INSTANT MESSAGING SPAM DETECTION IN LONG TERM EVOLUTION NETWORKS

Swagata Das

A thesis
in
The Department
of
Concordia Institute for Information Systems Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Applied Science
(Quality Systems Engineering)
Concordia University
Montréal, Québec, Canada

August, 2013

Concordia University

School of Graduate Studies

This is to certify that the thesis prepared

By: **Swagata Das**

Entitled: **Instant Messaging Spam Detection in Long Term Evolution Networks**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science
(Quality Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Chadi Assi	Chair
Dr. Mourad Debbabi	Supervisor
Dr. Makan Pourzandi	Supervisor
Dr. Amr Youssef	CIISE Examiner
Dr. Walaa Hamouda	External Examiner

Approved ______________________________
Chair of Department or Graduate Program Director

__________ 2013 ______________________________
Dean, Faculty of Engineering and Computer Science

Abstract

Instant Messaging Spam Detection in
Long Term Evolution Networks

Swagata Das

The lack of efficient spam detection modules for packet data communication is resulting to increased threat exposure for the telecommunication network users and the service providers. In this thesis, we propose a novel approach to classify spam at the server side by intercepting packet-data communication among instant messaging applications. Spam detection is performed using machine learning techniques on packet headers and contents (if unencrypted) in two different phases: offline training and online classification. The contribution of this study is threefold. First, it identifies the scope of deploying a spam detection module in a state-of-the-art telecommunication architecture. Secondly, it compares the usefulness of various existing machine learning algorithms in order to intercept and classify data packets in near real-time communication of the instant messengers. Finally, it evaluates the accuracy and classification time of spam detection using our approach in a simulated environment of continuous packet data communication. Our research results are mainly generated by executing instances of a peer-to-peer instant messaging application prototype within a simulated Long Term Evolution (LTE) telecommunication network environment. This prototype is modeled and executed using OPNET network modeling and simulation tools. The research produces considerable knowledge on addressing unsolicited packet monitoring in instant messaging and similar applications.

To, My Parents and Family.

Acknowledgments

This dissertation would not be complete without the help of several people who have assisted and influenced my quest for the unknown throughout the academic program. I would like to acknowledge and extend my earnest gratitude to all of them who encouraged me to successfully complete this research. Most of all, I owe a great debt of gratitude to my supervisors Professor Mourad Debbabi and Professor Makan Pourzandi. I consider myself fortunate to be working under the guidance of Dr. Debbabi and receiving affluent knowledge toward my research topic. I would also like to thank Dr. Pourzandi from Ericsson Software Research, who provided ample facilities and opportunities for this research. I do appreciate his conceptual insights, discussion and scientific comments.

The research described in this thesis is a part of a major project on the threat detection of Universal Mobile Telecommunication System (UMTS). This project is a joint collaboration of National Cyber-Forensics and Training Alliance Canada, Ericsson Research and Computer Security Laboratory at Concordia University.

I would also like to express my gratitude to my colleagues in Computer Security Laboratory, Mr. Elias Bou-Harb and Mr. Feras AlJumah in particular. They helped me by sharing their ideas and knowledge. Thank you all for your help.

Last but not the least, my deepest gratitude goes to my beloved parents, family, sisters and friends who unflaggingly love and support me throughout my life.

Swagata Das
August, 2013

Contents

List of Figures

List of Tables

List of Equations

List of Abbreviations

Abbreviation	Full Term
3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization, Accounting
ACE	Application Characterization Environment
APN	Access Point Name
AS	Application Server
ASN	Autonomous System Number
CDMA	Code Division Multiple Access
CSCF	Call Session Control Function
DNS	Domain Name Service
e-NodeB	evolved- NodeB
EPDG	Evolve Packet Data Gateway
EPS	Evolve Packet System
E-UTRAN	Evolve- Universal Terrestrial Radio Access Network
EV-DO	EVolution- Data Optimized
FSPAM	Forum SPAM
GB	Giga Byte
Gbps	Giga bit per second
GBR	Guaranteed Bit-Rate
GSM	Global System for Mobile Communication
GUI	Graphic User Interface
HSDPA	High Speed Downlink Packet Access
HSPA	High Speed Packet Access
HSUPA	High Speed Uplink Packet Access
HTTP	Hyper Text Transfer Protocol
ID3	Iterative Dichotomiser 3
IM	Instant Messaging
IMS	IP Multimedia Subsystem
IP	Internet Protocol
ISP	Internet Service Provider
ITU	International Telecommunication Union
K-NN	K- Nearest Neighbour
LAN	Local Area Network
LTE	Long Term Evolution
MAAWG	Massage Anti-Abuse Working Group
MAC	Medium Access Control
MaxEnt	Maximum Entropy
Mbps	Mega bit per second
MIM	Mobile Instant Messaging

Abbreviation	Full Term
MIMO	Multiple Input Multiple Output
MME	Mobility Management Entity
MRF	Media Resource Function
MRFC	Media Resource Function Controller
MRFP	Media Resource Function Processor
MSRP	Message Session Relay Protocol
NAS	Non-Access Stratum
NB	Naive Bayes
NS	Network Simulator
OFDMA	Orthogonal Frequency Division Multiple Access
OFDMA-CRT	OFDMA with Cognitive Radio Technology
PA	Presence Agent
PCEF	Policy Control Enforcement Function
PCRF	Policy and Charging Control Function
PDN Gateway/P-GW	Packet Data Network Gateway
PMIP	Proxy Mobile Internet Protocol
PUA	Presence User Agent
PUI	Public User Identity
QoS	Quality of Service
RTP	Real-time Transport Protocol
RWIN	Repeat Window
SAE	System Architecture Evolution
SC-FDMA	Single Carrier- Frequency Division Multiple Access
SDF	Service Data Flow
SDP	Session Description Protocol
S-GW	Serving Gateway
SIMPLE	Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions
SIP	Session Initiation Protocol
SMS	Short Message Service
SPIM	SPam over Instant Messaging
SPIT	SPam over Internet Telephony
SPLOG	SPam over BLOG
SVM	Support Vector Machine
SVM-SMO	SVM- Sequential Minimal Optimization
TCL	Tool Command Language
TCP	Transmission Control Protocol
ToS	Theft of Service
UAA	User Authentication Answer
UAR	User Authentication Request
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunication System
URI	Uniform Resource Identi?er
VoIP	VOice over Internet Protocol
WAN	Wide Area Network
WIM	Wireless Instant Messaging
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WRAN	Wireless Regional Area Network
WWAN	Wireless Wide Area Network
XMPP	Extensible Messaging and Presence Protocol

Chapter 1

Introduction

The evolution of message exchange techniques is bringing newer security concerns to Internet users and network managers. These concerns are largely stemming from the unsolicited and malicious information being transmitted to the users. The resulting damage has been seen in the form of excess data usage, network load imbalance, theft of privacy information, denial-of-service attacks and other cyber-crimes. The increasing popularity of unsolicited messages, commonly referred to spam, is due to the soaring business interest in reaching potential customers through Internet-based data sharing. The spam detection is crucial to Internet Service Providers (ISP) for their secured functioning, network load reduction and customer satisfaction. In this regard, the subject is widely studied for emails. Several content-based spam removal techniques are also commercially implemented and deployed. However, applications of these detection techniques are very limited in faster interactive communication, such as Instant Messaging (IM). In this thesis, we evaluate the suitability and deployment of a spam detection module in telecommunication networks especially applicable for messaging applications [41]. The proposed technique is based on machine learning procedure and tested on packet data in a simulated environment of Long Term Evolution (LTE) framework. The contributions of this dissertation are: (i) gap analysis of the requirements and existing techniques; (ii) evaluation of applicable machine learning algorithms; (iii) proposing architecture for an IM spam detection module in LTE; and (iv) experimental validation of the execution of such module in a network simulator.

1.1 Motivation

The detection and the removal of spams are subjected to well-known legal and technical challenges. There exist various legislative efforts worldwide, such as European Union Privacy and Electronic Communications Directive, US CAN-SPAM Act, etc. Canada's own anti-spam legislation has also been passed in 2010 as Bill C-28 [1]. However, technological complexity, remote location and caveats within the Internet often keep the spammers out of organizational reach. We summarize the major difficulties in spam identification as follows:

- Lack of profound spam detection/classification techniques in user and server platforms.
- Difficulty of identifying and removing actual spam senders (spammers),
- Speed, volume of and inter-active nature of message communication,
- Use of automated programs (for example: spambot, chatbot) to send spam messages,
- Frequent changes of spam delivery strategies at the sender's end,
- Lack of knowledge/attention from end-users against possible threats and exposures.

Spam removal is an important industrial and academic research issue in message communication. Removal of spam has been first addressed in emails. Renowned statistical reports [3] indicate that, in 2011, approximately 3.1 billion active email accounts existed in Internet. According to the Message Anti-Abuse Working Group (MAAWG), email-spam constituted as 88-90% of the total email messages sent in 2011 first quarter [75]. Such a large volume of email spam generates exorbitant financial loss by using Internet bandwidth unnecessarily [26, 60]. In 2005, this loss was estimated as 50 billion of US currency [60]. Such an enormous volume of spam and associated financial loss called for implementing robust mechanisms for spam detection at industrial scale. In the last decade, ISPs and independent software giants invested large amount of funds to ensure spam-free emailing services. On the other hand, given the current growth of 11%, the number of IM user accounts are expected to reach 3.5 billion by 2015. In 2004, Claburn statistically illustrated that almost 8% of all the corporate instant messages are spam [68]. Furthermore, they informed that due to fast, interactive and low-cost communication, IM is soon going to be an attractive platform for the spammers to deliver commercial messages. Spammers are seen increasingly targeting near real-time communication software to deliver messages. Recently, Facebook® acknowledged that more than 1 billion IM messages are being exchanged per day in its network [58]. Multiple independent reports are already published stating the requirements of efficient SPIM detection mechanisms [68]. However, only a very few mechanism exists to reduce message spam for IM services. In absence of such mechanisms, the spammers and cyber-attackers are constantly exploiting this inadequacy. In 2002, MassMess, a Russia based company, sent 10 million unsolicited commercial messages on Yahoo Messenger users [48]. In 2004, New York police arrested a teenager in CAN-SPAM act for sending 1.5 million SPam over Instant Messages (SPIM) on mortgage refinancing and adult pornography [74]. In 2005, the IMlogic report indicated 1,693% increase of IM threats from the last year [19].

Securing IM environment requires elaborated research in various directions. In this thesis, we elaborate a new technique for SPIM detection from the packet flow inside telecommunication framework. The proposed detection module uses machine learning algorithm to detect SPIM over different instant messaging communications. The detection time and accuracy of the proposed module are successfully measured in a prototyped LTE framework deployed on OPNET simulation environment. These results reveal sufficient detection accuracy and small detection time for some machine learning algorithms. We expect that our finding can be used to modulate packet flow among Internet users in order to achieve overall user satisfaction and network load balance.

1.2 Overview of Message Spams

The need for messaging lies in the necessity of information sharing among distributed entities that are engaged in collecting, processing, producing and storing information on heterogeneous platforms. Spams are seen in almost every areas of information sharing, such as email, Voice over IP (VoIP), instant messaging, blog, newsgroup, forum, Short Messaging Services

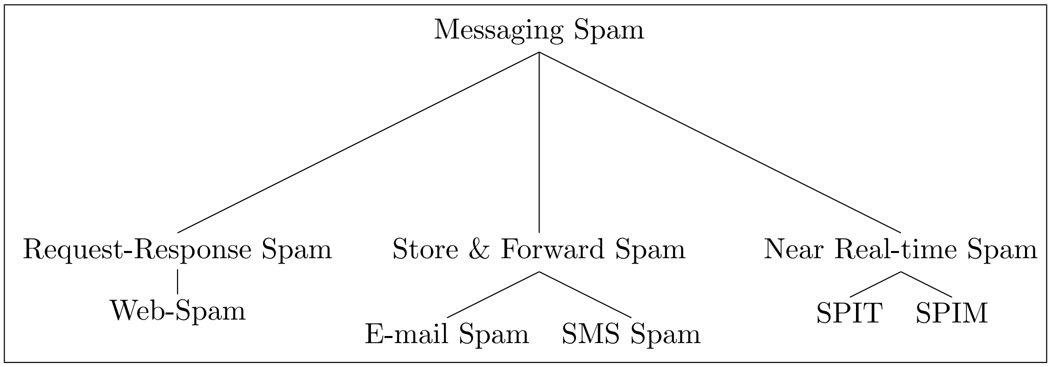


Figure 1: High-Level Taxonomy of Spam

(SMS), etc. Security research communities attributed different names to the spam based on their area of activities [14]. While email-spam is widely known, spams are also called as SPIT, SPIM, FSPAM, SPLOG in the area of VOIP, IM, forum and blog respectively[1]. In this dissertation, we are mainly focused on detecting instant messaging spam (SPIM).

Message spams are intentionally created texts that are indiscriminately sent without the consent of the recipients [55]. Different taxonomies exist on the message spams in the literature [22, 23, 55]. Figure 1 presents one such spam taxonomy from the data communication perspective. We consider different message communication techniques namely: request-response, store-forward and near real-time. We classify web-spam as request-response type. E-mail and sms-spam are store-forward data communication while SPIM and SPIT are of near real-time. In the literature, few articles also consider web-spam as message spam, as it comes as a result of Internet user's request. It includes different techniques such as term spamming, link spamming, hiding, etc. as detailed by Gyngyi and Garcia-Molina [55]. Bookmark spam, comment spam, blog-spam (SPLOG) and social network spam are also some examples of web-spam. These types of spamming are indirect and they do not really require to establish sessions with victims. In contrary, the other two types of spam are directly exchanged between spam-senders and receivers.

Most commonly, spams are sent through emails by writing text, adding unsolicited attachments or putting links of the propagandas and malware. The unsolicited electronic

[1] see *List of Abbreviations* for more details

content is often sent as a bulk to multiple recipients. There are different techniques involved in email-spamming such as image attachment, blank email, backscattering, etc. A good number of methodologies have also been established to classify the email-spam. The other type of store and forward spam is called Short Message Service (SMS)-spam that refers to the propagation of unsolicited texts usually containing advertisements through short message service. The distinguished characteristics of these spams include their small size and frequent use of non-dictionary words. SMS-spam is not as spread as its email counterpart. Strict rules, restrictions, and monetary charges have been imposed worldwide over user connection to set limits in such spam propagation. Both of these store and forward spams travel through intermediate servers, that also keep a copy of the messages. Consequently, these messages (including spam) can be delivered to the active, as well as, currently inactive users after their next activation. Consequently, this type of spams can be classified through reputation-based and content-based techniques [22] on the storages before or after delivery.

In contrast, instant messaging spams are generated through ubiquitous spamming techniques with high potential danger. SPIM can be delivered only to a registered "online" recipient though the instant messaging applications. The spam message comes through a chat window and therefore bypasses almost all security settings through a pre-installed messaging application in a user device. The window pops up advertisement, links to viruses and spyware, etc. It is also able to deliver applications, such as Trojan Horse, that are capable to install themselves inside the user device. recently, Security experts in Governments, corporations and ISPs have been continuously warning against SPIM because of its high intrusive character owing to its design to pass local security settings. SPIT refers to similar unsolicited "spam" calls using Voice over Internet Protocol (VoIP). Spammers use automated calling application (bots) for the purpose of telemarketing, prank-call and other abuses. As the VoIP is continuously changing the conventional telephony by low-cost communication, spammers are increasingly targeting this platform to reach out to large group of callers. SPIT is delivered by exploiting pitfalls in the underlying protocol, namely: Session Initiation Protocol [53]. However, their detection is not easy due to real-time communication and associated legal challenges concerning call privacy [28].

The impact of spamming is threefold. First, it affects the privacy and security of the spam recipients. Secondly, it creates vulnerabilities in the whole network that is hosting these user devices. Thirdly, it has a larger effect on the corporate resources and infrastructure since a significant amount of corporate resource get wasted to serve these unsolicited messages. More recently, spam sending bots are seen in attempting social engineering, gathering intelligence, mounting phishing attacks, spreading malware and thereby threatening the usability and security of the collaborative communication platforms. In the 3rd Generation Partnership Project (3GPP), technical specification group quantified that approximately 250 GB of SPIT traffic per month can be generated from only one SPIT bot [28]. In absence of effective filtering policies, SPIM is also seen generating significant potential revenue [68]. In 2005, Steve Roche reported in his book that 5% of the IM is SPIM [90]. In overall SPIM, 70% of messages carry links to pornographic websites, 12% contain "get rich" schemes and 9% promotes product sales [90]. Therefore, high academic and industrial interest are required to bridge up the gap.

