Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Telecommunication Engineering Graduate Program

Enhancing Mobile Banking Service Availability Using Machine Learning

Prepared by:
Said Ahmed Said

Advisor:
Dr. Murad Ridwan

A Thesis Submitted to School of Electrical and Computer Engineering in Partial Fulfillment for the Degree of Master of Science in Telecommunication Engineering.

October 2018
Addis Ababa, Ethiopia

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Telecommunication Engineering Graduate Program

This is to certify that the thesis prepared by Said Ahmed Said, entitled: *Enhancing Mobile Banking Service Availability Using Machine Learning* and submitted in partial fulfilment of the requirements for the degree of Master of Science (Telecommunication Engineering – Telecommunication Information Systems Track) complies with the regulation of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Dr. Yalemzewd Negash	_____	_____
Chair or School Dean	Signature	Date
Dr. Murad Ridwan	_____	_____
Advisor	Signature	Date
Dr. Yalemzewd Negash	_____	_____
Examiner	Signature	Date
Dr. Surafel Lemma	_____	_____
Examiner	Signature	Date
_____	_____	_____
Director of Post Graduate Program	Signature	Date

Declaration

I, the undersigned, declare that this MSc thesis is my original work, has not been presented for fulfillment of a degree in this or any other university, and all sources and materials used for the thesis have been acknowledged.

Said Ahmed	_____	_____
Author	Signature	Date

This thesis has been submitted for examination with my approval as a university advisor.

Dr. Murad Ridwan	_____	_____
Advisor	Signature	Date

Dedication

This research work is dedicated to my mother Merema Abdella and father Ahmed Said.

Abstract

One of the main obstacles for adoption of mobile banking is that of security concern. This concern is becoming a reality in the case of mobile core inter-node protocol, Signaling System number 7 (SS7). SS7 was developed with the assumption of trusted network within and among operators. With growing number of value-added service providers and roaming partners connecting to operators, the trusted network is no longer a closed network. Attackers continue to exploit vulnerabilities of SS7 network to conduct attacks that compromise confidentiality, integrity and availability of mobile banking users and mobile network operators. In Ethiopia, Short Message Service (SMS) and Unstructured Supplementary Service Data (USSD) are mainly used for mobile banking. These services are both vulnerable to availability attacks.

This thesis is an effort to detect SMS availability attacks on Mobile Application Part (MAP) layer of SS7. To mitigate these attacks, machine learning techniques using real SMS traffic data from ethio telecom is used for adaptive detection of abnormal SMS. A novel approach of using aggregation of Message Origination (MO) error codes is proposed for class feature extraction. A combination of expert judgments, literature reviews and information gain are used for optimal feature selection. As a result, it is recommended to use origination, destination, and mobile switching center address and write time as optimal features. To solve the problem of attack message detection, PART, Random Forest and J48 algorithms are compared. It is found that J48 has a superior performance with an accuracy of 98.6465% and model build time of 3.71 seconds.

Keywords: Mobile Banking, SS7, DoS, DDoS, Availability, Machine Learning, SMS, USSD

Acknowledgment

First, I would like to thank ALLAH for all the success in my life. This thesis would not have been successful without the support and assistance of my advisor, Dr. Murad Ridwan. I would like to express my gratitude and sincere appreciation for his guidance, valuable advice, supervision, and encouragement.

It is my pleasure to thank the School of Electrical and Computer Engineering and all the academic staff for their strong commitment and support. I also wish to thank my classmates for their valuable advises throughout my study.

Especially, I want to thank Negasi Fikre (IT technical specialist of value-added services) from ethio telecom for providing me all the necessary short message data, documentation and expert judgment on short message service. I would also like to thank ethio telecom for sponsoring my studies.

I would like to thank my family who supported and encouraged me throughout my study and my research work.

Finally, I would like to thank all of you who have been supportive throughout my study and this thesis work. Many thanks to you all, even if I did not mention you by name here.

Table of Contents

List of Figures

List of Tables

List of Annexes

List of Acronyms

2G/3G/4G/5G	Second/Third/Fourth/Fifth Generation
3GPP	3rd Generation Partnership Project
API	Application Program Interface
AuC	Authentication Center
BSC	Base Station Controller
BTS	Base Transceiver Station
CAPS	Call Attempt per Second
CDMA	Code Division Multiple Access
CN	Core Network
DoS	Denial of Service
DDoS	Distributed Denial of Service
CRISP-DM	Cross Industry Standard Process for Data Mining
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communication
GMSC	Gateway Mobile Switching Center
GW	Gateway
GT	Global Title
HLR	Home Location Register
HTTP	Hyper Text Transfer Protocol
HSS	Home Subscriber Server
IDS	Intrusion Detection Systems
ITU	International Telecommunication Union
Kc	Encryption Key
KDD	Knowledge Discovery in Databases
MAP	Mobile Application Protocol

ML	Machine Learning
MNO	Mobile Network Operator
MS	Mobile Station
MSISDN	Mobile Station International Subscriber Directory Number
MSC	Mobile Switching Center
NBE	National Bank of Ethiopia
NIDS	Network Intrusion Detection Systems
PART	Partial decision tree
PIN	Personal Identification Number
RAND	Random Number
ROC	Receiver Operating Characteristic
SDCCH	Slow Dedicated Control Channel
SEMMA	Sample Explore Modify Model Assess
SGSN	Serving GPRS Support Node
SIGTRAN	Signaling Transport
SIM	Subscriber Identity Module
SMS	Short Messaging Service
SMPP	Short Message Peer-to-Peer Protocol
SRES	Signed Response
SS7	Signaling System Number 7
STP	Signaling Transfer Point
TN	True Negative
TP	True Positive
TPR	True Positive Rate
UMTS	Universal Mobile Telecommunication System
USSD	Unstructured Supplementary Service Data
VPN	Virtual Private Network
VLR	Visitor Location Register
WEKA	Waikato Environment for Knowledge Analysis

1. Introduction

Mobile banking is a term used to refer to conducting banking transactions using a mobile device and a mobile network. According to National Bank of Ethiopia fourth quarter report for 2017, one branch in Ethiopia serves 22,164 people on average and of the total 4,257 branches thirty-three percent are in Addis Ababa [1]. The few number of branches show that Ethiopia has a large population that is unbanked. On other hand, ethio telecom has 57.34 million [2] subscribers which can be potential mobile banking users. Mobile banking has significant contribution in banking the unbanked. This has an impact on the national economy of the country.

The main obstacle in the adoption of mobile banking is the perceived notion of the lack of security [3]. One way to clear out this perception is by conducting research on existing infrastructure and services used for mobile banking and proposing mitigation strategies. One of the security concern areas whose vulnerability has been a focus of media [5,6] and research, in recent years, is the protocol used for communication inside an operator's core network which is Signaling System number 7 (SS7). The services used for mobile banking in Ethiopia are Unstructured Supplementary Service Data (USSD) and Short Message Service (SMS), which both use SS7 as backend protocol.

Ethio telecom is the sole telecom service provider in Ethiopia, which provides mobile network and services for mobile banking service providers. It is the responsibility of ethio telecom to ensure that mobile banking network and services are secure. In this thesis, assessment of application level SMS availability attacks is made with the intention of proposing a method for detecting attacks on availability of SMS service. Machine learning techniques are used for proposing an adaptive detection using optimal set of features and best performing algorithm.

1.1 Statement of the Problem

SS7 is the backbone protocol for inter-node communication in a mobile network. This protocol has been designed with the assumption of a closed trusted network. As a result, no security layers have been added to this protocol. However, due to the growing number of mobile roaming agreements that require SS7 communication between operators, direct connection of value-added service providers to an operator's core network, and the installation of Femtocells in physically unsecured areas has opened the closed SS7 network of an operator. Once an access is obtained to an operator's SS7 network, attackers can conduct SS7 attacks [4]. The vulnerability of SS7 poses a threat to subscriber's data confidentiality and integrity as well as availability of an operator's network and services [4]. It has also been reported that mobile banking frauds are being conducted using these vulnerabilities [5, 6]. Several studies [7, 8, 9, 10] also indicate that USSD and SMS information are visible inside an operator's network including sensitive information such as Personal Identification Number (PIN) number.

In ethio telecom, it is observed that there are large number of unusable messages/session attempts that are overloading both USSD and SMS centers. Complaints are also received from mobile banking service providers that they are receiving large amount of request that are not valid. These unknown large session and message attempts have an impact for ethio telecom as it consumes mobile originating (MO) license and could compromise the availability of the services leading to denial of service if license threshold is reached. In addition, it can also force ethio telecom to expand its license without a usable valid cause. Considering this and gradual findings of more SS7 vulnerabilities and new mobile malwares, it is necessary for ethio telecom to have an adaptive detection and mitigation method. In this research, the problem of detecting abnormal SMS application level messages is addressed. These messages have an effect on service availability and are causing message blockage. Messages originate from SS7, mobile malwares and intentional attack from mobile users.

1.2 Objective of the Study

1.2.1 General Objective

The general objective of this research is to detect SMS application level abnormal messages that have effect on SMS service availability using machine learning techniques.

1.2.2 Specific Objectives

To achieve the general objective of this research the following specific objective are identified:

- Critically review current SMS application level security issues on availability at ethio telecom.
- To identify types and source of SMS availability attacks.
- To study relevant literatures and select optimal important features that can assist in distinguishing abnormal messages from normal messages.
- To study relevant literatures and select three candidate machine learning algorithms for performance comparison.
- To capture real data from ethio telecom network for data analysis by selecting and pre-processing the final data set from initial raw data.
- To train and build models for application level availability attack detection using optimal set of features.
- To evaluate performance of selected algorithms, propose the best algorithm for detection, point of detection and show effect of detection.
- To propose architecture for detection deployment.

1.3 Scope and Limitation

1.3.1 Scope

This research focuses on enhancing mobile banking service availability security by detecting attacks on availability of SMS as a showcase. Detection stages that are covered in this research are data collection, pre-processing and detection.

1.3.2 Limitations

In this research, applications, air interface, mobile banking server as well as issues of confidentiality and integrity of mobile banking security are not considered. In addition, this thesis does not deal with attacks that occur in network and transport layer as well as mitigation after detection.

1.4 Contribution of the Study

The study and result of this research will enable ethio telecom to detect SMS availability attacks on Mobile Application Part (MAP) layer of SS7. This will further create understanding of the need for deploying an adaptive intrusion detection and mitigation method.

A novel approach of feature extraction using error code aggregation is presented. Result will provide an optimal set of features; suitable algorithm as well as a model that can detect abnormal messages with superior accuracy and minimum model build time.

As a result, ethio telecom will be able to enhance SMS service availability. Ethio telecom will need not expand its SMS license for commercially unusable traffic. This will result in capital expenditure savings. By detecting availability attacks, it is expected that message origination rate will increase and blocking rate will decrease

significantly. In due course, mobile banking service availability will be enhanced. This research will also build confidence for both ethio telecom and mobile banking service users based on the security efforts being undertaken by ethio telecom. As a result, it is expected that mobile banking adoption will increase in Ethiopia. This will further increase ethio telecom usable traffic and revenue.

It is also expected that this research will fill some gaps of previously conducted researches in SMS availability attack detection and point out areas of future work.

1.5 Research Methodology

To meet the general and specific objectives of this research the following methods are followed:

- Systematic literature review is done to understand the significance of the problem as well as the limitations of existing methods used for detection.
- Use CRISP-DM (Cross Industry Standard Process for Data Mining) is used as a process model. Its recommended phases of machine learning processes are followed through.
- Datasets that contain both normal and abnormal messages are captured from SMSC (Short Message Service Center).
- Systematic literature review of existing machine learning algorithms used for classification to select the three candidate algorithms; namely, PART, Random Forest and J48.
- Use WEKA (Waikato Environment for Knowledge Analysis) open source machine learning tool's explorer application for data pre-processing, feature extraction, and feature selection.
- Feature extraction is done by aggregating message origination (MO) error codes of SMS using expert judgments and literature reviews.

- Features that help in the detection of application level messaging attacks are identified from literature review and interview of expert judgment at ethio telecom.
- Features reduction is done using WEKA unsupervised feature selection methods that uses information gain with gradual feature reduction and evaluation.
- Use WEKA explorer for training, building models and abnormal messages detection for the selected three algorithms.
- Use model build time, accuracy, F-measure and Receiver Operating Characteristics (ROC) to compare algorithms.
- Use WEKA experimenter to evaluate detection performances of the three selected algorithms as per the selected performance metrics.
- Literatures are reviewed to propose a point of detection and architecture for deployment.

1.6 Document Organization

The remainder of this paper is organized as follows: Chapter 2 covers literature review to give fundamental understanding of the topic addressed. Chapter 3 covers related works. Chapter 4 covers data understanding and pre-processing of the traffic data to be used for modeling. Chapter 5 covers results, evaluation and performance of the selected algorithms as well as analysis and Chapter 6 covers conclusion and future works.

2. Literature Review

To create a clear understanding of this thesis, basic principles of mobile banking, GSM core network, Short message service, USSD service, SS7 protocol, attacks on SMSC, network intrusion detection systems, machine learning techniques, CRISP-DM process model and selected algorithms as well as algorithm evaluation metrics will be discussed in this chapter.

2.1 Background of mobile banking in Ethiopia

Mobile banking using SMS and USSD is widely used at the "bottom of the pyramid", which is used by most users in developing countries [9]. This is since SMS and USSD services are supported by all phones including basic phones. In Ethiopia, all banks and financial institutes that provide mobile banking use USSD and SMS for mobile banking in addition to internet banking. SMS and USSD services should be secure enough to provide financial transaction. One of the key dimensions of security is availability. Ethio telecom being the sole provider of SMS and USSD services needs to insure availability of these services for secure transaction.

2.2 GSM Core Network

The GSM has network has two subsystems, that of the access network and the core network. The access network allows a subscriber to connect to a mobile network through the nearest Base Transceiver Station (BTS) that is managed by a Base Station Subsystem (BSS). The BSS handles the connection to and from the MS and further sends and receives information to the Core Network (CN). The details of the access network are outside the scope of this thesis. In both 2G, 3G and 4G mobile network, CN provides the main functionality of integrating services necessary for providing value added

services to mobile subscribers in its network. The core network has many essential entities that are required to provide mobile network services. These nodes are defined and standardized by the 3rd Generation Partnership Project (3GPP) [13]. Figure 2-1 below shows a typical GSM architecture containing some of the elements defined by 3GPP and relevant interconnections with short message center. The different elements in the CN handle different tasks to provide services to subscribers that will be discussed below.

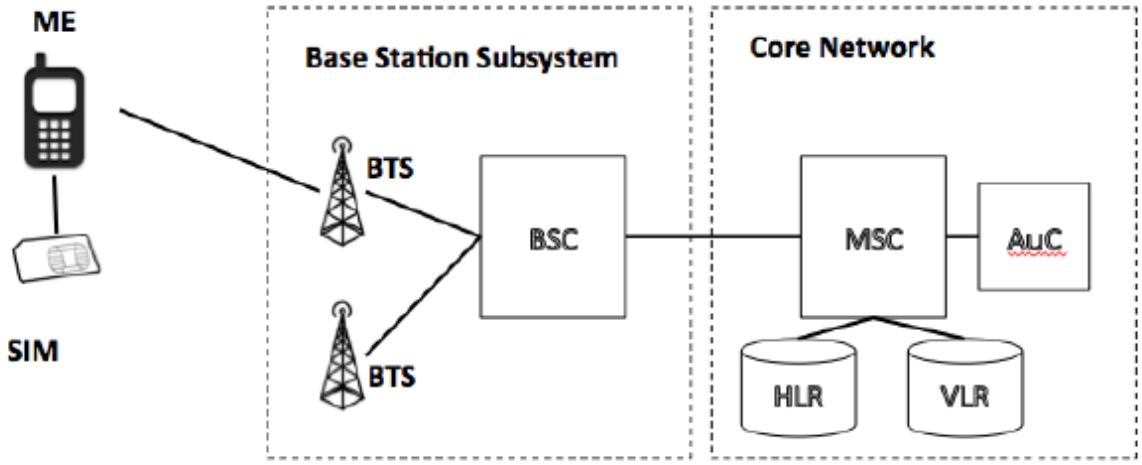

Figure 2-1: Typical GSM Architecture. Inspired by [13]

2.2.1 Mobile Switching Center

The Mobile Switching Center (MSC) performs all the necessary functions to handle circuit switched services to and from Mobile Station (MS) [13]. Its main functionality is to route calls and SMS, and other functions such as handover operations when a subscriber is changing location during a call. The MSC is used to translate user-network signaling to network-network signaling. When an MSC is used as gateway to another network or network elements such as SMS center and USSD gateway, it is called Gateway MSC (GMSC) and is located at the border of the core network. It has similar functionality

in that it appropriately routes a call or SMS to an MS located in another network or to access supplementary service providing platform such as SMS center or USSD gateway.

