SELECTED FORENSIC DATA ACQUISITION FROM ANDROID DEVICES

____

A Thesis

Presented to

The Faculty of the Department of Computer Science

Sam Houston State University

____

In Partial Fulfillment

of the Requirements for the Degree of

Master of Science

____

By

Khushboo Rathi

December, 2018

SELECTED FORENSIC DATA ACQUISITION FROM ANDROID DEVICES

By

Khushboo Rathi

APPROVED:

Umit Karabiyik, PhD
Thesis Director

Narasimha Shashidhar, PhD
Committee Member

Qingzhong Liu, PhD
Committee Member

John B. Pascarella, PhD
Dean, College of Science and Engineering Technology

DEDICATION

I would like to dedicate my work to my beloved parents and brother for their love and trust, to my husband and daughter, for their immense support and patience. Also, I dedicate this to all my friends and colleagues who helped me throughout my study and research.

ABSTRACT

Rathi, Khushboo, S*elected forensic data acquisition from android devices*. Master of Science (Computing and Information Science), December, 2018, Sam Houston State University, Huntsville, Texas.

In recent times, amount of data stored in the smartphones have increased phenomenally. A smartphone is as powerful as a laptop or a desktop where people store their person data or do daily activities, as a result it can act as an important evidence for law enforcement while solving the cases such as, in case of accident, malicious exchange of text messages, photos or videos taken during mass shooting incident. This act as an important forensic interest to the investigator.

Some people may be willing to give their phones to the investigator, but they would like to make sure that their privacy and their data privacy have been taken into consideration, meaning that only data relevant to the case under investigation should be analyzed and collected. Even supreme court have passed that ruling to preserve the users and data privacy. In this research study; a new forensic tool is developed which can do selective extraction of data from an android device. The input to this tool is based on the consent form which is filled by the witness/victim who voluntarily hands over his/her phone to law enforcement and investigator extracts data within those limits.

This tool does the extraction on metadata and content based filtering and export the extracted data along with the hash values to a bootable drive in a forensically sound manner. State-of the art machine learning models are used to perform content based filtering. As a result, a robust and efficient tool is built to solve the real time cases while preserving the users and data privacy.

KEYWORDS: Android forensics, Selective extraction, Machine learning, Tensorflow, Privacy, Data acquisition

ACKNOWLEDGEMENTS

First of all, I thank God for His blessings that have steered me through this thesis. I wish to sincerely thank my advisor Dr. Umit Karabiyik for his insightful comments and guidance. Without his help, this study would not have been completed. His patience, motivation and knowledge helped me to articulate my project in a better shape. In addition, I would like to thank my committee member, Dr. Narasimha Shashidhar and Dr. Qingzhong Liu, for their advice and feedback.

I am gratefully to my colleague, Nicholas Guerra, for working along as a team member and for his help during my research. I have truly shared and learnt from his company and gained a good friend. I would like to thank National Institute of Justice (NIJ) for this grant project (Grant Number: NIJ-2016-8976), and giving Sam Houston State University to work along with other universities, Florida State University and Florida Agricultural and Mechanical University (FAMU). I have learned to work better in a team, gained and shared knowledge throughout the whole process.

I would like to thank Computer Science Department for providing the facility for my research. Finally, I would like to thank all the professors, who transformed me to a responsible student by guiding me throughout my master's degree, which made the whole process an exciting and valuable experience.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER I

Introduction

According to a report from Statista (Statista, 2017) Android operating system holds 85% of smartphone market's share at the end of 2017. With this pace of increased in demand and popularity, smartphones have become the core repository which helps to store user's personal data or keeping track of daily activities. Some of the way that users store these activities can be photos or videos, exchange of texts communication, making calls and setting up calendar reminders. As a result, this information can act as an important evidence during a forensic analysis. The act of accidents, gun shooting incidents, abuses through malicious messages or sharing pornography images can be easily documented on the smartphones. This information can act as an important evidence to solve cases by the investigator. People are hesitant to come forward to handle over their smartphones as they fear their privacy and their data privacy will be violated. As this is a reasonable concern, courts have passed a ruling stating that only data relevant to the case under investigation should be collected and analyzed during investigation (Riley v California (573 U.S 2014)). This can be achieved by taking a signature on the consent form from the witness/victim who voluntarily hands over their mobile device and setting the search limit for the law enforcement agencies.

Significance

In 2014, US Supreme Court passed a ruling after a (Riley v California (573 U.S 2014,)) case and subsequent rulings arising from this case, that only selective data which is relevant to the case and for which the consent has been granted by the user should be collected and analyzed. As per our research, there is no such tool in the market which does selective extraction. Hence in this study, a tool is built to acquire data from the smartphones with metadata and content based filtering techniques. Metadata filtering will have date range, location and name options to filter the data from the smartphone. Content based filtering uses deep learning techniques on the photos stored in the smartphone galley to retrieve those photos using image classification. There are many commercial tools in the market which help to collect and analyze data in the smartphone and some of these tools such as Cellebrite UFE (Cellebrite_Wiki, 2018) and Magnet Axiom (Magnetaxiom_OfficialSite, 2018) are widely used by law enforcement. But these tools retrieve data as a whole and do not have the same selective use cases which our tool is built on. The main aim to develop this new forensic tool was to preserve the privacy of the user and their data, and the investigator against to legal consequences. This tool tries to ensure that the users are less fearful and more willing to come forward to handover their smartphones for evidence collection purposes, and protects investigators accidentally extending their search beyond the consent given by the users.

Statement of purpose

The goal of this research work is to develop a system which does selective data extraction from Android phones in a forensically sound manner, driven by the input provided by the first responder or forensic examiner which is based on a consent form agreed by the victim or witness.

To extract data from the Android device, some of the applications may require a root access. Rooting an Android device means attaining a privileged control or getting the super user permission. Rooting grant the permission for altering or replacing the applications and its settings, helps in running specialized applications that require administrator-level permissions, or application which are unable to run with normal user level permission.(Wikipedia & Rooting). As the phone is voluntarily given by the users to the law enforcement agencies, we do not want to void the warranty with rooting the device, and also don’t want to cause any potential harm to the phone. Hence, logical extraction techniques are only used for this study and no rooting is performed.

This tool does selective extraction with the main filtering categories, metadata and content based filtering. To perform metadata filtering, Android’s Application Program Interface (API) is used. This API will help in extracting the information from applications database files. To share this extracted information among different applications, content providers are used, which is the services provided by Android platform to share data among applications (developer.android.com).

Other filtering option, content based, uses machine learning technique which classifies photos using image classification models. By the help of these two filtering processes, our tool is able to extract only the relevant files from the devices. The input on the system is driven broadly by the contents of a consent form and by using the user interface the investigator inputs it in the system.

The extracted data after filtering can be readily visible to the investigator on the display screen and it can be exported to the investigator’s laptop with the system designed to carry out the export from the mobile device to the laptop. The details of the design and implementation is discussed further in the following chapters.

Assumptions

This study was conducted with the following assumptions.

1. The Android phone was willingly handed over to the investigator and hence it was in switched-on state. The device was unlocked or password was provided to the investigator before handing the device.
2. User data on the device does not change during between extraction processes.
3. Airplane mode was enabled during investigation so as to make sure that data was not modified while keeping it isolated from the network.
4. Windows 10 boot from the USB drive on which the extracted data resides.
5. Forensic examiner should enable USB debugging and unknown sources settings on the Android phone to enable the extraction.
6. Native applications such as calendars, message, phonebook etc. comes pre-installed on the device.
7. The deleted data was not recovered during the investigation.
8. The system works for both rooted and unrooted devices however rooting option is not required
9. At present, data from native applications from the Android phone is extracted.
10. Content based filtering uses image classification for categories such as weapons, drug detection, vehicles, and skin exposure.

Limitations

1. Android OS is being adapted by various manufactures such as HTC, LG, Lava, Motorola, Samsung, Google etc. But the testing for this study were performed

on two devices, Motorola Moto G3 with Android version 6.0 (Marshmallow) and Samsung Galaxy S7 with Android version 7.0 (Nougat).

2. The content based filtering is only available to extract the image data from the Android device.
3. The artifacts collected during this study are limited to Text messages, Call logs, Pictures, Videos, Calendar events and Contacts (phonebook).
4. There are different operating systems which work on smartphones in the market, but this system will only work for Android devices.
5. Third party applications are apps which are developed by individuals or companies other than the provider of the mobile operating system e.g. WhatsApp, Viber etc. Message database from such third party applications were not considered during data extraction process.

Summary

The motivation to design a system to extract selective data from an Android device was presented. Importance of data and users privacy issue arising during data collection in any investigative process is discussed. Metadata and content based filtering help to achieve selected data extraction are introduced. An outline of the study was discussed defining its scope, limitations and assumptions.

CHAPTER II

Literature Review

This chapter discusses the evolution of mobile phones, challenges in mobile forensics, Android architecture, file storage and machine learning techniques which can be integrated with Android applications. It also covers the related work on the forensic analysis of applications on Android devices.

Evolution of Mobile Devices

The evolution of calling a mobile device from cell phones to smartphones has seen a rapid growth over years. The first Mobile phone DynaTAC 8000X was developed by Martin Cooper (Morum de L. Simão, Caús Sícoli, Peotta de Melo, Deus, & de Sousa Junior, 2011) and was introduced in the market by Motorola in 1973. Over the years mobile phones have become more powerful and now they have taken a crucial role in our lives. The initial mobile phones were only designed to provide voice communication but now the horizon has expanded to features such as camera, external memory, biometric (fingerprint scanner, facial and iris recognitions), applications (calendar, contacts , Short Message Service (SMS), photo gallery, Multimedia Message Service (MMS) , online banking etc.).

In 2007, Apple introduced its smartphone iPhone with easy access to Internet and a number of other features (Morum de L. Simão et al., 2011) the Open Handset Alliance (OHL) formed a groups with more than 84 mobile technology group from mobile services, manufacturing, hardware, software and semiconductor manufactures and launched an Android operating system (OS) (Lessard & Kessler, 2010) Android OS is built on Linux system. As per the report on May 2017 (Techcrunch, 2017) Google announced that there were 2 billion active devices which run on the Android operating system.

Mobile forensics and its importance

Publication (Ayers, Brothers, and Jansen (2014)) at National Institute of Standards and Technology (NIST) on guidelines on mobile device forensics states that "Mobile forensics is the science of recovering digital evidence from mobile phones under forensically sound conditions using accepted methods" (p. 9). Forensically sound condition means that extraction process should not alter the data of a mobile phone memory from its original state.

Importance of mobile forensics has increased with the involvement of mobile devices been used to commit crimes. Recovered data from the mobile phones can be considered as an evidence in a court of law which acts as a crucial role (Ahmed and Dharaskar (2008)). (Grispos, Storer, & Glisson, 2011) discussed several cases where mobile phone evidence played an important role in the investigating crime related to social media. It also discussed about the different tools used to collect evidence from the mobile phone.

Mobile phones are capturing the telecommunications market with a rapid speed coming up with different varieties of operating systems and large storage capacities. Frequent updates on the version of operating systems, lack of standard framework for collecting and analysis, and evidence volatility are the major difficulties that are being faced by mobile forensic examiners these days. Mobile forensics has become an important field with the increase in the number of mobile phone users, advancement in technologies related to mobile phone and rampant use of mobile phone while committing a crime.

Data storage locations in mobile devices

There is different storage medium where the data in the smartphone can be stored. The basic information on the mobile phones are implemented in the Subscriber Identity Module (SIM). Other storage, internal memory where most of the applications or system data are stored and external memory in the form of in-built or extended SD (Secure Digital) card can act as an additional memory unit attached to the mobile device. A SIM card stores very limited information which includes mobile subscriber's information, text messages and controls. But it should be noted that SIM card doesn't store this information in continuous memory. As retrieving this information does not give meaningful results hence internal memory was introduced (Yngvar Willassen, 2003).

Internal memory was introduced with EEPROM (Electrically Erasable Programmable Read-Only Memory) chip but later it was replaced by flash memory (Yngvar Willassen, 2003). Flash memory is designed using the NAND technology (Yngvar Willassen, 2003). Non-volatile storage technology such as NAND doesn't require power to save the data. Most of the applications stores their data on the internal memory. Recently, the features in the smartphones are expanding and hence require much more additional space. This additional space is fulfilled with the help of external memory. Internal and external flash memory have made the mobile phone as a goldmine of information, and tools used to image these mobile phones are becoming valuable. (Y. Willassen, 2008) worked on locating different data items on the internal memory of a mobile phone by de-soldering the memory chip from the phone using embedded test technology. The data recovered using this technique was very useful for forensic investigation.