1.3 Problem Statement

In this thesis, we primarily focus on understanding and identifying the scope of intercepting IM data traffic at the LTE /IMS networks. We also investigate for a suitable technique to classify SPIM from large LTE traffic with adequate accuracy and small classification time. After a thorough analysis, we realized that a complex analytical solution with qualitative or quantitative measurement of SPIM turns impractical to match the near real-time flow-speed of huge incoming information. In this setting, machine-learning may offer a better alternative to classify information based on predefined rules. These rules are often formed by mining large-scale information and used in intelligent detection of matching patterns on the incoming packets. Consequently, we evaluate different existing machine learning algorithms and measure their performance to classify spam after training with small packets (in contrary to email spam detection where the data content can be big). Finally, we stress on accurate feature selection from packets such that the chosen algorithms can perform well also in presence of encrypted data content.

In practice, SPIM detection is extremely difficult because of various complexities associated to IM applications, messaging architecture and protocols. First, instant messaging applications are often peer-to-peer applications where the application server is only needed for session binding. Secondly, IM is a near real-time communication with multiple services (presence, messaging, content sharing, etc.). Therefore, online SPIM inspection is required to be fast and accurate. Thirdly, few messaging protocols allows transferring of short and symbolic texts along with voice and video messages through the same application that can be hardly distinguished in the external network [27]. Finally, third-party IM applications often do not follow industrial messaging standards and include additional data protection by using proprietary protocols, encryption, etc.

Our proposed solution is very generic, however, throughout the thesis, we precisely focus on identifying peer-to-peer IM communication. As the model of the data generating process is too complex and partly unknown (as expected in our case), creation of a perfect spam detection technique is impossible. Therefore, we expect a small amount of wrongful attribution in classification of the incoming IM packets. We envision a possible use our detection module in regulating the data flow from SPIM sending applications where the removal of packet is not an acceptable solution due to technical and legal issues.

1.4 Objectives

In this thesis, we investigate for an effective and efficient SPIM detection approach in LTE mobile framework. The objectives of this dissertation are as follows:

- Studying data communication in LTE and analyzing the scope of SPIM detection within this telecommunication network.
- Finding the location and design of SPIM detection module in LTE.
- Analyzing the suitability and the complexities of different machine learning algorithms for the proposed SPIM detection.

- Selecting important features of SPIM classification from packet data.

- Designing and deploying the proposed module using an existing network simulator.

- Conducting experiments and presenting ‘online’ spam classification time and accuracy among chosen machine learning techniques.

Our technique involves the use of machine learning algorithms on the packet features that are mainly payload independent or associated to the packet headers. We use an online SPIM detector at the Packet Data Network-Gateway inside an LTE component over downlink packet stream. In this thesis, we evaluate four machine learning algorithms namely Naive Bayes (NB), Support Vector Machine (SVM), AdaBoost, and J48. SPIM detector may output a general insight about percentage of SPIM in the incoming traffic over specific time-period. This insight can be later translated to regulate the packet-flow from various IPs. The presented study mainly concerns about the feasibility of machine learning technique in SPIM monitoring with respect to time and accuracy. We consciously ignore the details of packet data processing or packet control in the mobile framework.

1.5 Thesis Organization

The rest of this thesis is organized in four more chapters. Chapter 2 introduces an overview of the literature including remarkable previous research work in this field. It also covers the state-of-the-art for existing infrastructure and a background study on instant messaging. Furthermore, different messaging protocols are discussed in relevance to instant messaging protocols. In Chapter 3, we describe our approach of SPIM detection and analyze the choices for our machine learning algorithms. It also presents a set of features that are suitable for SPIM detection from incoming network traffic using algorithms of machine-learning. Chapter 4 elaborates on the validation of our approach by presenting and analyzing results found from the simulation runs using a renowned network simulator [10]. Finally, Chapter 5 summarizes our contribution along with specific limitations of the approach. Consequently, we also outline future steps that may enhance and consolidate this approach.

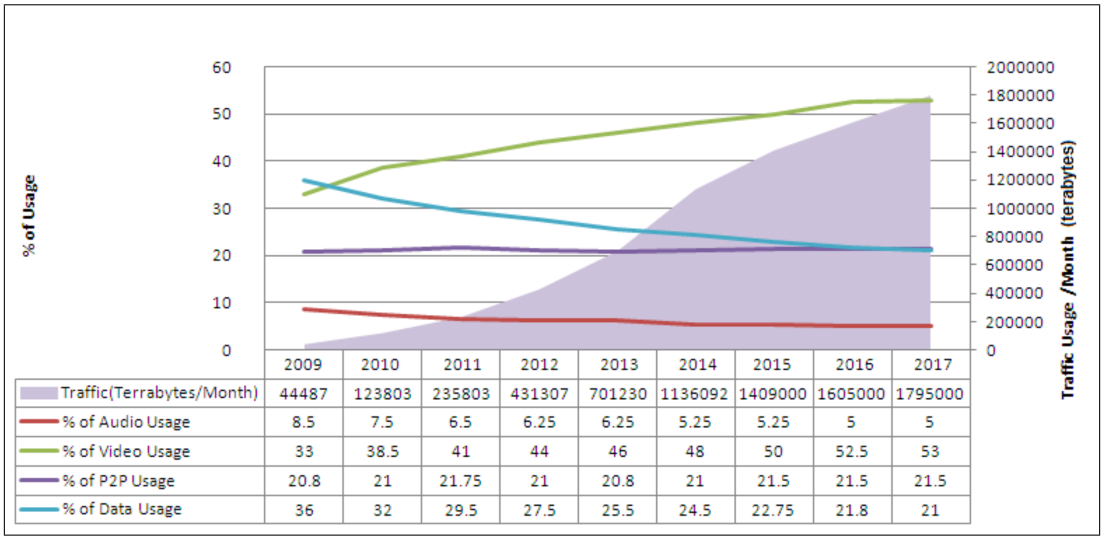
Chapter 2

Background

In this chapter, we elaborate on the state-of-the-art. We start by presenting an overview of necessity for SPIM detection in advanced telecommunication networks followed by a short review of previous research work in this direction. Afterward, we illustrate the target platform (LTE) and the specific message communication (IM) of our interest. We illustrate necessary LTE telecommunications network components aiming to find a suitable location for the SPIM detector. We also discuss IM-services along with different protocols. Finally, we describe on various machine learning techniques in connection to our approach.

2.1 Overview

As the telecommunication is becoming more crucial for the information exchange, the demand for faster data sharing is also rapidly growing in the industry. Spammers are taking advantage of such exchanges to deliver the unsolicited messages or spams. Spams demand a significant amount of corporate resources and attentions as they represent a potential cause of various cyber-threats. However, keeping the telecommunications system spam-free is probably impossible. In this thesis, we aim at finding a proper set of technologies and algorithms to develop a module for on-line spam detection applicable to telecommunication networks. We expect that our insight can be transformed into key decision input to reduce spam messages flow for Internet Service Providers (ISP). The detection mechanism is



	2009	2010	2011	2012	2013	2014	2015	2016	2017
Traffic(Terrabytes/Month)	44487	123803	235803	431307	701230	1136092	1409000	1605000	1795000
% of Audio Usage	8.5	7.5	6.5	6.25	6.25	5.25	5.25	5	5
% of Video Usage	33	38.5	41	44	46	48	50	52.5	53
% of P2P Usage	20.8	21	21.75	21	20.8	21	21.5	21.5	21.5
% of Data Usage	36	32	29.5	27.5	25.5	24.5	22.75	21.8	21

Figure 2: Forecast of Different Mobile Broadband Traffic

designed to function inside the LTE network from topological and functional perspectives.

Figure 2 renders the expected worldwide information sharing pattern for the near future. The graph is produced by combining the statistical information published by Coda Research Organization in 2009 [2]. In their published reports, the company estimated that almost 80% of the world's three billion broadband subscriptions will turn to advanced mobile telecommunications by 2013 and there will be huge increase in mobile data traffic especially in LTE and 4G networks. The global smart phone sales will touch 2.5 billion over the period starting from 2010 to 2015 as they assessed. As depicted in Figure 2, the traffic will also increase from 44,487 terabytes per month in 2009 to approximately 1.8 Exabyte per month in 2017. The graph also provides the trends of different type of traffic including audio, video, peer-to-peer applications (such as instant messaging) and pure data sharing (such as emails, browsing etc.). Recent surveys by Nielsen and eMarketer [24] also supported similar predictions for United States. According to the report, 31% of mobile phone users possessed smart phone in 2010 and it will reach to 43% by the end of 2015.

2.2 Literature Review

In 2007, Park *et al.* published on a number of threats in 4G networks [81]. These threats include but not limited to malformed message attacks, denial of service, buffer overflows, etc. In the case of instant messaging, network operators are highly concerned about the Theft of Service (ToS) by automated program generated messages. A significant part of these messages is unwanted and exploits IM vulnerabilities to reach the end-users [33]. Unfortunately, these operators cannot detect or stop the SPIM senders, which are mostly malicious software (called as bot) residing remotely in the Internet. Therefore, an appropriate mechanism to detect spam/SPIM messages is essential to the modern telecommunication networks.

The SPIM classification is a challenging task for the instant messaging application providers in telecommunication systems due to its peer-to-peer architecture and use of heterogeneous communication protocols. Aiming to solve the problem of SPIM classification, researchers across the world stress on some of the key metrics in verifying security of the telecommunication networks [81, 82]. Interoperability, usability, Quality of Service (QoS) guarantee and cost-effectiveness of secured solutions are marked as important factors. Secondly, researchers also suggest for continuous monitoring of various features and anomalous events to detect new attacks [97]. Recent articles have proposed various multi-layered architectures to block automated spammers to send messages to the networked users [82]. Symantec Corporation has implemented messaging gateway to protect network against the SPIM threats [20]. This detection mechanism blocks SPIM message from infected external users based on heuristic based rules. It uses SPIM signature and monitors uncommon messaging activities [20]. The application is, however, successful in limited scenarios.

In the literature, many research initiatives can be found in tackling the problem of SPIM filtering. Liu *et al.* proposed a new architecture for real-time SPIM defending and filtering in a personalized setting of various IM gateways [68]. They tested a number filtering methods that include Black/White list, collaborative feedback based filtering, content-based technique, challenge-response based filtering, IM sending rate, content based SPIM defending techniques, fingerprint vector based filtering, text comparison filtering, Bayesian filtering, etc. [68]. After a number of experiments they claimed that the blacklisting spammers based on user feedback produces most efficient blocking with least error rate. The work also mentioned a challenge and response-based filtering technique in the same article. Another unique challenge generation technique has been patented by Satish, in 2005, that can be implemented in network and/or user devices [94]. In his technique, the detection manager sends a special message to the sender once it suspects an incoming instant message as a SPIM. The detection manager then analyzes the response of that message to identify SPIM. Mannan and Oorschot focused on using an encryption module on client chat application for secured communication [70]. Inter-organizational collaborative feedback technique also provides benefits to large-scale IM applications to handle SPIM [68]. Damiani *et al.* addressed privacy-preservation and privacy guarantee for secure peer-to-peer IM [39].

Information security research communities have proposed SPIM removal by handling the botnet and worms since these applications mostly use IM to deliver their campaign messages. Change point detection [51], Virus Throttling [71] are some of the known techniques. In 2010, Maroof published a technique to distinguish bots from human users by mining user characteristics [73]. The features in his collected dataset use word-length, message length, URLs, Capital and small letters, etc. However, this work implements SPIM detection at the user end. In [71], authors classified the chat BOTs into four different types: periodic, random, responder and replay BOTs. However, detection or removal of remote bots from Internet is challenging from the perspectives of ISPs in many ways. Therefore, in this thesis, we decided to focus on packet filtering instead of sender identification.

Cormack *et al.* [36] worked on feature engineering for short messages where he found interesting message characteristics by mining alphanumeric characters, words, and sequence of words and letters in word. Sequence of letters in words over a specific size may generate several attributes than the word itself. It has been found helpful in dealing with short messages as well as in finding sequences of letters closely carrying the meaning of a word [37]. The work elaborated higher classification accuracy using regression based and SVM filters over large data of short messages. Yang and Pedersen [110] described some standard tests to verify importance of the features in the context of text categorization. This work is useful in classification as it provides mathematical means to understand the importance of various document features.

Given the limited memory and low computation power in currently available mobile user equipments (UE), many of the aforementioned user-centric techniques cannot be well implemented in the telecommunication user equipments to adequately deal with SPIM from near real-time IM communication. On the other hand, network centric techniques are feasible for SPIM detection but require to handle huge amount of packet related characteristics. In this trade-off, machine learning techniques are most useful to classify SPIM packets from incoming packet streams. The choice mainly stems from the requirements of analyzing large amount of observational traffic data in a very short time. Machine learning employs fast rule-based prediction where the rules are typically generated off-line by elaborated training on previous data. We assume that the incoming information is too complex for analytical solution generation in near real time. In such case, machine-learning may be beneficial appropriate detection technique by matching patterns from the data features.

2.3 LTE 4G Network

In 2007, the 3GPP proposed LTE as the standard specification of future all-IP mobile broadband networks that will converge with WWAN technologies. LTE provides faster web-browsing, rich Internet applications, real-time message communication along with advanced telecommunications of audio and video. In a short period of time, LTE has been accepted worldwide [7]. Often times, LTE is considered as pre-4G telecommunication network, however, according to the specifications, LTE is not a true 4G standard. It supports lower data speed (300 Mbps downlink) in compare to the 4G requirements (1 Gbps downlink). LTE specification is therefore expected to be enhanced to bridge the gap [101].

In LTE, communication technologies have been primarily optimized for efficient throughput and higher-quality of services. They are designed to cope with all previous mobile communication standards in heterogeneous network environments [101]. The use of advanced antenna technology along with 2x2 and 4x4 channel configuration of Multiple Input and

Multiple Output (MIMO) offers reduced packet latency and therefore provides faster data communication. LTE is also known for its superior quality enhanced air interface, high spectral efficiency, flexible radio planning and high data rate [104].

2.3.1 Evolution of LTE 4G Network

Aiming to support greater opportunities, quality of service and efficient real-time communications with various user equipments, Universal Mobile Telecommunication Systems (UMTS) have been upgraded and migrated from circuit switched network to packet data core network. The transition for the technological enhancement has taken wireless network through different generations. The wireless communications have been developed under two different types of technologies: Wireless Local Area Network (WLAN), and Wireless Wide Area Network (WWAN). WLAN has been serving as a de-facto wireless distribution method to wirelessly connect supported devices to the wider wired networks and Internet. The data communication rules are governed by Internet Protocol (IP). IEEE 802.11, popularly named as Wi-Fi, has turned out to be universally accepted wireless communication standard. WWAN is another set of wireless communication techniques to serve in large geographical area and therefore supports higher mobility to end users. Mobile telecommunication in cellular network uses this technology as standardized by the International Telecommunications Union (ITU) [65]. The aim of WWAN was to move analog phone system to IP-Based packet switching communication and to offer higher mobility and security. The goal of the 4G advanced networking is set to converge WLAN and WWAN under the same technology and to proceed ahead with unified wireless communication standard. However, there exists another parallel improvement in technologies for Wireless Personal Area Network (WPAN) [107]. WPAN is a typical ad-hoc technology designed to connect devices within close proximity. Examples of WPAN technologies include Bluetooth network devices, Infrared Data Association, etc. The WPAN maintains IEEE 802.15 standard [46].