2.2.2 Home Subscriber Server

The Home Subscriber Server (HSS) is the master database for subscribers of an operator. It is the entity containing the subscription-related information to support the network entities handling calls/sessions [13]. HSS is in a subscriber's home network, its main functionality is subscriber identification (numbering and addressing information), user security information (network access control information for authentication and authorization), and user location information at inter-system level (supports user registration and stores inter-system location information) as well as user profile information. HSS also is responsible to support for call control and session management entities of the different domains and subsystems. There could be several HSSs depending on the number of subscribers.

The HSS consists of two main components:

- The Home Location Register (HLR), which handles roaming information such as where the subscriber is always located so that calls, SMS and USSD can be routed correctly.

- The Authentication Center (AuC) is responsible to authenticate subscribers trying to connect to the network. It generates and stores user security information keys for mutual authentication, communication integrity check and ciphering [13].

2.2.3 Visitor Location Register

The Visitor Location Register (VLR) controls a Mobile Station (MS) roaming in the area covered by an MSC and is usually co-located with the MSC. When a subscriber is roaming to a new MSC area, the VLR handles the registration procedure that includes exchange of information between the VLR and the subscriber's HLR. The VLR will inform the HLR of the subscriber's location and will in return get information required to handle calls and other services. The VLR handles different elements such as the International Mobile Subscriber Identity (IMSI), used to identify a subscriber in the network, and the Mobile Station International Subscriber Directory Number (MSISDN), which is the subscriber's telephone number [13].

2.2.4 Short Message Service Center

The Short Message Service Centre (SMSC), or more formally the SMS Service Centre (SMS-SC), is an entity that handles routing of SMS messages in the CN. It operates by querying routing information from an HLR and routes the message to the appropriate SMSC or MSC in order to deliver an SMS to the intended subscriber [14].

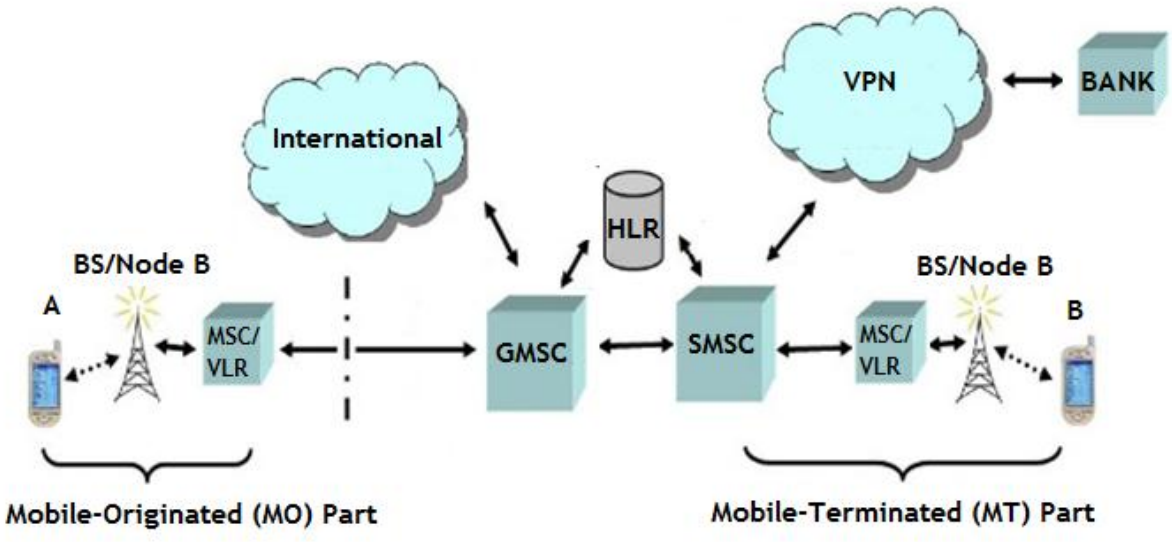

Figure 2-2: Simplified SMSC Architecture

A typical diagram of SMSC above shows the network connections that lead to an SMSC. The radio access is the base station subsystem (BSS) and MAP part starts at MSC. Both GSM network and Code Division Multiple Access (CDMA) networks converge at SMSC to send and receive messages. Mobile banking service providers are connected to SMSC through virtual private network (VPN) using Short Message per to per Protocol (SMPP). Since the focus of this research is on application level, detection is to be done at GMSC.

It is to be noted that SMS is a service that provides a connectionless transfer of messages with at most 160 characters using signaling channels in mobile networks. Figure 2-2 illustrates the basic SMS architecture in a GSM-based system. A short message sender (on the left bottom corner) uses an *originating MS* (Mobile Station) to send a short message to a receiver (on the right bottom corner in Figure 2-2). The short message is delivered to a nearby BSS (Base Station System) through the GSM signaling channel and then the MSC (Mobile Switching Center) associated with this BSS. The MSC first checks with a VLR (Visitor Location Register) database whether the originating MS can receive the short message service. The VLR database temporarily stores subscription information for the visiting mobile stations so that the associated MSC knows what services should be provided to them. If the originating MS can use SMS, the MSC further routes the short message to SMSC, a dedicated store-and-forward server that handles SMS traffic [15].

The SMSC is responsible for forwarding the short message to the targeted mobile device, also called terminating MS (on the right side in Figure 2-2). To do that, it queries an HLR (Home Location Register) database, which keeps information about mobile subscribers, such as their profile information, current location, billing data, and validation period. The HLR responds by sending back the serving MSC address of the terminating MS. Thereafter, the SMSC forwards the short message to that MSC, which further queries its associated VLR database for the location area of the terminating MS. Once the location of the terminating MS is found, the short message is delivered to it through its nearby BSS [15].

2.3 Signaling System No. 7

The protocol used in mobile network inter node communication is called Signaling System Number Seven (SS7). SS7 is a suite of protocols that were standardized in the 1980s in ITU-T Q.700 series. Additional protocols were added in the 1990s and 2000s by ETSI and 3GPP to support mobile phones and the services that need (roaming, SMS, data...) [4, 12]. This section covers overview of SS7 protocol stack and exposed intrusions.

2.3.1 SS7 Protocol Stack

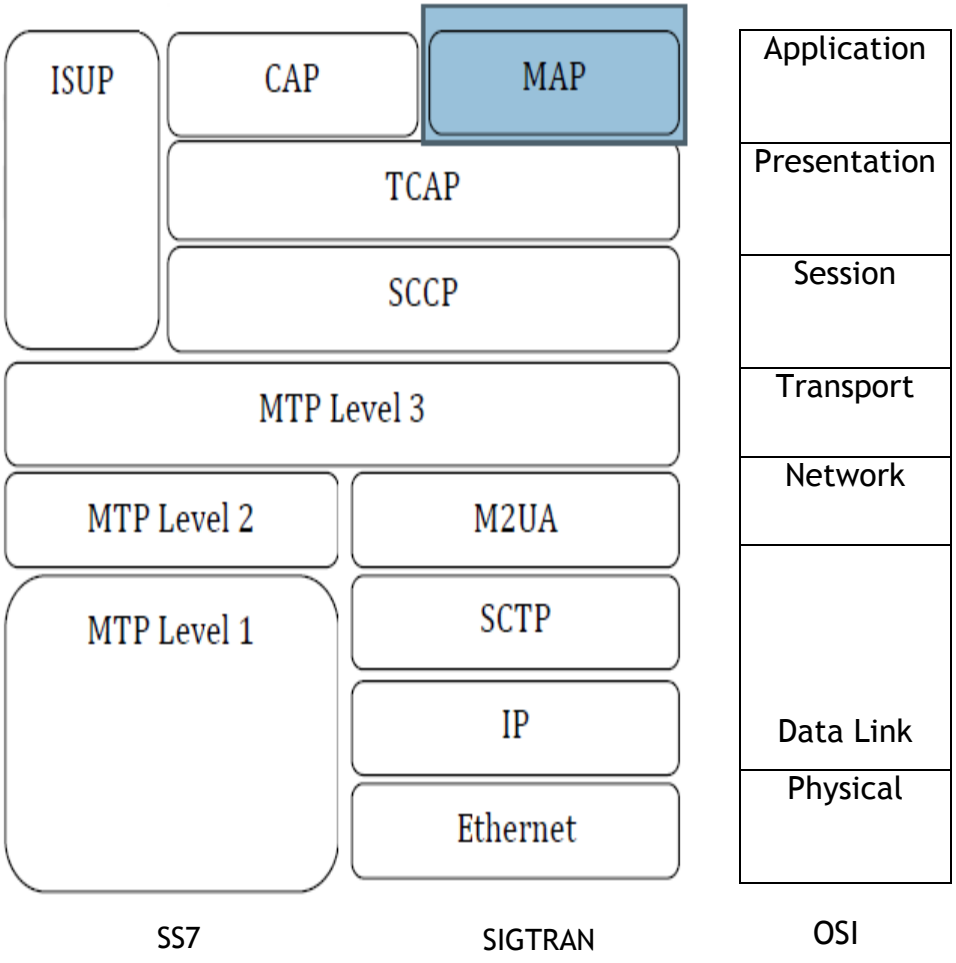

Figure 2-3: SS7 and SIGTRAN Protocol Suite with OSI [14]

The implementation of SS7 through E1 is known as SS7 and when it is implemented over Ethernet, it is called SIGTRAN (Signaling Transport). As shown above, the difference between SS7 and SIGTRAN is only below the network layer. Application layer, which is of interest for this research, is Mobile Application Part (MAP) in both cases.

As highlighted in Figure 2-3 above, MAP is an SS7 protocol that provides an application layer for the various nodes in GSM and UMTS (Universal Mobile Telecommunication System) mobile core networks and GPRS (General Packet Radio Service) core networks to communicate with each other to provide services to mobile phone users. MAP is the application-layer protocol used to access the Home Location Register, Visitor Location Register, Mobile Switching Center, Equipment Identity Register, Authentication Centre, Short message service center and Serving GPRS Support Node (SGSN) [12]. The MAP protocol is responsible for providing the following functionality: Mobility Management, Operation and Maintenance, Call Handling, Supplementary Services and Short Message Service [16].

SS7 is based on the assumption of being a trusted closed network. Keeping this in mind, no additional security layer was added in this protocol. Thus, data moves through the network after BTS in plain text format within an operator's network. Abusing SS7 insecurities can have a severe effect; the nature of the protocol allows access to information such as user location and call/SMS details. Financial services and authentication systems were built based on the trust of the services provided by such protocols. Denial of service attacks abusing those insecurities can be devastating to the telecommunication infrastructure of nations [12].

Section 2.3.2 examines intrusion of SS7 and some of the attacks that were announced against SS7. The following section bases on the work done by security researchers [4] in the areas of call and SMS interception, location tracking, fraud, and denial of service.

2.3.2 SS7 Intrusion and Attacks

After moving through the air interface, mobile banking data transverses through the operator's radio and core network before reaching the service providing platforms (SMSC and USSD). As explained in [7], USSD platform has weakness since it does not encrypt transactions inside the operator network and servers. This is also the same for SMS [11]. Encryption in GSM is only applied between a mobile and a base station; but across the rest of the network, no encryption is applied making data vulnerable when intercepted [11].

If an attacker can access the operator's signaling network, an attacker will be able to listen to everything that is transmitted, including the actual phone call as well as the air interface security triplets which are Random Access Number (RAND), Signed Response (SRES) and Encryption key (Kc). Access to operator's core network can be achieved through several ways. According to [4], it can be bought from telecom operators with roaming agreements or roaming hubs for a few hundred euros a month. Some network operators are also known to leave their equipment unsecured on the internet and Femtocells are part of the core network that have been shown to be hackable [4].

An attacker who has managed to access an operator's network can manipulate SS7 vulnerabilities by creating fake MSC [4]. Using a fake *update Location* message an attacker claims that the victims MS is connected to their MSC. In this case, the subscriber SMSs will be forwarded to the attacker's SMS center to be delivered to the MS [4]. In addition to intercepting personal SMSs of the target, this attack can be used against authentication systems that utilize SMS verification (SMS token, Facebook verification, etc.) and could lead to the compromise of the target's identity [12]. Unfortunately, the *updateLocation* message has a legitimate use case when the subscriber is roaming outside the operator's network. Thus, it cannot be filtered at network borders. *UpdateLocation* is one example of legal message that is available and

rule-based intrusion detection will not be efficient since legal messages should not be blocked.

USSD is used popularly for mobile banking transaction service. In this service, a subscriber typically sends certain USSD codes (such as #100#) to fulfill certain mobile banking transactions. Using a *processUnstructuredSS* message, the attacker can send USSD codes on behalf of the customer, possibly authorizing a credit or money transfer transaction from his target [4]. Unfortunately, in many cases the operator allows receiving this message from external networks, in case their roaming subscribers need to access these services while visiting another country, which makes filtering such message at the border very hard [12]. In the case of USSD, *processUnstructuredSS* message is one example of legal message that is available and rule-based intrusion detection will be difficult since legal messages should not be blocked.

In SMS intercepting using fake MSC attack, an attacker impersonates to HLR as an MSC that is serving the victim. Since SMSC inquires HLR for destination MSC details during Mobile Terminated *ForwardSM* procedure, attacker can successfully intercept all the SMS messages intended for victim [17]. This attack is represented in the Figure 2-4 below.

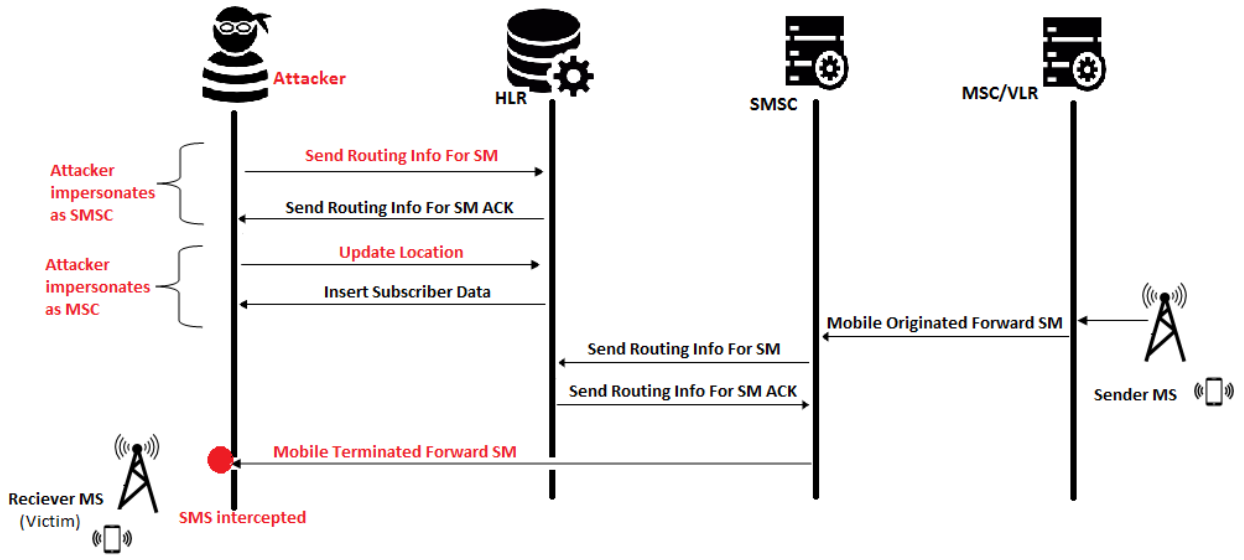

Figure 2-4: SMS interception on the receiver end using fake MSC [14]

An explanation on the flow of messages depicted in Figure 2-4 is as follows. As pre-condition for this attack, the attacker impersonated as SMSC as described in Section 2.3.2 to learn the MSC GT and IMSI of victim. Having known the MSC GT and IMSI of a victim, an attacker impersonates as an MSC serving the victim and sends update location message to HLR by providing victim's IMSI. HLR responds back to the attacker with insert subscriber data message and this shall be ignored. Let us assume that a subscriber A is sending an SMS to the victim (subscriber B), the MSC serving MS A will initiate Mobile Originated *ForwardSM* to SMSC. SMSC contacts the HLR using send Routing info for SM message to know MSC GT and IMSI of subscriber B. Since HLR has stored the attacker as the MSC responsible for subscriber B, it sends attacker's MSC GT to SMSC using send Routing info for SM ACK message. SMSC thinks that is the legitimate MSC of subscriber B and hence dispatches the SMS message using Mobile Terminated *ForwardSM* message. Attacker can now receive all SMS message that is destined to subscriber B, the victim. This attack is possible until the victim changes location area. However as long as HLR stores attacker's fake MSC as the legitimate MSC serving the victim, the attacker can intercept all the SMS chats, one-time passwords, confirmation codes and password recovery messages which increases the threats to personal privacy [14].

As recommended by [12] some of the vulnerabilities of SS7 can be managed using best practices. However, since vulnerabilities of SS7 are not yet all known, and some are legitimate messages as shown above. Hence, the need for adaptive intrusion detection and mitigation method such as machine learning techniques becomes essential.

2.4 Short Message Service Attacks

This section focuses on the different types of SMS attacks that are done using the vulnerabilities of SS7 and malwares. Although there have been many SMS attacks reported, the discussion here focuses on those that use SS7 and those that originate from mobile users as result of malicious attack or malwares.