The possible data type that could be stored on the mobile phone's internal memory, SIM card and external memory cards in files are listed below:

- Contacts list
- Call Logs
- Calendar Entries
- Audio files – Music and voice
- Internet History
- MMS – Multimedia Messages
- Pictures
- Videos
- System Firmware Information
- Text messages

Common Acquisition Methods

There are three different type of acquisition methods: manual, physical and logical (Ayers et al., 2014). In manual extraction, the investigator manually interacts with a mobile device and collect the data either by taking the picture of the evidence in the devices or physically interacting with it and collecting it. Logical acquisition involves capturing a copy of logical storage objects, such as directories and files that resides in file system partitions (Ayers et al., 2014). Logical acquisition techniques can only extract data from allocated space assigned by the operating system. Physical acquisition is bit-by-bit copy of the entire internal flash memory. The deleted files are also recovered with this method.

There are different commercial forensic tools available in the market which help in extracting the data from the device. Tools based on logical technique access memory by using the command and respond based protocols. They extract data by sending the queries to the operating system which in turn communicate with the phone memory. As this method relies on the response of the mobile operating system, the pointer to the deleted files are erased and hence deleted files cannot be recovered.

NIST maintains the list of databases of all the logical forensic tools. NIST is an organization which provides guidance to forensic examiners on the tools, software and hardware specifications required for the forensic examination. The tools which are based on the physical techniques are based on flasher boxes, Joint Test Action Group (JTAG) and Chip-off. (Y. Willassen, 2008) "JTAG method extracts complete physical image of a mobile phone if the JTAG port of the mobile phone is connected to a computer using a JTAG emulator. Extracted image is analyzed by using WinHex or any other binary files analyzing software" (Ayers et al., 2014). Chip-off method requires to remove the flash memory physically from the devices Program Control Block (PCB). The data from the separated flash memory can be read by inserting the card into the flash reader.

Physical and logical techniques both have their advantages and disadvantages. Logical acquisition requires less time and works fast. Extracted data is easy to analyze without much of training and very helpful in forensic investigation. The drawback is that the deleted and unallocated data cannot be recovered. Physical techniques extract complete internal storage data and hence can extract more information and deleted data as well but it takes longer time to analyze, parse and derive the extracted data.

Android Forensics

The smartphone market is abound with several different types of smartphones running different types of operating systems. Smartphones have great computing power and are equivalent or sometimes more powerful than personal computers. It was found that about 97% of U.S. mobile market (D. Reisinger, 2017) can be divided into Apple iOS or some version of Android operating system from Google. Among these Android is gaining popularity and users for Android phones are increasing every day. There are about 1.5 billion Android users worldwide (D. Reisinger, 2017). There are more than 60 mobile manufactures using Android OS in mobile, tablet or laptops. Android devices comes with the official pre- installed app call 'Google Play Store' which provides access other application hosted on it. It allows users to access applications, music, movies etc. According to the report from Statista (StatistaReport, 2017) Android play store features over 3.5 million applications .

Android was initially developed by Android Inc., which was later bought by Google in 2005. The first Android based phone was released in 2008 and marked with the name HTC-G1. Since then the operating system has undergone multiple major releases. The current version at the time the study was conducted was version 8.1 also known as Oreo. To perform forensic analysis on Android devices it is important to understand the Android architecture, data storage structure (user space, system space) and retrieval mechanisms based on tools that are available and method of extraction.

Android Architecture Design

Android stack is divided into five layers (Google Inc.). Figure 1 discusses the major components of the Android platform. Android is an open source operating system and it is designed to support a wide variety of hardware. As Android is based on the Linux based software stack, it helps in supporting wide array of devices and form factors.

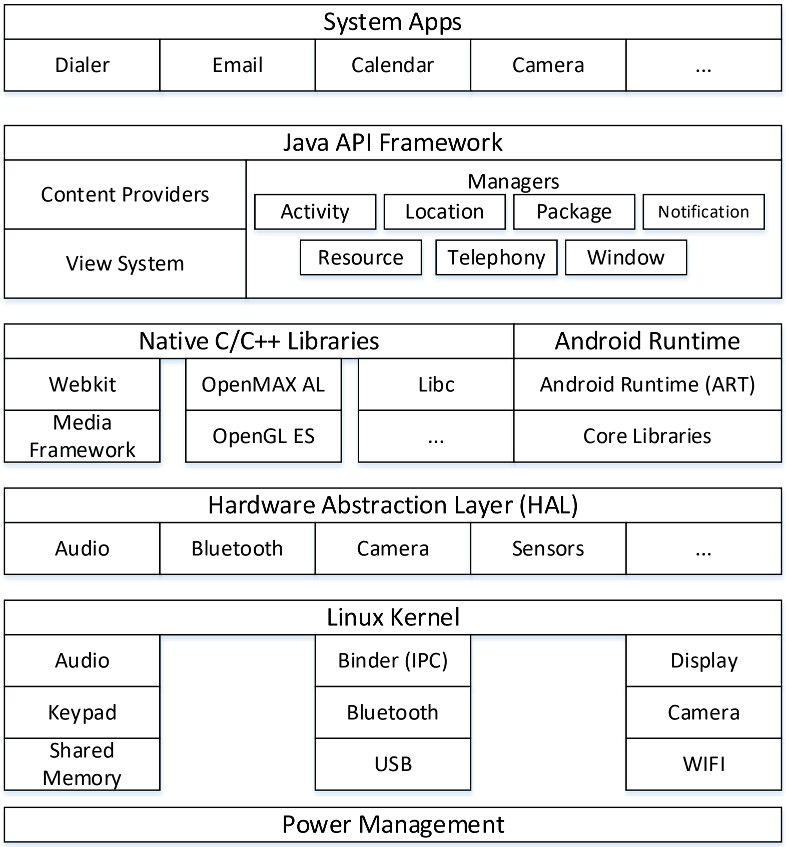


Figure 1. Architecture of Android operating system

Linux Kernel: This layer marks the foundation of the Android stack. Android is built on the 2.6 version of Linux kernel. The main functionalities of this layer is memory, process management. It allows Android to take advantage of the security features and allows different manufactures to develop hardware drivers.

Hardware Abstraction layer (HAL): Through this layer, Android exposes the device hardware capacities to the upper Java API Framework. Most of the library modules for the Android drives such as Camera, Bluetooth, and Audio etc. are stored in HAL. Whenever framework API make a call to access these device hardware's, Android loads the library module for that hardware component.

Native Libraries layer: Android runtime is comprised mainly of core libraries and the Dalvik Virtual machine (Google Inc.). The core features of the Android is taken care by the native libraries. Like Java Virtual Machine, Dalvik VM is also a register based Virtual Machine that provides the necessary optimization for running in low memory environment. Dalvik VM converts the Java byte code into the executable files of *.dex* extensions. It takes advantage of the core features of Linux kernel that includes multithreading, process and device and memory management.

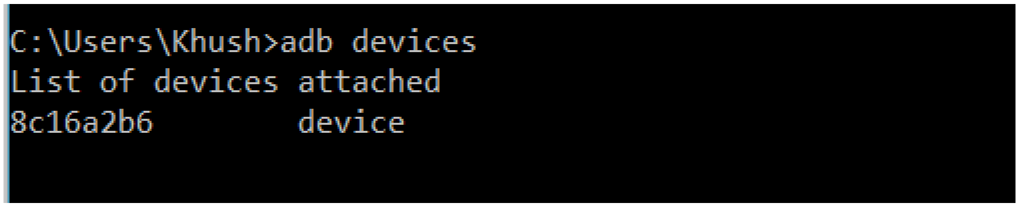
Java API Framework: The major services provided by this layer includes Activity Manager, Content Providers, Resource Manager, Location Manager, Notification Manager, View System and Telephony Manager. All these services are provided and used by the application with the APIs and high level services.

Android Applications: The top most layer in the Android stack comprises of the native applications (in-built applications) or any third-party applications. All applications can run simultaneously and most common applications are Contacts, SMS, Clock, Calculator and Internet Browser etc.

Android Software Development Kit and ADB Tool

Android Software Development Kit (SDK) is the collection of software development tools which helps in creating an application to run on different platform. As most of the Android application are written in the Java programming language, it also make use of Java Development Kit (JDK). After installing Android SDK on the workstation or examiners laptop, it allows the connected Android device to communicate and perform forensic analysis on the device under investigation.

Android Debug Bridge (ADB) is one of the tools available with SDK which allows access to the system partitions part of the device if the USB debugging mode is enabled. To enable the USB debugging mode on an Android device, the developer option in the settings of the phone can be activated by taping on '*Build number*' for seven times. ADB has command line interface where ADB commands allows to retrieve the active files from the device. *adb device* command helps to determine all the devices connected to workstation. Figure 2 shows the screenshot of the command while executing it on the command line. *adb install* is used to install the application on the Android device. *adb shell* gives access to the Android shell mode which allows forensic examiner to collect the information such as running process information, file locations, date, connection status etc. To access the system reserved partition, root access need to be granted, by default the devices are unrooted as to prevent it from malicious attack and to secure the user information.



```
C:\Users\Khush>adb devices
List of devices attached
8c16a2b6        device
```

Figure 2. Display of list of devices attached using adb devices command

Android File System Hierarchy Standard

Android file system is built upon Linux file system structure with the starting directory as *root.* Initially Android file system was implemented using Yet Another Flash File System (YAFFS2) (Vijayan, 2012). FAT and FAT 32 file system are supported on the external SD card of an Android device. As most of the current forensic tools are not compatible with the YAFFS2 file system so they faces problems during mounting the

Android partitions and accessing the data stored on the file system. (Barmpatsalou, Damopoulos, Kambourakis, & Katos, 2013).

YAFFS2 file system was replaced by a new file system named Fourth Extended File System. EXT4 file system supports the dual processor processes and also keeps a log of all the actions performed. It also provides the mechanism for error recovery. The EXT4 file system provides "acquisition of unallocated files and recovery option" (Barmpatsalou et al., 2013)

Android Application Data Storage

Android devices can store the data either in the internal or external memory (Vijayan, 2012). Applications on the Android devices may store data in any of these storage locations. APIs are used to store the data in the internal location where else there is no fixed procedure to store data on external location (Hoog, 2011).

Concept of sand boxing is used in Android devices which prevents different applications to see the implementation of each other and also place each application at the specific location in the file system. According to (Maynard Yates, 2010) call logs, text messages, calendar events and other items are stored in the internal memory. The default path to all the application in Android devices is stored in the */data/data* directory. All the database files are stored in */data/data* directory. In the forensic examination of an Android device, the database files and the libraries component plays a vital role (Lessard & Kessler, 2010). To access the */data* directory, super user permissions is required. The database files are stored in the SQLite data format (AndroidSQLiteInc.). SQLite is an open source and free of cost database structure.

Every application installed in the Android device is placed by default at */data/data/<package_name>/.* This location will store all the database files, libs and *shared_prefs* directories. The items in this location can vary with the different manufactures. The devices used during this study (Moto G3 and Samsung Galaxy S7) use the directory */data/data/<package_name>/* to store these files.

The photos and videos for the third party applications such as WhatsApp, Viber and WeChat are stored in the external memory (SD card) and rest of the other database files are stored in the internal location. This system extract files from both of the locations during extraction process.

Logical Techniques to Image the Memory location

Data can be extracted from the application in the Android devices by accessing the file system. (Mohindra, 2008) illustrated the Android file system. Figure 3 depicts the directories on the file system on an Android phone.

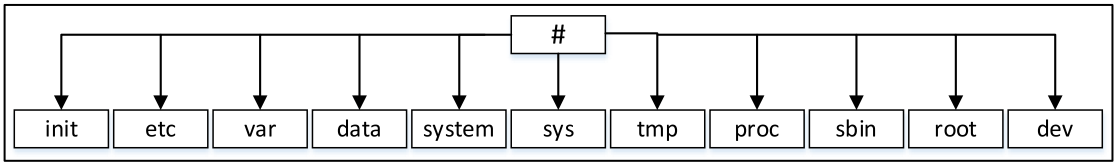


Figure 3. Directories of Android OS

Logical tools can extract the data from all the directories in the file system, but it fails to retrieve the deleted data. To use logical extraction, the examiner does not need to know the hardware details of the Android phone and hence it is quite easy and fast.

There are four categories to do logical extraction on the Android devices (Silla, 2015) :

1. Partition imaging
2. Copying files & folders

3. Content providers
4. Recovery Mode

In this research, content providers and logical imaging techniques are used to extract the data from different applications on the Android device. Content providers act in a special way by which sharing the data among the applications is achieved successfully.