Figure 3 presents a brief elaboration of WLAN evolution on the left side. WLAN communication primarily offers multiple access points within small area coverage. In its infrastructure a number of devices, having unique IP addresses, can use these access points

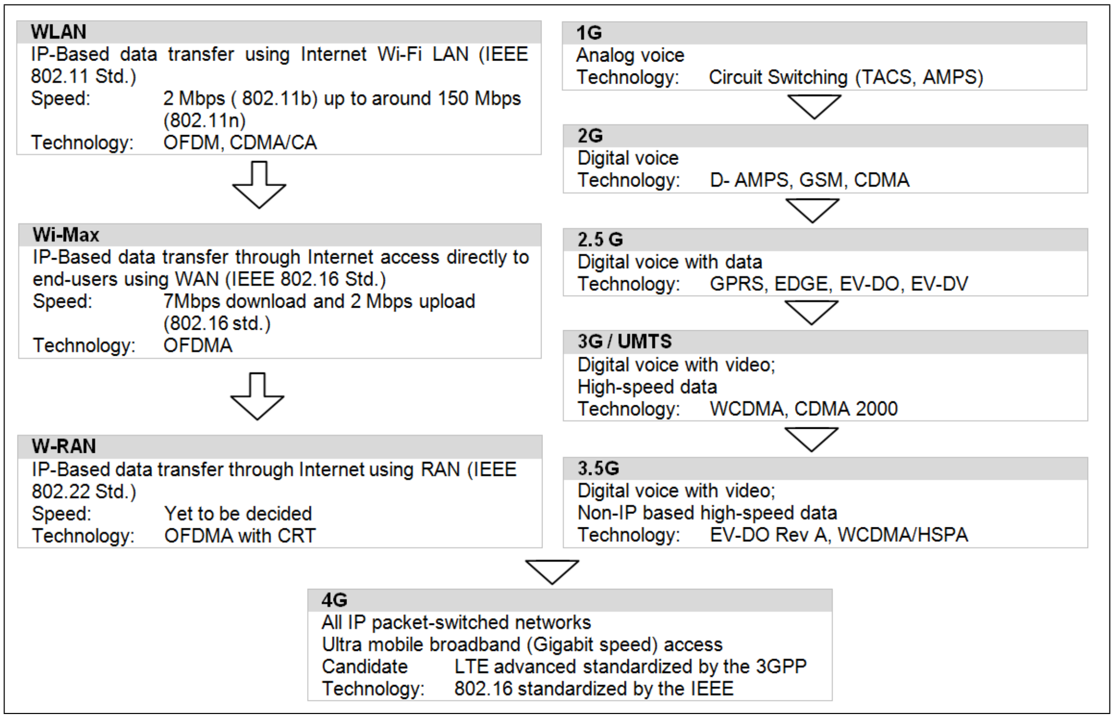

WLAN
IP-Based data transfer using Internet Wi-Fi LAN (IEEE 802.11 Std.)
Speed: 2 Mbps (802.11b) up to around 150 Mbps (802.11n)
Technology: OFDM, CDMA/CA
Wi-Max
IP-Based data transfer through Internet access directly to end-users using WAN (IEEE 802.16 Std.)
Speed: 7Mbps download and 2 Mbps upload (802.16 std.)
Technology: OFDMA
W-RAN
IP-Based data transfer through Internet using RAN (IEEE 802.22 Std.)
Speed: Yet to be decided
Technology: OFDMA with CRT
1G
Analog voice
Technology: Circuit Switching (TACS, AMPS)
2G
Digital voice
Technology: D- AMPS, GSM, CDMA
2.5 G
Digital voice with data
Technology: GPRS, EDGE, EV-DO, EV-DV
3G / UMTS
Digital voice with video;
High-speed data
Technology: WCDMA, CDMA 2000
3.5G
Digital voice with video;
Non-IP based high-speed data
Technology: EV-DO Rev A, WCDMA/HSPA
4G
All IP packet-switched networks
Ultra mobile broadband (Gigabit speed) access
Candidate Technology: LTE advanced standardized by the 3GPP
802.16 standardized by the IEEE

Figure 3: Evolution of Wireless Networks

in a simultaneous coordinated manner in order to connect to a wired Ethernet connection using network bridges. WLAN also offers ad-hoc peer-to-peer connection among these devices. IEEE standard 802.11 protocols are used in WLAN for the physical and MAC layers. Wi-Fi is the first set of standard technologies for WLAN. Depending on various operating frequencies and throughput, IEEE 802.11 family is named as 802.11 a, b, g, n, etc. WiMAX is a WLAN-successor wireless communications standard established in 2001 using IEEE 802.16 family of protocols. It has been designed to enhance the coverage of LAN by extending it up to 50 kilometers [17]. WiMAX offers mobile broadband connectivity for IP-enabled devices, VoIP, and similar services across cities. The latest development in the wireless communication standard is the Wireless Regional Area Network (WRAN) based on IEEE 802.22 family of standard protocols. WRAN has been designed with primary focus on TV frequency and aims at providing broadband data communication in remote areas. In physical layer, WRAN uses Orthogonal Frequency-Division Multiple Access (OFDMA) whereas in MAC layer it exploits Cognitive Radio Technology (CRT). Furthermore, LTE/4G network has more similarities with WiMAX than with WRAN [88]. On the right side of Figure 3, we elaborate on Mobile telecommunication technology evolution. This networking is commonly mentioned as WWAN. There are two main types of WWAN technology: Global System for Mobile Communications (GSM) and Code Division Multiple Access (CDMA) [107]. GSM establishes multiple channels in frequency bands and offers them to multiple users. CDMA digitalizes the calls one over other and uses sequence code to unpack them at the back-end [4]. However, WWAN is a non-IP based communication that has been evolved over multiple generations commonly named as 1G, 2G, 3G, etc.

The CDMA technology also evolved from first generation, followed by CDMAOne and CDMA2000 technologies in second and third generations respectively. Evolution-Data Optimized (EV-DO) was set as CDMA broadband wireless network standard in Third-Generation Partnership Project 2 (3GPP2) that increased the uplink and downlink data rate. The latest Revision of the EV-DO, namely EV-DO Rev A, is considered as the current technology with uplink and downlink data rate up to 1.8Mbps and 3.1 Mbps, respectively. GSM, on the other hand, is a 2G technology that was also evolved over time. Initially, Wide band-CDMA technologies were accepted by GSM in the third generation. Afterward, High-Speed Packet Access (HSPA) has been adopted to improve the performance of the UMTS. HSPA is composed of High-Speed Downlink Packet Access (HSDPA) and High-Speed Uplink Packet Access (HSUPA) telecommunications protocols that offer downlink and uplink speed up to 14.4 Mbps and 5.76 Mbps, respectively. Currently available technology is HSPA+ that provides enhanced speed (42Mbps and 22Mbps) using MIMO technology and higher-order modulation in antenna [107].

2.3.2 LTE Network Components

To support greater spectrum efficiency and reduced latency, LTE uses Orthogonal FDMA with downlink data rate of 300Mbps and Single Carrier FDMA (SC-FDMA) to offer 75 Mbps of uplink data rate using MIMO system. All-IP supported 4G/LTE network allows communication with the other existing 3GPP, as well as, non-3GPP telecommunication networks. Precisely, LTE network architecture pertains to the interconnection of different

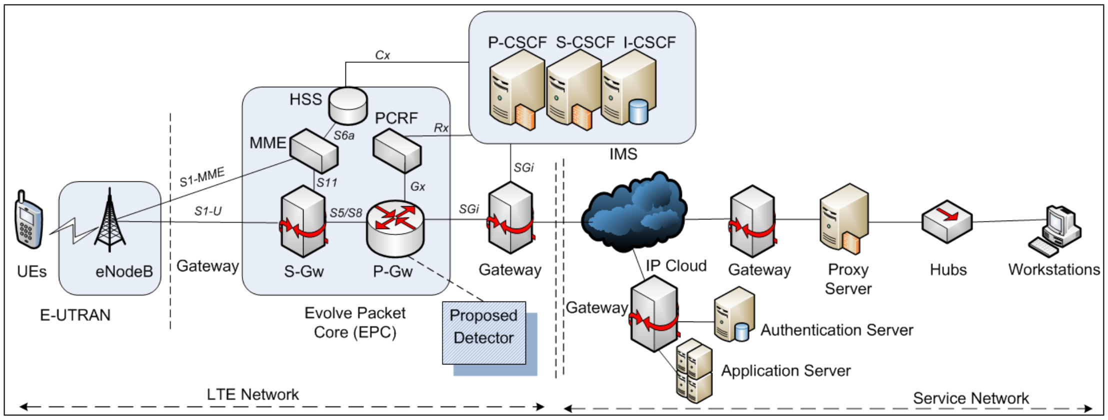


Figure 4: LTE Architecture for Instant Messaging [101, 104]

architectural components for enterprise-scale telecommunication networks. Term “LTE” is used to represent a broader technological aspect in the world marketplace, while in accordance to the developer’s term, it only means the radio access part of the whole network.

Figure 4 depicts the high-level architecture including LTE-IMS framework and the Internet service network. LTE is also termed as Evolve Packet System (EPS). EPS contains LTE access network and System Architecture Evolution (SAE). A variety of user equipments (UE), ranging from simple handheld cell-phone to laptop computers associated with mobile broadband adapter connect to the radio access network by LTE-Uu air interface. A UE mainly acts as the end user which initiates, connects, controls and manages the call and the session. The radio access network of LTE, also known as the Evolve-Universal Terrestrial Radio Access Network (E-UTRAN), mainly reduces latency for all air interface operations. E-UTRAN is composed of advanced antennas and a set of evolved-NodeB (eNodeB) to connect wireless devices. The eNodeBs are interconnected by X2 interfaces to the EPC nodes by different S1 interfaces. The functionalities of these nodes are decentralized/flat. They are responsible for radio resource management such as scheduling and allocating uplink and downlink resources, compression of IP header, encryption of user data. They also play an important role during handover by coordinating with other eNodeBs [96]. E-UTRAN is connected to the Evolved Packet Core (EPC) system at one side through IP-based connectivity whereas E-UTRAN relays data in radio connection to the other side. Thus, it can guarantee higher amount of packet flow to the UEs. The core part of SAE, responsible for the rapid and efficient adaptation of mobile broadband, is called Evolve Packet Core (EPC). EPC supports multi-service convergence and multiple access technology. It supports both GSM and CDMA networks and represents the common core network for fixed mobile network convergence. There exists four types of logical nodes in EPC, as depicted in Figure 4, that are inter-connected by means of interfaces (S1-U/S11 etc). These nodes are identified as follows:

- *Mobility Management Entity (MME)*: This node acts as a control node for the bearer (IP packet flow with defined QoS from gateways to UE) establishment, maintenance and release while ensuring at the same time session connection and security. The underlying functional layer in UMTS protocol stack from this node to the core network is named as Non-Access Stratum (NAS).

- *Home Subscriber Server (HSS)*: It is a master database that stores SAE subscription data of user addresses for connecting PDNs (as Access Point Name(APN)), holds dynamic information such as the currently attached MME, physical location of users and generates vectors for authentication and security keys. HSS provides registration data, presence, roaming, buddy-list among multiple access networks to the IMS [43].

- *Policy and Charging Control Function (PCRF)*: This node makes decision on charging and control QoS based on policy rules from the technical details of Service Data Flow (SDF) that will apply to services for the users. Then, it passes these rules to the Policy Control Enforcement Function (PCEF) located in the Packet Data Network Gateway for the enforcement purposes.

- *Serving Gateway (S-GW)*: This type of nodes plays the role of the mobility anchor during inter-eNodeB handover and other 3GPP technology handover. It helps in routing and forwarding all data packets. Among its other responsibilities, it remembers information about the bearer during UE idle mode. It also temporarily buffers downlink data while the Mobility Management Entity (MME) initiates paging of the UE for bearer re-establishment, collects charging data in the visited networks, etc. [101].

- *Packet Data Network Gateway (PDN-Gateway or P-GW)*: This node offers UE connectivity to the external packet data networks and also anchors the mobility between 3GPP and other non-3GPP technologies. However, in practice, multiple P-GWs may exist for Internet or SIP-based instant messaging. One UE is also allowed to maintain simultaneous connection with more than one P-GW [96]. It performs policy enforcement, packet filtering for each user, packet screening, charging support and lawful interception [77]. P-GW allocates an IP address for the UE and enforces the flow based charges from PCRF. As the S-GW in LTE network acts as the *handover anchor* for inter-working with GSM and UMTS networks, P-GW takes the responsibility for seamless handover to non-3GPP networks such as CDMA2000 or WiMAX by the interfaces based on Proxy Mobile Internet Protocol (PMIP).

There exist other logical nodes in EPC network that are kept to support non-3GPP trusted / untrusted nodes such as: Evolve Packet Data Gateway (EPDG), 3GPP-Authentication, Authorization, Accounting Server (AAA) etc. However, their detailed functionalities are beyond the scope of interest in this thesis. We propose to deploy our SPIM detector in P-GW to analyze incoming packets on downlink. The detector functionalities are discussed in Section 3.

2.3.3 IP Multimedia Subsystem Network Architecture

To support real-time IP-based multimedia services such as: messages, video, audio and text, from heterogeneous platforms, LTE framework includes IP Multimedia Subsystem (IMS) network. IMS is the convergence platform of network, device and services for the establishment of peer-to-peer or peer-to-content connection between IP enabled devices. IMS guarantees better QoS and account pricing (irrespective of the user location) including additional services in relation to multi-media [31]. It is responsible for establishing, managing and terminating sessions, and presence also. It supports standard interfaces for multiple service developers for both wireless and wired network [31].

On the top of data access network (fixed-line, packet-switched, radio, etc.), IMS has three distinct functional layers according to 3GPP Release 5, namely Transport, Core-Control and Service/Application Layers. These layers invoke different functions to perform tasks for managing applications, mobility, and compliance constraints. The transport layer is a combination of protocols including Real-time Transport Protocol (RTP) and Session Initiation Protocol (SIP). It includes media gateways with Media Resource Function (MRF), signaling gateways and media servers, which altogether facilitates registration of devices, in-band signaling, and shared applications (voice mail, push-to-talk, interactive response systems etc.). MRF is functionally separated by Media Resource Function Controller (MRFC) and Media Resource Function Processor (MRFP) [43].

Figure 4 also denotes logical components of IMS control layer. This layer connects to a central user database server "HSS" through the session control severs. Session control is performed using SIP. HSS routes the SIP messages, authenticate and distribute of IMS traffic between transport and service layers [43]. Logically, IMS control layer consists of a Proxy Call Session Control Function (P-CSCF), a Serving CSCF (S-CSCF) and an Interrogating-CSCF (I-CSCF) with visible IP addresses in Domain Name Service (DNS). The IMS control layer components are detailed as follows:

- *Proxy Call Session Control Function (P-CSCF)*: P-CSCF is the first point of contact for a UE within IMS. P-CSCF secures the media flows by accepting the SIP register/invite methods and allocating resources for the purpose. Afterward, it determines the next passing destination either to route the information to the right S-CSCF or to the home network [109].

- *Serving Call Session Control Function (S-CSCF)*: S-CSCF provides session states control services for an ongoing session. It performs SIP registration by accepting user requests and relays the SIP request and response between the sender and the receiver. It also acts as a proxy server between users and other CSCFs or SIP servers [98].

- *Interrogating Call Session Control Function (I-CSCF)*: I-CSCF is another contact point for all connections that interrogates the location and the suitability of the S-CSCF to the HSS within the network of operator. Consequently, it assigns S-CSCF address to the registering users if match is found. There may be multiple I-CSCFs in a network of an operator. I-CSCF is also used to hide the internal structure of the operator network from the external network [98].

Also behind the IP-Cloud, as shown in Figure 4, there exists an application server that runs the IM application. The IM application can be shared by several users from different LTE frameworks or Internet connected workstations (using legitimate proxy servers). The authentication server verifies and validates the authenticated users of the IM application. UEs mainly require the IM application server for connection establishment (connect, login, logout functions), presence management, and session binding purposes. Up next, we are expanding our research interest in the details of instant messaging services and protocol.

2.4 Instant Messaging

Messaging is a general form of asynchronous near real-time communication between two or more users communicating using different peripherals in their respective networks. As mentioned in Section 1.2, different types of messaging techniques exist in various communication applications ranging from e-mail to Short Message Services (SMS). Exchanged messages are mostly text-based with possibilities to carry audio and video data. Instant messengers are one of the most popular type of interactive application that communicate packet data directly between known (and unknown) people. The procedure is grossly known as chatting. However, chatting covers a more general concept than instant messaging (IM). Chat often refers to communication between known and unknown people in a multi-party platform while instant messaging is mainly performed between known people from their contact lists. Chatting is commonly a Client-Server communication (example: yahoo messenger, AOL messenger) while IM is mostly peer-to-peer communication (example: Skype). However, some messengers do not follow peer-to peer instant messaging for their business interest. A handful number of client-server based instant messaging applications exist also, such as: Jabber. With the technological developments, these applications are integrated in commercial and educational websites, email applications, mobile phone devices, etc.