As described in Section 2.3.2, SMS protocol includes two parts [14]. When an SMS is initiated from subscriber A, the MS will contact its MSC, which in turn connects to SMSC using Mobile Originated *ForwardSM* message. SMSC looks up for MSC GT and IMSI of the SMS receiver, and forwards Mobile Terminated *ForwardSM* message towards destination MSC. The Figure 2-5 below illustrates the two stages of SMS.

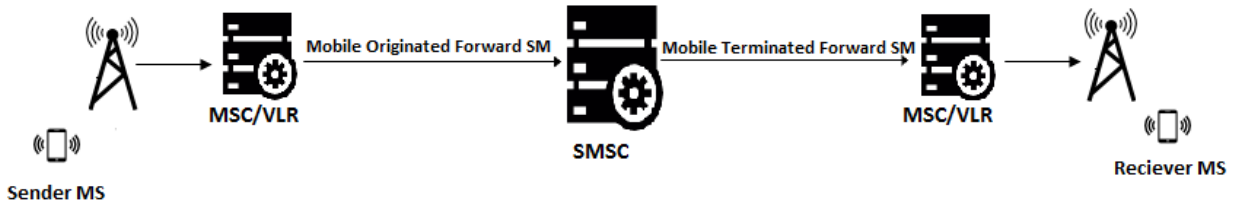

Figure 2-5: *ForwardSM* Stages [14]

Since Mobile Originated *ForwardSM* and Mobile Terminated *ForwardSM* messages are not checked for their authenticity, vulnerabilities in the SS7 protocol gives rises to the chances of SMS based attacks [14].

2.4.1 Sources of Attack

As explained in Section 2.3.2 one of the sources of attack is through exploitation of SS7 protocol vulnerabilities. An attacker who has managed to create an intrusion into the SS7 network can send a flood of messages to create availability attacks. Hence, one of the sources of denial of service attacks that affects availability is SS7 intrusion.

There are several ways an attacker can deny service to subscribers. Using *insertSubscriberData*, or *deleteSubscriberData*, the attacker can remove critical services, or activate call barring for the target. Using a *cancelLocation* message, the attacker can trick the network into removing the subscriber's connection to the network, and hence calls and SMSs cannot be delivered as shown in Figure 2-6 below. As result, mobile banking user will not be able to use mobile banking service or will not be able to receive notification receipts through SMS [12]. However, these types of attacks focus on a subscriber or group of subscribers and the effect is not to all subscribers.

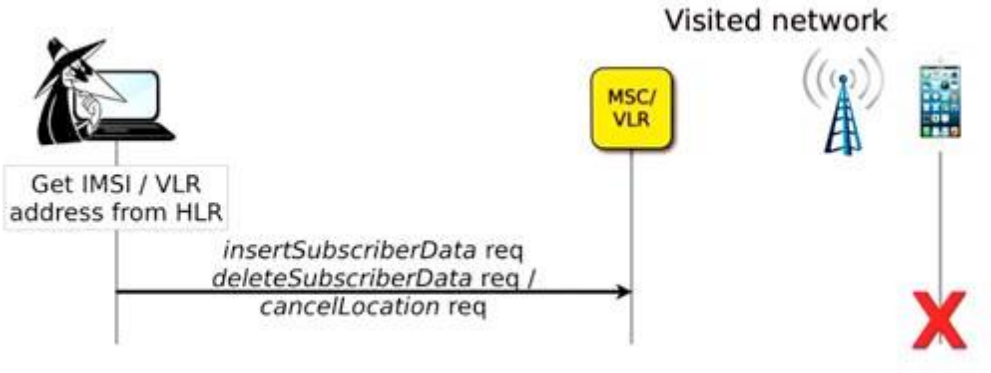

Figure 2-6: Denial of Service Using SS7 [4]

Having access of SS7 network, a more serious type of attack an intruder can carry is to send flood of messages to SMSC to congest the SMSC service causing message blockage. Since SMSC resources are shared, a congested SMSC will block any attempts by mobile

banking service providers. Considering the magnitude and number of subscribers affected, this type of attack is the focus areas of this research.

Another source of attack is from mobile users due to mobile malwares or intentional mobile message flooding. Mobile bots have different intentions when it comes to host-based characteristics such as intercepting SMS, sending SMS, making phone calls, stealing passwords and hijacking an application. However, they all have a common feature, which is establishing network connection between the device and attacker [18]. Malwares often enter a user's mobile through unknown sources or sometimes through official android markets such as Google Play. [18] were able to install 14 out of 49 most malicious mobile malwares in the world from the official Android market known as Google Play.

In the above-described cases, large number of messages arrive at SMSC that are not usable both from the mobile operator's point of view as well as from mobile banking providers. This research will use machine learning techniques to be able to distinguish between abnormal and normal messages originating from the above-mentioned sources.

2.4.2 Types of attack

As explained in 2.3.2, Mobile Originated *ForwardSM* and Mobile Terminated *ForwardSM* messages are not checked for their authenticity. This allows an attacker to send illegitimate SMS messages such as fake SMS, spam SMS as well as flooding SMS. To perform SMS attacks using Mobile Originated *ForwardSM* message, the only information the attacker has to know is MSC GT and IMSI of destination MS [14]. A basic flow of such an attack is represented in the Figure 2-7.

An attacker with access to SS7 network can send SMS messages using vulnerabilities in SMS delivery mechanism. An attacker can use this method to execute the following types of attacks [14]:

- Fake SMS: An SMS can be sent to any subscriber within the network by faking sender's MSISDN. Here an attacker can also spoof as some legitimate subscriber.

- Spam messages: Unsolicited SMS containing commercial advertisement, bogus contents can be sent by an attacker.
- SMS flooding: An attacker can send large number of messages to one or more destinations. The sole purpose of attacker is to slow down the network or to jam mobile stations.

As the focus of this research is to detect illegitimate messages that not usable. Flooding messages are the type is required to detect. Through SMS flooding, an attacker can send large number of messages to one or more destinations. The sole purpose of attacker is to slow down the network or to jam mobile stations.

By only knowing MSC GT and IMSI of destination MS, an SMS attack can be conducted using Mobile Originated *ForwardSM* message [14]. A basic flow of such an attack is represented in the Figure 2-7. In this attack, to know MSC GT and IMSI of destination, the attacker uses location privacy breach attack described in Section 2.3.2. Attacker can then impersonate as MSC and sends MSISDN of sender in Mobile Originated ForwardSM message to SMSC. If SM is successfully stored in SMSC, an acknowledgement is sent back to attacker with Mobile Originated ForwardSM ACK message. To deliver SM to the destination, SMSC has to know the MSC location and IMSI of the recipient that is stored in the HLR. SMSC sends MAP *SendRoutingInfoForSM* message to the HLR to query the MSC GT/location and IMSI of the recipient. HLR encapsulates IMSI and MSC location in *MAPSendRoutingInfoForSMACK* message and sends it back to the SMSC. Based on this

information, SMSC routes the SM to the recipient MSC that in turns delivers it to the mobile user using Mobile Terminated Forward SM message.

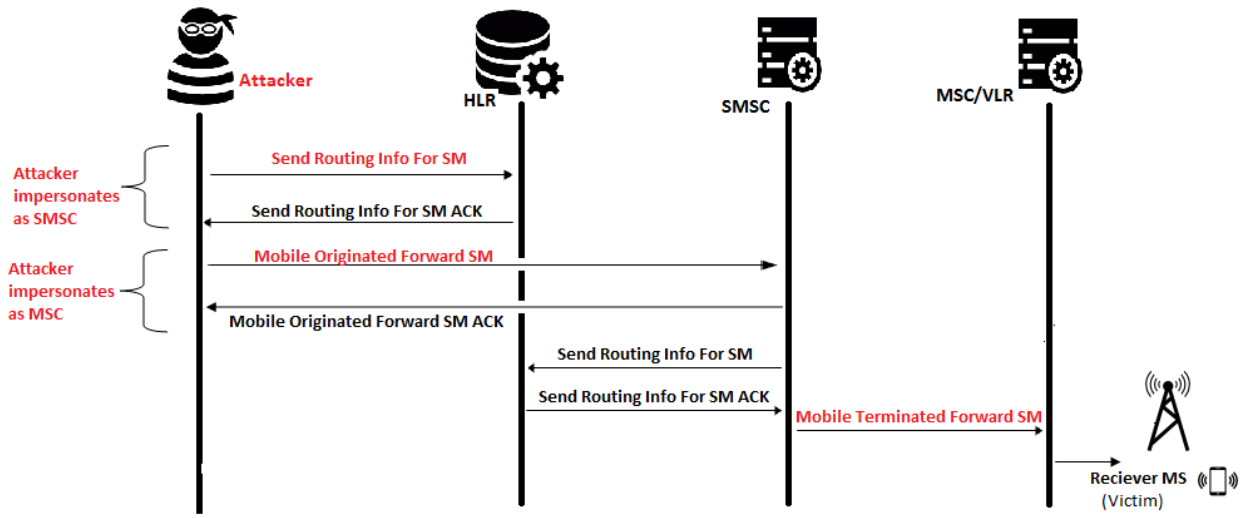

Figure 2-7: Sending illegitimate SMS using MO Forward *SMmessage* [14]

This enables an attacker to an SMS successfully send to a desired destination. The SMS will be billed to the sender, which can be any legitimate MSISDN that the attacker has chosen.

A second type of attacks are messages that originate from mobile users. These types of messages are often either intentional by attackers or due mobile malware in a user's mobile phone. In fact, since 2004 many malware targeting mobile phones have already emerged [19]. These worms or viruses can propagate and infect vulnerable smart phones through all kinds of mediums including Internet [19], Storage Cards [20], SMS [21], MMS [22], and even some local wireless protocols like Bluetooth [19].

A mobile botnet to launch a DDoS attack against the core infrastructure of the cellular network is proposed by [23]. Their simulation and analysis demonstrate that their attack can cause nation-wide outages with even a single-digit infection rate, demonstrates a about the great destructive power of mobile botnets [23].

2.5 Network Intrusion Detection Systems

To monitor network or system events and to detect malicious activities, software or devices called Intrusion Detection Systems (IDS) are used. When the focus is to detect remote attacks on a host or a network, Network Intrusion Detection Systems (NIDS) are specifically used. NIDS are often placed in networks to collect relevant network traffic that can be used to detect an attack that has happened or one that is being executed. There generally exists two types of NIDS, based on their method of detecting attacks: misuse (signature) based detection and anomaly detection [24, 25].

2.5.1 Signature based detection

Misuse detection systems detect attacks based on a previously known pattern related to the attack, also referred to as the attack signature. A network intrusion detection system using misuse detection will look for these signatures in the network traffic to identify an ongoing or upcoming attack. The signatures explain what to look for in the traffic [25].

Signature-based detection searches for a particular type of network traffic or sequence of bytes that are known to be destructive. These types of systems collect statistics from current data set. In addition, it also captures details regarding a variety of attacks and system exposure. The acquired expertise is used to discover anomalous traffic behavior and generate alarm if a known attack is identified. It compares an incoming traffic with an existing stored signature to identify existing types of attack. However, this type of detection is effective only when an attack is of known type. It will not detect a new type or slightly modified attack. This is because such attacks do not have signatures or patterns in the database. The primary benefit of signature-based detection method is that, patterns or signatures are simple to understand and develop if it is known that what network behavior is being identified [26]. If we consider one of the sources of

attack in terms of this research, the detection of malware originating messages, mobile devices have adopted traditional approaches such as the antivirus. This tactic is not as efficient [27] against mobile device malware, as it requires continuous signature database updating and mobile malware is constantly modified to bypass the various detection methods.

In addition, an antivirus also needs to continuously monitor a device's activities closely and requires frequent signature update processes to detect new malware. As a result, these procedures demand excessive memory and power usage, which degrade mobile device performance [28].

2.5.2 Anomaly detection

On the other hand, systems based on anomaly detection tries to assess what behavior is "normal" and look for deviations in the network activity. Typically, a policy is created based on what is normal and the NIDS uses this policy to detect abnormalities in the network that are referred to as anomalies. Anomalies might for example include a sudden rise in network traffic, or a sudden rise in use of an application protocol [25]. The main goal of an anomaly detection approach is the detection of outliers. Typically, an anomaly detection algorithm will be trained using normal data; it is then the task of the algorithm to decide if input data is close to or far from the defined normal. The algorithm will look for activities defined as normal, and report on any deviation from normality. As anomaly detection is a central part of this thesis' approach to SS7 security problems, an introduction will be given to the techniques available to detect anomalies in a network.

Anomaly-based detection is another DDoS detection method that can detect unidentified and new attacks. This method analyzes benchmark network output and evaluate with the incoming data instances. While the variance among an expected and observed performance reaches a predefined threshold, the detection system produces

an alarm of anomaly; thus, an attack is revealed. Because of uncertainties present in the acquired data and due to varying nature of network behavior, anomaly-based approaches generate many false signals. Still, anomaly-based IDS have a benefit over signature-based IDS as a new attack for which a signature does not survive can be detected if it falls out of the normal traffic patterns [26]. Unlike misuse based, the anomaly-based IDSs do not necessitate signatures to detect intrusion. Besides, anomaly-based IDS can identify unknown attacks based on the similar behavior of other intrusions [29].

2.5.3 Source side detection

Actual detection of DDoS attacks on availability can be done on source, network and server side. Source-end detection methods are organized at source of attack to stop network users from producing DDoS attacks. In this method, source devices rate-limit or filter the traffic and, they identify the malicious packets in outgoing traffic [26].

2.5.4 Server side detection

In server-side detection (victim-end defense) method, the victim system filters, rate-limit and detects the malicious incoming traffic at the routers of a victim networks [26].

2.5.5 Network side detection

In an intermediate network or core-end defense method, the identification of the malicious traffic and filtering or rate limiting the traffic is done by any router in the network independently. It also balances the trade-offs between [26]. In this research, this method is proposed in relation to mobile core network.

2.6 Machine Learning and Algorithms

Machine learning (ML) is a subfield of computer science that is used for data analysis and acquisition of knowledge. The basis of machine learning is "the automatic modeling of underlying processes that have generated the collected data" [30].

According to [31] machine learning aims to make sense of data and is more formally defined as: “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.” In other words, if a computer program performs some task T, say playing chess. The programs ability to play chess can be determined by how many games it wins, which is the performance measure P. If by playing additional games of chess the computer program wins more games, it is said to have improved its performance based on its experience E. The program is said to have learned and therefore improved.

If we have large data sets, data mining is used to discover patterns out of the available big data. To do this, it uses methods derived from artificial intelligence, machine learning, statistics and database systems. In doing so, data mining has the goal of extracting information from a set of data and store it as an understandable structure for further use [32].

2.6.1 Machine learning basics

A system based on learning to achieve a task is referred to as expert system in machine learning. An expert system holds knowledge that is critical to make decisions. An example of a decision can be determining as to which class an input belongs in the case of classification [33].

For each input, the learning algorithm to perform its task uses a set of measurements referred to as features. Each set of features is called an instance and this instance is a row in a table containing the input data. Just like variable, a feature can be an integer, a string or a binary value and it can be either discrete or continuous [33].

Classification is one of the tasks that machine learning uses to solve a problem. In the case of classification, an input data is to be classified to a specific class that is part of a discrete set of classes. To achieve this goal, a machine learning algorithm will initially learn from a *training set* before being presented with new inputs that should be classified. A training set will contain several *training examples* containing features that is used by the classifier in the classification task. A training example will contain the features relating to a *target variable* that the features describe [33]. For example, as per the focus of this research a target variable short message may be explained by features such as originating and destination MSISDN. Whereas, if it we it was an internet network it could be source, destination IP or even port number.

ML algorithms that have been used for traffic classification generally fall into the categories of being supervised or unsupervised. In the case of unsupervised (or clustering) algorithms, traffic flows are grouped into different clusters based on the similarities in the feature values. It is the responsibility of the algorithm to cluster based on statistical nature and there are no pre-defined clusters. On the contrary, the class of each traffic flow must be known before learning for supervised algorithms. Based on the class of traffic flow, a classification model is built using a training set of example instances that represent each class. The resulting model will be able to predict class membership for new instances by examining the feature values of unknown flows [34].

2.6.2 Selected Algorithms

To solve the problem of abnormal message detection, a suitable machine learning technique and algorithm must be chosen. There are many algorithms and techniques available for machine learning and anomaly detection [35]. To select a most suitable approach, many issues need to be considered. The first is the choice of the type of machine learning technique. As discussed in Section 2.6.1, there are mainly two techniques for anomaly detection. These two techniques are supervised learning and unsupervised or outlier learning. To be able to choose one of these techniques, one must consider and decide based on available data [35]. If a dataset consists of many known normal and abnormal instances with many labeled normal and abnormal instances, a supervised algorithm would be best. On the other hand, a dataset consists may consist of many known normal instances with only a few abnormal instances. In such cases, there might be large number of different anomaly types. If this is the case, an unsupervised detection algorithm is best fit to solve the problem [35].