Open Source Tools Utilized

AFLogical by NowSecure (formerly viaForensics) is the open source tool which is used by the investigation agencies. It extracts data using content providers which is similar to the most of the commercial tools using the other logical extraction techniques (Hoog, 2011). All these tools extract the whole backup from the Android devices. The system in this study is based on the AFLogical techniques but it does the filtering before extracting the data from the applications hence ensuring that only selective data is extracted.

Content based filtering, includes classification of the photos which are stored in the gallery application uses machine learning models which are included in this new study. This powerful feature is missing in the AFLogical tool.

There are other open source tools which help in extracting the data logically from the mobile devices. They are:

1. Android Debugging Tool (ADB)
2. Backup analysis

This study makes use of above tools to compare results. ADB *install* command, is used for installation of our application to an Android device.

Commercial Tools

There are various commercial tools available in the market which make use of the logical extraction technique to retrieve the evidence file from the devices. Few of the most popular ones are:

1. Cellebrite UFED
2. Encase Neutrino (Encase_official, 2018)
3. Paraben Device Seizure (Paraben_official, 2018)
4. Mobile phone examiner (MPE+, 2018)
5. Magnet Axiom

All of the above commercial tools except Cellebrite take complete backup of the data from the devices and does not perform selective extraction use case. There is a feature in Cellebrite which helps to take the backup of individual category such as call logs, contacts or photos etc.

Machine Learning and Deep Learning

The term "Machine Learning" was coined by (Samuel, 1959). It is a branch of artificial intelligence in the field of computer science in which machines often utilize statistical analysis to predict results by gaining knowledge progressively from the data on their own. It is the study of algorithms which learn from the data and make predictions on it without the need of explicitly programming the task (Samuel, 1959). The efficiency of a machine learning algorithm depends on the sample data used *(Mohri, Rostamizadeh, & Talwalkar, 2014).* It is also closely related to computational statistics, mathematical optimization and data mining.

Machine learning algorithms have been deployed in pretty much any technology used in our everyday life. Often, work done in industries creates data which is then used in learning models to make predictions. The quality of the results depend highly on the quality of the data fed to the algorithm. (Mohri et al., 2014) lists significant number of applications which are accomplished using machine learning algorithms such as text classification, speech recognition, image recognition, games, driverless vehicles, personalized advertisements, etc. These are only some of the applications that we observe in everyday life. The scientific community is ever trying to increase the reach of machine learning algorithms every day.

Learning tasks can be identified into three main categories – supervised, semi-supervised and unsupervised. In supervised learning, the training data set consists of inputs and corresponding outputs and the algorithm maps how the two quantities are related. Once this relation is established, predictions can be made using the model. When some of the outputs are missing from the data set, semi-supervised learning is performed. However, when no outputs are present and all the inputs are used to make sense of the data and find hidden relations, it is called unsupervised learning.

Deep learning is a sub-category of machine learning algorithms which are mainly used for pattern recognition (Li & Dong, 2014). In this research, deep neural network algorithms are also utilized to recognize and classify images. These algorithms are inspired by the communication channels in the nervous system, neurons, to process information. The algorithms consist of several layers which progressively refine data and characterize any hidden patterns in it.

Machine Learning Models

Machine learning (Murphy, 2012) and its applications have gained a lot of attention lately. Deep learning (LeCun, Bengio, & Hinton, 2015) has been successfully applied for building systems which help in image recognition (Krizhevsky, Sutskever, & Hinton, 2012), pattern matching to natural language processing (Cho et al., 2014) . Neural networks can help in feature extraction from images to predict photos after training them on a training data set. TensorFlow (Abadi et al., 2016) which is an open source framework released by Google can be used to implement deep representation learning using neural networks easily. TensorFlow Lite (TensorFlow.org) is specially designed to integrate machine learning models on the mobile devices. The advantage of using TensorFlow Lite is that it uses the optimized model and hence after integrating trained model with it in Android application it reduces the size of the application and makes it easier to use. There are different frameworks such as CAFFE (Jia et al., 2014) and Theano (James Bergstra et al., 2011) that are developed to be used in deep learning representation. Furthermore, with the advent of smartphone that are equipped with state of the art processors, it is now possible to run trained models for deep representational learning on these phones for several different tasks such as face detection, image analysis, and classification using frameworks like Inception (Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2015), MobileNet (Howard et al., 2017) and NSFW (Mahadeokar, 2017).

Transfer Learning

Many image classification models have millions of parameters and training them will take a huge amount of time and computation power (hundreds of GPU hours). Transfer learning is a way to use already trained model on a related task and reusing it in a new model (TensorFlow.org). Figure 4, discuss about the Inception V3 model (Szegedy et al., 2015) and how transfer learning is applied on the last layer to identify the images. "*Inception v3 is a widely-used image recognition model that has been shown to attain greater than 78.1% accuracy on the ImageNet dataset*" (Szegedy et al., 2015). After the feature extracted from the image the final layer is re-trained with the training data set. The training data set is preprocessed before being used to train the last layer.

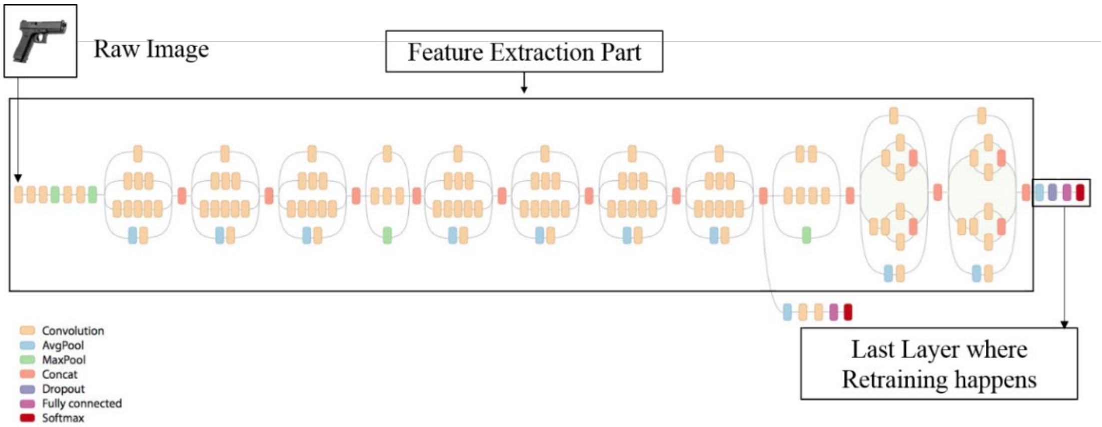


Figure 4. Example of InceptionV3 Transfer Learning Model (Source: (Model))

CHAPTER III

Design Model and Implementation of a system

This chapter describes the design and implementation of a novel forensically sound system that does selective data extraction. It also discusses about the subsystems such as data identification, data acquisition and data validation in detail.

A recent ruling of the US Supreme court (Riley V California (573 U.S. [2014]) and subsequent rulings arising from this landmark case indicate that in order to search a smartphone it may not be enough to have a warrant for the search, but it may also be required to restrict the search to specific data on the device that relate to the crime being investigated. This chapter describes the design and implementation of a novel forensically sound system that does selective data extraction. Commercial tools such as Cellebrite, Paraben and Magnet Axiom has great utility but they target a different use case and their search feature does not currently support many of the capabilities, including Artificial intelligence capabilities, that this research study integrate. The Android phone used as the test phone to conduct experiments and the test procedure used are also discussed in this chapter.

High Level System Design

The system is broadly consisting of three different subsystems, namely: The data identification system, the data acquisition system and the data validation system. The data identification system is responsible for identifying the most relevant files based on the metadata and content based filtering. The input to this subsystem is driven by the contents of a consent form and fine-tuned by the investigator using a *user interface*, designed for this purpose. The identification system uses state of the art algorithms in machine learning, natural language processing and data mining to analyze the file content during content based filtering and hence helps to extract the relevant files.

The data acquisition system will interact with the identification system to retrieve selected files in one or more phases in a forensically sound manner. Acquisition will also include data collection so that after acquisition, analysis can be performed on desired evidence. The data identification system and data acquisition system work with each other, one describes what data to extract and other actually exacts that data.

The verification system will extract the files along with their hash values. The extracted data is collected in the JSON format. Report is presented to the investigator. The verification system will ensure the integrity of the extracted data files obtained by the identification and acquisition systems and also ensure that no data through the extraction process is added to the mobile device.

Data Identification System

The data identification system provides the core functionality of the system. This system is responsible for identifying the most relevant files to be extracted from the device with the metadata filtering and content based filtering. The input to this system is based on the consent form which is turned in by the witness/victim. The data on smartphone can be found in different types, the basic categories are listed below.

- *Pictures, audio and video*
- *Call Logs*
- *Contact – Phonebook*
- *Text Messages*
- *Calendar*

Each of these categories are associated with metadata filtering that describes data which are based on location (place where the evidence was taken), data/ time (when was the evidence captured) and sender/receiver (for text and multimedia messages). Notice that by metadata it simply means information about the data in that category.

Content based filtering is achieved with the help of machine learning models. Photos from the gallery are filtered out which are broadly classified into weapons, vehicles, drugs and skin exposure. For forensic examination phase, the goal is to be able to identify files that may contain significant evidence of the purported crime. Depending on the type of crime being investigated, several different types of machine learning algorithms can be used either on their own or in conjunction with each other. In general machine learning algorithms can be divided into two broad classes, namely, supervised algorithms and the unsupervised algorithms for accomplishing the task in hand. Here for classification of images we use supervised learning neural network, namely, Convolutional Neural Network (CNN). Images are successfully analyzed and classified by the deep learning, feed forward network called as CNN .The MobileNet model are based on a streamlined architecture that uses depth wise separable convolutions to build light weight deep neural network (Howard et al., 2017). The mobile net is trained on the ImageNet (ImageNet_official) database. It is based on the WordNet hierarchy with total of approx. 14 million images with total of 1000 classes. The detail about the machine learning models used will be discussed further in the chapter.

Data Acquisition System

Data Acquisition System is used to collect and retrieve targeted data from devices. This phase interacts with identification system for its filtering categories. The process follows forensically sound manner for acquisition. The data retrieved will act as an important evidence to solve the case. System developed for data acquisition in this research is divided into two parts: System-on-chip called as “Targeted Data Extraction System (TDES) manager” which resides on a portable bootable device. The USB memory drive is used for this purpose, other part is the app, call “TDES app” which is deployed on the smartphone. First, the manger boots up in windows 10 OS when connected to any laptop or computer. Second the targeted smartphone is connected to the same workstation laptop and the TDES app is installed on the device. The investigator officer is presented with the graphical user interface in which he can input the filtering categories which is based on the consent form signed by the victim/ witness. Finally, the data from the targeted devices is exported to the TDES manager on the USB drive.

There are two main categories on the TDES app to filter data and are discussed in detail below.

On-device metadata based filtering Data on the phone can be classified into different categories and can also be filtered using different metadata filtering options associated with it. In Table 1, it is shown how the TDES app can extract data using the different metadata filtering, when deployed on the targeted devices. The first part of the table shows what all can be extracted from the TDES app and second part shows what cannot be retrieve currently with the application. Access to these native applications metadata is granted by the content provider. Content provider is an Android framework which helps an application to manage access to data stored by itself, stored by other apps and provide a way to share data with other apps. (developer.android.com)

Table 1

On-device Metadata based extraction

Category of Data	Metadata Type	Retrieved
Photos	Date & Time	Yes
Photos	Location	Yes
Photos	Album Type	Yes
Videos	Date & Time	Yes
Videos	Location	Yes
Call Logs	Date & Time	Yes
Call logs	Incoming Calls	Yes
Call Logs	Outgoing Calls	Yes
Call logs	Missed Calls	Yes
Call logs	Contact Name	Yes
Messages- SMS/MMS	Date & Time	Yes
Messages- SMS/MMS	Contact Name	Yes
Messages- SMS/MMS	Contact Number	Yes
Contacts	Name	Yes
Contacts	Number	Yes
Calendar	Date & Time	Yes
Notes	Date & Time	No
Web history	Date & Time	No
Emails	Date & Time	No
Third Party IM apps	Date & Time	No

TDES application interact with Cursor loader and Content resolver to retrieve data from the database files with the help of Content providers. Content resolvers method provides the basic CRUD (create, retrieve, update and delete) functions of persistent storage. Cursor loader helps to run the asynchronous query in the background and helps to accessing a content provider from the user interface. Figure 5 discusses the flow from application to database storage through content providers to access the metadata filtering categories.