Mobile Instant Messaging (MIM) enabled devices are mostly equipped with proprietary or carrier independent messenger. On the other hand, Wireless Instant Messaging (WIM) users require internet to connect with other WIM users or fixed IM users (AOL, MSN, Yahoo! etc). The spam detection in instant messaging requires a thorough understanding of messaging procedure itself. In the following, we describe services and protocols associated with the peer-to-peer instant messaging, especially in our context of LTE framework.

2.4.1 Services

Instant messengers offer three main services [69] namely *presence service*, *messaging service* and *content management*. A brief description of these services is presented in the following.

- *Presence Service*: The presence service intends to accept, store, and distribute presence information among the IM clients [42]. The users of the presence services are namely *presentities* and *watchers*. Presentities provide presence information to the presence service at the server, while watchers request presence information about presentities from the presence service. A watcher can be *subscriber*, *fetcher*, or *poller*. A *subscriber* sends a request for presence information of others such that the service remembers the subscriber and its track of subscription. It also sends notification of change in presence information for other subscribers of interest. A *fetcher* uses request-response based communication to retrieve presentity information for one time. A *poller* fetches presentity information multiple times possibly after periodic interval [54]. The presence service is associated with two logical agent entities, namely: Presence User Agent (PUA) and Presence Agent (PA). While the PUA is involved in manipulation of presence information for presentity, PA works inside proxy to receive and send subscribe and notify messages. The latter is implemented in SIP and has the ability to retrieve and to process presence information as well in the server.

- *Messaging Service*: Messaging service is responsible for message delivery. It also helps in access control, group usage, management, access control, etc. In IM applications, this service is designed in many ways. In Section 2.4.2, we later discuss messaging for two standard protocol implementations, namely: eXtensible Messaging And Presence Protocol (XMPP) (IETF RFC-3921) and Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE) (IETF RFC-3428 [32]).

- *Content Management*: IM content is consistently managed by effective policies as described in multiple RFCs (RFC 4745 [95], RFC 5025 [92], RFC 4825 [93]). Content management offers to minimize the risk of malware infection, virus, etc. from the attachment files. It also hinders private and confidential information to be used by software implementing institution. Finally, content management enforces in the messaging application compliance to governmental laws and regulations.

In the next section, we present two application layer protocols for messaging session initiation, presence notification, and message delivery.

2.4.2 Protocols

Architecturally, IM applications are of two main types: Client-Server (example: Google Talk) and Peer-to-Peer (example: Skype). Enterprise messenger applications use proprietary and standard protocols for session initiation and data communication. In this thesis, we focus on the two industrial open standard protocols: XMPP and SIMPLE.

XMPP is a robust, presence-aware, inter operable client-server protocol with XML-based data-transport. It enforces message streaming through servers to meet potential business interest. An example of such a service is *Jabber* [6]. The adherence to XML structure limits the inclusion of voice and video through the same protocol and requires separate technology such as: *Jingle* [18]. Corporations, like Google, HP, Sony, Hitachi have implemented the concept. In XMPP, message delivery is implemented using two client types: *senders* and *instant boxes* [54] according to the IETF model released in the RFC 2778. The sender provides message data to a messaging service including information of an instant box. The underlying service attempts to deliver the message to a target instant box.

SIMPLE, on the other hand, is a presence-aware, peer-to-peer text-based protocol on top of SIP used by Microsoft, IBM, etc. Although there is no popular Internet scale application, SIMPLE is a promising protocol for high-speed LTE framework because of its ability in voice, video, and content management in concise text-based format. In SIMPLE, there exists two different modes namely: *Pager* and *Session*. In pager mode, instant messages are sent over SIP using the MESSAGE method extension (RFC 3428 [32]). When using session mode messaging, a session is established using SIP, after which the Message Session Relay Protocol (MSRP) is used for exchanging instant messages within the session directly between the users (RFC 4976 [61]). Table 1 shows a comparison of the aforementioned two protocols with respect to their features and capabilities.

2.5 Machine Learning

In recent years, simultaneous sharing of enormous information via web, mobile devices, satellite etc. necessitates automated processing of large amount of data while managing high quality-of-service, data control, security and privacy. Consequently, the autonomous learning and consolidate decision making techniques based on available sample data gained enormous research interest. We intend to use classification based on machine learning techniques to detect SPIM from instant messaging in LTE framework. A number of machine learning techniques are available in literature. Most of them are actually originated from the prior used off-the-shelf detection mechanisms in the industrial arena. The machine learning algorithms are capable of recognizing the regularities in the data and predict an approximate result. In a nutshell, machine learning techniques are used to study a large amount of previously stored data and generate pattern-based output to train detection module. The trained detector is later applied during execution in order to identify pattern(s) during the testing phase from the incoming data. Machine learning techniques has been studied based on different aspects of interest. Dredze [45] pointed out two main types of machine learning, namely output-based and function-based, as depicted in Figure 5. The performance of the applied technique depends on the characteristics of the data and the overall scope.

The output-based machine learning techniques are classified with respect to the types of the output. The most popular technique in this regard is *Classification*. *Classification*

Table 1: Comparative Study of XMPP and SIMPLE

Features	**XMPP**	**SIMPLE**
Architecture and data transport	Client-Server protocol with XML-based	Peer-to-Peer protocol with text-based
Data transfer	Streaming based	Two modes: Paging and Session
Presence	Presence-Aware	Presence-Aware
Audio/Video Support	Difficult to extend	Easy to extend
Audit/Logging	Easy to audit/log ongoing communication	Difficult to log communication
Deployment	Large-scale proven implementation available	Large-scale applications under development
Code	Open-source code available	Unknown
RFC	IETF RFC 3921	IETF RFC 3428
Users	Google, Hewlett-Packard, Intel, Wireless Computing Group, Sony and Hitachi	Microsoft, IBM, Yahoo, Sun Microsystems, Oracle
Messaging Example	<message from=‘juliet@example.com/foo’ to=‘romeo@example.net’> <body>Hello, How are you? </body> </message>	<sip:...@example.net> SIP/2.0 Via: SIP/2.0/TCP x2s.example.com;branch=z9hG4bK776sgdkse Max-Forwards: 70 From: Peter Saint-Andre <sip:...@jabber.org>To:<sip:...@abc.net> Call-ID: <Hr0z...@example.com> CSeq: 1 MESSAGE Content-Type: text/plain Content-Length: 35 Hello, How are you?
Presence Example	<presence from=‘juliet@example.com/foo’ to=’romeo@montague.net/bar’/>	NOTIFY sip:192.0.2.2 SIP/2.0 Via: SIP/2.0/TCP x2s.example.com; branch=z9hG4bKna998sk From:Peter Saint-Andre <stpe...@jabber.org> To:<sip:...@example.net>;tag=yt66 Call-ID:<j4s0...@example.com> Event: presence Subscription-State:active;expires=599 Max-Forwards: 70 CSeq: 157 NOTIFY Contact:<sip:sipgate.example.com;transport=tcp> Content-Type: application/pidf+xml Content-Length: 192 <?xml version=‘1.0’ encoding=‘UTF-8’?><presence xmlns=‘urn:ietf:params:xml:ns:pidf’ entity=‘pres:<jul...@example.com>′><tuple id=‘x’> <status><basic>open</basic> </status></tuple></presence>

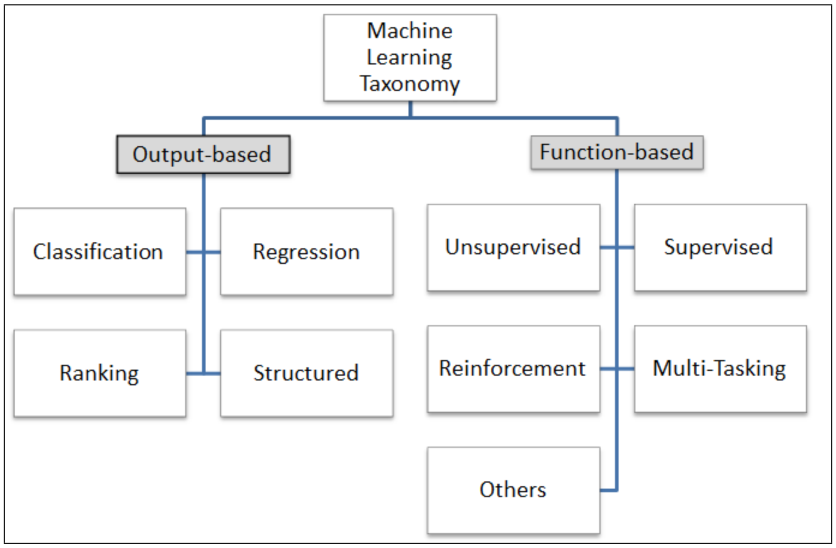


Figure 5: Taxonomy of Machine Learning Techniques

predicts the association of input data to an output group. The prediction can be binary, multi-class or even hierarchical. Loss of accuracy is a major concern in these techniques. Another output learning technique is *Ranking* [21]. In *Ranking*, classified groups are partially ordered among themselves. The process of ordering the detailed measures into a sequence of ordinal numbers helps in evaluating large volume information according to certain criteria. *Regression*, (see Figure 5), is the third technique to formulate training output into an equation or mathematical model that helps in predicting status of live input information. The *Structured learning* is associated to build sequence, data structure, segmentation from the output and classifies the live input data accordingly.

The taxonomy of machine learning can also be seen from a functional perspective. The two main functionality-based categories are *unsupervised* and *supervised*. Unsupervised machine learning does not involve training phase to build the model. It creates groups for new information instance by finding hidden structure based on a given set of measurements and observations from unlabeled data in order to differentiate the data. Association, clustering, density estimation are few general examples of unsupervised machine learning. The supervised learning technique works in two phases, namely training phase and testing or classification phase. In the training phase, a model is trained from the known examples with pre-labeled classes. In the testing phase, also known as the classification phase, this trained model predicts categorical class labels for new data instances. The prediction is usually nominal or discrete. In the case of continuous output prediction from trained model, the procedure is called *Reinforcement*. Different algorithms exist in the literature to train the inferred model function with labeled input data such as support vector machine, boosting, k-nearest neighbor algorithm etc. Hybrid semi-supervised learning mechanisms are also available in the literature. Furthermore, training of machine learning model can be achieved through functionally distributed agents with/without *multi-tasking* abilities.

In the next chapter, we evaluate some of the promising machine learning algorithms with established performance measurement metrics to justify our choices of the most profound algorithms for spam detection in telecommunication networks.

2.6 Summary

Instant messaging applications are highly attractive to the spammers in order to communicate spam directly to the targeted users. The underlying technologies are much faster in delivering message in compare to the emails and have intrinsic limitations in identifying and separating spam messages. As a whole, they possess higher threats that range from unsolicited data delivery to complex cyber-attacks. However, the advantage of the SPIM detection in mobile communication network is that the IM packets always pass through the mobile network establishment before reaching the user equipments. The contribution of this chapter can be seen in a thorough review of the research background. In the aforementioned study, we illustrate the need for SPIM detection in this high-speed mobile telecommunication network. We elaborate our target telecommunication network, illustrate the scope of the SPIM detection module and discuss various machine learning techniques to identify the SPIM. We also cover major previous research and development efforts toward this direction.

Chapter 3

Proposed Approach

In this chapter, we propose a module to use machine learning techniques for spam classification from IM packets. First, we identify the location of this SPIM detection module in LTE network architecture. Then, we present a high-level architecture of the detection module followed by a description of the proposed technique. Afterward, we select a number of machine learning algorithms from different categories. Their suitability in our context are verified from theoretical perspectives and experimental verifications. Finally, we discuss a set of IM packet features that can be used to classify the packets containing spam.

3.1 Location of SPIM Detector

As depicted in Figure 4, we propose the location for the SPIM detection module in PDN-Gateway. The proposed detector in this gateway will classify incoming packets on the downlink to find SPIM. The underlying reasons for choosing PDN-Gateway are as follows:

- PDN-Gateway has existing techniques for policy enforcement, lawful interception, packet filtering and packet screening.
- Incoming IM packets in LTE telecommunication networks mostly pass through PDN-Gateway and therefore can be intercepted, monitored and analyzed as per requirements.

- PDN-Gateway may share information and knowledge with other third-party non 3-GPP networks.

- It also supports a number of user charging mechanisms for additional services.

- The number of PDN-Gateway is far less than other types of LTE components in the packet data flow. Therefore we expect to receive considerable amount of coherent training data to prepare the classifier.

However, it should be understood that the LTE data load is very high on the PDN-Gateway. Therefore, it requires very fast technique to analyze packet from this huge volume of traffic. Machine learning is one of the most suitable techniques in this regard.

3.2 Proposed Technique

Many off-the-shelf packet sniffers and filters, such as: Snort[91], use combination of rules during Deep Packet Inspection (DPI). They apply regular expression, efficient string search algorithms (ex. Boyer-Moore), etc. We rely on stateless deep packet inspection techniques to get comparatively faster inspection than its stateful counterpart especially in LTE network [66]. However, we expect intelligent spammers to be situation-aware and adaptive. They often bypass these rules by changing string and symbolic representations. In this respect, adaptive machine learning algorithms offers significant advantages in SPIM detection.

Figure 6 depicts our proposed SPIM detection module. In offline, we prepare the ‘user model’; a trained classifier for packet classification. We carefully differentiate between two types of IM packets: small signaling packets (<100 bytes) and large data packets (An absolute size limitation of TCP packet is 64K but IM uses much smaller packets usually less than 1500 bytes). We first pre-process the SPIM and the legitimate datasets from previously captured and labeled databases available offline. This preprocessing includes selection of attributes/features and representing each packet of the message as a vector of attributes. Selected learning algorithm can then be used to train the classifier for online IM packet classification. Our classification objective involves binary determination of SPIM

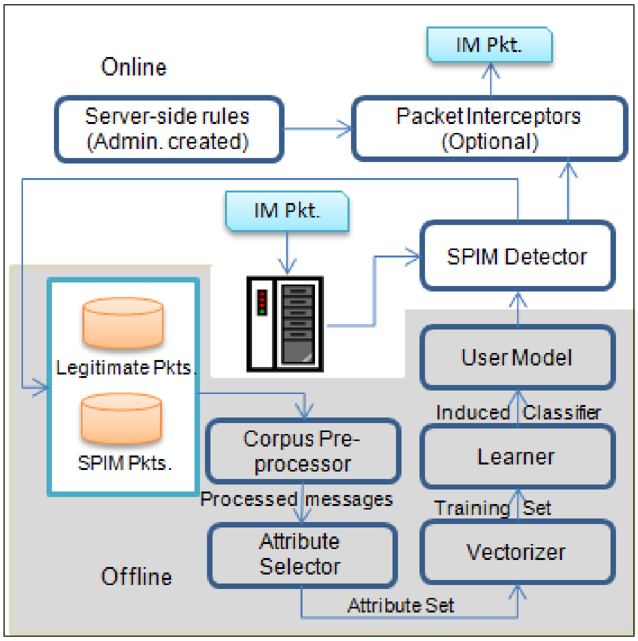


Figure 6: Component and Data Flow in SPIM Detection Unit

and provides network administrators with statistical information about packet senders. We assume that the vectorized packet information is instantly prepared by an external DPI plugin. Although strong encryption may result in failure to understand the packet payload but packet flow, structure and message header related information can be mined. Zhang *et al.* [111] discussed spam detection accuracy from message header. Using e-mail corpus, they found that the classifiers using features from the message header alone may achieve comparable or better performance than classifiers using features from message body. In the next section, we investigate and compare some of the suitable machine learning techniques for SPIM classification from theoretical perspectives and implementation standpoints.

3.3 Selection of Machine Learning Algorithms

The following discussion includes a selected number of machine learning algorithms from three major perspectives of deployment. First, we focus on the spam detection accuracy. Second, we analyze the offline training time of these supervised machine learning algorithms.

Finally, we compare their spam classification time to judge their appropriateness.

According to the existing research performed in this area, the supervised learning methods have shown a higher percentage of success compared to other techniques in the context of spam filtering, particularly in email-spam detection and text categorization [44, 105]. In the supervised learning technique, for example, a set of messages (stored in inbox) which is divided into spams and legitimate, trains a classifier offline from one or more data sources (also known as corpora). The training procedure starts learning priorly identified set of features from the flow, packet header and packet content that are priorly pruned from the data. It should be noted that the significance of feature selection varies greatly from a classifier to another. In this research and development, we restrict our search among four commonly used spam filtering algorithms, namely Naive Bayes (NB), J48 decision trees, Support Vector Machine (SVM) and AdaBoost (Ada). We also discuss two more algorithms, namely, Maximum Entropy and K-Nearest Neighbor (K-NN). However, we do not select them to implement in the simulation environment. The selection of these classification algorithms are made based on their distinguishably different learning techniques and their suitability for the targeted environment. Specifically, the focus is on the following criteria.

