5c71d21e9d6a69d5021a52602ec55a627d470e9c2039 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13044| Hash :: c71464d1f12f269c8e733bb0e8d4fe937e2015c09b8eebd8d9584cd262f7a228 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13045| Hash :: ca00da13f4dd9d20898bee4a0735276216d89a6d4a7eef80bc380a97fe24ae3c |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13046| Hash :: cada149e1ba36ba36ea953614a0233dd9f5ab6cae1b013a98bb4dbec7ba01c64 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13047| Hash :: c1becfd90472e2e710d76211ff5d5892a84f7c3bcb716b70fa72da4de55eb6d3 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13049| Hash :: 45aea60cb22dbc48e6c5f338a9dc779d733b58aa451ada15f84092a68b9f4a71 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13051| Hash :: 9e1230c5780594791cc6af3ac23514ea38c0064919df5b69b1a56faa9a6e5649 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13052| Hash :: d1d5f55b964a4551ddf527c3fc8ace305ddf9e47e1f480dc5171b3599eb90604 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13054| Hash :: cae4f1bf028421e6202ff0dbd3fcfe33cfac0acd0f36bfa37c79403da735a9bb |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13056| Hash :: 26d75f292e3a7bf62d9aa3a05042cb73beb2482701e1ff7cc7e7482df88fdfde |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13058| Hash :: c792f4d7c40757eaa3ae7b7334ca5aa982a017aa3875d29894957e5bfc674de6 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13060| Hash :: c6dbdc6fd198becb4953d208b225f1618b38cdf11a2f7369bd6c769ded8cac18 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13061| Hash :: be405bc7a704c6d0dffeb72e88d45d1eff5b8987c58634e53bfd96e8dbc5f9e7 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13062| Hash :: 1994140eab07181629e1090159e64c4d3324240d6a1677a2342e31731db91ce5 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13063| Hash :: d0a894a44435256e87401d7dc7fde4df58169fd81d49dc14a7ba3124f7ca2ac8 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13065| Hash :: 96cad65f77ae26eba046a9e7b148e8c88eedaf1ca29d2b753ea81828f5e6bca3 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13066| Hash :: 06ffd3880789ac1d08141e2823e95c0ee7d99e7488fda7a08ac18dc078d5238a |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13067| Hash :: 792ac2de4a2dd544a2176d2261b21be46b9935583eddb0e333d001d276635f8a |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13070| Hash :: 0834a50f1e53872c3140a4a9da472603dbc0032aa74e8ac4c20ca7fb91effb3e |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13071| Hash :: cae0679d56432eca2751c69c8cdaa611a66e35b1763d49881befd86c3bcf1708 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13072| Hash :: 06e0215ae28ed983631e9519a8482f01cf82b7c67b763200effafaeb24a290b1 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13073| Hash :: 5aa19d58ccda518979ecec97db9ee7a6a1de606fd661ef9238e65f3b533f170c |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13074| Hash :: c7d7487314bdf86739bf8bf5118448d8fa24560619b0ce22ac5e9ec1124a3e8b |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13075| Hash :: c82f19b816a235f8e3a8e4995d3c1bbe507f4f0162aaee511e8d4ec04dd94b5b |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13078| Hash :: c8b39c924bda8a1f48deb188c75eab06ba351b8542f5d859c0e1e27c51928884 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13079| Hash :: 90aa80c19b1f57d64384542ae0ee67db24d5335acff7c461548cd98e859411c9 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13082| Hash :: cc4f2a2fd87b1d556ad4d755372d403271577d0a4608230ec93279b3e2a63adb |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13083| Hash :: c935c5d20668633f879f44179e837aa2c3e16436349a87a0e3f0b05a229ae159 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13086| Hash :: 8f584dcdaba3eba690de8f22b6b9641f2ebe129cbed661e1f20e0a22e5a9ad09 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13088| Hash :: c72d80077c28c3b5479c6f017a4f45e9827bc45dfb2c15f18add58335b7d99cc |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13089| Hash :: 68abdebf27be552c6eedbc12ab0053c2bcd26e77fc1671d0f03e27f133984ae7 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13092| Hash :: f870666aae1019ad7d6375d2f2a30e5a892f148cdc293714258a9df5ba524cea |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13096| Hash :: 31e616facba512e5736a5f92e3b4b399b34a22658cebf99cfdaf7f2a2d7f1b32 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13097| Hash :: 5fb1033c8515f1409511f6556782c942d33654a8cfbdd84382c86b27202f6241 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13099| Hash :: 30aa5bfe84acd4a4170db016dc7a8f00e730cb9d7fe3a0ca483a45c621c3986c |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13102| Hash :: 9fc23a4e6686c8295af7147ec494f37bb1570c0c78452bb652332f97a36b3bb9 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13103| Hash :: c9a1e589b60d795e277c8812671a668821047775f85c8f10ceed0d8d593271b4 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13105| Hash :: c7360df49501334ff8b04339790e6db39eaae72c98e4dc4b0a7ef793e6d594d0 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13106| Hash :: 30ae05d32b757f1af3f28069c9e6c5fc9efa537e62ff6a0857ab6e030d92b8ef |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13107| Hash :: a5aa466f34193493238a1b1c04dbd8e38b3bdedc4e9c3a7b083ce73591f92df9 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13109| Hash :: 0fba16cbc915b65e0c6aca71c628fdd79a85aab762b57a618a78a2dacd6286c8 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13112| Hash :: c84093271930996382156ccef3136f109c6a8f2ae3d60087829acabfeadcadf6 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13117| Hash :: 018547257cf26cda09c4c940d9d1601ea9a47da9d8101aeed97f2bfa13308f07 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13119| Hash :: d2ce94c20a8aabb4d39a1fd3647813f7331d44d4b0beb34c832a4617126d3973 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13120| Hash :: d62d58304917ca8ddb25dbc76b34c837b70bdc0ae75304c54f823770335224b6 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13124| Hash :: edf606c17e34185253943b1de949b5e4bfa5047aff3e46017d5d603b592da863 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13128| Hash :: 2e2aa59e1f209ef57e27a2e296c87d31e9b4118be516a6cca48cac07a54f5097 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13131| Hash :: cace2ffc4707dd5a2c309e7393b9d825455602c1992ffddec4c4cb93a2447cf7 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13135| Hash :: 6093d381dbc71e81e97ce1b828ea78ce2602ebfc3cf91bd9ce57a50e8620b22c |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13136| Hash :: ca81453c4caed849bebac2fce3cf54a774c0105656062c3919f6fa0abc4814ba |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13138| Hash :: c90a5fc02c1bdcddd2c314ded91b1edc080aeda38a5bcf4e3c2894e31825979f |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13139| Hash :: c740955d375451e433a5e0ce64b44c8ebbdb4c6965468e47bfc26778bd1230a2 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13141| Hash :: 9121f1a39bf195bc313a63dd21d9ecb1fd3e4edad07f6affcbf14ec2cf6a8146 |Prediction :: 1 | Actual val :: 1

Ticket_id :: 13142| Hash :: 87bf02dc36ed4f2bd0c5000abff9ba73c0a32d47edd31cecce0dfdccc6832c5b |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13143| Hash :: 8b95cef2685389ff015201a9a1efa6864635c0aa9f13c371cd8713ceca710023 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13144| Hash :: 6434448b875d9fc3be0f58c3ebd3cc5413227028bc898c679bd55fd9fea160a1 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13145| Hash :: 0dc8ce5c31f45ba232af475e4e952eee67f904cfd9cd57996af9a9285cb8c1a8 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13148| Hash :: 9529df20bbfa8cedd19e4d918b1c4ae12410ef73c149ff3b5b0cafaee5e56e0f |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13149| Hash :: ca51f2d36818415b116bc041911852a11bb12fdb5fb90ee861adf78260572ac2 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13150| Hash :: ed890e654f184b3313531c7810331eee0fa4a28e5e5daa14f35abe187692affe |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13154| Hash :: 7b8375f375fbb62ffecb80decd8072f7090e5c524a8b8e0d5617c048ad37fb81 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13156| Hash :: c75e560162fe07dae762c5673cd578b9ffb9ab6991b47c6592d3df0390785fd9 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13158| Hash :: 1e5e2d0ed69ec2a2e23c83ddc33a77b28b0b28464c988877907bdfc28f7af7c2 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13159| Hash :: 85a93e1155430ecebf16c68ee8e5f32ec7ed72441691e3117a641ab217e20ba6 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13160| Hash :: c98fb3012c5156e66768113f425b2adb499419ad7289a9a378edf6065b663028 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13165| Hash :: 9d3f984f1e43fd955dfe0c04592998dfbee6b3ea4ab4d1b42f94815150abe062 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13166| Hash :: 6203cc2b175c1bd9a8a809c7cfe6af9b6554e9f0629ccf5195557575892925bc |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13171| Hash :: e6d228a381fbb5c944db4e4af145cafbb436d5705ede883b41e606dc469f74a9 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13172| Hash :: c26b1db4ec74863aedad22616b555e35c25174b57f60efdba23b13d1af064330 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13173| Hash :: c8319035a211236d5b94f91853a3424cb4079044b4af8559272535c6b72e549f |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13174| Hash :: c9d84e273f9775ad5f7bd5cf6ddcc9bba36890ef7cd64b392e20560e3ff004f4 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13175| Hash :: a498182d3b2c0dd02cc051e67e90c2f2a8869a94df00c610655abde03fabd128 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13177| Hash :: c97afcfeea73327f86109b76fb444253489f67709a399e9cf8301fd35d831ff1 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13179| Hash :: ca79ca1ac41b7dcbb045219f1b57f1cb5db20aca57cf9a468a8e0faae7bfb10e |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13181| Hash :: 6e625a18a84826c1e77cefea728bc21bffbb2a0db0ed4bb33c66f58be1c499aa2 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13182| Hash :: c9acc2c6ec44f6ff555eb0369cfbd33d1cd7205d230d142072caf4bf1b3c08f7 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13185| Hash :: c949f8755eac881ae3b3636f7b64b943efad6a8e0bf72add74db1e9b991a149c |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13187| Hash :: c79f478c4653d1f7d875030d3498ee31fdb8ada6986d762f761407b218070d5b |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13188| Hash :: c097bb9d830f2ecf1148737acdfb7577335e0adb0d0d8977be1f03b0605fbba9 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13190| Hash :: c97df17fcff00ea4bbf201ca3fcbe1f7348c740e9255befc64f6fef5ecd751fb |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13191| Hash :: a75ee0b0a3e7218c8a947e7d4eaa8a67477b6cc569a13b99224cc4f1d741e9c3 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13194| Hash :: 16250445cbac3c527499670ec83e9cf9964335ddec5a46d6e4739a444ad77f2b |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13196| Hash :: 9e187be9efa0bc84ffe73ac96c8051296ac9df09cc64180a4f73067445cf50bc |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13199| Hash :: 66c2e8026dcbb24e2083583166bff5e95deed4b1a4931c585401c6a0516ff880 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13201| Hash :: d6cd2931aca245bede61b12c28bcc625ce7ae4aec29c2a4eda0fc6ed415edd10 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13203| Hash :: c7ea852ca302029c0f835de9943ddb111da9e67087dccbd7e154d916e676ce63 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13204| Hash :: f4e0651136465528412 59bc98219b2fa4ddd26af734264537bfd97d3504c495e |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13209| Hash :: 36334bf29f3670487fe767c41d5ea49d1c249190a56253a6115c96b1f7b29954 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13213| Hash :: 12f57f478f1d29b25c4709140ddd1b024467146d5ee72984fc38f16ac2265c43 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13214| Hash :: 26b111b01485fe4ebdfabf1e7d5ad3f28f38542942ca1123f5d1211aa56d7457 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13215| Hash :: c7ae98089be4d64845ae114d481f4649fde391f4d6249e3d5b5f01fcf3d2414f |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13216| Hash :: 369905cac0a7e802ef897939caa326b14acf4bee6ac5409fe74a0398bcef7c9d |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13218| Hash :: c973787cc74d01aad38e986c840dcee00524b26fa011c0643bb975f1792be521 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13219| Hash :: c8cf6cb557ffda56ae785b04493cd9f0a6f3fa8cf50b2c1544a871bdff6cba3a |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13220| Hash :: 60fc5ba70b3d637e197d6ca242c7ee24cdd195f93096f3930acfb73aff92177a |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13221| Hash :: 552f3777ad038306b0735dad8126f78c1e4405b3692c41a14bde84ddd911b749 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13222| Hash :: c9e8f3d4065baa89592ceb8085a0ac96b743761502dae079b85ce71bba9644ff |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13225| Hash :: 3a6b6679eea8596b0b7c9eb1a6253e492cc8cf38201de5f2ecea83fec1e51f7e |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13226| Hash :: d1d32a3728ce418fe96cbc86ec4948660aec847643a045e9191998dba281050d |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13229| Hash :: 98633eddd37c3bd9877531491c5ab996e0bf041c3cb7db5ab1036ac2359b92c3 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13230| Hash :: ca71d010b0522e73b24807f0c98a3047e43731dc826230dd09452c212d7dbf17 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13232| Hash :: cc5c6ded37561a5e0e5d94d2df07b354a5d77f4a6479cf11a47cb88ea25fee23 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13233| Hash :: 2f5e3e7b1d5c40e9ff7030305e22b6aa2ab0174ea074c9c2f217ee828620b28a |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13234| Hash :: 7aff63e11abe2f1dae60225bdcbfc038eccf8eebee89c3841f2bb6ac1316c932 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13235| Hash :: ca3af7542eb7984c470a6ca40512c515174082ab9451908641618 64a951b5c17 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13236| Hash :: dd75e6e22a5f2176c1a26bee3e9df821493cc35180ce3351d56186dbc690efe6 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13237| Hash :: cc11db192acac72e1f927201d34c744c98dc947dd2d01d365898f8475a322eef |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13238| Hash :: 224282288637b643dcfec31f925d18b6f66bc28f47a40011dc3b3af36a78355a |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13241| Hash :: c6fd9094571712d4c581f89a7b137214a9467238a6f7aac5ac2d0f21fd4a605a |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13242| Hash :: 0c95345ffb2eb5251757812db8143dab4424bc95bf0b46584571e693c9413388 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13244| Hash :: 5b64a800ae2adbdb8f88872e18d45ea0d719e4f006c338ef41cfbabbc855686b |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13246| Hash :: b9fb60b50ae68335639d9cb270e57a716efb3cac7ae81e506144227321db71e7 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13248| Hash :: 706f86f614ad5e2c909ec49f16a7d5519958e30030124964b5921d2924c5452d |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13252| Hash :: da81636c1b187e013b3b8a61bc8ab7e1bfa2aa45aebb79ee943208a91bc0d5d9 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13255| Hash :: ca6b95002d280ea78b777108376745beb4d2124dc7a838cf87be169e56a2b7cb |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13256| Hash :: 1a6f4502ed7e14ef98842c7c4ee65476ef42d191fd10ea265c201698e9a52253 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13257| Hash :: cbb64c5bbaea0a8051bdaf4070a641bf0888c6d4d159031305e103f1e63fc8ea |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13259| Hash :: c853560050bdf0b2b303bdb4e7589d81511365b06dccfcd1b53b6c771d260b76 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13262| Hash :: 3543f4efdb4e64705c1e694ac2ee4ac3c31f8709afb7ba8b283bd6a438150088 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13266| Hash :: 020b575d0f24652642b3456b0d5b4044ffe172316fd8b84508bcb22367b22b7a |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13271| Hash :: cb5aa3575998bd7ae2377e8c8e8085d06f89dd4bb9f0ade2d55666322edc751a |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13274| Hash :: c8a50400e482881b985fd1776a5edd1bbd5290111ef9535139ad67b3f583502e |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13275| Hash :: b941cb6156793decf657f3fd01d81934b0a37535889e2bb4ad6bd3e78c181a0a |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13277| Hash :: 109f056a2694064f96d1eb9aa7a3f10ec941a5b147a115cabc604b6de1a8c86c |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13278| Hash :: 137832555be7b905cf31cedd4cbf0bf011dbf5067d79164d0ccdcc7feb8f249a |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13279| Hash :: c71306c144c27c616cd925f19fb118ecb5016c324e9bb324a218fc39d2dff7ca |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13280| Hash :: e1366693b703d2e057a9e81b10bdd00642c4cdbec397e3f0b55224d1fa3f5025 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13281| Hash :: 380061bd4833624e069904945009e5fec314319aa4e2e94617034663a1a81ad1 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13282| Hash :: 4e3e8c4d3bce8c16ece4a4bce3d3379e5db62c92541281677c5296d973a8f984 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13284| Hash :: 6f264d59ada8e8f1e1f46771b832e8f1940f4924d81fecfac4d8379ce747c2f6 |Prediction :: 0 | Actual val :: 1