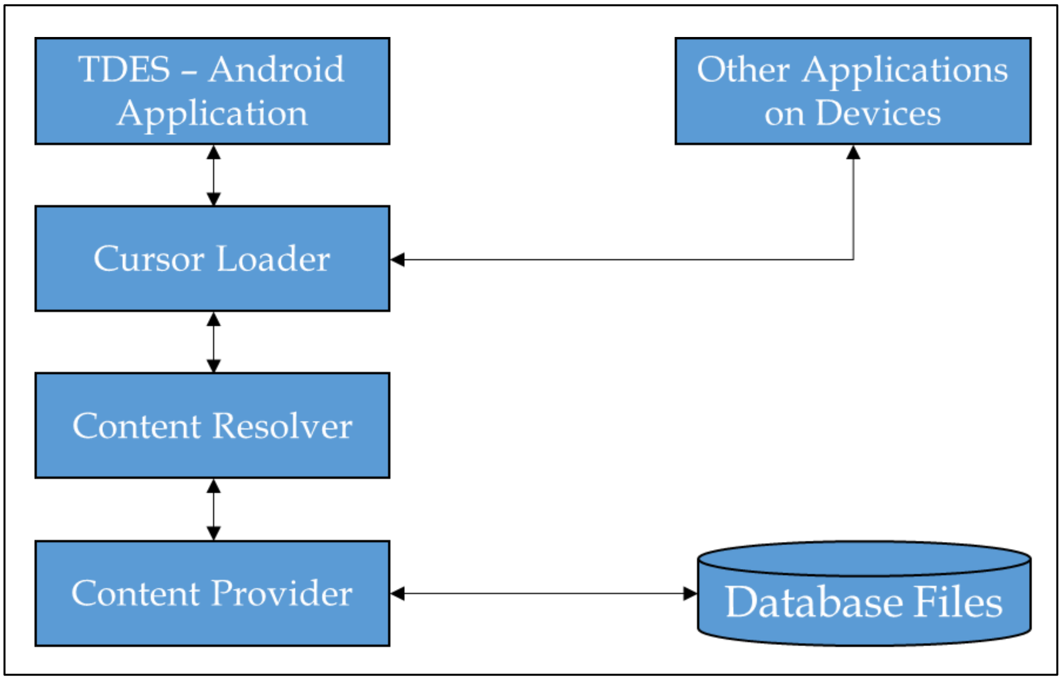


Figure 5. API connections in Android framework.

Each application on the Android devices has a unique content provider name associated with it. In our system, we used applications such as contacts, messages, call logs, media (images and videos) and calendars. Table 2 present each content provider's name which were used to retrieve data from these applications on Android devices.

Table 2.

Application and associated content provider

S.no.	Application Name	Content Provider Name
1	Contacts	ContactsContract.Contacts.CONTENT_URI ContactsContract.CommonDataKinds.Phone.CONTENT_URI ContactsContract.CommonDataKinds.Email.CONTENT_URI
2	Messages	content://mms-sms/conversations?simple=true
3	Calendar	content://com.android.calendar/calendars
4	Call Logs	content://call_log/calls
5	Images	MediaStore.Images.Media.EXTERNAL_CONTENT_URI
6	Videos	MediaStore.Video.Media.EXTERNAL_CONTENT_URI

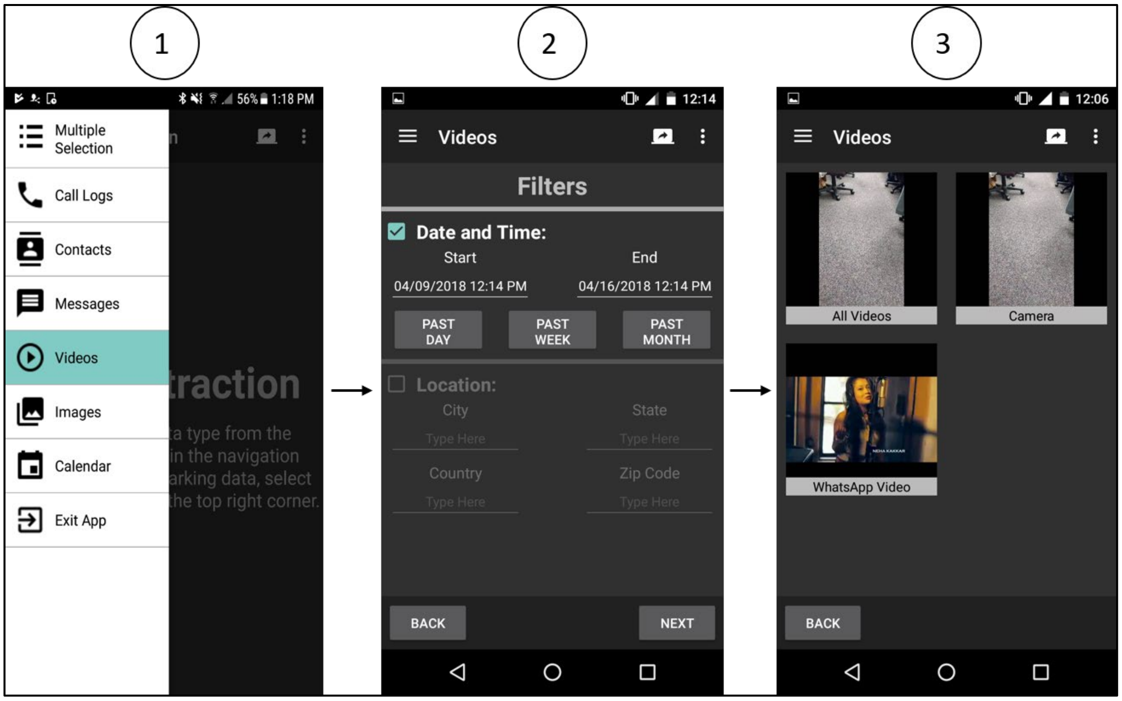
Root directory will have AndroidManifest.xml file of every application. The manifest presents essential details about the application to the Android system, these are information which is required to run the application. There is different permission level need to be set in the Android manifest file to use functionality of the content provider in the application. Table 3 discuss the permissions for each of these applications which needs to be included in the android manifest. The user interface for video, images and calendar is discussed in detail for the metadata filtering. Metadata filtering is based mainly on – date, time and location (City, State, Country and Zip code).

Table 3.

Application and associated permissions in Android manifest

S.no.	Application Name	Content Provider Name
1	Contacts	android.permission.READ_CONTACTS
2	Messages	android.permission.READ_SMS
3	Calendar	android.permission.READ_CALENDAR
4	Call Logs	android.permission.READ_CALL_LOG
5	Images	android.permission.READ_EXTERNAL_STORAGE
6	Videos	android.permission.READ_EXTERNAL_STORAGE

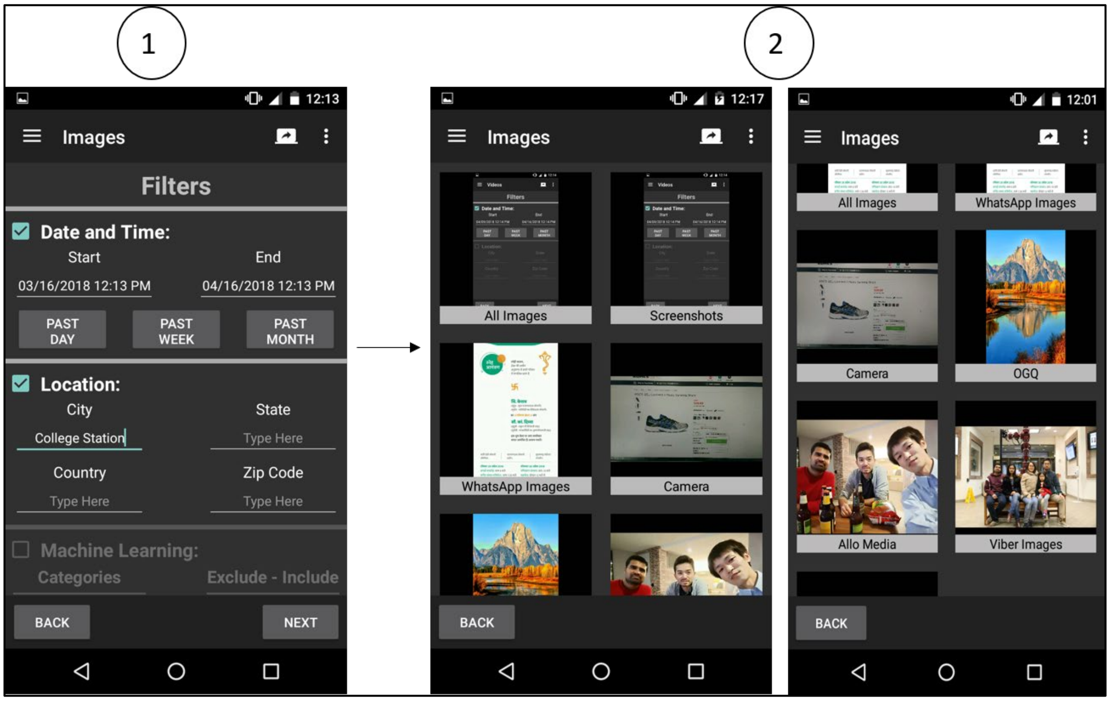
Figure 6 shows the user interface for the TDES app where videos are being selected on the main front page, denoted by screen (1). After selecting the video tab, second screen (2) shows the metadata filtering page, where the evidence can be filtered using date and time or/and location. In the date and time filter, a specific date range can be selected, or investigator have the option of selecting dates – past week, past month or past day. In the following example the filtering is done on the specific date range filtering. The third screen (3) display on all the videos falling in that date range. It displays the videos in folder, where *all video* folder has all the videos found during the filtering and other folders displays videos in specific categories such as camera, WhatsApp videos etc.



☑ Date and Time:
Start: 04/09/2018 12:14 PM
End: 04/16/2018 12:14 PM

□ Location:
City: Type Here
State: Type Here
Country: Type Here
Zip Code: Type Here

Figure 6. User Interface of TDES app showing video interface



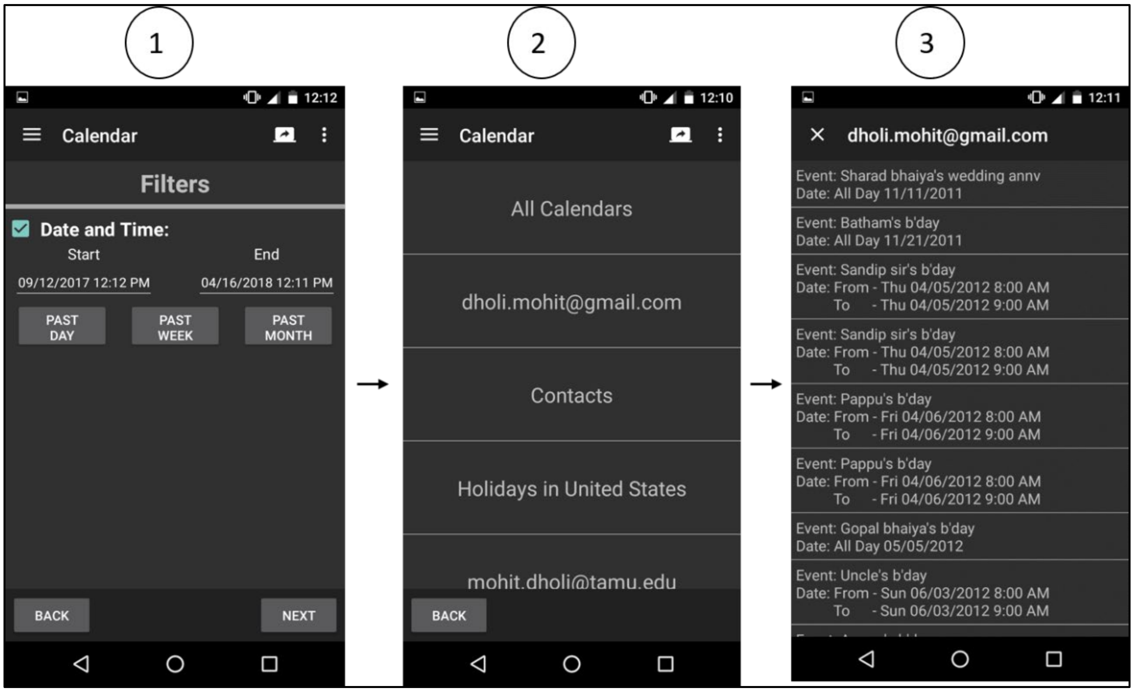
☑ Date and Time:
Start: 03/16/2018 12:13 PM
End: 04/16/2018 12:13 PM

☑ Location:
City: College Station
State: Type Here
Country: Type Here
Zip Code: Type Here

□ Machine Learning:
Categories
Exclude - Include

Figure 7. User Interface of TDES app showing images interface

Figure 7 shows the user interface for the TDES app where images category being selected on the main front page. In screen (1) filtering is done on the date and time and location. In location, city is being selected. Screen (2) shows all the images which are pulled from the different third-party application such as WhatsApp, Viber, Allo etc. It also collects images taken from the phone camera application which is a native application.