- Compliant with real-time tracking with low processing overhead;
- Deployable in the LTE/IMS architecture and cellular infrastructure;
- Highly accurate in prediction with small size packet content;
- Cost effective in terms of network hardware and software requirements.
- Applicable to different protocols;

The following sections describe each of these selected supervised machine learning mechanisms and their various aspects in spam classification.

3.3.1 Naive Bayes

Naive Bayes (NB) is one of the most popular algorithms that is implemented in various commercial and open-source anti-spam email filters [72]. According to Bayes' theorem, the conditional and marginal probabilities of two events A and B are related as follows:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

Eqn. 1 formulates the conditional probability of A, given the probability of B. The rule can also be used to compute the probability of the class $c \in \{spam, legitimate\}$ from a document vector $\vec{d_j}$ where $\vec{d_j}$ can be seen as j^{th} document with a collection of independent features such that $\vec{d_j} = \{w_{j1}, w_{j2}, \ldots, w_{j|\vec{d_j}|}\}$. Let n be the size of $\vec{d_j}$, such that $|\vec{d_j}| = n$. In fact, each document feature is conditionally independent with other features and therefore can be represented as *Naive* condition in the conditional probability:

$$P(\vec{d_j}|c) = \Pi_{i=1}^{n} P(w_{ij}|c) \tag{2}$$

Hence, the Naive Bayes probability of document class can be expressed as:

$$P(c|\vec{d_j}) = \frac{P(c).\Pi_{i=1}^{n} P(w_{ij}|c)}{\sum_{k \in \{legitimate, spam\}} P(c_k).\Pi_{i=1}^{n} P(w_{ij}|c_k)} \tag{3}$$

where $\sum_{k \in \{legitimate, spam\}} P(c_k).\Pi_{i=1}^{n} P(w_{ij}|c_k)$ is a scaling factor, often termed as evidence. In the training phase, the procedure helps to create an inferred function with a determined output of the threshold t to recognize a spam from other incoming messages during the classification stage. Several research articles have been published to enhance performance of NB [76] using variants in conjunction to Bernoulli theorem, multinomial term frequency, Gaussian distribution, etc. In this thesis, we consider only multinomial NB with Boolean attributes (representing whether a feature is present or absent in a message) as this technique is known for low computational complexity and comparatively high performance [76].

3.3.2 K-Nearest-Neighbor

K-Nearest Neighbor (K-NN) is a non-parametric inductive learning algorithm that projects every message instance in a well-defined space during the training session [49]. In the testing phase, it computes "distance metric" of the given message sample with respect to its nearest neighbors. The computation is also referred as voting of neighbors. The parameter k actually determines the number of neighbors to be considered in the computation related to the classification process. From the basic functions, it is clear that K-NN needs large memory, computationally intensive recall and it is highly susceptible to the curse of dimensions phenomenon [25]. Moreover, the learning procedure is lazy [49] as the classification procedure requires the actual computation of the distance metric to classify a message instance to its nearest neighbor(s). Although the implementation is analytically tractable and simple to implement, we do not consider this algorithm for our experimental purposes.

3.3.3 Maximum Entropy

The Maximum Entropy (MaxEnt) considers that if a probability distribution offers optimum information entropy for a message then it is the actual probability distribution with respect to that message classification [112]. In other words, the most uniform model that can be generated from the labeled training data is assumed to be able to best predict the class of a message during testing phase. Let us consider a set of training samples $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, where x_i is a real-value feature vector and y_i is the target class. So, we hereby consider a set of testable information $X = \{x_1, ..., x_n\}$. For a test/classification over a new message x, the first constraint may be defined as the sum of probability of its classification to different class members which is equal to 1.

$$\sum_{i=1}^{n} P(y_i|x) = 1 \tag{4}$$

Now, let $f(x, y)$ be an arbitrary feature function with data variable x and class variable y that we choose to model. The true value of this function provides an indication that a certain expectation is true for a given situation. Therefore, the empirical distribution $\tilde{P}(f)$ of our interest may be described by the following equation:

$$\tilde{P}(f) \equiv \sum_{x,y} \tilde{P}(x,y).f(x,y) \Rightarrow P(f) \equiv \sum_{x,y} \tilde{P}(x)P(y|x).f(x,y) \tag{5}$$

$\tilde{P}(x,y)$ is the expectation of x and y to occur simultaneously. The mathematical measure of uniformity for conditional distribution $P(y|x)$ is provided by log-likelihood $H(P)$ on the training data, also known as entropy:

$$H(P) = -\sum_{x,y} \tilde{P}(x)P(y|x).log\ p(y|x) \tag{6}$$

The MaxEnt message classifier is a discriminative model that is trained to maximize the (log) likelihood of the class labels conditioned on the features in the training examples. Therefore, it is a primal constrained optimization problem to optimize the entropy:

$$arg\ Max_{p\in C}H(P) = arg\ Max_{p\in C}(-\sum_{x,y} \tilde{P}(x)P(y|x).log\ p(y|x)) \tag{7}$$

Zhang et. al. [111] described the use of MaxEnt classifier in text classification and mentioned its remarkable ability to freely incorporate features from diverse sources into a single well-grounded statistical model. This work [111] shows a good success in using MaxEnt. Improved iterative scaling algorithm [52] is often used for training the model. However, the time complexity for training such a classifier is observed high [40, 52].

3.3.4 Support Vector Machine

Support Vector Machine (SVM) is a powerful binary classifier that follows supervised learning based on the structured risk minimization principle [111]. SVM is widely used for classification and successfully performs binary spam detection since a message belongs to binary class: spam (+1) or legitimate (-1). We consider a set of training samples: $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, where x_i is a real-value feature vector ($x_i \in \mathbf{R}^n$) and y_i is the target class ($y_i \in \{+1, -1\}$). If ϕ be a function that maps training document vector $\mathbf{x}$ to higher, possibly infinite, dimensional space, SVM finds one hyper plane in that high dimensional space with maximum Euclidean distance to the closest training examples. This distance is often termed as margin. The concept can be represented through a minimization problem where optimal weight vector W is required to find the optimal linear separating hyper plane (see Eq. 8). Mathematically, the problem can be represented as:

$$Minimize\ argmin\ \frac{1}{2}W^T.W,$$
$$subject\ to: y_i(W^T\phi(x_i) + b) \geq 1\ for\ any\ (i = 1, ..., n) \quad (8)$$

where W^T is the transpose of the matrix W. b is used to capture the offset ($\frac{b}{||W||}$) of the hyperplane from the origin. In practice, we always consider a soft margin by using a slack variable ξ_i that allows for the existence of a non-separable training set by which training error can be measured. The optimization problem can be refined as follows:

$$Min_{w,b,\xi}\ argmin\ \frac{1}{2}W^T.W + C\sum_{i=1}^{N}\xi_i,$$
$$subject\ to: y_i(W^T\phi(x_i) + b) \geq 1 - \xi_i\ for\ any\ (i = 1, ..., n)\ \forall c \quad (9)$$

During the training session, the SVM solves a quadratic problem and therefore the training time complexity is non-linear. However, given the context of spam detection, we envision to use faster training of the classifier. A linear training technique is published by Joachims [62]. The implementation of this training technique is known as Sequential Minimal Optimization (SMO) [83]. We intend to use SMO for our classification purpose.

3.3.5 AdaBoost

AdaBoost or Adaptive Boost algorithm is introduced by *Freund and Schapire* in 1996 [50]. This learning framework efficiently constructs highly accurate classification rule by boosting the accuracy of a weak rule of classification in multiple rounds. Assuming each rule has an error rate less than 50%, the algorithm simulates a weak learner on multiple distributions over the sample space of available data. By taking the majority vote on how the current set of rules is classifying the training documents, the framework finally yields a strong multi-class classifier in a fixed number of training rounds. For our purpose, we may consider a set of training samples, $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, where x_i is a real-value feature vector ($x_i \in \mathbf{R}^n$) and y_i is the target class ($y_i \in \{+1, -1\}$). We start at time t with a weak hypothesis h_t and distribution of weights D_t [78] , such that:

$$h_t : X \rightarrow \{+1, -1\} \tag{10}$$

where, the correctness of the weak hypothesis can be measured inversely from the existing error (e_t) which is defined as follows:

$$e_t = \sum_{i:h_t(x_i) \neq y_i} D_t(i) = \sum_i D_t(i) \frac{1 - y_i h_t(x_i)}{2} \tag{11}$$

In each round, the algorithm progresses as follows:

$$D_{t+1}(i) = \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t} = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t}, & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t}, & \text{if } h_t(x_i) \neq y_i \end{cases} \tag{12}$$

where, $\alpha_t = \frac{1}{2} ln \frac{1-e_t}{e_t}$ and Z_t is a normalization factor defined over $\sum_i D_t(i) \exp(-\alpha_t y_i h_t(x_i))$.

After τ rounds, we find a strong classifier that can be determined as:

$$H(x) = sign(\sum_{t=1}^{\tau} \alpha_t h_t(x)), \tag{13}$$

where, *sign* is special function that generates -1 for any negative real number, 0 for number 0 and +1 for any positive real number.

3.3.6 Decision Tree J48

J48 is a Java implementation of a decision-tree based learning algorithm that is often used in classifying categorical data. The categorical data refers to non-numeric type of information where the class for a group of information is classified through distinguished names (example: professor, student, etc.). The research work on a tree-based learning algorithm was pioneered by Hunt [59] and subsequently followed by Quinlan [85, 86, 87]. Quinlan published the Iterative Dichotomiser 3 (ID3) algorithm, that uses a divide and conquer method of classification through a suitable tree structure. The tree is designed based on information entropy from the available data. In each round, the algorithm computes Information Gain (IG) from every unused attribute of the dataset and selects an attribute with smallest entropy value in the tree structure. Quinlan created a C program of an improved ID3 algorithm, named as C4.5, that was later implemented in Java as J48 [87]. Given n possible values of an attribute in dataset, S where $f_s(j)$ is the frequency of value j of the same attribute, the entropy of the attribute is computed as:

$$E(S) = -\sum_{j=1}^{n} f_s(j) log_2 f_s(j) \tag{14}$$

The log_2 is associated to binary search in a tree traversal. The dataset with higher entropy is usually split into smaller subset of information. A class of data with entropy 0 produces perfect classification. However, in practice, larger decision tree size is more susceptible to classification over-fitting problem. Therefore, smaller appropriate size of decision tree is used for the classification purpose. In run-time, sample data get classified through the traversal of a tree branch. The J48 classifier yields performance benefit to classify categorical group of information, especially IP address and domain names. However, J48 training is proved to be expensive while its classification time is short.

3.4 Comparative Study of Time Complexity

In this section, we present a comparison for the aforementioned algorithms and analyze their suitability for our on-line SPIM detection. This analysis is verified from several documents and modified as per our requirements. We evaluated the theoretical training and testing time complexities of these algorithms and verified them through experiments. We also compared the detection accuracy during the process.

For the theoretical evaluation, we assume that there exists a set of pre-labeled training documents identified as D, where average length of each document is L_{avg}. If V represents the set of total vocabularies, we are interested in all the $|V|$ features. In average, each document contains M_{avg} features from the vocabularies. At the end of the classification, we determine class c of the document where $c \in C$ and $C = \{spam, legitimate\}$. At the classification, we test a particular document of size L_a containing M_a features.

3.4.1 NB Time Complexity

In case of Naive Bayes, the training time can be computed in two different parts [72], namely, a) Document count and vocabulary deduction and b) Training algorithm. In the first part, we assume that the lengths of the documents in D are $l_1, l_2, ..l_{|D|}$. Considering average length L_{avg}, the vocabulary detection requires a time complexity of $O(|D|L_{avg})$. For the training part, the Naive Bayes algorithm should be applied to understand the probability of each member class in C with respect to each feature in the vocabulary V. Thus the time complexity of this part can be clearly deduced to $O(|C||V|)$. Therefore, the total time complexity is $O(|D|L_{avg}) + O(|C||V|)$. However, $|C||V| \ll |D|L_{avg}$, so NB training complexity is $O(|D|L_{avg})$ [72].

Similarly, the classification/detection procedure has two parts: vocabulary detection from the incoming document and the class identification. In case of feature detection from document of size L_a, the time complexity is $O(L_a)$. Likewise, the time complexity of main classification is $O(|C|M_a)$ where M_a is the number of available features in the test document. Therefore, the total time complexity is $O(L_a + |C|M_a) \approx O(|C|M_a)$ [72].

3.4.2 K-Nearest Neighbor (K-NN) Time Complexity

K-NN is a distinguished classification algorithm that can be used with or without supervision. K-NN determines class of a document based on the voting from K neighbors. K-NN training is used to select an optimal value for K, i.e. the number of neighbors to be selected for voting. If K is a pre-selected value, the classifier does not require any training time [72]. In order to determine the value of K, the preprocessing of the document has similar complexity as of NB, i.e. $\approx O(|D|L_{avg})$.

However, in the classification or testing phase, K-NN is considered costly in terms of time. The computation of the nearest neighbors requires first to read classification document of size L_a (processing time: $\approx O(L_a)$) and then compute voting of M_a features with respect to M_{avg} features from all training documents of size $|D|$. Simply, the total processing time is therefore $\approx O(L_a) + O(|D|M_{avg}M_a) \approx O(|D|M_{avg}M_a)$. In absence of training phase, we are required to replace M_{avg} with L_{avg} as no prior featureization is performed before.

It should be noted, the K-NN testing time is independent from the number of classes $|C|$ [72]. In our case, the significance is less as our classification is binary, however, it can be very useful for multi-class classification purposes.

3.4.3 Maximum Entropy Time Complexity

The MaxEnt learning is usually performed based on two main algorithms: Generalized Iterative Scaling (GIS) [40] and Improved Iterative Scaling (IIS) [52]. Typically, it is a discriminative model that is designed to find the class boundary rather than the entire instance space. The training session begins by defining features templates, followed by creating the feature set and finally finding the optimum feature weights via GIS or IIS. In practice, we follow the iteration for bounded number of rounds (T) to obtain near optimal value. In our complexity study, we consider IIS for its improved training time.

As mentioned before in describing MaxEnt, in the training, the scaling happens for each of the existing features (M_{avg} on average) for $|D|$ documents. As a whole, the training is performed for $|C|$ classes for $|V|$ vocabulary elements. So, the training complexity of MaxEnt in each round is $\approx O(|V||D||C|M_{avg})$. Given that T rounds of training happens to get exact weight for each feature, the training complexity of the classifier is $\approx O(|V||D||C|M_{avg} \cdot T)$. On classification, however, MaxEnt is faster than others. Basically, the trained classifier is tested over M_a features for $|C|$ class types. It yields a testing complexity of $\approx O(|C|M_a)$.

3.4.4 SVM Time Complexity

SVM is a known binary classifier that finds the class boundary through solving an optimization problem based on quadratic function. The standard formulation of an SVM is a minimization problem. The complexity of solving a quadratic optimization problem, such as SVM, is time cubic with respect to the size of the data set $|D|$. *Kozlov et al.* [72] proved the approximate time complexity of $\approx O(|D|^3)$. However, Joachims later proposed an efficient training method for SVM with complexity $\approx O(|D|^{1.7})$ [62]. Such empirical training complexity allows to train SVM linearly for near optimal result. The detailed explanation of such a complex algorithm is beyond the scope of this work. We evaluated the total training complexity as $\approx O(|C||D|^{1.7}M_{avg})$. The trained SVM classifier creates a hyperplane to classify a document. Given a document of M_a features for $|C|$ class types, it offers a testing complexity of $\approx O(|C|M_a)$.

3.4.5 AdaBoost Time Complexity

In boosting technique, each hypothesis is trained to offer a strong classifier from a weak initial one. If the average number of features in a document is M_{avg} then for $|D|$ documents, the hypothesis is tested using an algorithm of time complexity of $\approx O(|D|M_{avg})$. AdaBoost, like MaxEnt, uses T rounds of training session to strengthen the classifier. Thus, the total time complexity of training is $\approx O(|D|M_{avg} \cdot T)$. In case of testing, AdaBoost is reasonably fast. It only compares M_a features from the input document. In comparison with Eq. 13 in Section 3.3.5 , the classification complexity can be computed as $\approx O(M_a \cdot T)$.