In the case of abnormal short message detection, it is found that there are large number of abnormal messages. If labeling of instances can be insured as discussed above, supervised learning is more suitable. Hence, this research will use supervised learning. In the next section, the theoretical background of the selected algorithms will be presented briefly.

2.6.3 PART Algorithm

PART (Partial decision Tree) algorithm is a rule-based algorithm. PART is a developed version of C4.5 and RIPPER algorithms. The main advantage of the PART algorithm is that it does not need to perform global optimization like C4.5 and RIPPER to produce the appropriate rules. PART is an indirect method for rule generation. Using separate-and-concur strategy, PART generates a pruned decision tree for each of the iterations.

From the best tree, the leaves are translated into rules. PART adopts the divide-and-concur strategy in that it builds a rule, remove the instances it covers, and continue creating rules recursively for the remaining instances until none is left. It uses C4.5 statistical classifier to generate a pruned decision tree [36].

2.6.4 Random Forest Algorithm

Random forest is a blend of random subspace and bagging method proposed by Tin Kam [37]. Random forest ensembles classifiers using many decision tree models. A different subset of training data is selected with a replacement to train each tree. The remaining training data serves to estimate the error and variable importance [38]. This classifier is a logic-based algorithm that is proved to produce high-accuracy result as shown in [39] and for malware detection [40].

2.6.5 J48 Algorithm

J48 is a predictive machine learning classifier that decides the target value of a new sample based on various attribute values of the available data. This classifier consists of dependent and independent variables. The dependent variable is the attribute to be predicted, while the independent variable is the attribute that helps predict the value of the dependent variable. To classify new samples, a decision tree based on the attribute values in the training set must be created. These attribute values determine data instances to classify and have the most information. [41] Applied this classifier and obtained very high detection results. J48 is also a type of logic-based learning.

2.6.6 Machine Learning Process Model

There are many machine learning process models. These models enable to structure a machine learning research. Among the most famous once, we have the CRISP-DM, KDD (Knowledge Discovery in Databases) and SEMMA (Sample Explore Modify Assess). A parallel view of these models has been done by [42]. The author asserts that SEMMA and CRISP-DM can be viewed as an implementation of the KDD process. It also asserts that CRISP-DM is extremely complete and documented. It is well organized, structured and defined, allowing that an undertaking to be easily understood or revised. Considering this, this research will follow CRISP-DM as a machine learning process model.

The various steps to be followed in CRISP-DM are explained in [43] as below.

Business Understanding. This is the first phase of the model and it focuses on understanding the objectives on the undertaking and requirements from business perspective. In this phase, the knowledge gained is converted into a machine learning problem definition, and a preliminary plan is designed to achieve the objectives.

Data Understanding: This phase starts with initial data collection. It is followed by activities to get familiar with the data, to identify data quality problems, to discover first insights into the data, or to detect interesting subsets to form hypotheses for hidden information. Business understanding and data understanding have a close relationship. Hence, a clear understanding of available data is required to formulate the machine learning problem.

Data Preparation: The focus of this phase is to cover all activities that are required to construct the final data from the initial raw data that will be applied to a modeling tool. Data preparation is often done multiple times with no order to get a proper data

set. In this phase, the tasks include table, instances, and feature selection, data cleaning, construction of new features, and transformation of data for a modeling tool.

Modeling: In this phase, different modeling techniques are selected and applied. There are many possible techniques for the same machine learning problem. As with other phases consecutive phases, there is close relationship between data preparation and modeling. Some algorithms might require different data format and this data problem can be observed while modeling. As result, it might be necessary to construct new data.

Evaluation: By this phase, one or more models are built from previous phase having high quality from data analysis perspective. The models and the steps to construct the model are deeply evaluated in this phase to ensure that initial objectives are met. The main goal is confirming that important objectives are considered and no objective that is not sufficiently addressed. By the end of this phase, a decision on the use of a machine learning model should be reached.

Deployment: The final step of this process model is deployment. The knowledge gained as result of the modeling should be organized and presented that can be usable. Depending on the objectives set, the deployment phase can be simple generation of a report or a complex as implementation of a repeatable machine learning process. This step in often done by a user who will carry out the deployment process. In this phase, a clear understanding of the important activities required to use the created model in created.

In this research, CRISP-DM is used as process model for the problem of abnormal short message abnormal message detection. A pictorial representation of it as well as implementation of each of the steps is presented in Chapter 4 of this research.

3. Related works

In this chapter, a selected review of existing literatures on availability attack and detection mechanisms are analyzed. Researchers try to differentiate normal traffic from abnormal or attack traffic in their work for attack detection using different techniques.

DoS attacks is a major concern threatening the availability of the Internet. As a result, many studies have been conducted to detect attacks in application level of IP networks [53]. However, the direct application of these solutions for short message service is not suitable because the IP flow and short message flow have different properties. SMS-based attacks pose serious security threats to both mobile users and mobile networks, including battery exhaustion and network congestion. Effective countermeasures, unfortunately, are still lagging.

The general solution for detecting intrusions normally use signature-based and anomaly-based detection. In signature-based technique, attacks are discovered using predefined list of known attacks. This solution tries to detect malware from mobile devices with techniques inspired by their counterparts in IP networks [50]. For instance, signature-based detection schemes are proposed to examine mobile network traffic [48] or power usage of mobile applications [49] for signatures that are extracted from existing mobile malware instances. This solution has the potential for detection, but it requires a constant updating of the predefined signature database. In addition, this method is less successful in detecting malicious activities since there is a constant change mobile malware [18]. According [51], current software can detect up to 79.6% of mobile bots in their dataset. The same is true with intrusion since SS7 vulnerabilities are still being discovered. As a result, signature-based techniques are not effective for new types of attacks [12].

The serious effects of availability attacks in short message has been shown by many researchers. In [53], it shown that there could be a severe congestion when the SDCCH channels are full utilized as they are shared by SMS, call setup and location updates. Capacity analysis was conducted in [50] using a queueing model to show that the SDCCHs can be a bottleneck which increases the blocking probability of SMS as well as voice calls during increased message loads. Furthermore, the possibility of an attack exploiting the limited and shared property of the SDCCHs was addressed in [53]. In addition, it is shown in [45] that SMS-capable mobile networks are vulnerable to an SMS flooding attack. These attacks are possible when a large rate of SMS messages is sent from the Internet to local cell phones in order to saturate the SDCCH capacity. This has been shown in [46] where an increased performance of this attack is modeled and simulated. However, it did not address on how to detect flooding attacks.

Previous research conducted by [50] proposed a SMS-related attack detection scheme named SMS-Watchdog that detects abnormal activities of SMS users by checking deviations from their normal social behaviors. Their approach is applicable to SMS flooding attacks because the attacker's behavior may change from the behavioral profile trained before the attack starts. SMS-Watchdog's work is to detect anomalous SMS behaviors by observing that there are window-based regularities inherent in behaviors of typical SMS users over a five-month period. Experimental results show that it has detection approach that can detect more than 92% of SMS-based attacks with false alarm rate 8.5%, or about two thirds of the attacks without any false alarm. However, SMS-Watchdog is not able to detect attacks related to SS7. For example, SMS-Watchdog is not able to detect SMS faking attacks, as such, attacks simulate the behavior of SMS switches and the illegitimate SMS messages do not go through the SMSC of the originating terminal, where the SMS-Watchdog is deployed [50].

The proposed detection method however is to be deployed on gateway MSC and hence all originated SMS will be routed through it before causing blockage on SMSC.

Similarly, [51] identified the best classification method out of six classifiers, namely J48, Naïve Bayes, Bayesian Networks, k-Means, histogram and logistic regression, using the Andromaly framework. The framework adopts feature selection methods such as Chi-square, Fisher score and information gain to enhance the detection accuracy. As a result, they managed to achieve a high accuracy rate with the decision tree (i.e. J48) classifier and information gain method [52] but did not address the issue of SS7 intrusion.

In the absence of attack traffic traces, [53] analyze normal SMS traffic to distinguish the difference between flash crowds and flooding attacks. They found through analysis that a mobile user replies to a message from a close friend with high probability and is unlikely to answer a message from an unknown number. Therefore, they conclude that if the reply rate for a handset that sends messages into a congested network is lower than a threshold, it is likely to be a malicious handset attempting to congest the control channels [53]. They also propose a mitigation technique that classifies SMS traffic as normal, suspicious, or malicious and separates the traffic into three distinct queues with decreasing priorities to reduce the blocking caused by attack traffic and allow for fast identification of malicious handsets. The blocking of the normal handsets' traffic is efficiently diminished since a higher priority for obtaining the limited control channels is given to the normal handsets rather than the suspicious and malicious handsets [53]. To distinguish malicious handsets, [53] consider the reply rate to messages sent by a handset. If the reply rate of a certain handset is lower than that expected for a normal handset, the handset is likely to be an attacker. The replay rate is used for classification in this research and will not hold valid, as user replay rate is every change. In addition, they only consider a mobile-to- mobile attack in this paper and do not consider messages that could come from SS7 intrusion.

Recent trends computer networks show that machine learning techniques are being used for DDoS detection. A machine learning-based approach is developed in [54] to catch mobile malware by discriminating behaviors of normal applications and malware

at the level of system events and API (Application Program Interface) calls. Unfortunately, although [51] achieved great accuracy, they used self-written malware to test their framework on, which could have produced unrealistic results [52]. In contrast, this research uses real traffic to detect availability attacks of SMS that arrive at SMSC. Detecting DDoS attacks using traditional methods is not much efficient. These days machine learning based techniques for detection of DDoS attacks has received much attention [55]. ML classifiers have been playing significant part in the development of intelligent systems for many years. It acquires a labelled dataset and produces a model as output, which can manage new data. To build a model, classifiers learn from a large amount of input and labelled output. As such, adopting machine learning classifiers is proven to enhance detection accuracy [56].

This research proposes an anomaly machine learning technique to detect abnormal messages to identify SMS flooding attacks and a mitigation technique to lower the blocking rates. The objective of this work is to be able to detect abnormal message that are influencing availability of SMS as well as mobile banking service. I tis expected that this research in a gap where SMS availability attack messages coming from different sources are managed using aggregated SMS error codes to label the status of message. Hence, this research will add value in that it will address flooding due to malware, intentional flooding and SS7 attacks detection using real SMS network traffic. As presented in this chapter most attempts using ML have focused on malware detection. In addition, the use of real traffic from an ongoing operator is expected to add insight into classification of the nature of messages and anomalies. This will further enable operators to manage some of the attacks through reconfiguration of their mobile switching centers as well as their short message service centers.

4. Data Understanding and Pre-processing

In this chapter, the tasks of business understanding, data understanding and data pre-processing of SMS traffic data is done as an input for next chapter of modeling, performance analysis and deployment. To undertake these tasks, CRISP-DM process model is utilized. Figure 4-1 bellow illustrates the series of steps to follow while utilizing the CRISP-DM model.

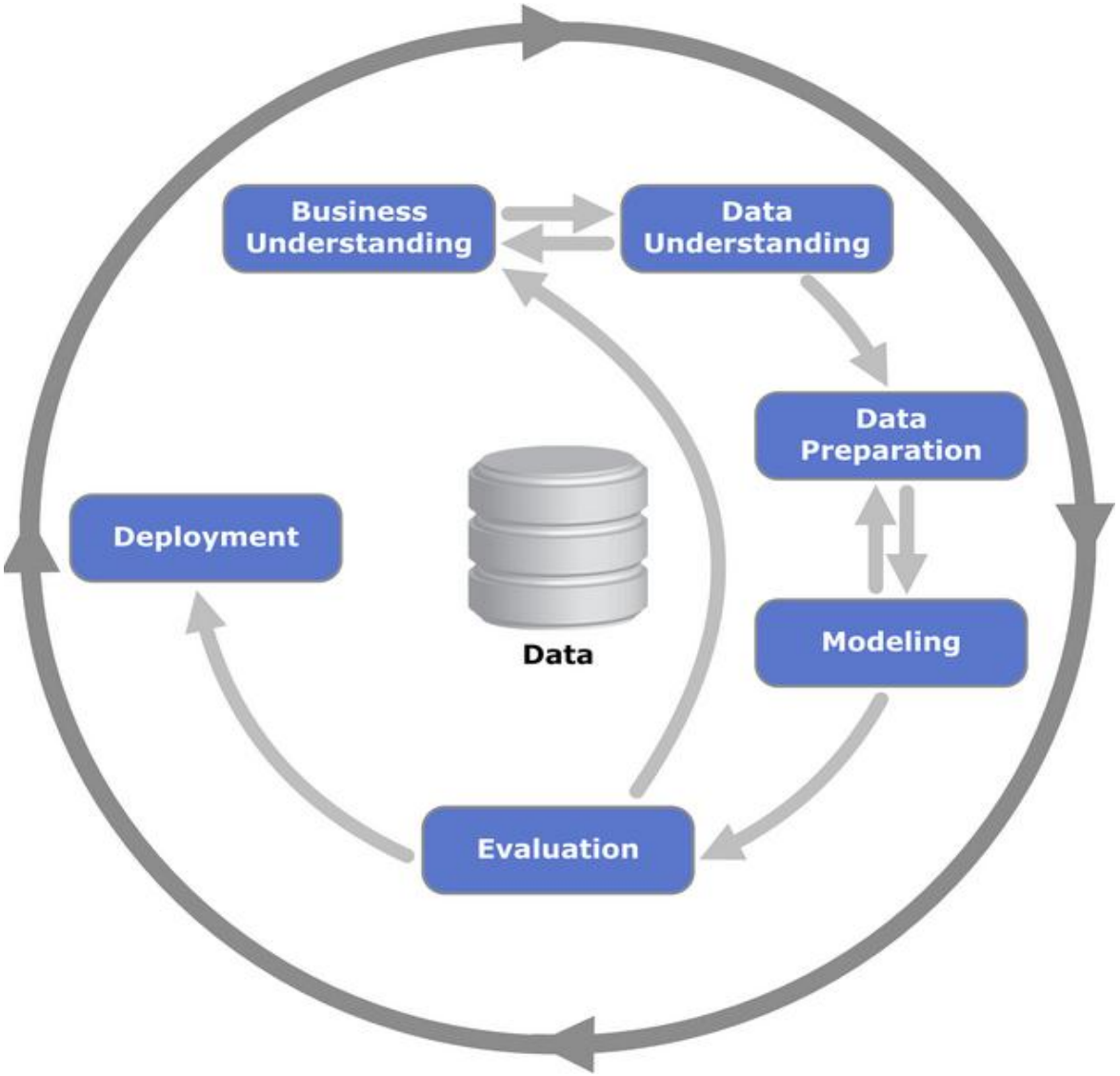

Figure 4-1: CRISP-DM Reference Model [42]

4.1 Business Understanding

This initial phase focuses in assessment of current situation to create a clear understanding of availability attacks and their impact from business perspective. The problem of the declining MO success rate and decrease in availability is demonstrated using real statistical data from SMSC. In addition, the opportunity because of detection is also discussed. The knowledge gained in this phase is converted into machine learning problem.

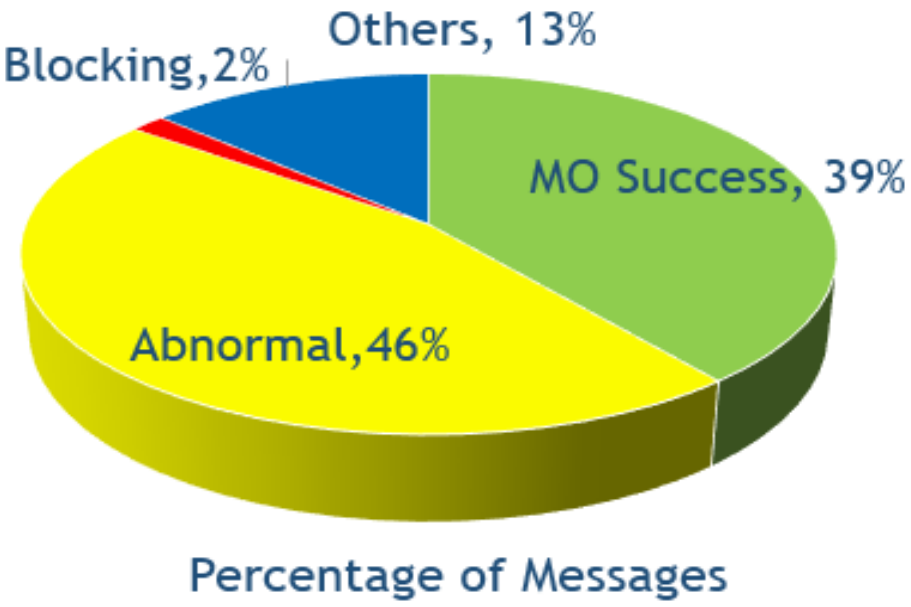

Figure 4-2: Monthly SMS Mobile Origination Message Statistics

As derived from ethio telecom SMSC performance report the impact of illegitimate/attack messages is illustrated by the pie chart above. From these statistics, we can see that the bulk of the messages are abnormal/illegitimate messages that utilize license but have no commercial revenue for the operator. Furthermore, these messages cause two percent blocking which results in messages being blocked before even being received by SMSC for processing. As results, messages to and from banks is blocked causing lack of availability of service for mobile financial users. Detecting these illegitimate messages before they reach SMSC will decrease license utilization by up-to forty six percent and decrease the blocking rate to zero.