☑ Date and Time:
Start 09/12/2017 12:12 PM
End 04/16/2018 12:11 PM

Event: Sharad bhaiya's wedding annv
Date: All Day 11/11/2011

Event: Batham's b'day
Date: All Day 11/21/2011

Event: Sandip sir's b'day
Date: From - Thu 04/05/2012 8:00 AM
To - Thu 04/05/2012 9:00 AM

Event: Sandip sir's b'day
Date: From - Thu 04/05/2012 8:00 AM
To - Thu 04/05/2012 9:00 AM

Event: Pappu's b'day
Date: From - Fri 04/06/2012 8:00 AM
To - Fri 04/06/2012 9:00 AM

Event: Pappu's b'day
Date: From - Fri 04/06/2012 8:00 AM
To - Fri 04/06/2012 9:00 AM

Event: Gopal bhaiya's b'day
Date: All Day 05/05/2012

Event: Uncle's b'day
Date: From - Sun 06/03/2012 8:00 AM
To - Sun 06/03/2012 9:00 AM

Figure 8. User Interface of TDES app showing calendar interface

Figure 8 shows the user interface for the TDES app where calendar category is selected on the main front page of an application. In this example, date and time filter is used to retrieve all the calendar events on the mobile.

On - device Content-based filtering

Content based filtering is performed by deploying the supervised machine learning models in the TDES application on the mobile devices. Trained models can be used which are built using the deep neural network. Models can also be used that have been provided open source or it can be trained from the scratch by training it with the large amount of training data sets. A trained model can be viewed as a data structure which helps in classification of the images with the help of feature extraction techniques. In this study, both pre-trained model and retraining the last layer of the model was performed to test for better accuracy. Figure 9, the basic machine learning approach used during this study is discussed. Machine learning model, which is used to perform the image classification by using different neural network. After selecting the model, the training data set image is feed into the model. Next step is the feature extraction which deals with edge detection, sharpening of images. Once the model is trained on the last layer with the given trained images. It is ready to prediction. This process is achieved through a process call transfer learning. Transfer learning is process where model trained on one task is re-purposed on the second related task (TransferLearning.org) . The testing was performed using two models, Inception – v3 which is the model released by Google and the other, MobileNet. Both the models were tested and deployed using TensorFlow and TensorFlow Lite framework on the mobile devices. It was found that TensorFlow Lite outstands the other in terms of speed and space size of the application. The details steps about how the training part was performed is discussed in further chapter. Currently the tool can identify photos containing weapons, vehicles, drugs and shin exposure.

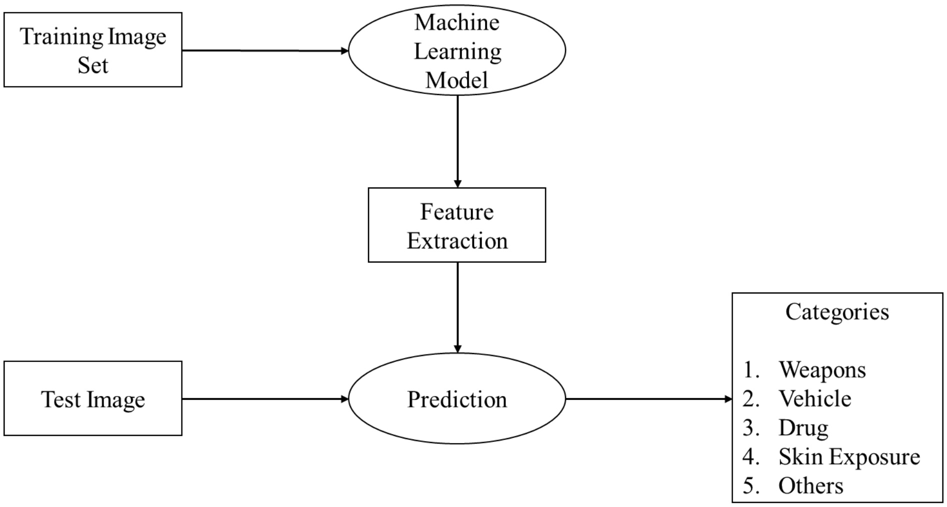
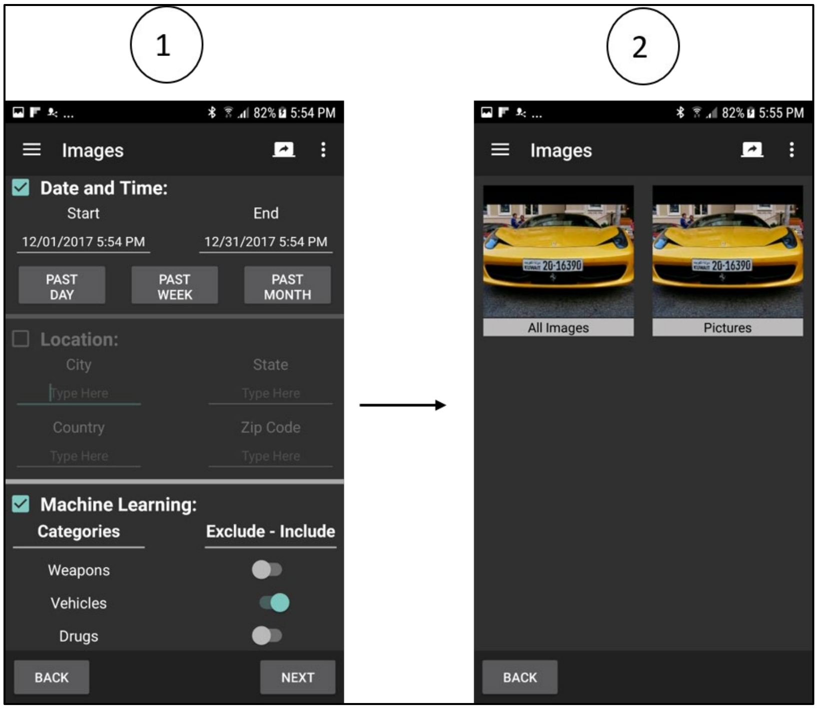


Figure 9. Machine learning approach

After last layer of the model is trained, testing Images from the mobile gallery are passed on it. The images falling into the given categories (weapon, drugs, vehicle, and skin exposure) can be passed and filtered out and images not in the category will come under the category of others. The model was integrated into the TDES app and content based filtering was performed on the images. Figure 10 and 11 shows the filtering results after selecting vehicles and weapons category respectively. The accuracy of the model is discussed in the later chapters.



☑ Date and Time:

Start: 12/01/2017 5:54 PM

End: 12/31/2017 5:54 PM

□ Location:

City: Type Here

State: Type Here

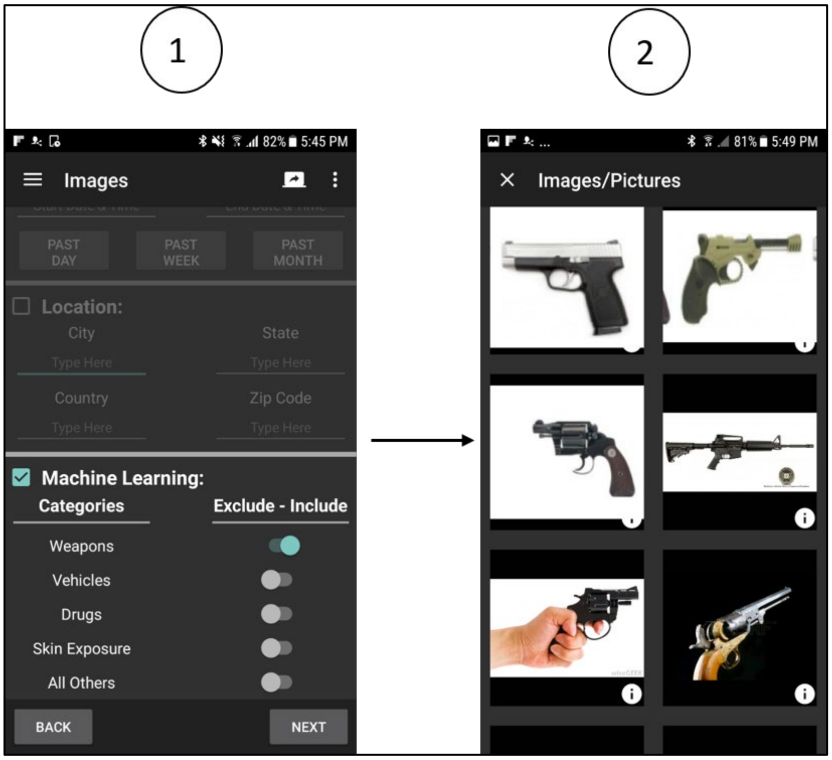
Country: Type Here

Zip Code: Type Here

☑ Machine Learning:

Categories	Exclude - Include
Weapons	○
Vehicles	●
Drugs	○

Figure 10. Content based filtering on weapons category



□ Location:

City: Type Here

State: Type Here

Country: Type Here

Zip Code: Type Here

☑ Machine Learning:

Categories	Exclude - Include
Weapons	●
Vehicles	○
Drugs	○
Skin Exposure	○
All Others	○

Figure 11. Content based filtering on weapons category

Data Validation System

Data Validation phase ensures that the data transferred from the TDES application after filtering to TDES manager on the USB drive follows a forensically sound manner approach. This is achieved by including appropriate hashing to insure the integrity of the data. The hashing technique used here is SHA-1. Server and Client communication is established between the TDES application and USB drive (TDES manager) during the data transfer. The data transfer is also written in the JSON file format through which the report can be generated which could be easily read by an investigation officers. The JSON structure describes the extracted data as well as additional information such as when the TDES app started to run, time it took to extract the data, the mobile EMI number. The information about the case number and investigators name.

Location of TDES app in Android Stack

The Android operating system is collection of software components. These components are divided into five sections namely – application layer, application framework, libraries, Android runtime and Linux kernel. The TDES app which was created in this study was deployed on the first layer, application, along with the other native and third-party application on the device. It uses the services such as Content provider and other frameworks such as (Activity manager, Resource manager and View system) from the below layer. Content provider is used to perform the data collection from the other native application on the devices. Activity manager helps in building the user interactive pages through which the investigator can input the desire filters. As the system has the capacity for content based filtering it uses the machine learning models. These models are placed in the second layer of the Android stack along with the other frameworks. As these models does not come default with the Android stack and is only included as a demand for this project. In the figure 12 it is highlighted with a different color. Third layer, libraries, which helps in accessing and sharing the database files with the application through SQLite libraries. Media framework libraries such as Media store helps in accessing the media database files such as images and videos. Android runtime is useful for running the device with the version 5.0 or higher and helps in running the code on low memory devices by executing DEX files .The last layer which act as foundation of the Android stack is the Linux Kernel which allows the security features and the device manufacturer to develop hardware drivers.