3.4.6 J48 or C4.5 Time Complexity

A tree-based learning algorithm requires to build an information classification tree based on entropy (as published by Quinlan [85, 86]). In [108], discussing C4.5-based machine learning, the author mentioned the tree generation in three different sub-steps: building a tree, sub-tree replacement and sub-tree raising. The tree is usually built up from M_{avg} features from $|D|$ documents using an algorithm of complexity $\approx O(M_{avg}|D|log|D|)$. The sub-tree replacement cost is $\approx O(|D|)$. Each re-distribution of sub-tree costs $\approx O(log|D|)$ on average and every instance may be needed redistribution at every node between its leaf and the root. Thus, the necessary cost is $\approx O(|D|log|D|)$ [108]. So the total sub-tree redistribution complexity is $\approx O(|D|(log|D|)^2)$ and the total training time complexity is: $\approx O(M_{avg}|D|log|D|) + O(|D|(log|D|)^2)$. The classification time complexity on average is $\approx O(log|M_a|)$, same as any tree traversal algorithm. However, the implementation procedure may restrict on the tree level to overcome the problem of over-fitting. Table 2 summarizes time complexity of the aforementioned algorithms.

Table 2: Training and Testing Time Complexities for Machine Learning Algorithms

Algorithm	**Training**	**Classification**
Naive Bayes	$O(\|D\| \cdot L_{avg} + \|C\|\|V\|) \approx O(\|D\| \cdot L_{avg})$	$O(L_a + \|C\| \cdot M_a) \approx O(\|C\|M_a)$
Maximum Entropy	$O(\|V\| \cdot \|D\| \cdot \|C\| \cdot M_{avg} \cdot T)$	$O(\|C\| \cdot M_a)$
K-NN (With Training)	$O(\|D\| \cdot L_{avg})$	$O(L_a + \|D\| \cdot M_{avg} \cdot M_a)$
K-NN (No Training)	$O(1)$	$O(L_a + \|D\| \cdot L_{avg} \cdot M_a)$
Support Vector Machine	$O(\|C\| \cdot \|D\|^J \cdot M_{avg})$	$O(\|C\| \cdot M_a)$
AdaBoost	$O(\|D\| \cdot T \cdot M_{avg})$	$O(M_a \cdot T)$
J48 or C4.5	$O(M_{avg} \cdot \|D\| \cdot log_2(\|D\|)) + O(\|D\| \cdot (log_2(\|D\|))^2)$	$O(log_2(\|M_a\|))$

3.5 Experimental Performance and Detection Accuracy

From the previous section, we decide to discard the K-NN classifier for its high testing/classification time. Similarly, we also discard MaxEnt. classifier for excessively large training time and absence of publicly available implementation. J48 training time is also very high for large dataset but we expect SPIM detection module to face repeated training over small dataset of pre-classified packet information as the new SPIM sending patterns will evolve dynamically with respect to time.

3.5.1 Setup

With the knowledge of time complexities for different algorithms, we verify and justify each of these selected algorithms with a collected dataset of short messages. In the absence of properly labeled SPIM dataset from packet data, we perform experiments on the accuracy of the proposed machine learning techniques on a small pre-labeled dataset of SMS which is available online [13]. Although the experiments are performed taking features from message content, we assume that similar accuracy, time will be required for offline training and online classification even with packet data. However, processing of captured packets to generate feature vectors will incur additional delay. Therefore, our objective in these experiments is to choose right algorithms that impose minimum delay in classification.

In the following, we detail the preprocessing steps and the experimental setup used in order to predict the accuracy of spam classification and to analyze the performance of different machine learning classifier models.

We employ data preprocessing features using a Java-based open source machine learning tool, Weka, version 3.6.1 [16]. The preprocessing includes the following:

- Creation of intermediate attribute relation file format (".arff" file) from datasets;
- Extraction of features from .arff file by selecting attributes and removing stop words;
- Addition of @attribute $is\_spam \in \{0, 1\}$ to classify a file as per provided knowledge on the sms/email-types.

We use a pre-existing list of stop words that are commonly used in sentences such as: a, the, etc. During preprocessing, we randomize the categorical output in order to retain an unknown distribution of input. In the classification stage, we consider NB, J48, SVM, and AdaBoost algorithms for the SMS dataset. In order to avoid high training complexity of

SVM, we use efficient sequential minimal optimization (SMO) algorithm. We use the *10-fold cross validation* to evaluate the classification accuracy. In this technique, 90% of the total data is used for training of the algorithms and the rest 10% is used in classification. Each test is repeated 10 times with different testing inputs and the average is considered.

The following system configuration is used for the experiments:

- CPU:Intel Core i7-2600 CPU 3.40 GHz
- RAM: 16GB memory
- Operating System: Microsoft Windows 7 (64 bits)

The following subsection analyzes the experimental results from the perspective of the spam classification accuracy and performance for different machine learning classifiers.

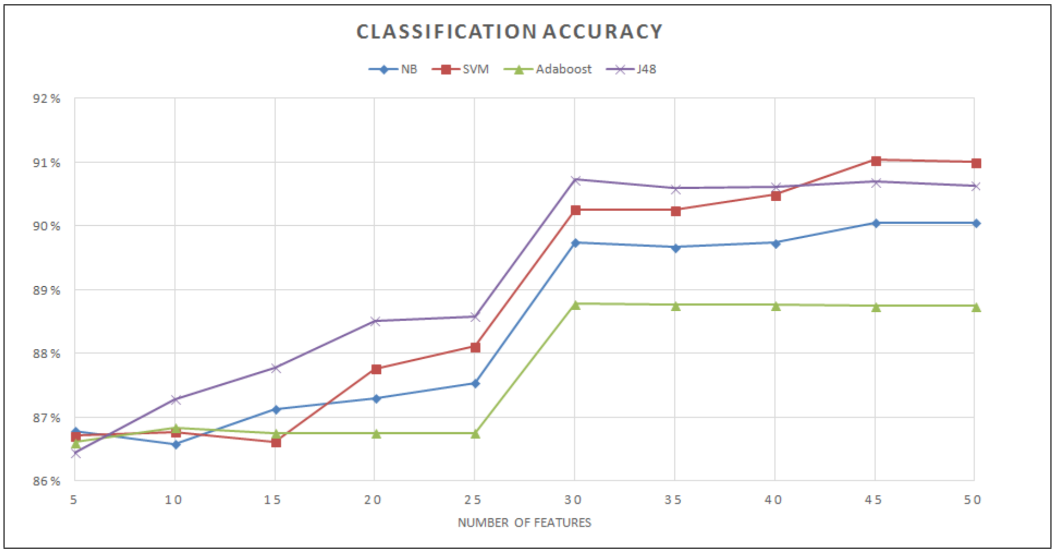


Figure 7: Spam Classification: Accuracy Vs. Number of Features

3.5.2 Detection Accuracy

The detection accuracy is defined over the percentage of correctly classified spam and legitimate documents with respect to the total documents. The accurate detection represents proper recognition of spam and legitimate data commonly represented as *true positive* and *true negative* respectively. The other two important quantifiers are *false positive* and *false negative*. In our case, false positive rate determines the percentage of legitimate documents wrongly identified as spam. False negative rate is the percentage of spam documents identified as legitimate. We intend to choose algorithms with low false positive rate, as large false positive rate may adversely impact a legitimate instant messaging communication. Figure 7 depicts the spam detection accuracy on the SMS dataset. The accuracy is judged over an increased number of features up to 50 features. It is noteworthy to mention that all the studied algorithms show high-level of accuracy even with small number of features. The accuracy of AdaBoost algorithm is particularly interesting where the level of detection accuracy is not affected by larger number of features used in the experiments. The detailed SPIM packet features are discussed in the Section 3.6.

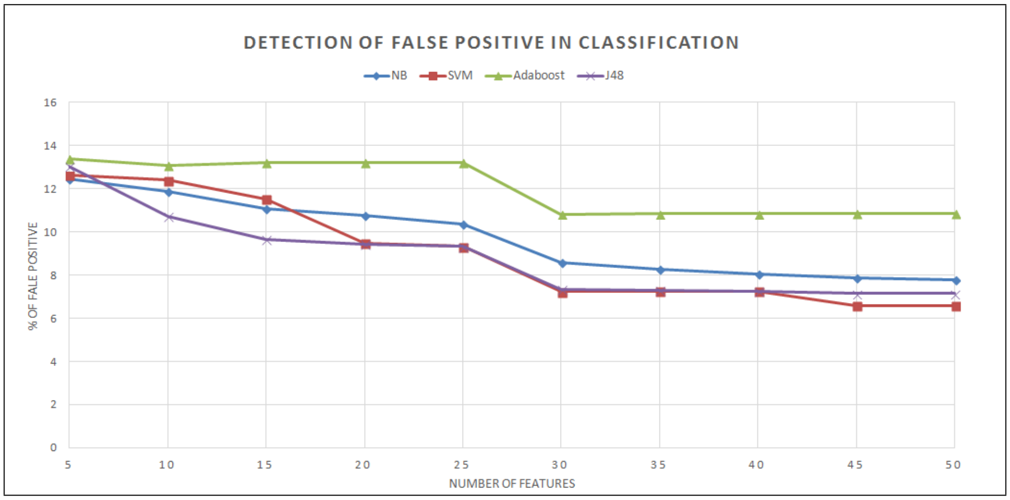


Figure 8: False Positive detection in classification over SMS Dataset

Figure 8 depicts a comparative study for the percentage of false positive found during the experiments. As the experiments incorporate consider higher number of features from the SMS dataset, the The false positive rate gradually decreases for all algorithms except AdaBoost implementation. In general, SVM classifies with low false positive ratio representing its gradual perfection in identifying legitimate messages correctly. NB and J48 follow higher percentages of false positive in this respect.

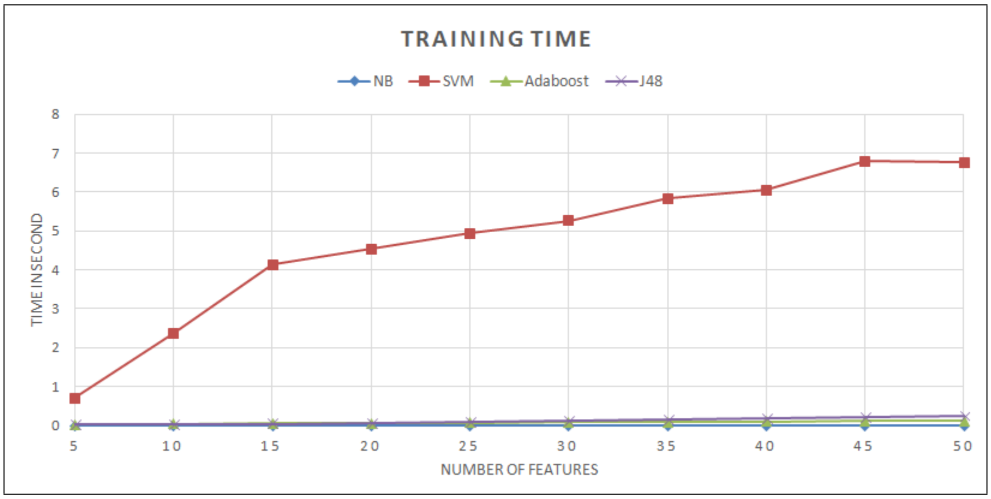


Figure 9: Comparison of Classifier Training Time over SMS Dataset

3.5.3 Training Time

Figure 9 compares the training time of the classifiers on the SMS dataset. The training of the classifier model is important in the process of machine learning. We expect SPIM detection environment to be fairly dynamic since the spam senders practice a range of deceptive techniques to establish connection and communication. Therefore, periodic incremental training may be required for better performance in SPIM detection. According to the experimental results (*see* Figure 9), training time is comparable for AdaBoost, J48 and NB in the small repository of SMS dataset. However, the training time of SVM (SMO implementation) is remarkably high with inclusion of more features. We suspect that, in practice, it will incur longer training process if SVM is used for SPIM detection.

3.5.4 Testing Time

Figure 10 presents a comparative study of online spam classification time. The classification of SPIM detection is expected to be near real-time. Therefore, machine learning models with very low testing time are critical for the SPIM detection purpose. Among all the algorithms tested over SMS dataset, AdaBoost and J48 show very promising result as they incur low detection time in compare to others.

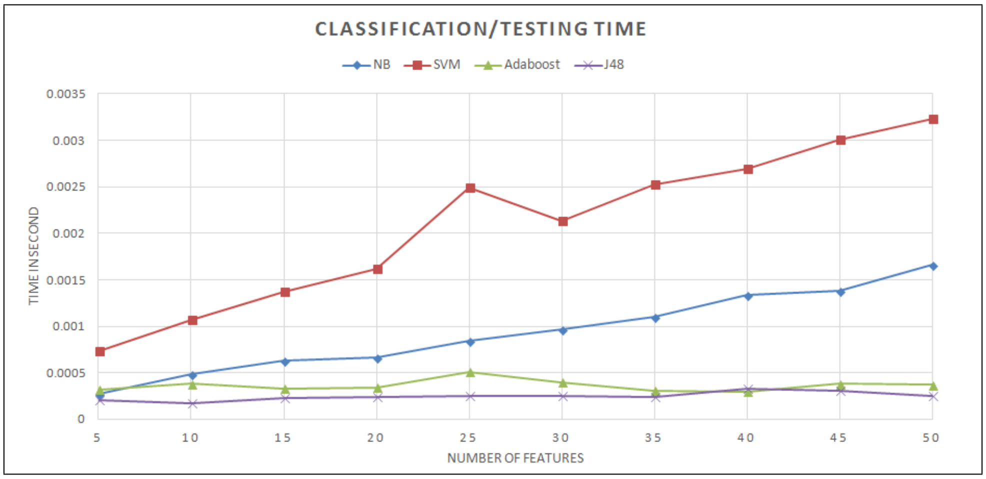


Figure 10: Comparison of classifiers: Testing Time over SMS Dataset

3.6 Feature Set Selection

An accurate selection of features is vital to detect SPIM by applying machine learning technique in the LTE network. In general, spam features are often chosen from previously stored offline message content especially in the detection of email-spam, blog spam, etc. In this respect, content-based spam detection has been extensively studied in literature [26, 111]. On the other hand, SPIM detection requires fast classification of network traffic including packet data due to its near real-time data communication setup. Our proposed location of SPIM detection module in PDN-Gateway is suitable to meet these requirements. In this thesis, we concentrate on finding a general set of features for network traffic aiming to identify SPIM packets from network traffic with acceptable classification/detection time and accuracy. A list of features has been selected below after a thorough discussion.

Korczynski [64] illustrated various possibilities of network traffic classification for IM in his thesis. The network traffic features can be important for traffic-based spam classification as well. The features can be categorized as content dependent or independent. The content based traffic data can be classified based on port and payload. Content-independent traffic features generally include host-behavior, flow related features, etc. Packet payload offers usually unencrypted header data with protocols and other application information. However, message content in IM traffic is often encrypted and transformed to almost similar size of data packets before sending. Besides, packet flow features include average packet sizes, packet inter-arrival times, flow durations, etc. In the work on classifying modern IM applications, Korczynski mentioned that single network feature-type does not help in classifying modern IM applications as these applications themselves try to bypass restrictive firewalls by randomly changing features like port, payload signature, etc. In our purpose, we emphasize on the payload independent features and header information of the payload. Previous researches showed that payload independent features maintain properties throughout a session as they are hard to be manipulated by spam senders [80]. On the other hand, header related features with IM traffic characteristics may offer significant insight on the class of IM packets if the classifier model is trained properly over such repositories [111]. In the following, we describe some payload-dependent network traffic features suitable for data mining purposes. It should be noted that the features are chosen considering their detection feasibility in the LTE downlink traffic.

- *Protocol*: The IM packet contains multiple packet headers in the network traffic. Our needed protocol information resides at the TCP or UDP packet headers. Sometimes IM packets use an additional HTTP header information to bypass firewall restrictions. The network devices at PDN-gateway mainly inspect IP header but are also capable to collect protocol information in TCP or UDP headers. This protocol information influences a set of predictable behaviors for other packet features such as port, packet size, etc. Protocol has been previously used in classifying packets [106, 38].