4.2 Raw Data Understanding

In this phase, initial data acquisition and exploration is conducted to become familiar with the actual data, asses data quality issues, and high-level insights into the data as well as to deduce necessary subsets for availability attack detection. Figure 4-3 below show the overall detection process model. Raw data extraction, pre-processing and eventually coming up with structured data is covered in this chapter. The training, testing and performance evaluation is covered in Chapter 5.

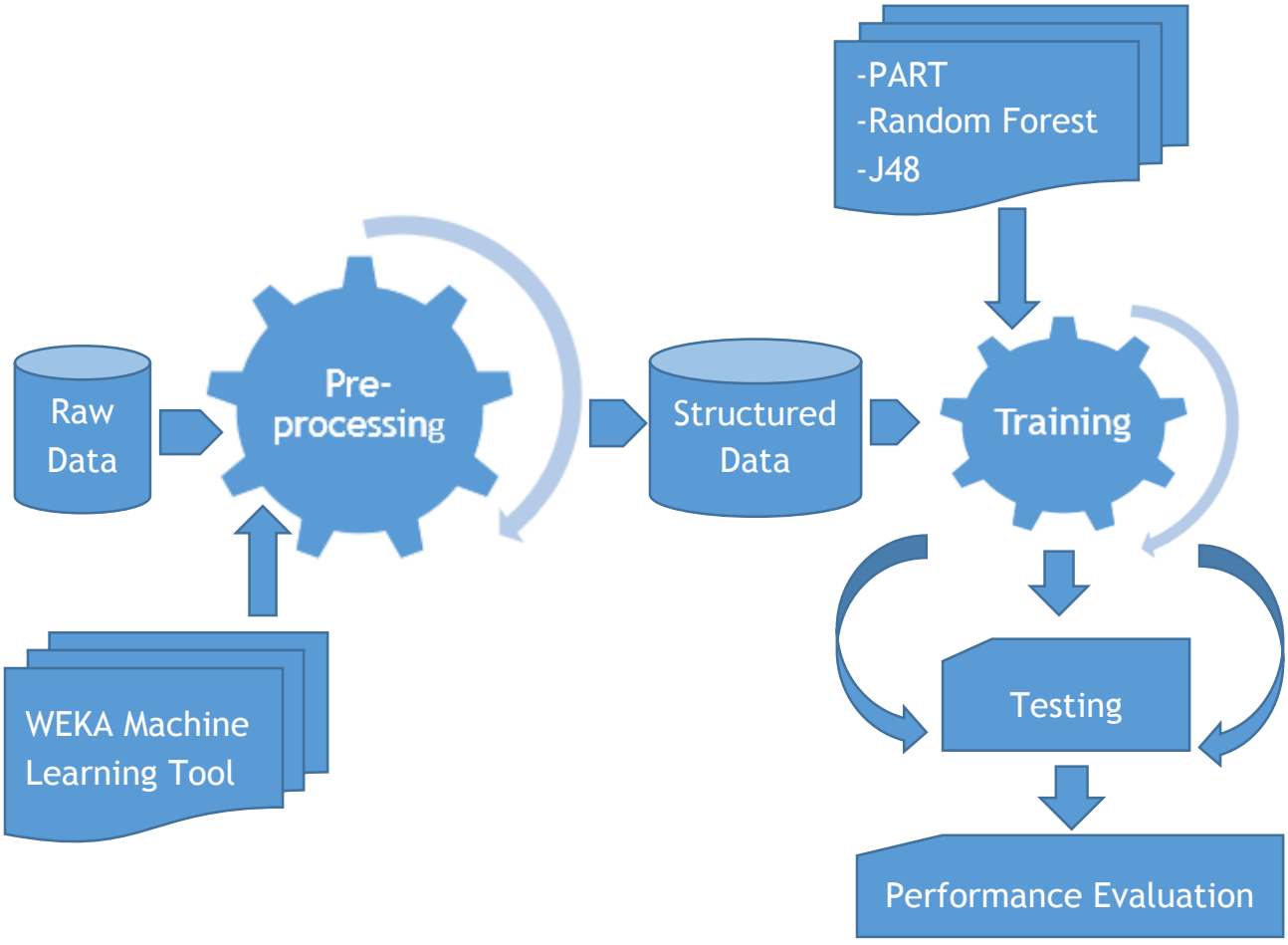

Table 4-3: Detection Process Model

4.2.1 Data Acquisition

There are many network elements that messages transverse before being successful received by the intended recipient as explained in Section 2.2.4. The key to data quality is properly identifying from which part of the network element to acquire data. The data for this research in obtained from ethio telecom SMSC since offline data is available of longer duration.

Ethio telecom has one SMSC that servers all MSCs. Each SMS transaction is stored on SMSC database upon successful receiving by SMSC. This same message is further forwarded to and from mobile financial service providers for mobile bank services. The focus of this thesis is on originated messages since they are the source of attack and once that consume origination license.

The data acquired for this research is extracted from SMSC database. For business understanding, annual statistics is extracted with monthly granularity. Data is extracted in real time as messages arrive during pick traffic and saved for offline processing. A total of *1,048,576* instances are extracted. Data is extracted having all possible attributes since it will be further processed. The raw data is originally extracted and saved in .csv format. These messages are explored below to understand the nature of instances that correspond to each message being received by SMSC.

4.2.2 Data Description

To create a better understanding, Table 4-1 below gives summary of some features with corresponding data type and description. The remaining complete features are as per sequence of appearance are presented in Annex 1.

Serial No.	Feature	Type	Description
1	MAP_ID	STRING	Identification of connected MSC
2	ORGAREA	STRING	Origination Area
3	DESTAREA	STRING	Destination Area
4	ORGSERVICE	STRING	Origination Service
5	DESTSERVICE	STRING	Destination Service
6	ORGISMOBILE	STRING	Originators Mobile
7	DESTISMOBILE	STRING	Destinations Mobile
8	SMID	DWORD	Short Message Identification
9	ORGADDR	STRING	Original Address
10	DESTADDR	STRING	Destination Address
11	MOMSCADDR	STRING	Message Origination MSC Address
12	SCADDR	STRING	Short Message Center Address
13	SCHEDULEMODE	STRING	Schedule Mode

Table 4-1: Description of Sample Features

4.2.3 Data Exploration

To pre-process transaction data, it is essential to explore extracted data and have high-level understanding of the nature of the data. Since the focus of this is thesis is on enhancing availability of mobile banking by detecting illegitimate messages, the extracted messages as intended is only of originated messages. Messages that have been processed for termination and those in retry scheme are not considered and the same is asserted from the extracted data. It is found that each instance of originated message has eight four features that describe an instance of transaction message.

4.2.4 Data Quality Verification

Having gone through the data it is verified that the data is appropriate and can be used to meet the set objectives. The only data quality issues observed are noisy instances and this is handled in Section 4.3.1. Overall, it can be said that the data is complete and relevant since no additional error is observed while loading data into WEKA. Selected algorithms (PART, Random Forest and J48) that are described in Section 2.6.3 to 2.6.5.

4.3 Data Pre-processing

In this phase, the actual preparation of data for modeling is done. Issues such as data quality and feature selection are addressed. Multiple iterations are carried out to ensure that there is clean data that will be fed into Weka machine learning tool from the initial raw data. Tasks included in this phase are data cleaning, feature selection, construction of new feature (feature extraction), data transformation and data reduction.

4.3.1 Data Cleaning

In this phase, data is cleaned to insure it does not generate errors for training and model building. In the first step of data preparation, WEKA tool is used for data pre-processing. In cleaning of data, it is observed there are noisy entries that contain special characters such as “, ;:'"()?!/-_><&#. “. It is also observed that there are missing instance values. The selected algorithms (PART, Random Forest and J48) can manage missing values. The focus is to clear out special character instances to make the data usable. To clean special characters, *StringToWordVector* unsupervised attribute filter is used. As a result, clean data is achieved.

4.3.2 Data Reduction

In this step, the data is reduced to align with objectives for abnormal traffic that originates from mobile users, SS7 intrusion and mobile malware. Computing on huge amount of data can take a very long time. Hence, reduction of data without losing data representation is a key step. To derive a reduced representation of the data set different data reduction techniques can be used. The derived data set is much smaller in volume but one that closely maintains the integrity of the original data. As a result, machine learning on the reduced data would be more efficient and produce the same or almost the same results.

In this research, from the different data reduction mechanisms *Feature Subset Selection* is used to reduce the data set features. *Feature Subset Selection* reduces the data features by removing irrelevant or redundant features. The purpose of *Feature Subset Selection* is to find a minimum set of features which will give result of probability distribution of which as close as to the original data distribution obtained using all set of features. In addition, reduction of features and instances will save the requirement of computational resources. The resulting model will also be easier to understand.

Data reduction is first done by considering only messages that contribute to abnormal traffic. In doing so, messages that originate from internal systems such as customer relation management (CRM), Convergent Billing System (CBS), system alarm management system (I2000) as well Voice Mail System (VMS) are filtered out. Message origination from these sources do not also come through GMSC but have direct connection to SMSC. These set of instances have no value for training and building a model for detection as eventual detection will be done before reaching SMSC. The Features used for reduction are shown in Table 4-2 below.

Serial No.	Feature	Possible values	Reason For Reduction
1	ORGSERVICE	Mobile ESME Internal Services	ESMEs have direct connection with SMSC through VPN and have their own authentication. Internal Services also have direct connection.
2	DESTSERVICE	Mobile ESME Internal Services	ESMEs have direct connection with SMSC through VPN and have their own authentication. Internal Services also have direct connection.

Table 4-2: Features used for data reduction

The internal message instances are removed by filtering using ORGSERVICE and DESTSERVICE features and values shown in Table 4-2. The remaining set of features reduced due to irrelevancy and redundancy is managed under Section 4.3.3 to arrive at an optimal set of features.

4.3.3 Feature Extraction

In this stage, a class feature is extracted for labeling messages as normal or abnormal. This is necessary to change the unsupervised anomaly problem into supervised anomaly machine learning problem. The feature extracted is “RESULT”. It is to be used for message detection.

After a short message (SM) is submitted to the SMSC, the SMSC processes the SM and records the submission result of the SM in the result field of an MO bill. In the case of a submission failure, failure causes are also recorded in the MO bill. This research uses this error codes to label the unsupervised data to a supervised machine learning problem. Having gone through the error codes the following error codes are found in data extracted. According to the failure reason error code messages are labeled with nominal instance of "Yes", this is abnormal message or "No" this is not abnormal message.

Serial No.	Error Code	Error Description	Abnormal Message (Yes/No)
1	0	SM Submission Succeeded: An SM is submitted successfully	NO
2	14	SM Content Error During Anti-Spamming Authentication: The error code indicates that an SM sent to the SMMC contains restricted characters.	NO
3	65	Maximum Submission Number Exceeded: An SM is submitted after the maximum number of SMs that can be submitted by the calling party and saved in the SMCAPP (Short Message Application) has been reached.	YES
4	66	Maximum Delivery Number Exceeded: A short message (SM) destined for a subscriber or service provider (SP) has been submitted after the maximum number of SMs that to be delivered to the subscriber or SP and can be saved in the SMCAPP is reached. A subscriber or an SP (Service Provider) has submitted an SM that requires a status report after the maximum	YES

		number of SMs that to be delivered to the subscriber or SP and can be saved in the SMCAPP is reached.	
5	146	Insufficient Balance of Calling Party: When the calling party of an SM is a prepaid subscriber and has insufficient balance, the error code is returned.	NO
6	150	Unknown SCP Error: The error code indicates that an error that cannot be identified by the SCP occurs.	NO
7	215	Delivery Route Not Obtained: In the MO SM process, the SMSC checks for the delivery route of an SM. If the SMSC finds no delivery route of the SM, the error code is returned.	YES
8	220	Authentication of Destination Address Failed: When the SMSC authenticates the destination address of an SM, if the SMSC finds no matched authentication data or the found authentication data indicates that the destination address is invalid, the error code is returned	YES
9	1066	Caller or Called Address same as SMSC Address: This error code is returned when the caller or called address is same as the SMSC Address.	YES

Table 4-3: Error codes used for Feature Extraction

4.3.4 Features Selection

To build model, a set of features should be selected using expert judgment and/or through carefully selecting the number and type of features used to train the ML algorithm by an automated process of feature selection algorithms. Feature selection algorithms processes are broadly categorized into filter or wrapper model. In Filter model algorithms, a certain metric to rate and select subsets of features is used. The

wrapper method evaluates the performance of different features using specific ML algorithms. As result, it produces feature subsets 'tailored' to the algorithm used [34]. In this research, filter method is used since it does not required a specific ML algorithm.

Having extracted real raw SMS traffic data from SMSC, the first step in data pre-processing is to ensure that only features that have value for the set objectives are used. In the selection of features, the main challenge is finding the most relevant features that will give the highest true positive rate. In doing so, it is necessary to filter or refine many features in the dataset. In this section, a combination of expert judgment and filtering techniques are used to insure the optimal number of features are selected. In the first set of steps, expert judgment through exploration of the data and filtering is done. In the second set of steps, information gain will be used to insure significance of selected features and feature subset selection is used to distinguish redundant features.

The first step in feature selection is to remove features that have no populated values. The reason for removing this set of features is that they have no data entry, as the functionality is not being used. Examples of such features include ORGAREA and DESTAREA. These features are not used since MO is not configured to manage message origination and destination by area. The complete list of features that have been removed using WEKA's functionality of filtering are presented in Annex 2. Forty-eight features are removed in this stage.

The second set of features that are filtered are those features that have same values. Features of same value are filtered out since they add no additional value in training as well as modeling. These set of features removed and the reason for removal is presented in detail in Annex 3. Nine features are removed in this stage.

The third set of features that removed are those that have no correlation with objective of abnormal message detection. Features such as ORGIMSI have been removed since

IMSI is not going to be used for detection. SCADDR is the address of the short message center that the user has configured. This is also filtered as many short message center numbers have been configured to maintain backward compatibility with old SIM cards on MSC. This has no relevance to the problem of detection. The rest of the features that are filtered and the reason for filtering is presented in Annex 4. Three features are removed in this phase.

The four set of features filtered are those features which are only available on SMSC. Detection is to be done before messages reach SMSC and license utilization. Thus, features that are created on SMSC cannot be used for detection. As result, these features are filtered out. The features filtered are SMID and DESTSERVICE. The SMID is the short message identification number created as messages arrive on SMSC and DESTSERVICE describes the destination number whether it is local number or international number. Abnormal destinations fall into both local and international numbers. Keeping this in mind these features are filtered out. Two features are removed and is presented in Annex 5.

The fifth set of features filtered are those features that contain redundant information in relation to other features. It is to be noted that redundant features increase computational time and reduce detection accuracy. Hence, redundant features are filtered out. RAWORGADDRESS is represented with complete set of values in ORGADDRESS, RAWDESTADDRESS in represented with complete set of values in DESTADDRESS, and MOMSCPREFIX is represented in MOMSCADDR with complete set of values. In this step, three features are filtered out and presented in detail in Annex 6.

In the final step, features that are removed are features that describe the nature of a message rather than the volume. This set of features describe parameters such as message type (MESSAGETYPE), reference number of a multi-packet short message, maximum number of packets of a multi-packet short message as well as sequence

number of a multi packet short message. The full list of this of features and the reason for their removal is presented in Annex 7. Fourteen features are removed by this stage.

The final set of features selected for the problem of detection are tabulated Table 4-4 below with their description and relevance.

Serial No.	Feature	Description	Answers
1	DESTADDRESS	DESTADDRESS: Destination address.	To whom is it destined?
2	ORGADDRESS	ORGADDRESS: Address of originator represented by MSISDN.	Who is the originator?
3	MOMSCADDRESS	MOMSCADDRESS: Message Origination Mobile Switching Center Address.	Through which MSC does message arrive?
4	WRITETIME	WRITETIME: Arrival and write time of message.	Arrival time of MO on SMSC
5	RESULT	RESULT: Status of Message	Effect of Message

Table 4-4: Selected Features

To insure the validity of the above deduction. WEKA feature selection filters have also been used. A specific filter called *InfoGainAttributeEval* from Weka machine learning tool was applied for feature ranking. The reason for selecting this filter is that [52] has managed to increase accuracy of malware detection. The result of *InfoGainAttributeEval* shows four features with one class feature as optimal after filtering out all features that relate to a property of a message. These four features derived from the earlier twenty-seven features are in line with literature reviews in Chapter 2, expert judgment and filter results. The *InfoGainAttributeEval* result shows

the significance of each feature in the order presented in Table 4-4 above. Features selection tests show that there is no specific feature that can make a difference in distinguishing normal and abnormal messages, but a group of features. It is a combination of all the selected features that help to find anomalies and not just a single feature. As an instance, specific originating addresses in a certain MSC to a certain destination could be an originator of an attack. The severity of the attack is shown by the frequency of messages. As a result, the combination of different features results in the anomaly detection. It is expected that a classifier will study the message based on a group of features and build a behavior pattern to detect an attack. Any behavior pattern that deviates from the normal pattern as classified under class feature RESULT will be detected as an attack.