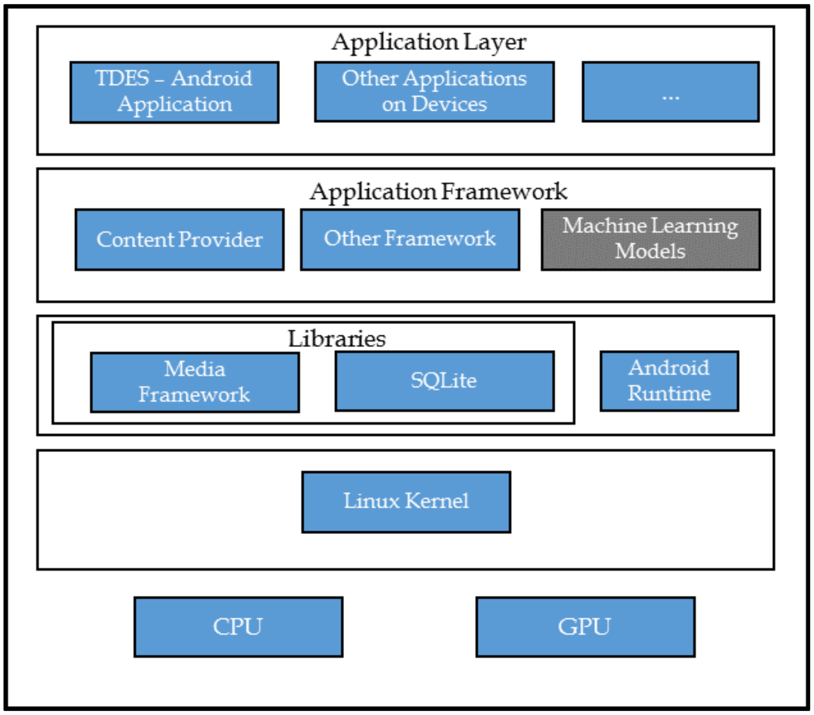


Figure 12. Android architecture stack

TDES Communication Paradigm

The system built during this study as a whole is called, TDES system. One of the important aspects of TDES systems is the communication between the TDES application, which reside on the targeted smartphone and TDES manager, which is responsible to carry out the data export. Figure 13 shows this communication. Investigator is provided with a portable TDES boot drive (USB drive), that is preloaded with the Windows 10, TDES manager app and other necessary tools such as Java JDK, ADB tool and device drivers required to install the TDES app to the target smartphone and get the extracted data back to the TDES manager. Investigator can use any laptop to boot the OS from the USB drive from TDES manager which run off an isolated environment on the portable TDES boot drive. The Internet connection is only needed to install the phone drivers if they are not already installed in the Windrow 10 which is booting from drive.

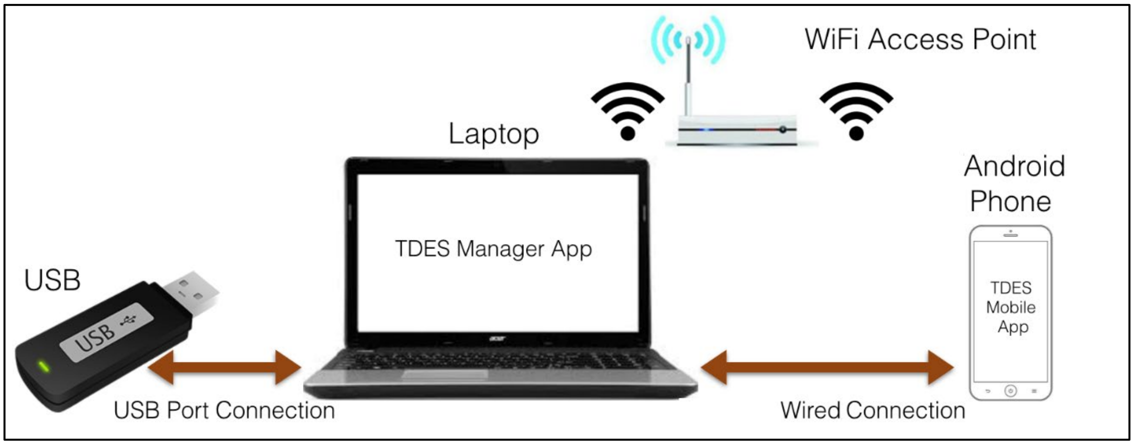


Figure 13. TDES communication paradigm

The steps followed for the selected data extraction are as followed:

1. The bootable USB drive containing the TDES manager and act as data repository for the extracted data is inserted into the workstation

computer/laptop and windows 10 OS boots up and the TDES manager app starts running.

2. A wired connection using the USB cable is connected between the Android device and the workstation laptop/computer. TDES app is installed automatically on the smartphone with the help of ADB install command coded in the TDES manager app.
3. A wired two-way communications channel is setup between the manger and the TDES app for the data transfer.

The selected data from the application is exported from the smartphone to bootable drive with all the hashes included. Each categories are transferred by making separate folders that is created on the file system of the bootable drive. The implementation ensure that eDiscovery principles are followed while transferring the files. It is taken into consideration that no copies of the data that to be exported are stored on the user's smartphone in any intermediate form.

Android TDES app Installation

The Android OS requires that every application being installed to a device must be signed. This signature process hashes the files within the application in turn generating a versioned application. This process ensures that if an application was maliciously tampered with and then attempted to be redistributed as an update, it would fail because there is no feasible way to replicate the key used in the signature process. Therefore, as long as the application developed is signed and does not attempt to update another application, it can be self-signed. The key used to sign the application is a Java Keystone (jks), which can be created using the Java Key tool. Once this has been generated, a versioned application can be created through Android Studio.

The output of the completed compilation is an apk file, which is the standard Android OS application extension and is used for installing any application on the Android device. In order to install the TDES app, its apk file must be on the target device. This can be done either using a wired or a wireless connection. Note that for Android no other authentication is necessary. Once the apk file is on the target device, the TDES app can be installed. For simplicity and ease this system uses Android Debug Bridge (ADB) a command line utility provided within Android SDK that allows for communication between the host computer and a target device, for installing the TDES app automatically. The only restriction to using ADB is that the target device must first be in USB Debugging Mode and once the installation is complete, this mode will be turned off.

TDES Data Transfer protocol

During the data transfer between the app and the manager, a major concern is to ensure forensic integrity, meaning no modifications should be done to the data. This is the data that is to be extracted using the app user interface. This is achieved through hashing files while exporting it to the manger. The hash values will ensure that no modification was done.

ADB that allows for communication between the host computer and the target device can be used. Simple file transfers can be done once communication is established but ADB also provides more capable commands to achieve this with ease. In particular, ADB allows for something called port forwarding. Simply put, this redirects data passing through the specified port on the host computer to the specified port on the target device.

With this setup completes, the target device can now create client sockets as needed to transfer data as many times as necessary. Android applications are natively written in Java so standard networking packages can be imported on both the target device app and host computer application. Specifically, we import "java.net.*" which allows for Server Sockets and Sockets to be created. The Server Socket waits for a client Socket to connect from the target device and we use input and output streams to gather the data.

User Interface of TDES application

The apps user interface helps the investigator to input the data with respect to the consent form which is filled by the victim/witness. An interesting and useful feature called bookmarking has been implemented in the TDES app. This feature works as follows. Suppose a dataset has been extracted using a set of filters. The investigator setting up these filters can display the results of the filtering and to do a quick data review on the phone itself before deciding what data to export to the TDES manager. For example, if a set of images of weapons in a certain time range has been selected, the investigator can do a review of the images to decide which subset of these are relevant to the investigation by selecting the relevant set.

Figure 14 shows the following:

1. Home page of the app.
2. There is a sliding selection menu which lists the categories: Call logs, Contacts, Messages, Videos, Images and Calendar through which the investigating officer extracts the data. There is an exit button too by which investigator can exit out of the application. After selecting any one of the categories will lead to the next screen.

3. Further filtering on the basis of metadata filtering can be done on such as Date and Time, Location and Name & Number. Images can be further filtered using the content based filtering based on weapons, vehicles, drugs and skin exposure.
4. Investigator is given an option to move forward by using next button which gives the option to bookmark the selected data which is extracted after the filtering.
5. Data Review button display all the bookmarked items and there is an export button at the bottom of the review page which helps in exporting the extracted data from the TDES app to TDES manager on the USB drive.

Table 4 discusses in detail how each of the categories can be further filtered into metadata and content based filtering and data can be extracted. Call logs, contacts, messages , calendar, Images and videos are categoried into Metadata filtering which is based on date & time, Name & number and location. Images and Videos can be further filtered using content based filtering which have the categories such as weapons, vehicles, Drugs and Skin Exposure.

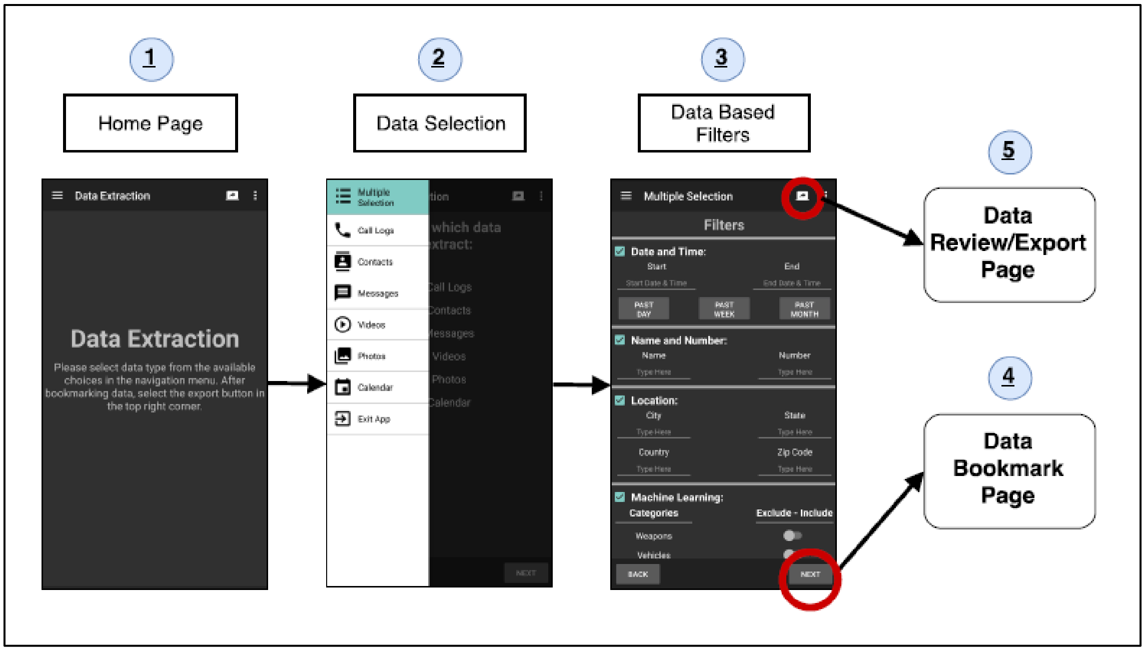


Figure 14. User Interface of Android application on the test device.

Table 4.

Categories classified into Metadata and Content based filtering

Categories/Filtering	Metadata Based			Content Based			
	Date & Time	Name & Number	Location	Weapons	Vehicles	Drugs	Skin Exposure
Call Logs	✓	✓					
Contacts		✓					
Messages	✓	✓					
Videos	✓		✓				
Image	✓		✓	✓	✓	✓	✓
Calendar	✓						

Exporting the Extracted Data

It's an important aspect that while transferring the extracted data from the smartphone app to the manager app on the bootable devices, its forensic integrity is maintained. In this system, JavaScript Object Notion (JSON) is used for the data transfer. JSON structure clearly describe all the extracted data, along with the hashes and other information related to the device which is used for the data extraction. All this information can be later used to generate a report. Generating a report is not included in this study but will be considered as a future work. Some of the information which will be included in this JSON files are: extracted information, time when the extraction was completed, phone information such as MAC address, MEID number, manufacture, model etc.

CHAPTER IV

Software Tools & Experiments Setup

This chapter describes the software tools and experiment setup which was employed in this research study. Android phones which were used during the test procedure are also discussed in this chapter.

Programming Languages

- **Java** is used as a programming language to code the TDES Android application to be run on the smartphone. Socket program written for exporting the data from TDES app to TDES manager also make use of this object oriented language. This language is used because of its portable nature.
- **JavaScript object Notation (JSON**) is a lightweight data-interchange format used to transmit the data items in plain text format to manager app on the bootable drive which could be later used for reporting purpose.
- **Python** scripts are used to retrain the machine learning models. This is helpful to do the image classification on content based filtering mechanisms provided in the app.

Software and Tool

- **Android Studio** is an official integrated development environment for Google's Android OS. Built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It also include Android SDK (Software Development Kit) and ADB (Android Debug Bridge) tool.

- **Java Development Kit** (JDK) is a software development environment used for developing Java application. This included Java runtime environment (JRE), an interpreter, compiler and an archiver (jar) etc. for java development.
- **Device Driver** is installed to support the particular type of device that is attached to a computer.
- **Windows 10** operating system is installed in the USB drive. This will act as a bootable drive attached to the laptop where the extracted data is stored.

Hardware

- USB drive, SanDisk ultra, 64GB, USB 3.0 flash drive is used for this experiment.
- USB cable, used for connecting the Android phone to the laptop.
- Laptop – HP Envy x360 m6 Convertible with i7 processor, 64 bit OS, 12 GB memory and 1TB storage is used.