- *Port*: Port is a key attribute that reflects the intent of a packet. A legitimate packet belonging to a specific messaging application connects to a predefined port while SPIM packets may be directed to other ports aiming to exploit vulnerabilities. For example, W32.Aplore.A@mm worm attacks the AOL messenger by starting an HTTP server on port 8180 [57]. Along with destination port, the source port of the packet can also be tested if it is an open port of a machine.

- *Packet Size*: The packet size is a necessary feature that most of the IM services keep under 1500 byte to pass through Ethernet. For example, SIMPLE protocol uses data packets of up to 1300 bytes approximately [8]. An unusual large packet size can be considered as a signature of SPIM trying to exploit a buffer overflow. Smaller packets can be ignored as they are used for IM signaling [38, 64]. We recommend to use average packet size and its variance in network flow as training features.

- *Inter-Arrival Time*: The packet inter-arrival time is a statistically important feature to classify the nature of usually bursty traffic of IM communication. Machine learning techniques are successfully applied using packet inter-arrival time in order to identify packet characteristics from the network flow [99].

- *Minimum Burst Duration*: Suh *et al.* [100] investigated the importance of burst duration in IM data flow over Skype traffic. The concept can be generalized for IM applications. Interactive messenger application receives messages in bursts based on sender's messaging pattern. The minimum burst duration in human interaction is expected to differ from auto-generated and targeted SPIM packets.

- *Average Packets per Flow*: Average number of packets per network flow is a good indication for the nature of the IM packet traffic. Usually, TCP offers some congestion control and IM applications also restrict number of packet delivery within a small interval. SPIM bots very often create a large number of packets to send multiple users for advertisement. They may also create denial of service attack by sending large number of packets to one user such as: SYN-Flooding. Monitoring average packets per flow in LTE downlink may help in identifying SPIM packets and forecast attack scenarios. Several research articles recommended to count SYN packets, RST packets and FIN packets in the IM traffic flow [67, 106].

- *Packet Arrival Order:* We consider packet arrival order as an important feature that will not only help in classifying SPIM packet but also determine the amount of data required for the classifier training. Buonerba also explained in her thesis the significance of packet arrival order to uniquely identify messaging application, protocol and the medium in between. [29]. A legitimate message traffic originating from a valid user will grossly differ in overall packet arrival order (within a small duration) from auto-generated spam messages originating from many different malicious applications.

- *Repetition on packet loss after TCP Repeat Window (RWIN)*: Usually, instant messenger applications repeat a packet, if lost, after the RWIN threshold. RWIN determines how many packets a user device can accept without notifying sender application. SPIM message from bots are not expected to resend packets.

We have also investigated a group of spatio-temporal payload independent features originally published by Hao *et al.* [56] to incorporate in the feature list. The expected values of these attributes can be gathered in a stored data tables separately inside the detection module. We claim that such an external verification will leverage the training of the classifier.

- *Distance between sender and receiver*: Once a packet is analyzed, the distance between sender and receiver can be calculated from their different IP addresses using a lookup table that contains relations among IP address and geographic location. Hao *et al.* [56] have claimed that most of the legitimate messages are exchanged among friends at nearby locations where as SPIM are sent from distant geographic locations.

- *Packet source neighborhood density*: The geographic locations of spam packets are expected to be close as the bots propagate through nearby computers. The distribution of legitimate packet sources, if not sent by same user, is expected to be distant.

- *Packet timestamps*: The time at the source of a packet or a group of packets in a network flow is also very important factor [56]. A bot is primarily a programmed application. Therefore, it is expected to send packets at a particular time pattern to a user as programmed. Furthermore, it is possible that multiple SPIM packets are sent to different locations under same or different ISPs at a very close proximity of time. A common source of knowledge can be used and updated to identify the class of packets from the packet time-stamp.

- *Autonomous System Number*: The unprotected ISPs are used by the SPIM senders. Autonomous System Number (ASN) provides the originating ISP of the packets that can be verified through a previously stored tables with IP addresses of the packet.

We have managed to capture various packets to analyze the aforementioned features using packet capturing tool namely Wireshark [35] (as depicted in Figure 23 and Figure 24). After a careful analysis of captured packets, it has been seen that a subset of aforementioned features can always be used for mining purposes. In Appendix, Figure 23 depicts the excerpt of data packets following Skype protocol. While the payload in this packet is typically encrypted, Skype (peer-to-peer communication) reveals the source IP address of the packet along with several other header information. On the other hand, Figure 24 elaborates an excerpt of a packet data communicated between two Yahoo users. Like other client-server based messaging architecture, packet source location is not clear from the packet as it is originated from a Yahoo server. However, the message content is unencrypted and can be used for content based machine learning. Therefore, we recommend to study the particular instant messaging protocol before choosing a final set of features for SPIM mining for specific applications. It should be noted that, in the PDN gateway, uplink packets pass unfiltered. So network flow characteristics from uplink packet features are unavailable to our classification purpose. Interestingly, we do not use IP address as a feature, because in mobile network, IP address changes with the change of locations [56]. Spammer may create packets with different IP address than his/her own. IPs are often dynamically assigned also.

A typical feature selection procedure attempts to choose a subset of available features to be used by a particular machine learning procedure [99]. The performance of machine learning algorithms vary from one feature set to others. The techniques of feature selection are mainly classified as filter method and wrapper method [99]. In filter method, features are selected based on statistical properties and metrics. It accounts to independent assessment of multiple learning algorithms. The wrapper method predetermines the mining algorithm and chooses a particular set of data features that fit the algorithm [99]. The filter method is faster with large dataset than wrapper method as the features are selected once. It also incurs low cost. However, wrapper method is comparatively more accurate if not over-fitted. We have clearly focused on the first method and evaluate performances of multiple machine learning algorithms on spam dataset.

3.7 Summary

SPIM filtering is an area of active research. In this chapter, we addressed a selected group of effective algorithms that can be used for SPIM filtering purposes. We evaluated the training and testing time complexity for each of these algorithms. We proposed a high level design of the SPIM detection module. Furthermore, we carried out a series of experiments with pre-classified data to conclusively determine the most efficient classifier for our purpose. The experimental results present the detection accuracy, training and testing time for each of the algorithms. While a faster classification is required to adapt the classifier with high volume of incoming information, the wrong detection of SPIM will penalize legitimate message passing. According to our experiments, J48 and AdaBoost will perform time-wise better than other algorithms with low false positive scores. With respect to SVM, we investigated a better training technique using SMO [62] to train classifier in linear time. AdaBoost is comparatively easy to train and test but has lower detection accuracy and higher false positive value in compare to others. We also realized that the reactive nature of the spammers is a major concern. Spammers bypass the filters by skewing the message statistics or by preventing proper featureization of the message using split-words or modifying message features. Placing extra spaces within a word may bypass a filter that classifies spam based on full words. Message content is also sometimes obscured from a filter by means of encoding. Then the detection has to be performed solely on packet header and packet features. In this context, we discussed a number of alternative features that can be used in classification algorithm. We constantly focused on packet and network flow features that are difficult to change for the SPIM senders.

Chapter 4

Simulation and Analysis

This chapter elaborates on the implementation of a proof-of-concept for SPIM detection module in OPNET network simulator [10]. We detail our findings from the execution of simulated scenarios and analyze the collected results. We hope that this elaborated insight along with verified results will encourage mobile network and service providers to consider for machine-learning based SPIM detection for large-scale SPIM detection.

4.1 Overview

We begin by modeling the simulation framework. The network modeling steps may also serve as implementation guidelines for the actual system deployment. The underlying design of the test-bed is achieved in four different phases, namely:

- Modeling LTE network along with IMS extension module, application server authentication server and other necessary components.
- Generating and characterizing network infrastructure and user equipments with different traffic flow patterns and user roles by applying specific QoS configuration while creating small-scale simulation environment.
- Designing scenarios based on core IM functionalities in mobile telecommunication networks through sequence diagrams.

- Implementing a proof-of-concept for packet data classification at a simulated LTE network. The packet snipping delay function is modified for SPIM detection based on packet features (such as: size, protocol, etc.), machine learning algorithm, classification time and other characteristics as documented in Chapter 3.

We decide for OPNET simulator from various network simulation engines available off-the-shelf in the software market. However, we acknowledge that the Network Simulator (NS), QualNet, GloMoSim, NetSim are also appropriate for our purpose. OPNET [10] is a popular network simulator with rich graphical user interface. It offers users a multitude of design, deployment, simulation, and network management possibilities. A large number of components, protocols are in-built in this software application. NS [11], on the other hand, is a discrete event simulator created and maintained by the University of Southern California. NS programs are specified in Tool Command Language (TCL) to run the simulation. QualNet [12] is another commercial software application to run what-if scenarios on simulated large complex wireless, wired and mixed platform network. The user interface is visual and feature-rich. Global Mobile information systems Simulation laboratory (GloMoSim) [5] is also a discrete event simulator but uses a separate compiler Parsec. The underlying network system of GloMoSim is built on the OSI seven layer model and able to run a number of simulations in parallel. NetSim [9] is an educational purpose software from TETCOS [15] that is built on top of a real-time network monitoring tool: Net-Patrol. NetSim simulates most of the basic protocols in connection to cellular technologies. Beside these software, a range of network simulators are available in Internet. Some of them are free and open source such as: OpenWNS, GTNetS, Cnet. Few are free for academic purposes such as: OPNET, OMNeT++ while OptSim is complete commercial. We also analyze the possibility of extending and using general purpose simulator tool such as: MatLab components for our network simulation.

We select OPNET network simulator for its superior quality of discrete event simulation, advanced Graphic User Interface (GUI), great performance, and license availability. In the following section, we briefly illustrate OPNET capabilities for the purpose of our interest.

4.2 OPNET Simulator

OPNET simulation engine is a fast discrete event simulator industrially available. This network simulator works in three different phases, namely: Modeling, simulation and analysis. In the first phase, it offers a number of communication network modeling features through its rich graphical user interface. The software application hosts many device configurations, a large set of communication protocols and several other features. The model design is hierarchical. It helps in designing large architecture and then offers to configure details for each component. The modeling is object-oriented and C++ like code exists behind each of these components. During simulation, the compiler creates the model from the design. This process is automated. However, advanced user may change directly in the code and recompile for desired modification. It can also run analytical and hybrid simulations, if needed. The core kernel runs in parallel with grid computation support for distributed simulation running. OPNET also offers unique data analysis interface with graphs and charts. This interface allows viewing the graphs with different criteria of user interest and exporting results in comma-separated values (.csv) format.

In our experiments, we use a separately licensed specialized LTE model from OPNET that offers to incorporate LTE components in network design and promises adequate data rates, improved system architectural performance and high spectrum utilization. The proof-of-concept was also required to specify the traffic flow. In this regard, we employ OPNET Application Characterization Environment (ACE) to design the traffic flow. The implementation work has been carried out to create a simple messaging application.Corresponding packet traffic are configured in sequence diagrams.

4.3 Instant Messaging Application Design in LTE

As presented in Section 2.4.1, IM works through three core services. In this design, we identify and summarize a core subset of functionalities from these services. Our design mainly accommodates the messages and activities used in SIMPLE protocol but it can be generalized for XMPP as well. We design our proposed application through six tasks in order to address the core activities. For the connection establishment and termination, we include three tasks: *connect*, *login* and *logout*. Other two tasks *register* and *subscribe* are designed to address presence of users. The *register* task registers and updates a user to the application server using a REGISTER message. On the other side, a registered user receives periodic updates on other users who s/he subscribed. The subscription is performed through SUBSCRIBE message. We followed SIMPLE based messaging sequence to design these two tasks. We also address the session binding and messaging activity through a combined *invite and messaging* task. The session binding is typically performed through an INVITE message between the application server and users. In peer-to-peer IM, once a session is established, users communicate through near real-time messages directly using their proxy(s). However, in client server architecture, messages pass through the application server. Detailed IM functionalities and protocols are publicly available in RFCs and literature [69].

Table 3: Description of SIP Request Methods

Request Name	**Description**
INVITE	Establishes a session
NOTIFY	Notifies presentity to user agent about a particular event
PRACK	Acknowledges the reception of a provisional response
PUBLISH	Uploads information to a server
REGISTER	Maps a public URI with current location of the user
SUBSCRIBE	Requests to be notified about a particular event
MESSAGE	Carries an Instant Message
UPDATE	Modifies some characteristics of a session

In the process of session establishment, several SIP/SIMPLE requests are used. Table 3 describes a subset of relevant SIP requesting methods. In the following, we address the proposed six activities at a high level for the mobile network.

4.3.1 Connect

Every UE in the mobile network first tries to connect to an instant messaging server through its SIP servers [63]. During this process, the SIMPLE protocol creates session-related messages using session signaling. The session signals help in binding connection with the IMS core SIP network (P-CSCF, S-CSCF etc) to the IM servers using SIP.

4.3.2 Login

When a user performs login through a device using user ID and password, an encrypted request is dispatched to the corresponding application server (AS) for secure login through the operating router. The AS redirects this login request to the authentication server and a dedicated TCP/IP connection starts for a communication session after a successful login. During the login procedure, operator's server checks both: the limit of session and the status of the server (busy or idle). Afterward, it notifies the user about the availability of the server. The authentication server sends the acceptance in encrypted and compressed format. At the same time, it creates a cache of the connectivity information for logging purposes and for maintaining the current session [89].

4.3.3 Register

In order to establish communication, secure profiling and preventing spoofing attacks [79], senders are required to be registered through a SIP REGISTER request in IMS. A registered user is named with a registered Public User Identity (PUI) in order to map its location. Registered users can subscribe to receive notifications on the up-to-date presence information of a requested contact upon approval of the other user. Figure 11 describes the registration procedure in IMS (*see* Page 341, [30]). The user logs in the IMS and sends register request to S-CSCF (steps 1-5). Diameter User Authentication Request (UAR) command (step 3) is prompted by I-CSCF to retrieve user information from HSS and to search for the address of an appropriate assignable S-CSCF [103]. User Authentication Answer (UAA) is received by the I-CSCF and forwarded as the registration request to an assigned S-CSCF. S-CSCF performs the registration as a third-party registrar in the IMS. After getting REGISTER request (step 11) from S-CSCF, AS generates a MESSAGE containing transcription of the pending messages to the UE through S-CSCF and P-CSCF [30].