We can also infer that the features selected from the dataset are derived from three feature categories. The categories are *destination identifying features*, *source-identifying features*, and *time*-based *feature*. The destination identifying features class is shown as per WEKA feature selection to be the most important traffic feature. The second category of source identifying features is shown to be the second most significant feature set. In this category, originating source number and originating MSC number are significant. The time-based feature in the third category keeps record of the frequency of messages from a certain originator to a certain destination. It is the fourth most significant feature. In the present study, four selected features and one class feature are proposed as shown in Table 4-4 for attack detection. The extracted features were stored as a sequence of *comma separated values* (CSV) files. Each instance consists of the summary data of four message features. MO error responses and expert judgment are used label the dataset information as either abnormal '*YES*' or '*NO*'. Finally, one of the features in the Weka pre-processor *weka.filters.unsupervised.instance.randomize* shuffled the records in the final dataset. Such data randomization ensures test validity.

4.3.5 Data Transformation

In this step that data is transformed to be in usable format as per objective set. The instances the data are transformed or consolidated into forms appropriate for mining. Table below shows the transformed features from their original format to a numerical usable format. This transformation helps the selected decision tree algorithms to manipulate, compare and compute on instance values to arrive at meaningful comparison and result.

Serial No.	Feature	Possible values	Existing Format	Transformed Format
1	ORGADDRESS	Up to 20 characters	STRING	Numeric
2	DESTADDRESS	Up to 20 characters	STRING	Numeric
3	MOMSCADDRESS	12 Characters long	STRING	Numeric
4	WRITETIME	Up to 19 characters	STRING	Time
5	RESULT	Up to 5 characters	STRING	Nominal

Table 4-5: Data Features Transformation

4.4 Structured Data

In previous sections of this chapter, different WEKA data pre-processing tools are used for Data Cleaning, Data Reduction, Feature Extraction, Feature Selection, and Data Transformation. From eighty-four features, five essential features have been selected. The structured data is converted from .csv to WEKA recommended format of .arff. Selected features for detection are to be used for abnormal short messages detection to enhance availability of short message service and eventually mobile banking service availability. The actual training, testing and performance evaluation of selected algorithms is presented in Chapter 5.

5. Results, Evaluation and Analysis

In this chapter, the pre-processed data from previous Chapter are used for training, modeling and finally detection of abnormal messages using the selected algorithms explained in Sections 2.6.3 to 5. The models and results from each of the three selected algorithms: PART, Random Forest and J48 the selected features as explained in Section 4.3.4. Each of the three models are used to detect availability attack messages and their respective results are tabulated. In addition, the detection results from these three algorithms are compared to propose the best performing algorithm in detecting availability attacks in SMS.

The tests were done using WEKA 3.8.2 open source machine learning software due to its simplicity and user-friendly interface. The Java Runtime Environment (JRE) version 1.8.0_181 operated the WEKA software. A laptop computer used for these tests has an Intel core i5-3360M CPU @ 2.80GHz with operating system (Windows 7), memory (4.0 GB of which 3.40GB is usable), having a system type of 32-bit operating system. The selected classifiers functioned in their respective default settings.

As presented in previous chapter, out of the possible eight-four features a gradual selection of features has given us four features and one class feature. The tests are conducted using these selected features. These features are used for training and detection for each of the three selected algorithms (PART, Random Forest and J48). The descriptions of the metrics used for evaluation are explained in Section 2.6.6.

5.1 Algorithm Performance Evaluation Metrics

To assess the results of this study performance evaluation of the selected algorithm is crucial. Many evaluation metrics can be used for algorithm comparison. To evaluate the detection performance successfully, it is necessary to identify appropriate performance metrics.

The first step before actual testing is to ensure that all algorithms are subject to the same method of training. There are three options. The first is percentage split. In this method, data is split into training data and test data in percentage. The larger percentage being for training. The second method is, k-fold cross validation. In this process, the data set is divided into k subsets. Each time, one of the k-subsets is used as the test set and the other k-1 subsets form the training set. Performance statistics are calculated across all k trials. This provides a good indication of how well the classifier will perform on unseen data. k=10 is used research as it is widely used [34]. The last method is supplying test set. Here a certain amount of data is kept for training and a new data set is provided as test set. 10-fold cross-validation and supplied test set methods are used in this research to obtain the best performing algorithm.

After training test is done using different metrics. The metrics that are widely used [34] and selected for this research are accuracy, model build time, True Positive Rate (TPR), F-measure and Receiver Operating characteristic (ROC). Accuracy is the percentage of correctly classified instances over the total number of instances. Model build time is the time required to train a classifier on a given dataset in seconds. True Positive Rate is the value of predicted classification that is correct. F-Measure is the harmonic average of the precision and recall. The mathematical expressions are represented as below.

Accuracy = $\frac{TP+TN}{N+P}$ TPR = $\frac{TP}{TP+FN}$ FPR = $\frac{FP}{TN+FP}$

Precision = $\frac{TP}{TP+FP}$ Recall = $\frac{TP}{TP+FN}$ F-Measure = $\frac{2 * Recall * Precision}{Recall+Precision}$

Where True-Positive Rate (the value of predicted abnormal messages classified correctly), True Negative (TN) (the number of messages correctly classified). False Positive (FP) (the number of normal messages classified as abnormal). False Negative (FN) (the number of abnormal messages classified as normal). Precision is also called positive-predictive value and it returns the rate of relevant results rather than irrelevant results. Recall is the sensitivity for the most relevant result. F-Measure is the value that estimates the entire system performance by combining precision and recall into a single number. The maximum value of 1.000 indicates the best result. ROC is created by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings.

5.2 Training and Testing

In this section, the selected modeling algorithms, namely PART, Random Forest and J48 are used for training and their respective resulting models are used for detection. To insure a balanced level of algorithms comparison, recommended default parameters are used. To come up with a model for each of the selected algorithms, the first step is to train using the structured data from Chapter 4.

The features extraction and selection process are presented in Section 4.3.3 and 4.3.4 respectively. The extracted class feature 'RESULT' is used for classification of a message as normal or abnormal. Messages are designated by nominal value 'YES' it is abnormal message or 'NO' it is not an abnormal message.

The test method for each of the algorithms is to use 10-fold cross-validation and by supplying test set so simulate actual detection. The accuracy and model build time are tabulated for each algorithm. Accuracy and model build time are primarily used to identify the performance of a model. To detail accuracy by class is also tabulated using true positive rate, f-measure and ROC. In addition, the confusion matrix for 10-fold cross-validation is also tabulated for each algorithm.

5.2.1 PART Results

The first algorithm on which test was conducted is PART algorithm. As described in Section 2.6.3 PART is a rule-based algorithm. This test is conducted using different test options namely 10- fold cross-validation and supplied test. The basic comparison for test method is model build time and accuracy of detection. The results of the models for each of the test methods is tabulated in Table 5-1 below. The detail test results using each of the algorithms is summarized in Annex 8.

Test	Test Method	Number of Features	Build Time (Seconds)	Accuracy (Percentage)
1	Cross-validation Folds 10	5	25.85	97.8601
2	Supplied Test Set	5	21.61	97.5662

Table 5-1: PART accuracy and build time

The best result from the test using PART algorithm with selected features is attained using 10-fold cross validation in terms of accuracy. In this test, 114,829 instances (97.8601%) are correctly classified from 117,340 instances. The incorrectly classified instances are 2511(0.9571%). The time taken to build a model is 25.85 Seconds.

	TP Rate	F-Measure	ROC	Class
	0.979	0.980	0.998	Yes
	0.978	0.977	0.998	No
Weighted Average	0.979	0.979	0.998	

Table 5-2: PART detail accuracy by class

The detail accuracy by class is shown in Table 5-2. It shows a True Positive of 0.979 for abnormal messages and 0.978 True Positive for normal messages with F-Measure of

0.980 and 0.977 respectively. ROC Areas for both normal and abnormal messages is 0.998.

Classified	Abnormal	Normal
Abnormal	60,819	1,313
Normal	1,198	54,010

Table 5-3*:* PART Confusion matrix

The confusion matrix for this model test shows that 60,819 instances are classified as abnormal and 1313 instances as normal that should have been abnormal. A total of 54,010 are correctly classified as normal but 1189 instances are classified as abnormal which should have been normal.

To simulate actual detection, external test instances were supplied. The time taken to build the model in this case is 21.61 seconds with an accuracy of 97.5662%. The summary indicates that out of 39,896 records 38,925 (97.5662%) are correctly classified and the remaining 971 instances are incorrectly classified which is 2.4338%. The detail accuracy by class shows that True Positive rate of abnormal messages is 0.973 and for normal messages 0.979. The F-Measure for normal messages is 0.974 and for abnormal messages, it is 0.977. ROC area for both normal and abnormal messages is 0.998. The confusion matrix for this model test shows that 20,679 instances are classified as abnormal and 574 instances as normal that should have been abnormal. A total of 18,246 are correctly classified as abnormal but 397 instances are classified as normal which should have been abnormal.

The comparisons above shows that the model using 10-fold cross-validation test gives an overall better result. This accuracy is taken as the best result using PART algorithm.

5.2.2 Random Forest Results

Random Forest is the second algorithm on which test was conducted. As described in Section 2.6.4 Random Forest is a decision tree algorithm. Just like PART, test is conducted using different test options namely cross-validation and supplied test. The basic comparison for test method is model build time and accuracy of detection. The results of the models for each of the test methods is tabulated in Table 5-4 below. The detail test results using each of the algorithms is summarized in Annex 8.

Test	Test Method	Number of Features	Build Time (Seconds)	Accuracy (Percentage)
1	Cross-validation Folds 10	5	33.13	98.5563
2	Supplied Test Set	5	21.61	98.2855

Table 5-4: Random Forest accuracy and build time

From table 5-4 above, the best result from the test using Random Forest is attained when using 10-fold cross validation. In this test, 115,646 instances (98.5563%) are correctly classified from the total of 117,340 instances. The incorrectly classified instances are 1694 (1.4437%). The time taken to build a model was 33.13 Seconds.

	TP Rate	F-Measure	ROC	Class
	0.987	0.986	0.999	Yes
	0.984	0.985	0.999	No
Weighted Average	0.986	0.986	0.999	

Table 5-5: Random Forest detail accuracy by class

The detail accuracy by class further is shown in Table 5-5 having a True Positive of 0.987 for abnormal messages and 0.984 for normal messages with F-Measure of 0.986 and 0.985 respectively. ROC Areas for both normal and abnormal messages is 0.999.

Classified	Abnormal	Normal
Abnormal	61,347	785
Normal	909	54,299

Table 5-6: Random Forest Confusion matrix

The confusion matrix for this model test shows that 61,347 instances are classified as abnormal and 785 instances as normal that should have been abnormal. A total of 54,299 instances are correctly classified as abnormal but 909 instances are classified as normal which should have been abnormal.

When supplying external tests set, the time taken to build a model is 21.61 seconds with an accuracy of 98.2855%. The summary indicates that out of 39,896 records 39,212 (98.2855%) are correctly classified and the remaining 684 instances are incorrectly classified which is 1.7145%. The detail accuracy by class shows that True Positive rate of abnormal messages is 0.985 and for normal messages 0.980. The F-Measure for normal messages and abnormal messages is 0.984 and 0.982 respectively. ROC area for both normal and abnormal messages is 0.999. The confusion matrix for this model test shows that 20,943 instances are classified as abnormal and 310 instances as normal that should have been abnormal. A total of 18,269 are correctly classified as abnormal but 374 instances are classified as normal which should have been abnormal.

The comparisons above shows that the model using 10-fold cross-validation gives an overall better result. This accuracy is taken as the best result using Random Forest algorithm.

5.2.3 J48 Results

J48 is the third algorithm on which test was conducted. As described in Section 2.6.5 J48 is a decision tree algorithm. This test is conducted using 10-fold cross-validation and supplied test. As in previous tests, the basic comparison for test method is model build time and accuracy of detection.

The set of tests done using J48 algorithm use five features that were selected in Section 4.3.4. The results of the models for each of the test methods is tabulated in Table 5-7 below. The detail test results using each of the algorithms is summarized in Annex 8 so as show an overall result.

Test	Test Method	Number of Features	Build Time (Seconds)	Accuracy (Percentage)
1	Cross-validation Folds 10	5	2.75	98.4379
2	Supplied Test Set	5	2.37	98.2028

Table 5-7: J48 accuracy and build time

The Table 5-7 shows that a better model is achieved when using 10 fold cross validation. The time taken to build the model is 2.75 Seconds with an accuracy of 98.4379%. The summary indicates that out of 117,340 instances 116,507 (98.4379%) are correctly classified and the remaining 1833 instances are incorrectly classified which is 1.5621%.

	TP Rate	F-Measure	ROC	Class
	0.984	0.985	0.997	Yes
	0.985	0.983	0.997	No
Weighted Average	0.984	0.984	0.997	

Table 5-8: J48 detail accuracy by class

The detail accuracy by class shows that True Positive rate of abnormal messages is 0.984 and for normal messages 0.985. The F-Measure for abnormal messages is 0.985 and for normal messages 0.983. ROC area for both normal and abnormal messages is 0.997.

Classified	Abnormal	Normal
Abnormal	61,130	1,002
Normal	831	54,377

Table 5-9: Random Forest Confusion matrix

The confusion matrix for test using this model shows that 61,130 instances are classified as abnormal and 1002 instances as normal that should have been abnormal. A total of 54,377 instances are correctly classified as abnormal but 831 instances are classified as normal which should have been abnormal.

Using Random Forest when external test sets are supplied, 39,179 instances (98.2028%) are correctly classified from the total of 39,896 instances. The incorrectly classified instances are 717 (1.7972%). The detail accuracy by class further shows a True Positive of 0.981 for abnormal messages and 0.983 True Positive for normal messages with F-Measure of 0.983 and 0.981 for abnormal and normal messages respectively. ROC Areas for both normal and abnormal messages is 0.996. The confusion matrix for this model test shows that 20,855 instances are classified as abnormal and 398 instances as normal that should have been abnormal. A total of 18,324 are correctly classified as abnormal but 319 instances are classified as normal which should have been abnormal. The time taken for this model is 2.37 Seconds.

The comparisons above shows that the model using percentage split gives an overall better result. This accuracy is taken as the best result using J48 algorithm.

In summary, in this section, tests have been conducted using 10-fold cross-validation and external supplied test sets. The selected three algorithms, namely, PART, Random Forest and J48 have all been tested with features that have been deduced in Section 4.3.4. The main performance measures used in this chapter are accuracy and model build time. In addition, True Positive rate, F-Measure and ROC have also been analyzed to confirm accuracy. The next section goes in depth into analysis and performance comparison using each of the selected algorithms.

5.3 Performance Evaluation and Analysis

Two models using PART, Random Forest and J48 have been built in Section 5.1 since they have demonstrated high accuracy. In this phase, the performance of resulting models is evaluated and compared to show which of the models has the best overall detection.

Using Supplied test set the maximum model build time and accuracy level difference observed using different algorithms is 29.66 seconds and 0.9174% respectively across the algorithms. The minimum model build time and accuracy level difference observed is same time between PART and Random Forest and 0.0827% respectively across the algorithms. Using the selected five features there is a slight better performance from Random Forest in accuracy but with a much longer model build time. The worst performing algorithm when test is supplied is PART in accuracy with similar time as Random Forest in model build time. Overall, it can be said J48 has the best performance when using supplied test set by considering both accuracy and model build time.

In general, for all most all test cases 10-fold cross-validation has given the best result but having larger model build time. The maximum model build time and accuracy level

difference observed using the different algorithms is 30.38 seconds and 0.6962% respectively across the algorithms. The minimum model build time and accuracy level difference observed using the different algorithms is 7.28 seconds and 0.1184% respectively across the algorithms. Using the selected five features there is a slight better performance from Random Forest but with a much longer model build time. The worst performing algorithm when using 10-fold cross validation is PART in accuracy and Random Forest in model build time. Overall, it can be said J48 has the best performance when using 10-fold cross validation by taking into consideration both accuracy and model build time and it is true when tests are applied. The detail view is tabulated in Annex 8. Furthermore, a comparison using the results of three different supplied test is summarized in Table 5-8 below.