Ticket_id :: 13289| Hash :: 7316db3cc57eea213bdb7c0fa950bbe06344f819349e26da47b82d5cea3255e9 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13293| Hash :: f70e3a9ce797669b59ddbeb778c5df96f26b8b952c2f2d39ae1f38cf4e3191f7 |Prediction :: 1 | Actual val :: 1
Ticket_id :: 13300| Hash :: 714e350fca01543976902eaba6a05f0f82936c6918d29f99cc85c82f99c7d47a |Prediction :: 1 | Actual val :: 1

Ticket_id :: 13303| Hash :: 3e90e36cf760c6d3606d9d979d1c897456021fde43fe2047f10ee95907417295 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13305| Hash :: 1774931ec9adc817db0a7d95795ab9a8971582d722738ceeb2c898056159ee4f |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13307| Hash :: 1692de232380168d92a9f8310bd99d933e90832e4bf5c90070f941fdaf984819 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13310| Hash :: 7e02833138d08ec013b1aed86dddf5dc9f05c6185d565093c432f5eb6661dfe7 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13311| Hash :: 8e30460dbb769e36fc0560b302b9976c950fc6dfdb7a029c53a20d6abbab9045 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13314| Hash :: 28e4c80ac5df696c5cb1d86d98a63c4a3ede6b0f823930e340ff1d7b108134d3 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13315| Hash :: dcc9b37b7308b2b1ec79ad54ec39129baa8d4b8a99bc0f5ee08097efd3b544d2 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13317| Hash :: 1f213eb36b7608ee3e77d71d4e355c7ae1bf62d38cab3e0c7b728632adc2f40d |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13320| Hash :: 1cc163bdd317319bdf985cfca4ec23b988e7480d340a6dc9dd33b28bf215a917 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13327| Hash :: 4765d1bb39698dfc47f7ab6e0fc3a4afe7b8505239ff2ced067b9585c6a2484c |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13328| Hash :: 1cd86c9e0454ba01ce49660f6060a6172e6532dff595d68e5ccd7b8f72881511 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13329| Hash :: 6280cadd5cb7a432679bbd976a89f3b6e4b9c88ee22cf6fd8a2747bd5428310d |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13331| Hash :: 598b4a635b92eb4bbcb6754481128737cf2cff3750b878bacf73c81f0a4dc0d4 |Prediction :: 1 | Actual val :: 0
Ticket_id :: 13333| Hash :: 9ff66304c096d9a44df748f4d3f53f953ab3dbe48fd0c80ad40b7e0c6608afa8 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13334| Hash :: 9996bcd2c918baf3f39c8a8513371ebb0bfc844ba1d93428d8fbbe47021a1bbd |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13337| Hash :: 5cf639f6bc24b2808026cfd5ea18f4e749151b543f59a1de86d28ae549104dfe |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13339| Hash :: 18db6c5b2ec3c1e37617ab1387b28f7899724b05efd9b10c89af8aeda1601996 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13340| Hash :: ffac33007622b422c34076cc255873fb19700d3508b257b40f9ca64fbf8409c8 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13343| Hash :: f137b1b147e30f9c92cda4532cd34a1a3e64c4d09ccc4993d5101eb1b3e55088 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13345| Hash :: d52763666ac1ce2b9b1202a7dea5b1a26379a582b7068818dd7fa8f48a6f0062 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13346| Hash :: f9674d0be6947df86b256cc994760e989b9391bff66f7430a3601516d8f95530 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13349| Hash :: a83669c22406607a25366ca62e5e715751945919d09307fc912b7e6f7369ca20 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13350| Hash :: 49a8d7b301c0ec3bc78ed003efb88035b4eed9e28d154f349d56a48c3492670b |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13351| Hash :: 3fbb4e88f46bd149590e9355cf6859538280e66bdf1d9996ba953a2a3b513057 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13352| Hash :: e2010d58cee00a897b25a1e47692ef64a01673e4265e2daa7755a897c5eb4d07 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13353| Hash :: c278722887fdf29ea7c2b4b82f85fddf1eb2a2bac96d635a22a35f4a08f94e25 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13354| Hash :: b8936612542053331f2521d66bfe0873a1fe7539de50291228d34b230d271007 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13356| Hash :: 7c4dc490af0c84ade3035bcae5e589b8f4851e158b8d42c3a94e91e31fb4b093 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13358| Hash :: 07ed0c1a916990970e37b75f1d3383144e30be47dc2339de2ec4f8d98305660b |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13359| Hash :: ec6a7b7f46e46e98e1f4b274f02bf7d8276252f40c714cb85f59e28ad7492041 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13361| Hash :: c4162a466804f179cff7cc57edda372831ccdadd0131bbd84de10e1755cd89ef |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13362| Hash :: 9b3a4c1cf4f55ebbfa0fc8a42ce1399dba549cf4e804fc173c7a48d50e9abdff |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13363| Hash :: 31fcbae7b7e516b4726775d45214681b11eb16ef39123a01ea0d119ec8dc843b |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13366| Hash :: 5b50501455d63ecb1133f88c5f614722851c09d1e67840543e55463c397d1159 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13369| Hash :: 9b90337df07789241184b4d9ea2143a403b6a39f12b3c3cb2875de2007741499 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13371| Hash :: 3bd76e569b80e6132c54973f3a4c5b1911cb651e443e845e2fa4d2713da00ad9 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13373| Hash :: db6fa33469900dd903d08c4a516bb2149520bc406f1bcb02a2afd61c804e1475 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13377| Hash :: 85d43cd6ea4047d9ccf4a83784dfbba4428790e0c7307c8d58ed5bb416952d36 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13378| Hash :: 139c1080c124c783dc2923cf296c050317560c120b65cd06e68fbd891dd2c2f4 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13380| Hash :: 3f23cfb03a5bb944a14758ed7c89a532b94c8de1e40eea6f43b7eb83343efa49 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13382| Hash :: db233d04b82794d317e47614eb42637535c9f14016d5784dfb6b9a9b39972049 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13384| Hash :: df49cd718b79771f8ec139ab3ffafa85811785bc17af2a93ecc98b8c36e2e0c7 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13385| Hash :: db4e5155f5a988f55c366fcf04a4a3309756df1c2cc8cf027e8a7229d605b19f |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13388| Hash :: 10d24393c519fa1db39230153b1ed12e80e4b7df56e6f59c8c864a522ebf8e89 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13394| Hash :: 04a559eab215934f09a78b37d8954c4f1ebc98ad27c2c51a06902bea52623d78 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13395| Hash :: 794f0f5bf8172476dfd89f44c63a7e30e90c27ecc7da64792148d3b6b9dd554e |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13397| Hash :: f03dca56e69ea7eeac7a85aa65456b86a8030e932f2d89aa99acf333897f321d |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13398| Hash :: 7abbfe0c8303e2beb9baafad682d844d3fc71a15ab35de03e3fa6076cd37c125 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13399| Hash :: f047393ef31b2040973f6c8816c97910fc2f359d71b4cf59acd345f2c6efaa7c |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13400| Hash :: 5331546f614e7e79973761440f1458e8f9fdc7e0ae81ef7e7148b83104739f61 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13402| Hash :: ec01d0aa5ee96d889fd7acf39ca2f7858f5d50898ba085ca6b8b526f3206e4fc |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13403| Hash :: e57320eb1cd909afd30a699b39d7ceafd3149e08703f44d8d097e9230c2bbc48 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13405| Hash :: e08d736c0776034736f4060cb572b86dc4af1b70ac7d037d13acbb315683c707 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13407| Hash :: 0dd33fe1cfda2118c28762603d6d0e99caa7aa001213412746a494ca84494002 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13410| Hash :: 9934024b3d65ed96c1b99a2fe08097743f1d497b9e16df9e774009ea29fa3502 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13411| Hash :: 941bd3a0aad0a831fea7a6695d9400d63658063058162067 0ce2ef541fdf42ba |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13412| Hash :: 75ad9ed35ec009be4d2b7f51ca2e7a71ef9343b2cc4c11d375ebbbaa17450e50 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13413| Hash :: 8da06740051127c07c23f3a9cdcfd5f37532cebcb00751e298fddada3a49a06d |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13414| Hash :: 10595d3c07784dd4bf1715e5a64eb9a2ccf870a02e829e8dfda6c96b920f3ba4 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13416| Hash :: 58e47d014f305d525c7ee5c652b2c651e1b288001f9895fd4f0bbe1d3e1d8dd7 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13417| Hash :: eac001128df0a90afcad94deaab50e9ed49f2644b57fe6db7644c4f3ca189ca0 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13418| Hash :: 88025b9308dc73f6b97cdbb5303974da1accc7361ba0c5b09696e32672099cca |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13419| Hash :: d5606d95a0e4997facec5f4ce95bc477678c61040d92a30ab1df5a286af869e3 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13420| Hash :: 198d1ea284e0d52654e8a8f9fe89220cf7677f4a720e39c1afc6a26f3819ebde |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13421| Hash :: f281c59f7fe8c2c06a2df40ea0af91eff9f4e5a4c6012c525a29adc55fb60550 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13422| Hash :: cc2fd258f96f207e52c01a12939c2899d360078d85b51f1852ea455080d7a54c |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13424| Hash :: f59a6f30a207492932480712b5b6b6863e58e51563cd13be22a23a21c707fdd0 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13425| Hash :: b02cbb5076775ef548ed9a16236a5c213250d29bce3d11040a33ffff148e932f |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13427| Hash :: 49e2ec4fd15e48ebcd51f55ccc62aa77e506f921732d6b8897a937da89bc9df5 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13428| Hash :: d76653ea29ce1352c09acc6c7ca9a7a76cc98b77e6f3e56a0df533b80400b667 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13429| Hash :: b0c64369d6a3a56a3a3c569c85c2e2956d84f9244b27efa38ca43406fe27452f |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13430| Hash :: b6a3a2207270684216b4543303152640d00703017f37058ef1517c8c3a55bf59 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13431| Hash :: 02dc10b0faf551d6509a935f8c848b7313069909a8809c69ee93d31280741291 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13432| Hash :: 2bdba370438f01f00f135695a40c531c104a59cf74d29dc97c7ad26e67394ec8 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13433| Hash :: fbec8c65e1d413f1cedb4a14ab9ea581c129951cfb2e10d1b5191d8fda57f3c9 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13434| Hash :: 1ad574ffb9da4cd1358056f25e1ad2a48668401901a59d28d690b1cb4603516f |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13435| Hash :: e739f887c8a539f9296444b24e0fa573073b74f8411b04c92e48d9580dca408a |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13437| Hash :: 6c4ad806d7109e1db3444a1c1b2f13bd34ee2d29ad510dbe2ed2f556561c3959 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13438| Hash :: 9df6180ab60cb173d534dd486726eb4be16572f8dd9d52bcfcfb3478f4dbeba3 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13440| Hash :: a948f1436442e84cbcb8d3dcdd01706401a6f8aca69fef434d172535153f5174 |Prediction :: 0 | Actual val :: 0

Ticket_id :: 13442| Hash :: 4d460987be628355c7ed196ec6675a85a494c414fa4fecf34b156836ff92de33 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13445| Hash :: ab250a1cc1592fd9cef8832fbceda473ae57d46d0489d3b44d3f088088c194a2 |Prediction :: 1 | Actual val :: 0
Ticket_id :: 13446| Hash :: 1a0206a0420990e12243b60b145bc5790beae5539dd362da25f6c563bbc43223 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13449| Hash :: 4fd30683cdd38ffac52fcf505e1b181a5b8b5cd4802bc87853d62c1e8729424f |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13454| Hash :: d8511ad51351e9018c9a6a1bb596026ee8589cb6ae35439110f5263e7b37d7da |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13456| Hash :: 809c98a9e3f2e47f494e16e7da0c246915c783eca7f84ba4ec6c1b775f857c4c |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13458| Hash :: d1ab8d531d66e19e581d80209abeaf8b0021870ebb46a4462e27c65c4e1593ec |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13459| Hash :: 48e453a570971ffad871ae7295403dab469f244612775fc63740ac7f05d14105 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13461| Hash :: 788621173fd4ff65865105ef9f4a96fe30aa747d143177e3e87d951068bff6e9 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13464| Hash :: a5f5cefe60eaf4b93229a790e18b6879f7a1cf903b915174f6994bd0bdf608a4 |Prediction :: 1 | Actual val :: 0
Ticket_id :: 13468| Hash :: 7d66a6831ffcbc8fa43704ecf99ab52d5879e6fdbf44078f3fc53b1504034882 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13469| Hash :: 80e20f03b77fd66b4245918b601902c17de3c4c6ee7d4447d96107996e7f50a2 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13474| Hash :: 2d780f57a5d47e72e23ae2677878af782f9e53cd266fea624355e2ca8329ffc0 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13479| Hash :: f5e0ece6c05214d5331c104bf0cff2fbb2557a9543a6de847d70521ab36bf735 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13480| Hash :: 71f94de4976fd88b59d9fe0ea626be65c801ae9b566de7ea5dae189672f2b4a7 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13481| Hash :: e5747006810e8db4f0b9fb4d97d7c1bc6ab0e1f33eaef7fbadd7a2b2843f2aec |Prediction :: 1 | Actual val :: 0
Ticket_id :: 13482| Hash :: c6fa1f9f010dcee33ba0380b820f8728131209a92a5cc85d6c5164614cbfd361 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13483| Hash :: 10426b3c114a79228bc50fda5e93eecc6fb31df3f4c991014e5e19a352b39d37 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13484| Hash :: eb02c596e511f868c71a6192203acfd39b7ab888424f750eca73b8b15b8befb8 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13487| Hash :: 1a30be8757e616d02fe65110a672c1601889171d4a05beb363f4627d199e274d |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13488| Hash :: 2eaf89f1f65cb721094a82b27958ed937a550a8ca107fc124f69812de89b4691 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13489| Hash :: 5871bb5556662321c4ac8a6f9e28e3a5c3e711c49560b424f3c0e257ada02c05 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13490| Hash :: d266a4da9fae54c7139e50cf7da38695e38dbd52f3ecd156a13d1e498efd8db3 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13491| Hash :: 18f652f39b756050baeda608f1dabb777aa57d1af3295c63144fa44d77bd94c5 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13492| Hash :: 7088fa6e83eaa8faa24569782c51c3e5617a06f89abc359ebd54357bb3ec7d5d |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13493| Hash :: f777ab0ed79dd4d8f0de96d7583cd976e9e79d7dbef7a8fdf3e6d449b9fc69b2 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13494| Hash :: cd5a412c2876373afc3f028a53835b41b463c7921bbd03f167488f29812b3fcb |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13495| Hash :: 81e002a3a649bd429d95522d3fe7e76eab4f05722cf9a2a0c2ca9f636313d836 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13496| Hash :: 8ecbff9bf6cba775246073240fc508bc8961823ca9e8daf1b6255a7d48a509b0 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13497| Hash :: 67400a05a38a5c08cff3532a3e2a26b07f29901d3c096261e8568cb236e53f82 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13498| Hash :: 034c5978611086e6a8617efca1dc0be36a23b4e6f54dd74477e1167ccf602fbf |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13499| Hash :: ecf85cbc9411b27abb9446e2cb3c72451d5a5efdb9a0241336c7f247361f62d8 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13505| Hash :: 29956c45c241498f7fe51b641aa591bb86a46382befd2bbbdb609c9f012cccf1 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13506| Hash :: 7f95651d744c2325acaec98a1fc4e576a24f1c81692348c91d29bc697544b4c6 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13509| Hash :: e45629086f95e68cba7126c20d3a9b23b080069f44738813e7c8e726253ba8b9 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13512| Hash :: 01038be502403caea433de605be701d16f1cc6849d03d6d2f1de9fee565370fd |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13513| Hash :: edc5699402a5653175d5bb490b65ae29edf95f5bbe83217df3f7ceab7fd82c97 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13514| Hash :: f33d71345b9cabf6c690486207e01883dff87e13ff5e86d38f86b6501e82ed7c |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13515| Hash :: c3d49ac4f5d72dcb8a378eeb016ebf45cc09ac17a57378b628b5e442146eee5a |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13516| Hash :: 997343bb42b9e6085122ec9cc7de6239ecc605e2e51d3cf4b9d02e12bdec03f7 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13519| Hash :: e310d996d3a930bd8755103e04dcbe886fd51deefd1366a4e3d35b2b143876f9 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13523| Hash :: 88c32523b7f08ed7dc49b681eb95840ddf8ae28fddb852d898cdd4218fc45873 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13524| Hash :: 733d749f443a794c2a4ec4d6ade577f13e9f225cf8af551f23f52f8e1b73588f |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13527| Hash :: 7b53fd1c0aa21c6c4461d0df0cd545eb8f7c9b1949d23c22ce1fd21976ff5f66 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13531| Hash :: 16b034dc4bfd187cff4c4bfc7444b972bbfbd6c829c13f68e1d0536f640e8f0c |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13533| Hash :: 2ce12517ca791251dee6dbb8b36e1ddba6672c69fc4d4111b21891fef7b9f80e |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13534| Hash :: 347b2956eb551abebba75fd806b9bcbd5c5f8d39f94e6276f0c2abaebcff8f32 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13535| Hash :: 902e0c037e0ddd29ec7d2317c877639c37cb09108958b27abcc91c2986cbbe65 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13537| Hash :: 7291722522b1a63a8c3d2745259e59f2b87c142b9d48102b3bbfc26ec11e7a54 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13538| Hash :: 30c5a445a27f8d08d4eea94121309fea229b09ab7032d467a58e6908cc74d5af |Prediction :: 1 | Actual val :: 0
Ticket_id :: 13539| Hash :: 3dab26a55881ffc84e6bda49cf1c8e0818c227f6ef6d589ca997c678c998144f |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13540| Hash :: e5afaec49152579d9fdd154ec1e98b47a0064a0036013a391e37fcd179551b62 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13541| Hash :: bf9ca9ab2dc713d1c7b6e4b6014808f5d01100d99d9088c263e97c1d2d599dd8 |Prediction :: 1 | Actual val :: 0
Ticket_id :: 13542| Hash :: 27afc538c1a5e4e2b294ad641c654292a1e28e2795f44eb2aec599217258039e |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13543| Hash :: e9cc53c5ec31cfd663f22f8b62e54008918f38d5e2df80823a00f988df1a7209 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13544| Hash :: 968e719a9fe362f2b71674a597fdd41f4cee486e27ae93ec4b587b0fbf8eac8d |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13545| Hash :: ed640f1dcfe29a74babc2ea02580245979aa42d83c0b40d8845d23f761a6a263 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13547| Hash :: fc1109bdd7a12375fce284dc04ff8884d622dc22aa068747c029f8a0e0862710 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13548| Hash :: cb92574905df48a723f3884de253fd41fcc95a9af85d84b31c1b2b24fbfb568f |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13549| Hash :: 1eb57711bcf5afeb9df91bc144de9021295837206c279ec529a3b040edaee647 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13550| Hash :: ab8e8599abec6d7693688a34ddac82f9257a67086ef316ffa92f7e094ea0bda5 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13551| Hash :: 9ec1ab2bbafe032f55a5cb996648d3e22f2f1cf9d29ec1a564dee9e4ccf841ef |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13552| Hash :: 92a92c162cc00246f0c2dee41729ca25a2e22cbaf53949de922cb5374ccd0d13 |Prediction :: 1 | Actual val :: 0
Ticket_id :: 13553| Hash :: ea5369c83e7bdf0dca000b462f0380703f4ce4e16f97c80e907c011c36c4f848 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13554| Hash :: 73ab4f53d008952d1fb95a64333c107034878bc39b32c87ea0f8ebf62e320aa6 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13556| Hash :: db9928d576a198b580dc3bc0bd9515e4978ebb7b0da72f4525cfd6aa5dd0b622 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13558| Hash :: b2c78b7c050e0fe2dd6bd2cd7788b641b3862ad13b58d73daaca730625875f5e |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13560| Hash :: 06fb8ecb5239593f31eb34011cb10020e07d0adacb9721e8a6d455284c718183 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13561| Hash :: 6c9a18e8b18ec17718945ceb731f19a1b18dded8e89a66b2d965ada743bc1432 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13562| Hash :: b82a93900b3338faadd490aef5d8878b4cdff88ce860866d64935e28a50bc16f |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13564| Hash :: 1cbfbae9be7f1858a115e3d0e67217387b6ffc5077f0000ddc94aa8ed73301d5 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13566| Hash :: 3cf7cb9c158c2e19efb2e463698ac41fef86babaf11fbed28afc24f289e9d882 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13567| Hash :: 4a0bd1efdb5e96624ddf9c5fc92efad89fdf9085b62b7cb465f87c5db4016cdb |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13568| Hash :: 36474406eb8c5b2a4db731e75d6b7cd1d817b8e931b236ac73943a38a2a69d70 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13570| Hash :: aa7a6b030c3e0f17f0bc08108e5059eed440cce31ef7871e0a9cbdeeebf46d2d |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13572| Hash :: 1cc82fd68ddcbda1db91c6378c3c9b8df63244239fb4b16640b2b6fb83270188 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13573| Hash :: 780f364a7deed4caa4adb27f9646f04e4ff390c147661adf1043bc4a68359ff4 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13574| Hash :: d480ef688c4955ace53c412c220fc2a3485e028a17a26785ca96d1d2a0f9a47b |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13575| Hash :: 450b756cb557b196386486cd417f7161883580083a4e1cf35373cf86a6ebf01b |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13576| Hash :: bc71ab47ad836f5efd36178bffe04dbb3fa92d79f443fe6d96d759a7d5e88c1e |Prediction :: 0 | Actual val :: 0

Ticket_id :: 13580| Hash :: b4fa3c33cbde230fe8d380077c466d8381c5e51abacc238cea0fdeab85d5a23e |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13581| Hash :: 8746920b74540fed19405b7eb74d3a59d202bae7c7c63d8a6625f9aeb7d54151 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13582| Hash :: 4a42ce21cf1d8920302a5e351a3e8731e0c1855afdbf0b495406668f5b40b18c |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13583| Hash :: 2c65aaa41048cfe4a73f37ae41b23132ac4f1bceab04452edeedcc2c7478c625 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13585| Hash :: e9b88a2d4fc78bf5fc76d73c9a758cef7eb4d2770062200272fc6b7da4e66c12 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13587| Hash :: 0b29d129e20a08730ae90f59e1588d12807e0fa2e0dcebc6a04cecaa1078c7e9 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13588| Hash :: b34e9a57d66de83d65f6d323dc27e4875c27e64054c26eb31bb7d56d6170d8af |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13589| Hash :: 4c3b9dc3294cd6e991daa512aaf0cc7363834c7d3f4e2ba24bce762658c2d7df |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13590| Hash :: 8b731027c557affe5b7701cd270e4a8e202e26dba901ea3cfe82db47b401c448 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13592| Hash :: dec371f63f7a5ca1eb78ec8b7b6921ca2699038d014868fe0c1a64e5893ea1c5 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13595| Hash :: 4e09da65078cb69ed5560298f5d81f3436ad30cf215d95f6cb151f22553c5528 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13596| Hash :: dd326397a6b156428c35f851a92bd427b95cef0c374f58f69472de3997577c4c |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13597| Hash :: 406dda3cfe89e5298e2e9043a05abe3d94f3d7baf471d9f72b98a75a38576335 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13598| Hash :: 2411ef63cb0dab056cc728222e0d4212ed35af6557231314ca572f84b13ce51d |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13599| Hash :: 4334dd973180b7edaf775e741a333194a5c4564857484928db75c37e9fab7beb |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13600| Hash :: 8e2512593c80856ef19758ed139487238102 79cf9ad06ca4a97b9de98c3484fd |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13602| Hash :: 77cade69c0bfbd7802cd184efb19375194296bf215c58d0cfb1b585185642830 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13603| Hash :: 86c87e3dc3afaa9e913c8b2864dd416775236d465f574cb865b480acd79731fb |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13605| Hash :: ebb875da699c8bb7d6faf9e6177204ac9cc4f8638185e8933fc12d6dc51bb687 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13607| Hash :: 70d0a20a6044d01f9c30944f7badc01c0562ed03b12211bbab0e629ea12a4183 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13608| Hash :: 79dcb7856fa20f22dafe299989e37cebba50173e598cbc7128f0a7cde6306ab9 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13610| Hash :: cbd08f9ee663d3dc4c014f42d2883be11c8afd4831e4935a245a29f81fc2c1d8 