Experiment setup

The main aim of this study is to extract the selected evidences from the Android phone and export it to a bootable drive with metadata and content based filtering. For conducting the experiments we divide the experiment phase into:

1. Test environment setup
2. Test data creation
3. Preparation for extraction

Test environment setup In this phase, setup of forensic workstation and selection of Android phone to conduct experiments are discussed. To consider the electronic evidence obtained from a phone, a forensically sound environment should be maintained which ensures the integrity of the evidences data hence all experiments are carried out keeping this principle in mind.

Selection of Android Phone At the start of this study, Android phones with the version 7.0 (Nougat) was the latest version. The Android phone available in our research lab during that time was brand new, Samsung Galaxy S7 with v7.0. We also conducted testing on other Android device, which was heavily used, Moto G3 running v6.0, Marshmallow, to test the application on different version and manufacture models. Both the testing phones were un-rooted. The test phone characteristics are included in Table 5.

Table 5.

Android Device used for Experiment

S.no.	Android Model	Version	Storage
1	Samsung Galaxy S7	“Nougat” - Android 7.0	32 GB
2	Moto G3	“Marshmallow”- Android 6.0	16 GB

Forensic Workstation setup As this system uses the USB drive to boot the windows 10 OS. We selected the USB- SanDisk ultra, 64GB, USB 3.0 flash drive to conduct the experiments. The USB drive needs to plug into the laptop/ desktop. This can be any laptop which the investigator can used. The laptop used for this experiment is HP. The USB cable is used to connect the Android phone and the laptop.

To prepare the bootable drive to recognize the test phone, Samsung and Moto G3 USB drivers for mobile phones were installed. The drivers were downloaded from Samsung and Moto G3 manufacture's website. Android SDK and Java JDK were installed on the bootable drive too. Installation of SDK on the workstation allows to use the ADB tool.

Test Data Creation As we brought a new Samsung phone for this study, it required to be populated. The test data was created by sending and receiving consecutive text messages, capturing images and videos. The contact list was loaded with random contact information. The calls were made so that the call logs can be generated. The communication was limited to four participants. The test phone, Samsung Galaxy S7 initiated the conversation in some chat and other responded and vice versa were also recorded. More than 100 images were stored in the gallery out of which 80 were taken from the phone and around 20 images were exchanged through third party applications or MMS. Some of the shared text messages and images were deleted intentionally to investigate whether content providers recover deleted activities. The activities performed on the test phone are listed below:

- Sent and received text messages

- Send and received MMS messages
- Installed third party application such as WhatsApp, WeChat, Viber and shared images.
- Received and dialed calls
- Stored the contact details on the phone
- Made calendar events
- Captured images, videos and audios
- Downloaded 84 images containing weapons, vehicles, drugs and skin exposure in them.
- Captured images and videos with the location GPS on.

Table 6 list the summary of count content of each categories in both the testing devices.

Table 6.

Count content of each categories in the testing devices.

Model	Photos	Videos	Messages	Call Logs	Contacts	Calendar
Samsung Galaxy S7	100	6	37	7	20	17
Moto G3	191	7	25,420	429	1889	780

Preparation for Extraction

To extract the data using the content provider along with ADB, Android phone must be preconfigured to enable *USB debugging* mode. The *Developer options* on an Android phone allows access to *USB debugging* mode. By default *Developer option* is hidden in some Android phones. On the test phone, the developer options were activated by using a one-time procedure: tapping build number for seven times enables the *Developer* *options*. *Build number* can be found under *apps → setting → About Device*. After enabling *Developer options, USB debugging* mode can be activated. Android test phone is now connected to the workstation with the help of USB cable to the USB port.

USB flash drive is attached to another USB port on the workstation desktop/ laptop. It is made bootable with the Windows 10 operating system. On this bootable drive ADB and JDK is installed. As to detect the testing phones, the respective drivers are also installed on this bootable drive.

CHAPTER V

Results

This chapter describes the data collected during the analysis and the main findings with respect to our research study. It also discusses the procedure followed during re-training of the machine learning models. The comparison study with the commercial tools is also presented.

Machine learning model used during experiments

The experiment was conducted to evaluate the system for accuracy and speed with respect to metadata and content based filtering. For the content based filtering, as discussed earlier, we use Machine Learning (ML) models for image classification. There are models available in the market which does the image classification but to narrow down our search to the specific categories which are specific to our research study and to also speed up the searching techniques we thought to retrain the models. After doing research it was found that the Inception v3 (Szegedy et al., 2015) and MobileNet (Howard et al., 2017)are two powerful open source model available which does the image classification with the higher accuracy when compare to other models.

Inception v3 and MobileNet both are trained on ImageNet database (ImageNet_official) which have a collected of 1.2 billion images divided into the 1,000 different categories. These categories are specified as label files in the model. We conducted the experiment with both the models with pre-trained and re-trained versions. It was found that MobileNet showed better results hence we decided to integrate the MobileNet model with our TDES app. TensorFlow platform was used to re-train the model.

Following section discusses the pseudocode followed to use the trained and re-trained model in our app.

Image classification Initially TensorFlow framework was used to integrate the ML model. Later with the release of TensorFlow Lite (TensorFlow.org) framework in 2018 which is specifically design to integrate the machine learning models with the smartphone were used to integrate MobileNet model within our TDES app. Our machine learning model is used to classify images into five categories: weapons, vehicles, drug, skin exposure and others (includes all images other than rest four categories). We have created a label file which are divided into these categories. Table 7 list the subdivided categories of label under each main categories.

Table 7.

Label categories for image classification

Weapons	**Vehicles**	**Drugs**	**Skin Exposure**
assault rifle	ambulance	medicine chest	bathing cap
letter opener	beach wagon	syringe	bath towel
cannon	bicycle-built-for-two	pill bottle	bathtub
chain	bobsled	beer glass	bikini
chain saw	cab	wine bottle	nipple
cleaver	container ship	face powder	brassiere
guillotine	convertible		shower cap
chopper	electric locomotive		snorkel
hammer	fireboat		swimming trunks
hook	fire engine		tub
missile	forklift		
power drill	freight car		
projectile	garbage truck		
revolver	go-kart		
screwdriver	golf cart		
	gondola		
	jeep		

(continued)

Weapons	Vehicles	Drugs	Skin Exposure
	jinrikisha		
	lifeboat		
	limousine		
	liner		
	minibus		
	minivan		
	Model T		
	moped		
	motor scooter		
	mountain bike		
	moving van		
	passenger car		
	pickup		
	pirate		
	police van		
	racer		
	recreational vehicle		
	school bus		
	schooner		
	snowmobile		
	snowplow		
	space shuttle		
	speedboat		
	sports car		
	steam locomotive		
	streetcar		
	submarine		
	tank		
	tow truck		
	tractor		
	trailer truck		
	tricycle		
	trolleybus		
	unicycle		
	warplane		
	horse cart		

Steps followed during the retraining and integrating the model in the app is listed in the form of pseudocode below.

Step 1 to 4, describe the process of retraining the last layer.

Step 1: Install the TensorFlow and its libraries on the workstation laptop.

Step 2: Collect the training data set (ImageNet and Google Image database). We used ~ 4000 – 5000 images for each of our four categories.

Step 3: Run the retraining python script from the TensorFlow framework.

Step 4: New graph file (Model file) and retrained label file is created.

Step 5-8, describes the process of adding the ML model in the application using TensorFlow.

Step 5: We need to have .so (shared object) file which is C++ compiled file and a jar file which will consist of JAVA API that will be calling the native C++. These 2 files need to be included in our Android project.

Step 6: Add the jar file in lib folder as library to the project.

Step 7: Add the .so file in the jniLibs folder in main director of the Android project.

Step 8: Add the label file and pre-trained/ re-trained model graph files in the assets folder of the Android project.

With the release of TensorFlow Lite Framework Step 5 – 7 is reduced into a single step. Step 9 describe this integrating of ML using TensorFlow Lite framework.

Step 9: Add the pre-complied TFlite Android ARR (*Compile’ org.tensorflow:tensorflow-lite:+’*). This is done by adding the dependency in the apps ‘build.gradle’ file in Android studio project.

Experiment results

After integrating the ML model to the app. We conducted a series of experiments to evaluate the extraction time and ML accuracy while using the content based filtering. The filtering on metadata based filtering was also conducted. The consent form which was filled by the victim/ witness was used by the investigator to input the data in the application and extraction process was based on it.

On-Device Metadata based filtering Table 8 shows the results for a series of experiments for on-device metadata based filtering for testing Android phones. For this experiment we used both the devices, Samsung S7 and Moto G3 and show the artifacts retrieved as a fraction of the total number of expected artifacts for the filter. The Disp. time is time it extracted data is displayed on the app and the Exp. Time is the time it takes to export the data from the app to manager. The Size column give the storage size after the export to the TDES manager app on the bootable device. It was found that on device metadata filtering had 100% accuracy.

Table 8.

On device metadata based filtering

Category	Meta-Filter	Samsung Galaxy S7 Dev I - Artifacts	Moto G3 Dev II - Artifacts	Disp. Time-I (sec)	Disp. Time-II (sec)	Exp. Time-I (sec)	Exp. Time-II (sec)	Size-I	Size-II
1-Photos	Date: 04/07/17- 02/05/18	100/100	101/191	2.06	0.83	12.03	13.13	10.3 MB	12.3 MB
2-Photos	Date: 02/03/18 - 02/05/18	22/100	2/191	0.86	0.31	4.65	2.32	9.50 MB	6.56 MB
3-Photos	Location: Current Location	4/100	1/191	0.56	0.63	2.5	10.58	4.50 MB	46.1 MB
4-Videos	Date: 12/19/17 -02/03/18	1/6	3/7	0.45	0.89	1.08	3.91	9.45 MB	16.6 MB
5-Videos	Location: Current Location	6/6	7/7	1.69	0.9	10.76	13.09	144 MB	190 MB
6-Calender	Date: 05/29/17 - 05/30/17	1/17	85/780	0.62	1.03	1.02	2.4	1 KB	13 KB
7-Calender	Date: 06/01/17 - 06/27/17	5/17	1/780	0.83	1.85	1.25	1.03	4 KB	2 KB
8-Messages	Date: 08/01/17 - 09/20/17	37/37	987/25,420	0.69	1.23	3.46	13.34	8 KB	16 KB
9-Messages	Name: aaabb	14/37	32/25,420	1.02	1.23	3.56	14.23	5 KB	7 KB
10-Messages	Number: +*(***)***-***	1/37	5/25,420	0.42	0.92	1.02	1.25	2 KB	4 KB
11-Call Logs	Date: 08/07/17 - 08/08/17	4/7	8/429	0.22	0.23	1.28	3.2	4 KB	5 KB
12-Call Logs	Name: aaabb	1/7	9/429	0.22	0.49	1.02	6.59	2 KB	5 KB
13-Call Logs	Number: +*(***)***-***	2/7	11/429	0.47	0.89	1.99	11.2	2 KB	6 KB
14-Messages	Number: +*(***)***-***	4/37	100/25,420	1.02	1.25	11.02	14.08	154.8 MB	199.1 MB
Photos	Date: 01/28/18 - 02/05/18	23/100	6/191						
Videos	Location: Current Location	6/6	7/7						
15-Photos	Date:12/20/17 - 01/16/18	2/100	5/191	0.56	0.98	5.02	9.68	58.89 MB	94.02 MB
Videos	Date: 12/20/17 - 01/16/18	1/6	2/7						
16-Messages	Date: 12/12/17 - 02/05/18	1/37	1000/25,420	0.44	1.02	0.98	3.89	3 KB	258 KB
Call Logs	Number: +*(***)***-***	1/7	8/429						

On-device Metadata and Content based filtering In Table 9, on device content based filtering is done on the Samsung galaxy S7 Android device to test the accuracy of the machine learning model. "Exp. Result" gives the count of the expected number of images output where else results column gives the actual results extracted from the app. False positive give the count of number of images wrongly identified in the given categories. Accuracy gives the correctness percentage of the MobileNet model. There we totally 84 images used for this experiment containing various object in them.