Test No.	Accuracy (Percentage)		Difference (Percentage)	Build Time (Seconds)		Difference (Seconds)
	J48	Random Forest		J48	Random Forest	
1	98.2028	98.2855	0.0827	2.37	21.61	19.24
2	98.1753	98.2705	0.0952	2.70	23.47	20.77
3	98.6465	99.1628	0.5163	3.71	33.37	29.66

Table 5-10: Comparison using three supplied tests

Table 5-10 shows that Random Forest shows a better accuracy ranging from 0.0827% to 0.5163% over J48. However, the speed in model build time is much lower. J48 shows a faster model build time with a difference ranging from 19.24 seconds to 29.66 seconds. From these results, it is inferred that J48 has an acceptable accuracy with a faster build time.

To reassure the performance analysis of the above comparison, the three algorithms have further been compared using the experimenter tool of WEKA. Experimenter is a WEKA application that allows a comparison of several algorithms as per desired metrics. The metrics used are F-Measure, ROC Area, and Accuracy in one side and Model Build Time on the other. The aggregation of these results gives the best performing algorithm. Considering the earlier result where J48 has shown overall better performance, J48 has been used as base algorithm for comparison. It is compared against PART and Random Forest. The comparison results are shown in Annex 9-12.

Random Forest shows a marginal better result over J48 with an accuracy difference of 0.12% with same F-Measure and ROC. On the other hand, J48 shows 3X (three times) faster model build time than Random forest and 2X (two times) faster than PART. Overall results show that J48 is the superior performing algorithm when both set of metrics are combined.

Based on the above results a decision to use J48 for detection is proposed having the features: DESTADDR (Destination Address), ORGADDR (Origination Address), MOMSCADDR (Message Origination Mobile Switching Center Address), WRITETIME (Time for Message Arrival), and the class labeling aggregated error result feature of RESULT (Used for classification of a message as normal or abnormal).

5.3.1 Impact Analysis

The application of the applied results to the detection of unusable messages has significant impact on availability and blocking rate of SMS. After removing internal message traffic, there is an average daily message of twenty million. As discussed in Section 4.1 of business understanding, 46% messages are abnormal (9,200,000 messages) which have no commercial benefit. Considering the highest detection accuracy of J48 (98.6465%), we can detect and block 9,075,478 abnormal messages. This is will create an opportunity for receiving same number of additional messages without causing blockage. As a result, the availability of SMS and mobile banking service is enhanced.

Considering a blocking rate of 2% discussed in Section 4.1, we can infer that up to 400,000 messages are blocked per day. If we convert this into revenue as per current rate of 20 cents per message, there is currently a revenue loss of up to 80,000-birr per day. Once abnormal messages are detected and blocked, the revenue loss incurred due to blocking will translates into a gain of 80,000 birr per day. In addition, any additional costs that would be incurred in MO license expansion can be saved. The effect is also felt in other supporting platforms. For example, convergent billing system will handle less none usable messages. In the message flow of the convergent billing system, air credit is deducted as soon as message is sent and refunded when not successful. As the number of non-usable messages, decrease the usable Call Attempt per Second (CAPS) increases. Thus, making room for more transactions.

In the absence of incomplete transaction due to network blockage, the level of confidence of mobile banking users is also expected to increase. Credit and debit transaction receipts can also be successfully received for every financially transaction. As the level of confidence on existing users increase, it is expected that potential customers will also start to use mobile banking services.

5.4 Deployment

The last phase of CRISP-DM process model is the deployment of the result. This section gives an overview of the repeatable process that is the outcome of this research. The output of this research, models, and rules can now be deployed by creating an interface with existing GMSC to detect availability attacks. The result of this research enables telecom companies, specifically ethio telecom, to use these machine learning models to detect abnormal messages. In addition, a set of rules and policies can be extracted to reconfigure MSCs and SMSC to enhance the detection of abnormal messages.

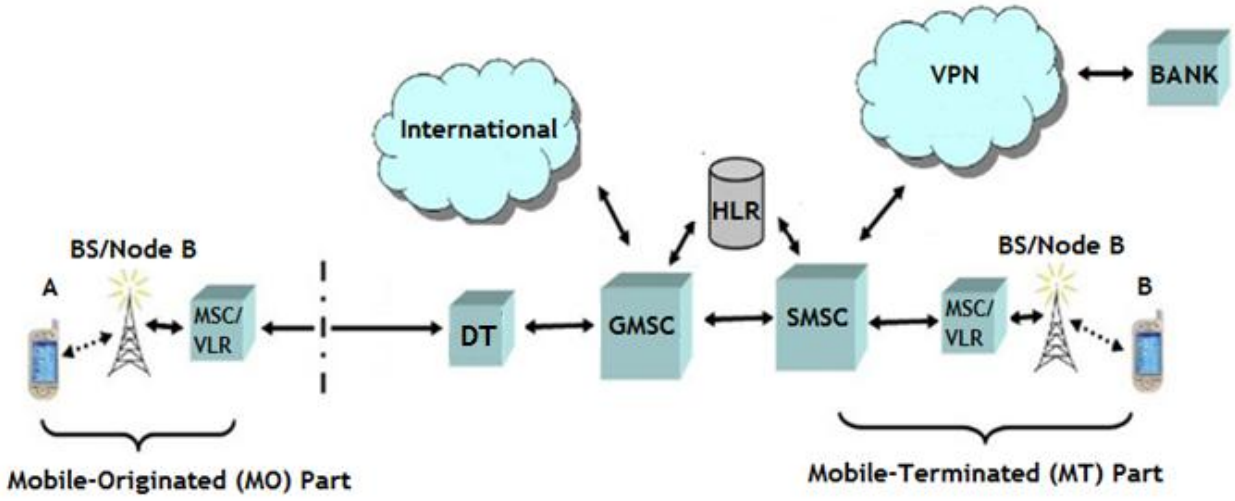

Figure 5-1: Proposed Detection Architecture

Figure 5-1 above shows the proposed detection architecture using the derived models. As explained in Section 4.2.1 messages are extracted from SMSC with their respective error codes. Error code aggregation and class labeling is carried out, followed by training and modeling. Considering the focus of the detection in on MAP application level, messages should be detected before GMSC (the first MAP supporting device) before reaching SMSC and causing damage. In doing so, both SMSC and mobile banking service providers are protected from availability attack messages, blocking rates will decrease and service availability is enhanced. The detection point is designated as 'DT' in Figure 5-1 above.

6. Future Works and Conclusion

Considering the methods and results of this master's thesis, additional questions and concerns arise. In this chapter, areas and recommendations for future work on enhancing mobile banking service availability is presented. In addition, an overall conclusion on this research is also presented.

6.1 Future Works

- The effect of large number illegitimate /attack messages on license of billing system also requires research. This is due to the fact for every messages billing is done in advance and re-funded when messages are not successfully sent.

- The approach used in this thesis used offline data for detection. To insure the real time applicability and feasibility, further research can be done using streaming of data. In addition, future works should also include other types of attacks not covered by this thesis.

- Further work can also be done to insure the performance and efficiency of the proposed machine learning method. Emerging machine learning algorithms as well as combination of algorithms can also be researched on to enhance the performance that was achieved in this thesis.

- The main objective of this thesis has been in detecting of availability attacks. This can be further enriched by implementing an API that can intercept attack messages and block before they reach SMSC.

- The effect of SMS attack messages on voice calls can also been researched since the same SDDCH (Slow Dedicated Control Channel) is shared for both.

6.2 Conclusion

In summary, to enhance availability of mobile banking, DoS attack on availability of mobile messaging should be detected considering features that address all sources of attack. In this research, attacks that originate from mobile phones, malwares, and attacks from SS7 intrusion are considered as discovered from different literature reviews. To address attacks from these three sources, it is found that the effective and optimal features for detection are origination source, destination address, MSC address, and write time, out of the possible eighty-five system features. A combination of expert judgment, relevant literatures reviews as well as WEKA features selection tool have been used for this selection.

Considering the selected features, detection algorithms have been researched that are suitable for detection. It is found that PART, Random Forest and J48 algorithms are effective for detection. Out of these algorithms, J48 has shown superior performance using the combination metric of model build time and accuracy. Modeling using J48 algorithm and subsequent tests using supplied test set has shown that it is possible to achieve an accuracy of 98.6465% and 3.71 seconds model-built time.

Because of this detection, sources and destination addresses can be blocked on GMSC to enhance the availability of messaging before they reach both SMSC and mobile banking servers. It is also shown that, mobile origination success rate will increase and blocking rate on SMSC and banks will decrease. As result, an enhanced mobile banking service availability is achieved.

References

[1] H. Arega, "Mobile Banking in Ethiopia: Challenge and Prospects," *Birritu* vol. 119, pp 10 – 22, February 2015.

[2] A. Shaban. (2017, November 16), *Ethiopia telecoms monopoly now Africa's largest mobile operator*. [Online]. Available: http://www.africanews.com/2017/11/16/ethiopia-telecoms-monopoly-now-africa-s- largest-mobile-operator//, accessed: 25.01.2017.

[3] A. Gugssa, "Assessment of Adoption of Agency Banking Innovation in Ethiopia: Barriers and Drivers," M.S. thesis, Management, AAU, Addis Abeba, 2015.

[4] T. Engel, "SS7: Locate.Track.Manipulate," presented at the Chaos Communication Congress, Hamburg, Germany, 2014

[5] B. Goodwin. (2015, August 14), *Security flaw exposes billions of mobile phones Users to eavesdropping*. [Online]. Available: http://www.computerweekly.com/news/4500251756/Security-flaw-exposes-billions-of mobile-phone-users-to-eavesdropping, Accessed: 25.01.2017.

[6] R. Moore-Colyer. *Criminals Drain European Bank Accounts Using SS7 Security Flaw*, [Online]. Available: http://www.silicon.co.uk/security/ss7-flaw banks-211127, Accessed: 25.01.2017

[7] A. B.Mtaho, "Improving Mobile Money Security with Two-Factor Authentication," *International Journal of Computer Applications*, vol. 109, no. 7, pp. 9–15, 2015.

[8] P. Cravo, "Securing USSD in Mobile Financial Transactions," M.S. thesis, Informatics, ULISBOA, Lisbon, 2011.

[9] P. Kruger, "Cellphone banking at the bottom of the pyramid," M.S. thesis SU, Stellenbosch, Western Cape, 2011.

[10] P. Raju, A. Gajwani, T.A. Gonsalves, R. Srinivas, "Analysis of Mobile Infrastructure for Secure Mobile Payments," *Mobile Payment Forum*, 2008.

[11] B. W. Nyamtiga, A. Sam, L. S. Laizer, "Enhanced Security Model for Mobile Banking Systems in Tanzania," *International Journal of Technology Enhancements and Emerging Engineering Research*, vol. 1, no. 4, pp. 4-20, 2013.

[12] H. Mourad, “The Fall of SS7 - How Can the Critical Security Controls Help?” *SANS Institute InfoSec Reading Room*, 2015.

[13] 3GPP TS 23.002. March 2018. Network architecture (Release 15).

[14] S.Rao, “Analysis and Mitigation of Recent Attacks on Mobile Communication Backend,” M.S. thesis, Institute of Computer Science, UT, Tartu, 2015.

[15] AdaptiveMobile. 2016. Shielding the core: An analysis of real-world attacks on the SS7 network. [Online]. Availabe: http://www:adaptivemobile:com/downloads/shielding-the-core, Accessed: 29.03.18.

[16] L.Dryburgh,J. Hewet, “Mobility Management. In Signaling System No. 7 (SS7/C7): Protocol, Architecture, and Services,” [Online], Available on: https://www.informit.com/library/content.aspx?b=Signaling_System_No_7&seq_Num=116, Accessed: 29.03.18.

[17] D. Kurbatov, S. Puzankov, "Cell Phone Tapping: How It Is Done and Will Anybody Protect Subscribers." 08 April 2014. [Online]. Available: http://blog.ptsecurity.com/2014/08/cell-phone-tapping-how-it-is-done-and.html Accessed: 30.03.2018

[18] F. Narudin, A.Feizollah, N.Anuar and A.Gani, Evaluation of machine learning classifiers for mobile malware detection, *Soft Computing*, pp, 343-357, Jan. 2016

[19] J. Hua and K. Sakurai, “A SMS-Based Mobile Botnet Using Flooding Algorithm,” *Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication Lecture Notes in Computer Science*, pp. 264–279, Jun. 2011.

[20] A.Lelli, “Security Response: A Smart Worm for a Smartphone-WinCE.PmCryptic.A,” [Online], Available on: http://www.symantec.com/connect/blogs/smartworm-smartphone-wincepmcryptica, Accessed: 30.03.2018

[21] A. Apvrille, Symbian worm Yxes: Towards mobile botnets? In: 19th Annual EICAR Conference, France, 2010

[22] C. Mulliner and G. Vigna, “Vulnerability Analysis of MMS User Agents,” *2006 22nd Annual Computer Security Applications Conference (ACSAC06)*, 2006.

[23] P. Traynor, M. Lin, M. Ongtang, V. Rao, T. Jaeger, P. Mcdaniel, and T. L. Porta, “On cellular botnets: Measuring the Impact of Malicious Devices on a Cellular Network Core,” *Proceedings of the 16th ACM conference on Computer and communications security - CCS 09*, Nov. 2009.

[24] S. Shamshirband S, N.B. Anuar, M.L.M Kiah, A. Patel, “An appraisal and design of a multi agent system based cooperative wireless intrusion detection computational intelligence technique,” *Engineering Applications of Artificial Intelligence*, vol 26, no. 9, pp .2105–2127, Oct.2013

[25] A. Ghorbani, A. A. Lu, W. Tavallaee, “Network Intrusion Detection and Prevention: Concepts and Techniques,” Chapter Detection Approaches, 27–53.

[26] S.Behal, G.K.Ahuja, “Detection of DDoS Attacks using Weka Tool: A Case Study,” *Proceedings of National Conference on Computing, Communication & Electrical Systems*, Nov. 2017.

[27] K. Sohr, T. Mustafa, and A. Nowak, “Software security aspects of Java-based mobile phones,” *Proceedings of the 2011 ACM Symposium on Applied Computing - SAC 11*, Mar. 2011

[28] Y. Lai and Z. Liu, “Unknown Malicious Code Detection Based on Bayesian,” *Procedia Engineering*, vol. 15, pp. 3836–3842, Dec. 2011.

[29] D.I. Curiac, C. Volosencu, “Ensemble based sensing anomaly detection in wireless sensor networks,” *Expert Systems with Applications*, vol. 39, no. 10, pp. 9087–9096, 2012

[30] I. Kononenko, M. Kukar, “Machine learning and data mining: Introduction to Principles and Algorithms,” *Choice Reviews Online*, vol. 45, no. 07, Jan. 2008.

[31] T. M. Mitchell, “*Machine Learning*,” New York: McGraw-Hill, 1997.

[32] I. H. Witten, *Data mining: practical machine learning tools and techniques*. Amsterdam: Morgan Kaufmann, 2005

[33] P. Harrington, “Machine Learning In Action,” Manning.

[34] N.Williams, S.Zander, G,Armitage, A Preliminary Performance Comparison of Five

Machine Learning Algorithms for Practical IP Traffic Flow Classification, ACM SIGCOMM Computer Communication Review, Vol 36, No. 5, October 2006

[35] Chandola, V., Banerjee, A., & Kumar, V. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3), 15.

[36] E. Frank, E., H.Witten, "Generating accurate rule sets without global optimization", Working Paper No. 98/2, University of Waikato, 1998

[37] Tin Kam H (1998), The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell 20(8):832–844

[38] Breiman L (2001) Random forests. Machine Learn 45(1):5–32

[39] M. Eskandari and S. Hashemi, "A graph mining approach for detecting unknown malwares," *Journal of Visual Languages & Computing*, vol. 23, no. 3, pp. 154–162, 2012.

[40] X. Su, M. Chuah, and G. Tan, "Smartphone Dual Defense Protection Framework: Detecting Malicious Applications in Android Markets," *2012 8th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, pp 153–160, Dec. 2012.

[41] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer, and Y. Weiss, "'Andromaly': a behavioral malware detection framework for android devices," *Journal of Intelligent Information Systems*, vol. 38, no. 1, pp. 161–190, Jun. 2011

[42] A. Azevedo, M. Santos, "KDD, SEMMA AND CRISP-DM: A Parallel Overview," *Proceedings of the IADIS European Conference on Data Mining*, pp 24-26 July 2008.

[43] R.Wirth, J.Hipp, "CRISP-DM: Towards a Standard Process Model for Data Mining," *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, pp 29-39, Jan. 2000.

[44] K. Jensen, T. V. Do, H. T. Nguyen, and A. Arnes, "Better Protection of SS7 Networks with Machine Learning," *2016 6th International Conference on IT Convergence and Security (ICITCS)*, 2016.

[45] W.Enck, P. Traynor, P.McDaniel, T. L. Porta, "Exploiting open functionality in SMS-capable cellular networks," *Proceedings of the 12th ACM Conference on Computer and Communications Security*, pp. 393–404, Nov. 2005.

[46] P. Traynor, W. Enck, P. Mcdaniel, and T. L. Porta, “Mitigating attacks on open functionality in SMS-capable cellular networks,” *IEEE/ACM Transactions on Networking*, vol.17, no.1, pp 40-53, Feb.2009.