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13611| Hash :: 9fc2a2b91114bdfb7867edb8e713bde9799559b6566650a50e04b786daaf4196 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13613| Hash :: 05aae0a0e11b473b6be4d81dd302ad7f26dcdb0b6ad90c259ee3a4590b73ee37 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13614| Hash :: 518ddc0e74379ac65c1397c0dcb21dd83553a3e81d40a7db12ecc2224694b95c |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13618| Hash :: 00ebf7aa2cd37161e438276784f2f73068b7cbdd8efb17329aa1da5864e29f51 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13619| Hash :: 42a02047511a16dbb2ea31e1d9a5b55b7071a75b26229ce26a74ba739b859d1e |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13622| Hash :: 6c491b74579119baf2f8390237ed9056106fa239ddde1010cd92b271243a404a |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13623| Hash :: 2d67408a79e7cd4af46b2ed92001159bb500a33a14e26c0602b15ab45540c575 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13625| Hash :: fdb7bc15519878e7d5dbf1f10253ee9f179ed5c0de50535cd6e60e04e8229c4a |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13626| Hash :: c6ffc7f6f8b18df387a570971c4d8f8b75e31866af1d2199491a2da37567c8b0 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13628| Hash :: 68bb34f4e504b700c11f7d73452a1806a1d473c3eff6535d45de341b1c88044b |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13629| Hash :: 6ab65982a1280c6c45d62b0ffb2dd2573d3d862e5ea36891308778a54c2d8beb |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13631| Hash :: 1880917c5751afe73831bb6626fbdb5152dd8fa38cdbd8deff76164254c3314b |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13634| Hash :: 2c40c6002e4869176f6031f296d2ebef1d8b0bce2ff95881cd446f9e03539464 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13635| Hash :: 682b6c7c4a0c8b69675add6effe0bbecfd229f5497f4fe587473aa047eabc685 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13636| Hash :: fc7d02a8068e2efdc7e2eab2d9418f6ebcf65665825bd0fe4e76309234815f25 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13637| Hash :: e34208573c6eb2f101074dc451a378a3fce9ab89f4fd23ee7a8bccd20638516b |Prediction :: 1 | Actual val :: 0
Ticket_id :: 13638| Hash :: 3da6330ec36cf400d34d5a81721e60a9b828d26879ea2c32fa9f9b35d8a2a546 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13639| Hash :: f322beaa5784e07b8e7d06f824cbea87ff2218437bdfd700d71b1c27019024c9 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13641| Hash :: 7588a0128b2bfc60b6062d2747fb3ab7f4143c71e9754b0a8730883ca76bbe99 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13644| Hash :: 1438aece19822328f1083ceaef5cd322047c36a404046bcc9459d5cc0e84e4b2 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13645| Hash :: 3263b547be6c1af4af12fc0e22613916b453ae96658d3e357aac3b433ea0e3a5 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13646| Hash :: 0e90720a7addb9302e77e5297a4b406ea784885a329a5737c690f81b8cdc20f1 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13647| Hash :: 80d96fcb92bb587f3433d3823300daf2ad875eb9b53304d84f8c5ca838c30e26 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13649| Hash :: 920b31e4b40e3091634dd54528e83f7fbbb78d311a6aa01d179bf3d74695cab4 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13650| Hash :: 9504fe262b629a94a46245c91fba91c95fe8b44bb88585f6146624594e3b386b |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13651| Hash :: 6311d76e917208347c3b33001a9607f10cc47699445b598caf0d6cb11b2f3ce0 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13652| Hash :: 3334f6569e05df22b538741a208b5ab0954d6977878d22478eec4120a664e939 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13654| Hash :: 52b3f97eb98dd65f2a6f53a581df9a1454799ab3e9965a6b630bab48dc77831a |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13655| Hash :: 46f089369bd407181888246e1c105c71126b92b3f8a14bb5f34a5b84408266b6 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13656| Hash :: 540c5064a46d34e3944cff164f0f6945c47f199405ffa644a55f3cdc7c079bc6 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13659| Hash :: 40fd67d4a466d894f9ebe8649a20a3a22ae34c5687316ddd66f6c6340693f66d |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13661| Hash :: 02957b2002a815e2ff33ad19c983c7387c6ebbd96b95abd2af7f3ff00a1e160a |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13662| Hash :: 890e6052be480b61981e9fa602215f12bfcfd41e99eccc9486799dc71241cf10 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13663| Hash :: fd26ca1af59096bc373611c555d3aa1c8cae12f348e23a69e36ce205f29e4f6d |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13666| Hash :: 0ec8c89d49a4c418621ee88ed8722b6921fb56478770499100 48c6046a1fb9a6 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13667| Hash :: ec8247d1721cb04d28f2efed8ea4581efcf3a7678b701ea2740b4b3003d68d39 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13670| Hash :: db71ea56b97c6f2e5e2d7e8b018053e35e1119e710b2f67edd562029fa62ad45 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13672| Hash :: 32bcf486c989f140d6496c24bad3de59bbe438e20516feac74644bd8562e97aa |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13674| Hash :: 873dbd3ef902bb7478b692f9d415a7565bc4d37c79750b95b907b5d47c99d9a6 |Prediction :: 1 | Actual val :: 0
Ticket_id :: 13676| Hash :: a9822d7d7a45d82fbb2e6afa488f9ca869cfa4e19a8876f488f3fbdb0925fbc0 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13681| Hash :: 06e17d07c49d3b93fe84bc641d7607fd61ec36a7eb5ea9aece275e31cd85eabc |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13682| Hash :: fde38705863198c5d44be0fe48ca59f3a600740adcf13347d997d66f0f05cfc8 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13683| Hash :: d47353df4baead38102761026ae00f0fc93b2eac477ce002737ce7a8371f8c8d |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13684| Hash :: f7caec54d03933e91e6a6af2f68713c2361f2988db8cbfaec688c724820fe93c |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13685| Hash :: 025d85e3d971cd9e97e253451e9790fdbbe3a6bf08bee2eafe3c848284b281c8 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13686| Hash :: 2003cdca3cabb9a07822a77ee5a6aba62c64a9551c6e1c2e6e5babded887d1ee |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13691| Hash :: fe515a2928e832f1a74e97de90a43d362d1c145604b09acd9de349d5c9ca8b71 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13692| Hash :: 1fb8482d3dcba74d91a8acdf3395d4416067951a876f70fcc6599c9aacdeb4c4 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13694| Hash :: 112971a398f6c679024eec033abca88dafda335e1d24867c52d19f725debdd92 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13695| Hash :: cc7a08f71e62fe2a59fd39bcc10f78c2bd39fdc35fb53a6acaf94b217a7e052c |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13696| Hash :: 895cc11640f486aed34e76c5d4193d32acc0621fe6154e422604869c5ee5d650 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13698| Hash :: d923d5583af9ee695df72dbf48cbf1bc62773a4b9757b7cc06510b3f110288f8 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13701| Hash :: 871607b349b03163899602893605c1a124cab4d0509a7316721887a31a43ea1e |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13702| Hash :: 4e72a98ac44c8d3290431eaeef02fd09eccc14f04d7be79bc54160e6ec104dbf |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13704| Hash :: ecba6b2073de9d77b3ac55745aba6d5a6b352534f086a8b2b33b475e72ea6dd5 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13705| Hash :: ea5e14d81d5467ea5cec11096c5e7b56aba5ecec1aee5aec0b15044367f8efd8 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13706| Hash :: 65e89be2ccbe5d5237be37392d6466f16e352f1453952b49cab87329df72485b |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13709| Hash :: 59b68792fb358f1d870413174297bc42a9c3cc3eb1a2d733c95477f4caadbcba |Prediction :: 0 | Actual val :: 0