Table 9

On-device Metadata and Content based filtering

Category	Meta-Filter	Content-Filter	Exp. Results	Samsung Galaxy S7 I - Results	False Positives	Disp. Time-I (sec)	Exp. Time-I (sec)	Accuracy Measure (%)	ML Model
1-Photos	Date: 11/12/17 - 02/02/18	Weapons	21	14	2	35	1.5	57	MobileNet
2-Photos	Date: 10/03/17 - 10/10/17	Weapons	3	2	0	1.6	0.33	66.6	MobileNet
3-Photos	Location: Current Location	Weapons	3	3	0	1.3	0.81	100	MobileNet
4-Photos	Date: 10/12/17 - 12/02/17	Vehicles	4	2	2	37.4	1.56	50	MobileNet
5-Photos	Date: 12/29/17 - 01/24/18	Vehicles	1	1	0	60	1.2	100	MobileNet
6-Photos	Location: Current Location	Vehicles	2	2	0	1.4	1.3	100	MobileNet
7-Photos	Date: 12/01/17 - 01/13/18	Drugs	6	1	0	34.69	1.2	17	MobileNet
8-Photos	Location: Current Location	Drugs	2	0	0	1.2	0	0	MobileNet
9-Photos	Date: 08/11/17 - 12/31/17	Skin Exposure	13	9	1	33.08	2.48	62	MobileNet
10-Photos	Location: Current Location	Skin Exposure	0	0	-	1.7	0	100	MobileNet

Comparison with Commercial Tools

We also compared the performance of the system against two commercial tools. Paraben (Paraben_official, 2018) and Magnet Axiom (Magnetaxiom_OfficialSite, 2018) which are currently in use by the law enforcement for extracting evidence from the phone. As we have already stated that as per our knowledge there are no tool in the market which does the selective extraction in the manner which is followed by our system. These tools do the physical extraction on the phone and then allow the investigator to analysis it off-line. We conducted the experiment on the Samsung Galaxy S7 devices. Results are discussed in Table 8. The app installation time (AIT) denoted time the testing devices is connected to the workstation laptop to the time the data is extracted from the app. Our system stores the data in the TDES manager app which is located on the bootable drive after extraction, but Paraben and Magnet AXIOM stores the data in the hard drive of the laptop.

Magnet AXIOM does the backup based acquisition, it means that it takes whole backup first and then the later the extraction can be performed. It should be noted that we need to calculate this overhead time for backup. Table 10 list this under Backup Acquisition Time (BAT). For example, it can be seen that it takes 29 min to do the backup and after that for example we need retrieve the contacts it takes additional 1 m 11 sec. When we compare this with our system it takes only 14 secs to install and then take only 1sec to retrieve the contact information from the device. Both the commercial tools can extract the data from the boarder categories unlike our system. Paraben has only one option to extract all media which includes videos, audios and images. However, in our experiment it was observed that the selecting this option resulted in extraction of only metadata information rather than actual media files.

Table 10

Export time comparison with commercial tools

Model	Summary	Export Time for TDES (USB)	Export Time For Paraben (HDD)	Export Time Magnet AXIOM (HDD)
Samsung Galaxy S7	App Installation Time (AIT)	14s	5s	NA
	Backup Acquisition Time (BAT)	NA	NA	29m
	Call Logs	1s	40s	1m 17s
	Messages	4m 9s	17m 3s	1m 21s
	Photos	42s	NA	14m 41s
	Videos	14 s	NA	1m 38s
	contacts	1s	2m 11s	1m 11s
	Calendar	6s	1m 5s	1m 14s
	All Media	NA	43 s (Metadata only)	NA

Summary

This chapter presented the results found after doing the metadata and content based filtering. It also discussed the accuracy of the system and compared the system with the two of commercial tool available in the market.

CHAPTER VI

Conclusion and Future Work

This chapter provides an interpretation of the results that were presented in the previous chapter and the conclusions that can be drawn from this study. It also discusses the future work which can be developed on this study.

Research Question

The research question for this study was to "How to develop a forensic tool that does the selective extraction from Android devices, which in turn preservers the user's and data privacy?" To answer this question, the content provider were used to extract data from the device on the bases of metadata and content based filtering. Exported data was transferred on to our system on USB drive with the respective folders and hash values included for the investigator to examine.

Analysis of results obtained during investigation

Consent form was filled by the witness/victim and turned in to the investigator agent. He then inputs the details in the TDES application and retrieves the information. It was found that the metadata filtering had 100 percent accuracy rate as it uses the built in API to extract the data. The content based filtering which is performed on photos in the gallery uses the machine learning model, Mobile Net, which was pertained on ImageNet dataset. Our system deals with classifying photos under specific categories. So re-training of the last layer on the machine learning model was done on the specific categories. It was found that the accuracy percentage varies between each categories, ranging from 57 to 100 percentage accuracy.

The exporting part of the system, which involves transfer of extracted data from TDES app to TDES manager app on the USB drive, results in robust speed. For example for the file size of 3 KB it takes approx. 0.98 sec to export the results.   The transfer rate out performs other commercial tools. The hashes values also confirm that the data was transferred without any modifications and followed the forensic principals.

Conclusion

The system built in this study was successfully able to do the selective extraction and is one of a kind of a system developed currently. It preservers user's and data privacy at the same time.

Future work

There is always a scope to increase the spectrum of this system which will help the investigator to extract more data from the targeted device. Below are some future research which can be performed:

- Study need to be performed to retrieve the messaging data from the Instant messaging application.
- The JSON file which was generated in this research can be used to build into report in the HTML format.
- The sentiment analysis on the text messages can be done so that the communication which might have malicious content can be filtered out easily.
- Collection of deleted files during the extraction can be performed.
- Adding content based machine learning filtering for video application.

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., . . . Zheng, X. (2016). *TensorFlow: A System for Large-Scale Machine Learning*. https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

Ahmed, R., & Dharaskar, R. (2008). *Mobile Forensics: an Overview, Tools, Future trends and Challenges from Law Enforcement perspective*. 6th international conference on e-governance, iceg, emerging technologies in e-government,m-government.

AndroidSQLiteInc. https://developer.android.com/guide/topics/data/data-storage.html.

Ayers, R. P., Brothers, S., & Jansen, W. (2014). Guidelines on Mobile Device Forensics. *Special Publication (NIST SP) - 800-101 Rev 1*.

Barmpatsalou, K., Damopoulos, D., Kambourakis, G., & Katos, V. (2013). A critical review of 7 years of Mobile Device Forensics. *Digital Investigation, 10*(4), 323-349.

Cellebrite_Wiki. (2018). https://en.wikipedia.org/wiki/Cellebrite.

Cho, K., Merrienboer, B. v., Gülçehre, Ç., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *CoRR, abs/1406.1078*.

D. Reisinger. (2017). *Tech.Rep*.

developer.android.com. https://developer.android.com/guide/topics/providers/content-providers.html.

Encase_official. (2018). https://www.guidancesoftware.com/encase-forensic.

Google Inc., D. https://developer.android.com/guide/platform/index.html.

Grispos, G., Storer, T., & Glisson, W. B. (2011). A comparison of forensic evidence recovery techniques for a windows mobile smart phone. *Digital Investigation, 8*(1), 23-36. doi:https://doi.org/10.1016/j.diin.2011.05.016

Hoog, A. (2011). Android Forensics. Investigation, Analysis and Mobile Security for Google Android.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., . . . Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR, abs/1704.04861*.

ImageNet_official. http://www.image-net.org/

James Bergstra, Frédéric Bastien, Olivier Breuleux, Pascal Lamblin, Razvan Pascanu, Olivier Delalleau, . . . Bengio., Y. (2011). Theano: Deep learning on gpus with python. *NIPS 2011, BigLearning Workshop, Granada, Spain, 3*.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., . . . Darrell, T. (2014). *Caffe: Convolutional Architecture for Fast Feature Embedding*. Paper presented at the Proceedings of the 22nd ACM international conference on Multimedia, Orlando, Florida, USA.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet classification with deep convolutional neural networks*. Paper presented at the Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, Lake Tahoe, Nevada.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*, 436. doi:10.1038/nature14539

Lessard, J., & Kessler, G. C. (2010). Android Forensics: Simplifying Cell Phone Examinations. *Small Scale digital Forensics Journal, 4*(1).

Li, D., & Dong, Y. (2014). *Deep Learning:Methods and Applications*: Now Foundations and Trends.

Magnetaxiom_OfficialSite. (2018). https://www.magnetforensics.com/magnet-axiom/.

Mahadeokar, J. (2017). opennsfw. *https://github.com/yahoo/open_nsfw*.

Maynard Yates, I. (2010). *Practical investigations of digital forensics tools for mobile devices*. Paper presented at the 2010 Information Security Curriculum Development Conference, Kennesaw, Georgia.

Mohindra, D. (2008). The android project: Incident response and forensics [Computer software manual]. *https://www.slideshare.net/peterbuck/the-android-project-a-project-by-incident-response*.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2014). *Foundations of machine learning.*

Morum de L. Simão, A., Caús Sícoli, F., Peotta de Melo, L., Deus, F., & de Sousa Junior, R. (2011). *Acquisition and Analysis of Digital Evidence in Android Smartphones* (Vol. 6).

MPE+. (2018). https://accessdata.com/product-download/mpe.

Murphy, K. P. (2012). Machine learning: a probabilistic perspective. *MIT press*.

Paraben_official. (2018). https://www.paraben.com/.

Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development, 3*(3), 210-229. doi:10.1147/rd.33.0210

Silla, C. (2015). WeChat forensic artifacts: Android phone extraction and analysis
Purdue University.

Statista. (2017). Market-share-forecast-for-smartphone-operating-systems

https://www.statista.com/statistics/272307/market-share-forecast-for-smartphone-operating-systems/.

StatistaReport. (2017).

Number of available applications in the Google Play Store from December 2009 to December 2017

https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). Rethinking the Inception Architecture for Computer Vision. *CoRR, abs/1512.00567*.

Techcrunch. (2017). Google-has-2-billion-users-on-android-500m-on-google-photos

https://techcrunch.com/2017/05/17/google-has-2-billion-users-on-android-500m-on-google-photos/.

TensorFlow.org. https://www.tensorflow.org/mobile/tflite/.

TensorFlow.org. *https://www.tensorflow.org/tutorials/image_retraining*.

TransferLearning.org. *https://machinelearningmastery.com/transfer-learning-for-deep-learning/*.

Vijayan, V. (2012). Android forensic capability and evaluation of extraction tools.
Unpublished master's thesis,Edinburgh Napier University.

Wikipedia, & Rooting. Android Rooting

https://en.wikipedia.org/wiki/Rooting_(Android).

Y. Willassen, S. (2008). *Forensic Analysis of Mobile Phone Internal Memory*.

Yngvar Willassen, S. (2003). *Forensics and the GSM Mobile Telephone System* (Vol. 2).

VITA

Khushboo Rathi

Education

Master of Science Student in Computing and information Science at Sam Houston State University, May 2016- present. Thesis title: “Selected Forensic Data Acquisition from Android Devices”

Master of Science in Software Engineering, 2006-2011(5 years Integrated) at Vellore Institute of Technology, Vellore, Tamil Nadu, India.

Academic Employment

Graduate Research Assistant, Department of Computer Science, Sam Houston State University, January 2017- April 2018. Research activities includes requirement analysis, designing, documenting and coding forensic tool for NIJ grant project.

Graduate Teaching Assistant, Department of Computer Science, Sam Houston State University, September 2016- December 2016. Responsibilities include: assisting professor with grading, proctoring exams, tutoring labs for undergraduate students.

Publication

Rathi K., Karabiyik, U., Aderibigbe, T., Chi, H., "Forensic Analysis of Encrypted Instant Messaging Applications on Android", 6th International Symposium on Digital Forensic and Security (ISDFS), Turkey. (Under print)

Targeted Forensic Data Extraction from mobile device – US Patent pending- filled in 2018

Academic Awards

Office of Graduate Studies Scholarship, Office of Graduate Studies, Sam Houston State University, Fall 2016-2017, Spring 2016- 2018

Graduate Student Scholarship, College of Science Engineering and Technology, Sam Houston State University, Fall 2017, Spring 2018

Scholarship from the Conference committee to attend, “Women in Cyber Security conference (WiCyS)” 2018 at Chicago.

Cisco Travel Fund to attend “Women in Cyber Security conference (WiCyS) 2018” at Chicago.

Travel Grant scholarship, Office of Graduate students, Sam Houston State University, Spring 2018.

Work Experience

Business Associate, Blend Financial Services, July’12-Nov’14, Chennai, Tamil Nadu, India. Responsibilities includes: ETL development, unit testing, and requirement analysis.

Program Analyst, Cognizant Technology Services, May’11- June’12, Chennai, Tamil Nadu, India. Responsibilities includes: software development, Requirement analysis, coding, and deployment in production server and documentation.

Software Engineering Intern, Dell EMC, May’18 - July ’18, Round Rock, Texas, USA. Responsibilities includes: Research and software development, OS Linux server POC projects.