[48] G. Hu and D. Venugopal, “A Malware Signature Extraction and Detection Method Applied to Mobile Networks,” *2007 IEEE International Performance, Computing, and Communications Conference*, 2007.

[49] H. Kim, J. Smith, and K. G. Shin, “Detecting energy-greedy anomalies and mobile malware variants,” *Proceeding of the 6th international conference on Mobile systems, applications, and services - MobiSys 08*, Jun. 2008.

[50] G. Yan, S. Eidenbenz, and E. Galli, “SMS-Watchdog: Profiling Social Behaviors of SMS Users for Anomaly Detection,” *Lecture Notes in Computer Science Recent Advances in Intrusion Detection*, pp. 202–223, Oct. 2009.

[51] Y. Zhou and X. Jiang, “Dissecting Android Malware: Characterization and Evolution,” *2012 IEEE Symposium on Security and Privacy*, 2012.

[52] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer, and Y. Weiss, “‘Andromaly’: A behavioral malware detection framework for android devices,” *Journal of Intelligent Information Systems*, vol. 38, no. 1, pp. 161–190, Jun. 2011

[53] E. K. Kim, P. Mcdaniel, and T. L. Porta, “A Detection Mechanism for SMS Flooding Attacks in Cellular Networks,” *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering Security and Privacy in Communication Networks*, pp. 76–93, Jan. 2013

[54] A. Bose, X. Hu, K. G. Shin, and T. Park, “Behavioral detection of malware on mobile handsets,” *Proceeding of the 6th international conference on Mobile systems, applications, and services - MobiSys 08*, Jun. 2008.

[55] K.Kaur, P.Kakkar, “A Review on Various Machine Learning Techniques for the Detection of DDoS Attacks,” *International Journal of Innovative Research in Science*, vol. 5, no. 9, Sep. 2016.

[56] N. B. Anuar, H. Sallehudin, A. Gani, and O. Zakaria, “Identifying False Alarm for Network Intrusion Detection System Using Hybrid Data Mining and Decision Tree,” *Malaysian Journal of Computer Science*, vol. 21, no. 2, pp. 101–115, Jan. 2008.

Annexes

Annex 1: Data Features Description

Serial No.	Feature	Description
1	MAP_ID	Mobile Application Part ID.
2	ORGAREA	Origination Area.
3	DESTAREA	Destination Area.
4	ORGSERVICE	Origination Service.
5	DESTSERVICE	Destination Service.
6	ORGISMOBILE	Originator Is Mobile.
7	DESTISMOBILE	Destination Is Mobile.
8	SMID	Short Message ID.
9	ORGADDR	Originator Addresses.
10	DESTADDR	Destination Addresses.
11	MOMSCADDR	Message Origination Address.
12	SCADDR	Short Message Center Address.
13	SCHEDULEMODE	Schedule Mode.
14	PRI	Priority of message.
15	RD	Reject Duplicate.
16	RP	Reply path.
17	UDHI	Whether the UDHI is contained in an SM.
18	SRR	Whether the status report is required.
19	MR	Indicates SM index.
20	PID	Protocol type of an SM.
21	DCS	Data coding scheme.
22	SCHEDULE	Scheduled time of delivering the SM
23	EXPIRE	Validity Period of SM calculated by SMSC
24	DEFAULT_ID	Predefined SM ID.

25	UDL	Length of an SM.
26	SMTYPE	Short Message Type.
27	RESULT	Result of Message Status.
28	FCS	MS failure cause.
29	WRITETIME	Write Time of Message.
30	UD	User Data (SM content).
31	SERVICETYPE	Service Type.
32	PPSUSER	Prepaid Subscriber (PPS) or not.
33	ORGACCOUNT	Originator Account.
34	SERVICEFLAG	Service Flag.
35	RAWORGADDRESS	Raw Originator Address.
36	RAWDESTADDRESS	Raw Destination Address.
37	SUBMITMULTIID	Submit Multiple ID.
38	ORGIMSI	Originator IMSI.
39	DESTIMSI	Destination IMSI.
40	ORIGINALGROUP	Original Group.
41	FEEFLAG	Fee Flag.
42	FEETYPE	Fee Type.
43	FEEADDR	Fee Address.
44	ORGCOMMANDID	Indicate the interface type.
45	RAWORGTON	Raw originating TON.
46	RAWORGNPI	Raw originating NPI.
47	RAWDESTTON	Raw Destination TON.
48	RAWDESTNPI	Raw Destination NPI.
49	MOMSCADDRTYPE	Message Origination MSC Address Type.
50	MOMSCTON	Message Origination MSC TON.
51	MOMSCNPI	Message Origination MSC NPI.
52	PPSFEEORGADDR	Prepaid fee originating Address.
53	PPSFEEDESTADDR	Prepaid fee Destination Address.

54	MESSAGETYPE	Type of Message.
55	ORGOPERATORID	Origination Operator ID.
56	DESTOPERATORID	Destination Operator ID.
57	RAWORGIMSI	Raw Originator IMSI.
58	RAWDESTIMSI	Raw Originator IMSI.
59	FEEIMSI	Fee of IMSI.
60	FEEADDROPERATORID	Fee address operator ID.
61	PPSFEEORGTON	Prepaid Subscriber Fee originator TON.
62	PPSFEEORGNPI	Prepaid Subscriber Fee originator NPI.
63	PPSFEEDESTTON	Prepaid subscriber fee destination TON.
64	PPSFEEDESTNPI	Prepaid subscriber fee destination NPI.
65	SSIPARTY	SSI Party
66	ORGPAYTYPE	Originator Payment Type.
67	ORGCUGINDEX	Originator Closed User Group Index.
68	DESTCUGINDEX	Destination Closed User Group Index.
69	SERVICECHARGEFLAGFORORG	Service Charge flag for originator.
70	SERVICECHARGEFLAGFORDEST	Service charge flag for destination.
71	SERVICECHARGEFLAGFORFEEADDR	Service charge flag for address.
72	UDENCRYPTTYPE	User data encryption type.
73	VPNORGADDR	VPN Originator Group.
74	VPNDESTADDR	VPN Destination Group.
75	SUB_ID	Subscriber ID.
76	DESTACCOUNT	Destination Account.
77	MOSUBMITTIME	Message Origination Submit Time.
78	RN	Reference Number of multi packet SM.
79	MN	Maximum number of packets of multi packet SM.
80	SN	Sequence number of current packets of a multi-packet SM.

81	ORGOPERATOR	Originating Operator.
82	DESTOPERATOR	Destination Operator.
83	MOMSCPREFIX	Message Origination Prefix.
84	ISORIGINALUSERROAMING	IS Original User Roaming?

Annex 2: Data Features with null value

Serial No.	Feature	Value	Description
1	ORGAREA	Null	Original Area.
2	DESTAREA	Null	Destination Area.
3	SCHEDULEMODE	Null	Schedule Mode.
4	RP	Null	Reply path.
5	MR	Null	Indicates Short Message Index.
6	SCHEDULE	Null	Schedule.
7	EXPIRE	Null	Expire time.
8	DEFAULT_ID	Null	Predefined Short Message ID
9	FCS	Null	MS Failure Cause.
10	UD	Null	Short Message content.
11	SERVICETYPE	Null	Service type.
12	PPSUSER	Null	Prepaid User.
13	SERVICEFLAG	Null	Service Flag.
14	SUBMITMULTIID	Null	Submit Multiple ID.
15	ORIGINALGROUP	Null	Calling network type.
16	FEEFLAG	Null	Fee Flag.
17	FEETYPE	Null	Fee Type.
18	FEEADDR	Null	Fee Address.
19	ORGCOMMANDID	Null	Type of the interface through which an SM is submitted.

20	MOMSCADDRTYPE	Null	MO MSC address Type (GT/DPC)
21	MOMSCTON	Null	TON of MO SC.
22	MOMSCNPI	Null	NPI of MO SC.
23	PPSFEEORGADDR	Null	Prepaid subscriber Fee Originator Address.
24	PPSFEEDESTADDR	Null	Prepaid subscriber Fee destination Address.
25	ORGOPERATORID	Null	Originating Operator ID.
26	DESTOPERATORID	Null	Destination Operator ID.
27	RAWORGIMSI	Null	Raw Originator IMSI.
28	RAWDESTIMSI	Null	Raw Destination IMSI.
29	FEEIMSI	Null	Fee IMSI.
30	FEEADDROPERATORID	Null	Fee Address Operator ID.
31	PPSFEEORGTON	Null	Prepaid Subscriber Fee originator TON.
32	PPSFEEORGNPI	Null	Prepaid Subscriber Fee Originator NPI.
33	PPSFEEDESTTON	Null	Prepaid Subscriber Fee Destination TON.
34	PPSFEEDESTNPI	Null	Prepaid Subscriber Fee Destination NPI.
35	SSIPARTY	Null	SSI Party
36	ORGPAYTYPE	Null	Originator Pay Type.
37	ORGCUGINDEX	Null	Originator Closed User Group Index.
38	DESTCUGINDEX	Null	Destination Closed User Group Index.
39	SERVICECHARGEFLAGFORORG	Null	Service Charge flag for Originator.

40	SERVICECHARGEFLAGFORDEST	Null	Service Charge Flag for Destination.
41	SERVICECHARGEFLAGFORFEEADDR	Null	Service Charge Flag for Fee Address.
42	VPNORGADDR	Null	VPN Origination Address.
43	VPNDESTADDR	Null	VPN Destination Address.
44	SUB_ID	Null	Subscriber ID.
45	DESTACCOUNT	Null	Destination Account of SM.
46	MOSUBMITTIME	Null	SM Submit Time.
47	ORGOPERATOR	Null	Originator Operator.
48	DESTOPERATOR	Null	Destination Operator.

Annex 3: Data Features with constant values

Serial No.	Feature	Possible Values	Reason For Removal
1	ORGSERVICE	Mobile ESME Internal Services	Only Mobile MO is used others are filtered out
2	ORGISMOBILE	0: No 1: Yes	Originator is mobile since ESME and Internal Service are filtered out.
3	DESTISMOBILE	0: No 1: Yes	Destination is mobile or not. No effect in assessing volume of messages.
4	PRI	0: Normal 1: High	Priority. Only one priority is found set. Value '0' is set for all

			messages. Hence, no effect on traffic.
5	SMTYPE	Message type. 0: common message 1: status report message 2: Anti-Spamming audit message	Since MO is researched only common messages are considered
6	DESTIMSI	IMSI of destination was expected	Destination IMSI address. No entries found after filtering.
7	RAWORGTON	Up to 3 characters giving origination TON	Raw source TON.
8	UDENCRYPTTYPE	Mode for displaying SM content, that is, the UD field. Value: 0, 1, or 2 0: in plain text 1: in AES-encrypted text 2: in hexadecimal text Default value: 0	User data encryption. All MS is set to 1. No added value since there is no value difference.
9	ISORIGINALUSERROAMING	Indicates whether the originating user is in roaming or not. 0: Original user is not in roaming. 1: Original user is in roaming	Data shows only none-roaming users. Hence, this feature is removed. Roaming also has no effect on origination.

Annex 4: Features with no correlation to detection

Serial No.	Feature	Possible values	Reason For Removal
1	ORGACCOUNT	Up to 15 characters configured to use either of the three Gateway_C Gateway_G Srvproxy_S	Submission account of the SM. Whether the message comes from GSM, CDMA, or service proxy. Has no effect in volume.
2	ORGIMSI	15 Characters long	Original International Mobile Subscriber Identity) IMSI. Many Missing values. Message source destination does not consider IMSI.
3	SCADDR	Up to 15 digits	Short Message Center address. Added to accommodate old SMSCs. Regardless SMSC number all messages routed to current SMSC.

Annex 5: Features not available on GMSC

Serial No.	Feature	Possible values	Reason For Removal
1	SMID	Up to 10 characters	Not available on MSC.
2	DESTSERVICE	Mobile ESME Internal Services	Not available on MSC but will be used for data reduction before removal.

Annex 6: Data Features with duplicated values

Serial No.	Feature	Possible values	Reason For Removal
1	RAWORGADDRESS	Up to 20 characters	Raw original address in the submitted message. Duplication of value as same value is found in source address.
2	RAWDESTADDRESS	Up to 20 characters	Raw destination address in the submitted message. Duplication of value as same value is found in destination address.
3	MOMSCPREFIX	12 Characters long	Message Origination MSC Prefix. Missing values and existing values are same us MOMSCADDR

Annex 7: Data Features describing message properties

Serial No.	Feature	Possible values	Reason For Removal
1	RD	0: Not Reject Duplicate 1: Replace Existing 2: Reject Duplicate 3: Delete the broadcast SM	Reject Duplicate (RD). Used for SMPP messages. It is an SM replacement Flag for messages in waiting for SMPP messages. No effect on message volume.
2	UDHI	0: No. 1: Yes. Other values: Reserved	User Data Header Indicator (UDHI). Used for messages processing and formatting. No effect on message volume.
3	SRR	0: No. 1: Yes. Other values: Reserved	Status Report is Required (SRR). Whether status is required or not origination is not affected.
4	PID	The value ranges from 0 to 255	Protocol Identification (PID). Protocol change does not show any traffic change.
5	DCS	Up to 3 characters. Code for Data coding scheme	Data coding scheme. Does not show any relevance to the amount of SM
6	UDL	Up to 10 characters	Length of Short message. No effect on volume.

7	RAWORGNPI	Up to 3 characters	Raw source NPI.
8	RAWDESTTON	Up to 3 characters	Raw destination TON.
9	RAWDESTNPI	Up to 3 characters	Raw destination NPI.
10	MESSAGETYPE	Up to 3 characters	Type of SM. It specifies the source and destination type. Interest is in finding abnormal messages regardless of message type.
11	RN	Up to 5 characters	Indicates reference number of a multi-packet SM.
12	MN	Up to 3 characters	Maximum number of packets of a multi-packet SM.
13	SN	Up to 3 characters	Sequence number of the current packet of a multi-packet SM
14	MAP_ID	Up to 3 characters	ID of Mobile Application Part gives the connected MSC

Annex 8: Test on group of Features

Algorithm	Test Type	No. of Features	Build Time	Accuracy
PART	Cross-validation Folds 10	5	25.85	97.8601
	Supplied Test Set	5	21.61 23.41 33.18	97.5662 97.4935 98.2454
Random Forest	Cross-validation Folds 10	5	33.13	98.5563
	Supplied Test Set	5	21.61 23.47 33.37	98.2855 98.2705 99.1628
J48	Cross-validation Folds 10	5	2.75	98.4379
	Supplied Test Set	5	2.37 2.70 3.71	98.2028 98.1753 98.6465

Annex 9: Model Build Time Comparison

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMatrixPlainText
Analysing:  Elapsed_Time_training
Datasets:   1
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       10/9/18 6:32 PM


Dataset                   (1) trees.J4 | (2) trees (3) rules
----------------------------------------------------------------
'3. 5 .Labeling Yes and N (10)   10.32 |   34.69 v   22.07 v
----------------------------------------------------------------
                               (v/ /*) |   (1/0/0)   (1/0/0)


Key:
(1) trees.J48 '-C 0.25 -M 2' -217733168393644444
(2) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698
(3) rules.PART '-M 2 -C 0.25 -Q 1' 8121455039782598361
```

Annex 10: Percent Correct Comparison

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMatrixPlainText
Analysing:  Percent_correct
Datasets:   1
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       10/9/18 6:30 PM


Dataset                   (1) trees.J4 | (2) trees (3) rules
----------------------------------------------------------------
'3. 5 .Labeling Yes and N (10)   98.44 |   98.56 v   97.86 *
----------------------------------------------------------------
                               (v/ /*) |   (1/0/0)   (0/0/1)


Key:
(1) trees.J48 '-C 0.25 -M 2' -217733168393644444
(2) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698
(3) rules.PART '-M 2 -C 0.25 -Q 1' 8121455039782598361
```

Annex 11: F-Measure Comparison

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMatrixPlainText
Analysing:  F_measure
Datasets:   1
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       10/9/18 6:28 PM


Dataset                   (1) trees.J | (2) tree (3) rule
----------------------------------------------------------
'3. 5 .Labeling Yes and N (10)   0.99 |   0.99 v   0.98 *
----------------------------------------------------------
                              (v/ /*) |  (1/0/0)  (0/0/1)


Key:
(1) trees.J48 '-C 0.25 -M 2' -217733168393644444
(2) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698
(3) rules.PART '-M 2 -C 0.25 -Q 1' 8121455039782598361
```

Annex 12: Area under ROC Comparison

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMatrixPlainText
Analysing:  Area_under_ROC
Datasets:   1
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       10/9/18 6:29 PM


Dataset                   (1) trees.J | (2) tree (3) rule
----------------------------------------------------------
'3. 5 .Labeling Yes and N (10)   1.00 |   1.00 v   1.00
----------------------------------------------------------
                              (v/ /*) |  (1/0/0)  (0/1/0)


Key:
(1) trees.J48 '-C 0.25 -M 2' -217733168393644444
(2) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698
(3) rules.PART '-M 2 -C 0.25 -Q 1' 8121455039782598361
```