Ticket_id :: 13711| Hash :: ae5594b7199c3a1bd9b7168de58735305ff5b094583b967f3efaa3766a80d303 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13712| Hash :: 5954fe89aa596744d5be2c34e47abef56d33265ab06a99c68ef37f9ff859df84 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13713| Hash :: 7e224a90861c7932163874671a4f2ded759a55c7d015c1a337a1a1c819027cef |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13714| Hash :: ab875a6c807b2e087312ee7fe51c6b8c5cfc4697da46365618f90719ba88d84f |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13715| Hash :: e4c035bc3fd34f64527c4b1eb2801705847316660f6fc08dbec7956fb9dc99a9 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13716| Hash :: af6cb9fa41d6a33820198617db0e56d8d70c9fe8075462bfb5a4255563472e34 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13717| Hash :: 95c2ad42c790844aa9d2cc92e4b8639a7ca3d9e0044b4c5cef79de1dc4ae0c4a |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13718| Hash :: efad5919d6d0de536743647cad0e0fb8b94fa4b754a535d4f7723bc326ecffc2 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13719| Hash :: e0bf01ab94960115dd236fc5dd584b43d4aeb0591ab0e757f56b9f8d2ee53269 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13720| Hash :: 0e453eabc673cece4272dbcffa74ff3839a95a0cb9766949c561d666f043c5f6 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13721| Hash :: c9e9880f2e6c2e48c12116e4b02ac5c1ab601a87c4fcaa09dcadde3c24533e8c |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13722| Hash :: 9b9f0ab1e5bede4a707555680a27980b7785a4b8926a75d1f9c807280b7b62f6 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13724| Hash :: 907f1535e194d3eb5d56fd8fcfa95fb19120cacdac1fcb39f6516f117433a691 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13725| Hash :: e37c1e6210ef8345b73e99f0d15a90777aba10f7a343f9610f0476eac538c260 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13727| Hash :: 6339dc6c8c3d04b2f6f6fe6437393905faa2d5cd546c5468bd960bb21276bdee |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13728| Hash :: 51c5fc326b12a391f2b042386ae4b74c5b5f7bc92357b58d1f93ac19b00894ab |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13729| Hash :: 001e6abbecd6ebef4d8a81aadbde51781308ed5a20b4ee010006ae23f5d09245 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13730| Hash :: f3d1b592e652d7981857fa040a5fa7ccfd92425d34149c526ab7f2096d1d47b2 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13734| Hash :: 1ed2d91e3522ee9c20f69ab94783c28e475764aeabf6f84aad8d4d11f5133870 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13737| Hash :: 18cb505e8fc5c04f783d4c59ac0222d147a84665136b2758b0002ea951673555 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13738| Hash :: c62719ea93e618e43f976eaf080e1c0979a8bc183ad5f0ce3167f1169c4e6dd3 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13739| Hash :: c140efe826c0e17a391ec84e38e25a984a8ddbe010920501c8ccc3551abfcc9e |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13740| Hash :: 47fef6214721c63149c57c599833b4d76ff9cf97607af7ab9adc604b4e8825ec |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13741| Hash :: 160a2bc5f20fb141cab880a1c49e73d5fc8d18a0346ced3e00a4d58870e308c7 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13745| Hash :: 5822fd0ef9cc56bf25dd1d6ffbdab0bfe0cba1d3b4ea5bf5450371b6d962eed3 |Prediction :: 1 | Actual val :: 0
Ticket_id :: 13746| Hash :: f8a40f7336350835ccf4335a63ba3540a46e631dfde2029c55f154e3a6b47065 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13747| Hash :: e2a72d40b2e3931a3266d560e5553243aea3737e388e2dec548a2a6bfa446d18 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13748| Hash :: 8f4dd32fb2b450d1acc275f276e0156f8abc03ec83e186a02140774d28842287 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13750| Hash :: 71d5f12f895f4143362d521cf044bf3758508062a4405d65d07a0b89f4f563d4 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13751| Hash :: abafe7c95bc7d172caab3feee963738ab1ec1e272e0d3ab587a737b05efbedb6 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13752| Hash :: 3b9321bd4b4ecf92c067999ddfa63fee0a10403755a83e26c617809130c73b00 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13754| Hash :: c629f114a51dca2560556150638004cce240e33d0d648233e22ee5c2c9f79bee |Prediction :: 1 | Actual val :: 0
Ticket_id :: 13755| Hash :: f646ecc288af7114a73e9c99bdf561daccc1a50ca72ffbaa66e642292c62b0d4 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13756| Hash :: b2f7b5f644de41d45f9d726db215dec310059db9e365141ab0697d941f659617 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13757| Hash :: 1af9cb417536a5dfc8e64ab736ebbd5b24054a5b65136faddec13daaf33ae8d7 |Prediction :: 1 | Actual val :: 0
Ticket_id :: 13758| Hash :: 66d99cc6d9b604443e995577c3e9596a1f601f71a3ed5b1f4dcf45c5214e0820 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13759| Hash :: 2b5043eecb068c49830cb67473ac9ae48b17989b9625fd2ca9a3685a8c4cbf1c |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13760| Hash :: 7768e79526672091fdbcc7e91230bee2cd01a4627e40bbd67620f4194ef7f3cf |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13761| Hash :: ff8c86030088e513d2f1b8571816b1efe9ce96c1d0e1806a9f2de2b434605cd1 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13762| Hash :: 3eb5e8784cac968ab3e296a54168834044560dfd096f7b3962696d33d3039739 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13763| Hash :: 994b402c59cd6d7afadf2ee5bb0ef6e1d9f7472f3cf3a3cd9f17f85bf6dba731 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13767| Hash :: 8d6831b3b18e68925fe32f4e25c143eecccd4bd2343831f87fed3ce341d43539 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13769| Hash :: 91bb7d549b2943d3a6b20751a47e233bd92860878f3e83c8db3b88a6cfb77cac |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13770| Hash :: 44a1ab822abe221750223d754a85c20ddde8f7c99b72d086746d356a7d6d50c5 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13771| Hash :: 46e39d701ecedc4f9b4e65904202f9dbf7038024b70eadb74a4e83c280bd23c2 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13773| Hash :: 303c403edbfc98bc78a3d0dfcabf568acae848be6a2516069b6c6fd254e657e6 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13775| Hash :: 3112078d3afbe2647c2f42847be25422ae0e17d9f1e8569da6e25eb365503140 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13778| Hash :: 4c335ebc907cc308c0e670129b1afcbb1ca959b4e3e844c01c97185eef695eee |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13779| Hash :: cc8d5366612c14e17ff3465b6954c65b988a2eeae0b6dde04605c50c929be2e4 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13780| Hash :: d3deb3f66eabf9fc7b8b4a3234e038cab597c73cbe72ed3b28f0cf47e3a073df |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13783| Hash :: 2a8a1f7c80907375d1e5e3961beecc63c9814192de4aa61a369ced7ff1f7f3e7 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13786| Hash :: afe1029448351f7f0cd962d059085cf2b00c4ffd005b6cc20c1c09cbb1693e5f9 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13787| Hash :: b903d843eeeedd851495eadbd9e4e701d14d76fc1eea4c158fe9e9039edc5fe2 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13788| Hash :: 1ffb80ab47a3a6ae87b87edf17856df48c4fa6eebfd7ba759206f06dabd6ceec |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13794| Hash :: 4578713c5d682c8bd8ae3c72e8f689de38e15cd8e577e0c78b2fa95b57ec42b6 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13795| Hash :: ca35567143cafd57384b8269d0bfcf162b5ec2f0a6fe390d00106e8e3cb891dc |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13797| Hash :: 8b9e83266b393f23143f2f5bc07f82ba9247572bc614231f188894abf5f3aa55 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13805| Hash :: a211ebb8298025a934736b515f48572d8f6fa21175722b9ef7b6bcb573869e36 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13811| Hash :: 82fcd3b27b3815908b4f0d49be8e22cb848e73795540eaf278dd853aaac738f1 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13837| Hash :: 61efb752ab2d57b095c88e844f321850236c0372ee18d0530aaa5ba4ace53de7 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13906| Hash :: f89f076e3e820a535579f04b6a094e9ba792d43a2b6b01c798bc0dcbf3feda63 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13947| Hash :: 3faee4491e4285cab3e863214bbdca2be426413f9b9c349037916192d10b2fa0 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13948| Hash :: 8bd8fa1861e3f2480693b15333d5d21429d459f2baf88dc9d0084c3e69782c03 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13949| Hash :: be69138ea74855d4b10f36f3cf72bcffcc90f9f13d2780d5d141751e2b889714 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13951| Hash :: b8f7ef6df38b078dc78cc0bd16de13a88b30cdcedc8d3f58f8aee2a24db5c95c |Prediction :: 0 | Actual val :: 0
Ticket_id :: 13952| Hash :: c60d9fd182e6d183e179631b8b78bd46ff92dc6e6dbf25039288386c6a349029 |Prediction :: 1 | Actual val :: 0
Ticket_id :: 13953| Hash :: deb7a0ba8faf9ca2e0614a6ff830def805309bfde4d29c6cfda48e5e40bb6c04 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14114| Hash :: db35642810d7f1dff96f11e87faaf88dfe0dcea369ef1ec67aaa52640ba193d8 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14115| Hash :: 258335e8a962fb3d2fe49b5ec9eb4a61f0a08f0c998dc133e8d3344743da285a |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14117| Hash :: 77de33bdfc5c9b3c3e5271632244023dab5129f69518f103e22b5297aec45fd4 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14118| Hash :: 87cfd7e566062c0f3634e9045530476ee6e6fc07d8550bf5f9d085fa50df392d |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14120| Hash :: f59ffccb3a00f8d160c1d85982658fc28e38fba96a1ef3a8d2509b7f8ea5437a |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14121| Hash :: 7ed4db26e7ba6c31ec78fc00ae7dbe41b662cd2b894c8cd89d1d1ee428d284b6 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14125| Hash :: 0c868f31353265b4c711c6183808ab068e9bc93951de15f99a0280203e09e697 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14126| Hash :: 4f08170d3fa0e243892f306a673f014338e44a14879fb19ec0a1bb7ef9893695 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14129| Hash :: 8cf92bd7039dc67a907340eb72b9db9d2a84e6b61c0ad74e241b8a397c52d1a9 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14130| Hash :: fa9ed376bb204ff3b3a23ff3241b954247748616c968c4225be215cdaf0fa6cb |Prediction :: 1 | Actual val :: 0
Ticket_id :: 14131| Hash :: cbad1ea3905b30bfadfeaeecd5b7c9eb3b4e7d818861e6ff70df4b408ce6bac3 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14133| Hash :: 1aa72bd1ee85693c9087dc29f64b283801c630ed19d9ebe04bbc6defad9bd871 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14135| Hash :: 20eb7e0577d4f6da3c8324a66a5c60c194626ba75f97afe06453d4a1487c20ce |Prediction :: 0 | Actual val :: 0

Ticket_id :: 14136| Hash :: 7ab348c24151db31305f9a838b6485d50282cbb32159a98ebb790291219b4a47 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14138| Hash :: 6a44b89bb55dfc573e0bdf518b87a40ab4b53883baeac0dd44d2c2f4a2fd8bfa |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14139| Hash :: b10deb13f38276dc704d4401942b7ee61668ddc232c9a6444ebbfd909d51ba7b |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14145| Hash :: 2c19938d709cefb4fa242ffa6fbf740f86ffc35f21beca35e8877144fb8e88f6 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14146| Hash :: 3bd6e2e629d191821ad20429f2d6c89262971096085ca3a0598f4f158dd8961c |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14147| Hash :: 22bd2dc90bc82a1460bf611899e92daf9249cc94c4c69ca7d8c65a0e9654545e |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14149| Hash :: 2e5fcaf9ae5e75da386fe489b8befb4bef21c90714096dd73a83e7bb2150fe33 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14150| Hash :: 3131bd2696034ed1cbcde20346cc3ea58670a79cb42dfc41df11591dc10637af |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14152| Hash :: afbdeb12c76974827841abb2e2822cf59171245a36e94bffe05705c036d85f1e |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14153| Hash :: 186772fa40d06f72a1258347eea937ce5a9a8996619af480cf136cd18fc2c927 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14155| Hash :: f049671d230a471329f34d3d9893fbbfcc6488038ede2627a5fcb57e93174b86 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14156| Hash :: de569c8133bd43c68616857d63de48f8bc331045234c64e1da16a47bcc478047 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14157| Hash :: aa5d6d960cb33a63742a621f8fe7e27b726731af818fc6aba7ecbda74fbc6d9d |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14158| Hash :: c0e77a9a1b9313f1e97924ed5ca4df4fb8e887dd4280e68a3967d6bf2c37b289 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14159| Hash :: d975733e5851d86cffc66e1a40e70a3af65bbf5cc4ad2509ded20f8ede4c080f |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14160| Hash :: ba4a6c6a91677a33bc74dc1a14e06914582fd4d70a5616ba965d1c498f1b4084 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14161| Hash :: fdea155356cd4fd6c961013b3c5adbe62ff8cca92361ff19266b56d193ec9be9 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14162| Hash :: aee631d034e427c4110a0fd6c2c48d617855b2daa9be4e6f3096b3cf63976f91 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14164| Hash :: 4d0ffb14cc9b69ef9a8b0e48b71c6b09503c82855d5f6f2f870417ff9ebbf6da |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14166| Hash :: b0c3d2fedf6a9277d3820638d0f8489e7b8b2e26aeedfdd430a2cb80556ee4b4 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14168| Hash :: 51baa9a72677f0323da36d95833f7268da5507984cac5d477ed925f41089770a |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14169| Hash :: 6535ccbe7dab021bb54c966644082664535aa425c77cd7ac26f79b29e3688c5a |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14170| Hash :: c103b8b94ccbeac36fa53896ab028ffe3d806417469bfed818ebb1aee07a2d0d |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14171| Hash :: 8d364ebf3d959c0041e7540288c4aa1c3ecaf3418922715c2e4e6d4965a737a4 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14172| Hash :: c184152e9936920724bacaf484909e454f5a2d61511138e65f7428648ff5e7d5 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14173| Hash :: 0e60420d6b98e526aedb141199e3b9447bf964ff5c0dfe2e8af07797ab79f780 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14174| Hash :: 06fac8d03f59f563e4e435577e02e6267a1c40d7fe3752169ed60d1b622ba41e |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14176| Hash :: f8c9db72dbddc1498fb2c04ff7664f629dc6570c334d0bae36976fb7b2ffe4a6 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14178| Hash :: be107cb52429384f9aab938778a506838ebd840210a3722fe6a6c997bea74936 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14179| Hash :: 7b3285c0af76e661a49d5772a636d69307731c6b5b3ded3fd653ad3b63beb145 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14180| Hash :: 70baf80d1b2da5e1526d849c16e93e76c8893e348c5baf00634688188ccf54f5 |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14182| Hash :: c9e7cadc92f256f93a9400e293fff023084190d660aa0b12644bd3cc8a81cf1e |Prediction :: 0 | Actual val :: 0
Ticket_id :: 14183| Hash :: b6ea44736fc5df1cb2466db24255e9e48f050debf1cb9fb18bdff77621c9963a |Prediction :: 0 | Actual val :: 0

10 Appendix B

Saved iptable Rules

```
# Generated by iptables-save v1.4.21 on Mon Apr 25 20:23:54 2016
*nat
:PREROUTING ACCEPT [539:83843]
:INPUT ACCEPT [539:83843]
:OUTPUT ACCEPT [7032:446659]
:POSTROUTING ACCEPT [4:292]
-A POSTROUTING -s 192.168.122.0/24 -d 224.0.0.0/24 -j RETURN
-A POSTROUTING -s 192.168.122.0/24 -d 255.255.255.255/32 -j RETURN
-A POSTROUTING -s 192.168.122.0/24 ! -d 192.168.122.0/24 -p tcp -j MASQUERADE --to-ports
1024-65535
-A POSTROUTING -s 192.168.122.0/24 ! -d 192.168.122.0/24 -p udp -j MASQUERADE --to-ports
1024-65535
-A POSTROUTING -s 192.168.122.0/24 ! -d 192.168.122.0/24 -j MASQUERADE
-A POSTROUTING -j MASQUERADE
COMMIT
# Completed on Mon Apr 25 20:23:54 2016
# Generated by iptables-save v1.4.21 on Mon Apr 25 20:23:54 2016
*mangle
:PREROUTING ACCEPT [264049:47887124]
:INPUT ACCEPT [264049:47887124]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [65378:591851790]
:POSTROUTING ACCEPT [65485:591859013]
-A POSTROUTING -o virbr0 -p udp -m udp --dport 68 -j CHECKSUM --checksum-fill
COMMIT
# Completed on Mon Apr 25 20:23:54 2016
# Generated by iptables-save v1.4.21 on Mon Apr 25 20:23:54 2016
*filter
:INPUT ACCEPT [119:26451]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [113:24770]
-A INPUT -i virbr0 -p udp -m udp --dport 53 -j ACCEPT
-A INPUT -i virbr0 -p tcp -m tcp --dport 53 -j ACCEPT
-A INPUT -i virbr0 -p udp -m udp --dport 67 -j ACCEPT
-A INPUT -i virbr0 -p tcp -m tcp --dport 67 -j ACCEPT
-A INPUT -s 192.168.0.0/24 -p tcp -m tcp --dport 27017 -m state --state NEW,ESTABLISHED -j
ACCEPT
-A FORWARD -d 192.168.122.0/24 -o virbr0 -m conntrack --ctstate RELATED,ESTABLISHED -j
ACCEPT
-A FORWARD -s 192.168.122.0/24 -i virbr0 -j ACCEPT
-A FORWARD -s 192.168.56.0/24 -i vboxnet0 -o em1 -m conntrack --ctstate NEW -j ACCEPT
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i virbr0 -o virbr0 -j ACCEPT
-A FORWARD -o virbr0 -j REJECT --reject-with icmp-port-unreachable
-A FORWARD -i virbr0 -j REJECT --reject-with icmp-port-unreachable
-A OUTPUT -o virbr0 -p udp -m udp --dport 68 -j ACCEPT
-A OUTPUT -d 192.168.0.0/24 -p tcp -m tcp --sport 27017 -m state --state ESTABLISHED -j
ACCEPT
COMMIT
# Completed on Mon Apr 25 20:23:54 2016
```

11 AndroSandX Product Source Code

Complete source code of the AndroSandX and other results data sets are available in the attached CD.