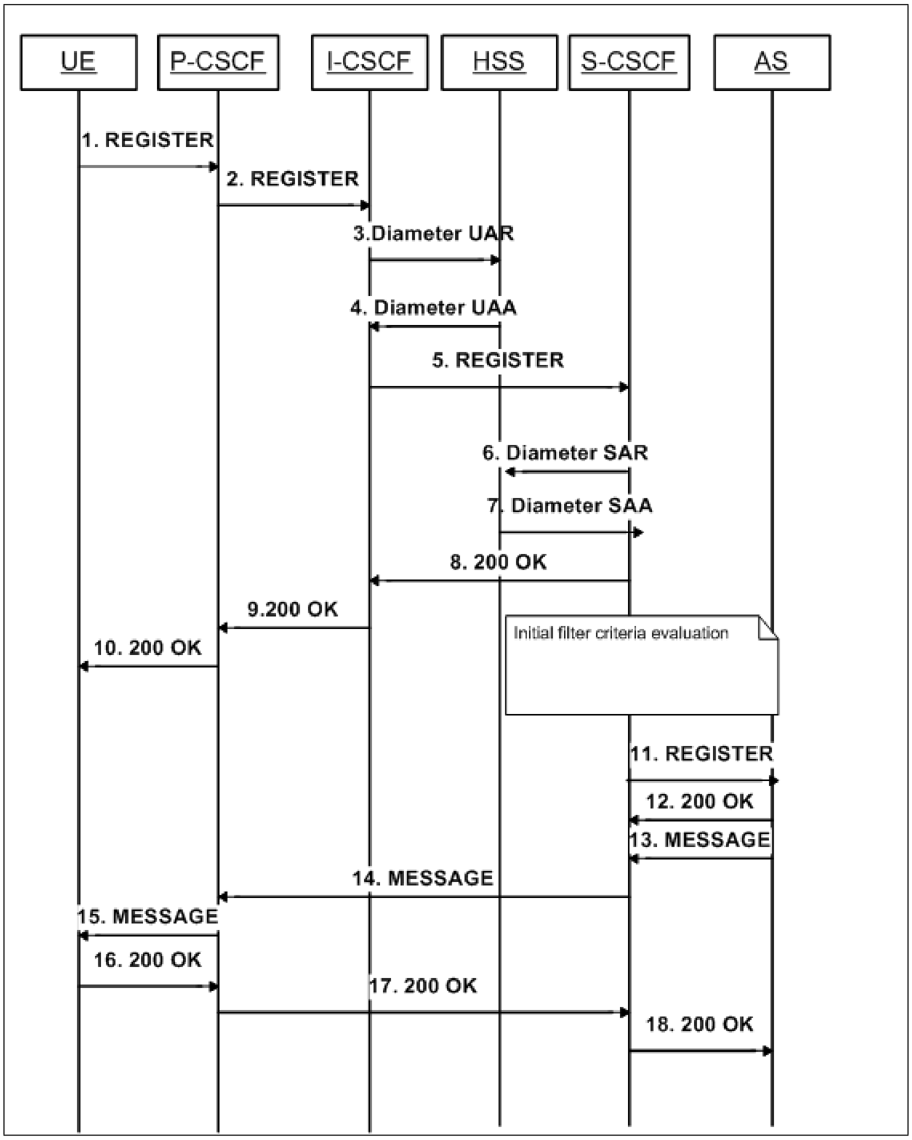


Figure 11: Sequence Diagram of *REGISTER* for Instant Messaging [30]

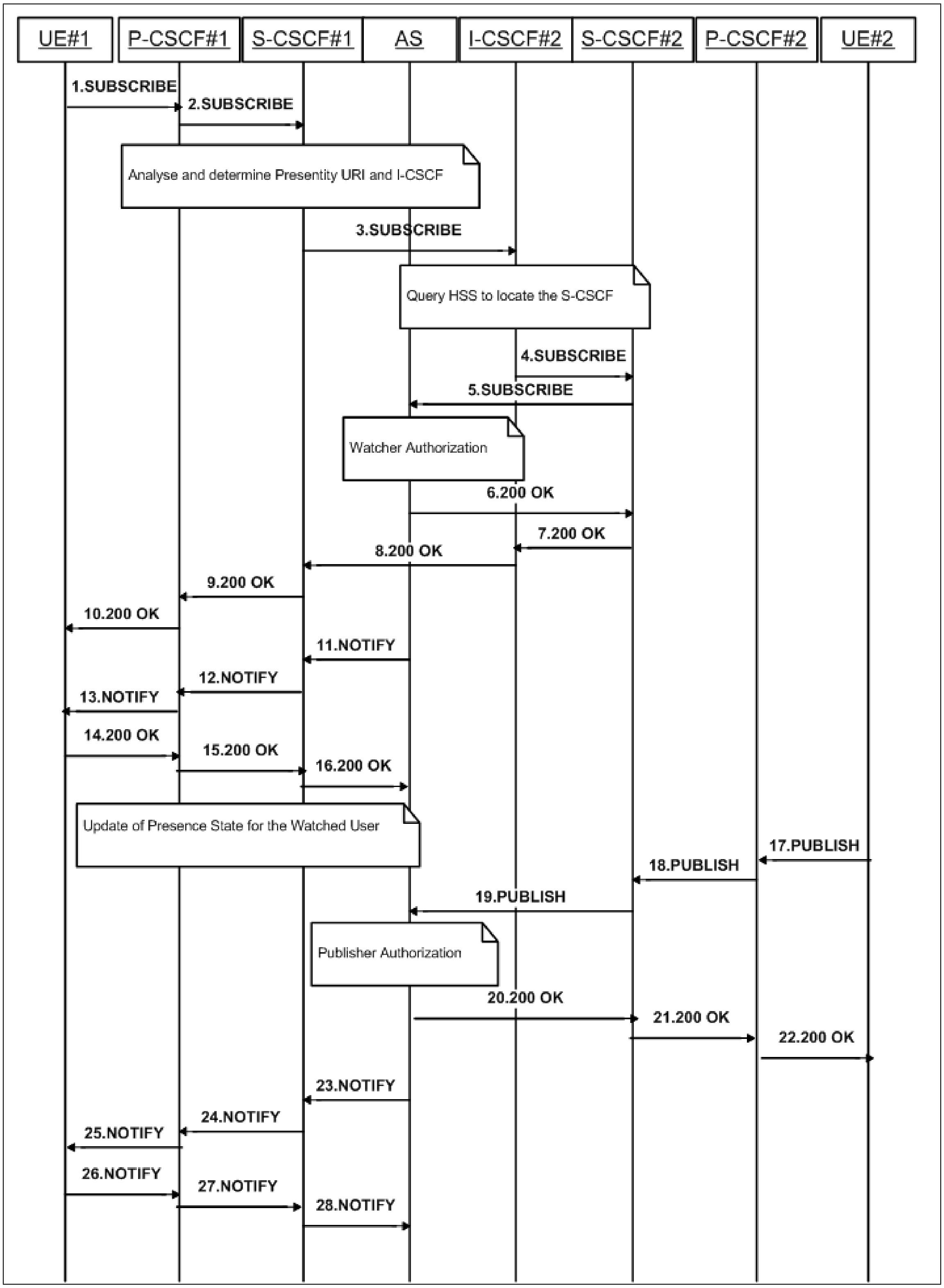


Figure 12: Sequence Diagram of *SUBSCRIBE* for IMS Presence Information [47]

4.3.4 Subscribe

Figure 12 elaborates on the reception of a SUBSCRIPTION request for the presence information from UE. We assume that there are two distinct UEs available in the network playing the role of two profiles UE-1 and UE-2 in the figure. P-CSCF-1 relays the UE-1 generated request to S-CSCF-1 along with necessary information stored during registration. S-CSCF-1 consequently transfers the request to the I-CSCF-2 at the destination network. Similar to the registration procedure, I-CSCF-2 sends query to the HSS to look up suitable S-CSCF-2 address and forward the SUBSCRIBE request. This request is further transferred to the Application Server (AS), which performs the final authorization. With a proper authorization, the affirmative response (200 OK) is sent all the way back to UE (steps 7-10). AS also sends a NOTIFY request (steps 11-13) just after. The UE-1 acknowledges the NOTIFY request using an OK message (steps 14-16) that goes back to the AS. Once the UE-2 publishes its presentity data using a PUBLISH message (steps 17-19) for necessary authorization checks, it comes back with 200 OK response to UE-2 (steps 20-22). The process thereafter notifies about the presentity to UE-1 by a NOTIFY message (steps 23-25) and ends up finally by passing 200 OK response message (steps 26-28) to AS [47].

4.3.5 Invite and Messaging

We present a sequence diagram for the establishment of an IMS session between two UEs in Figure 13 illustrating the use of invite and messaging requests. First, the SIP INVITE message traverses the IMS nodes (steps 1-12) through P-CSCF, S-CSCF etc. as a part of an active Session Description Protocol (SDP) for the instant messaging. These instant messages are sent end-to-end via Message Session Relay Protocol (MSRP) (steps 13-14). It is a text-based messaging session, hence, PRACK and UPDATE requests are omitted from the session flow. MSRP session has its own congestion control and it is not regulated by the size of the instant message [79]. MSRP is implemented in the IMS terminals, which resides near the AS. Messaging also runs on the media plane. All the messages generated during an IM session are related together in the context of its session [79]. A messaging session is

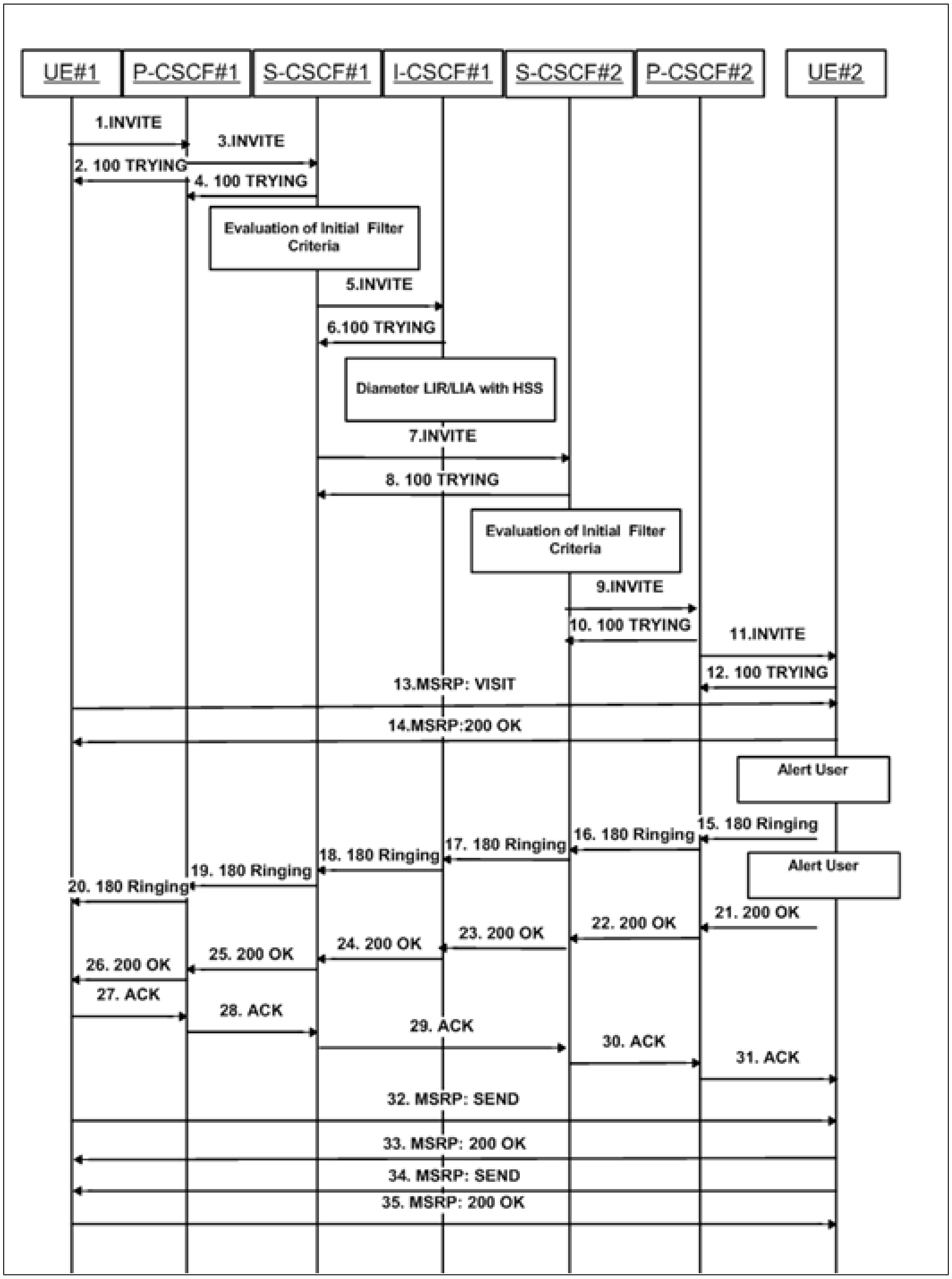


Figure 13: Sequence Diagram of *Invite and Messaging* [79]

maintained between users and can be terminated by either party. To terminate a session, a BYE message is sent to the IMS followed by a 200 OK response message sent back to the user.

4.3.6 Logout

The logout process ends all the activities between an instant messaging user with the server. The server correspondingly terminates all active messaging sessions for the requesting user and removes her/him from the current register list. Consequently, all the other users are notified about the absence of the user until s/he reopens a new session.

4.4 OPNET Simulation Setup

The OPNET simulator setup is composed of IM application configuration, user profiles and network components. In the following, we describe each of them with necessary details. Most of our chosen packet features do not change rapidly with the local movement of user equipments within short simulation run time therefore this setup omits to design the movement for the mobile agent. Furthermore, SPIM sending machines are usually stationary. We believe that such an omission will not cause a significant impact on the performance of the SPIM detection module.

4.4.1 Application Setup

Figure 14 highlights the proposed instant messaging application setup using SIMPLE protocol in a mobile telecommunication network, where traffic volume and speed have been configured according to LTE specification. In our simulation, we consider UE 1 and UE 2 as two IM client profile with messaging and presence service. We exclude the IM group management service (messaging of one-to-many clients) for simplification. Here, UE 1 starts communicating with a remote client UE 2. The IM communication is ultimately conducted at two different tiers in the LTE network, namely the SIP/IP core (which is mainly the IMS network) and IM/presence server in the packet-data network. For a successful session

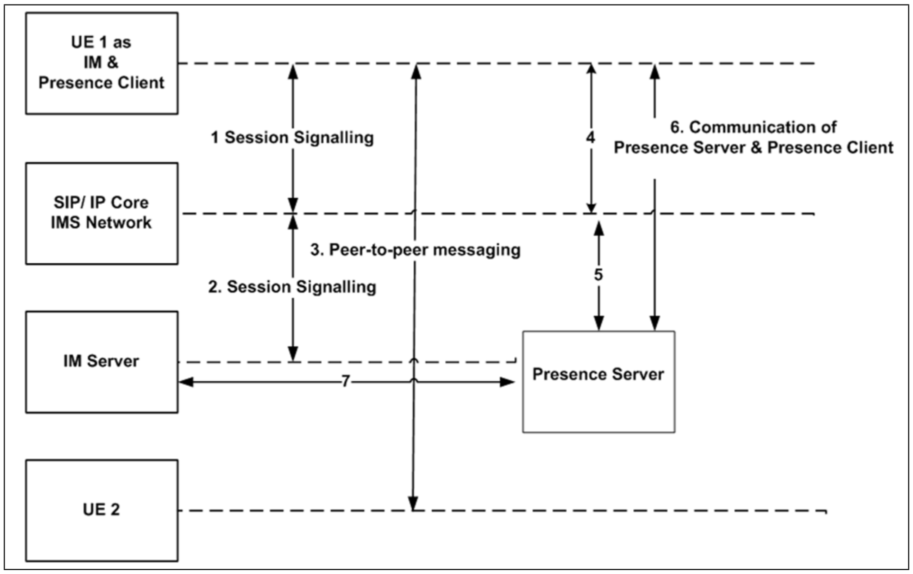


Figure 14: IM Architecture Using SIMPLE Protocol [63]

binding, a synchronous communication is performed with IMS Core network (step 1) using SIP protocol and then forwarded to the IM server (step 2). Once the session signaling is set successfully, peer-to-peer messaging occurs between two communicators using MSRP protocol. Furthermore, presence is updated and passed between presence server and presence client (step 4 and 5) using SIP. The same procedure of session initiation and messaging is repeated for all pairs of clients by randomly changing their profiles.

4.4.2 Profile Execution Setup

The procedure of setting IM applications inside the simulation execution is illustrated in Figure 15. Simulation run is bounded by a time period where several UEs execute in parallel while maintaining either UE 1 or UE 2 profile in any application instance with different bearers (an IP packet flow with different QoS between the gateway and UE) . Each profile serially runs instances for an IM application. Precisely, a client serially performs the proposed six IM activities (*connect* to *logout*) and generates packet traffic within an active application instance. Once the client/UE profile performs *logout*, the application instance terminates. UE profile executes the same activities again in the subsequent application

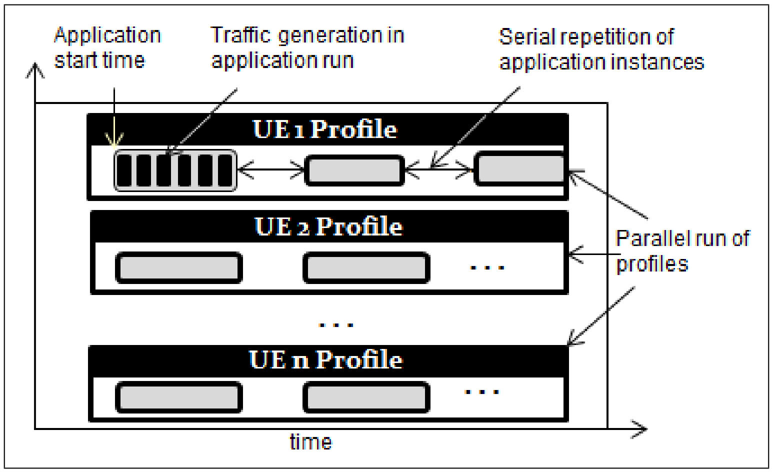
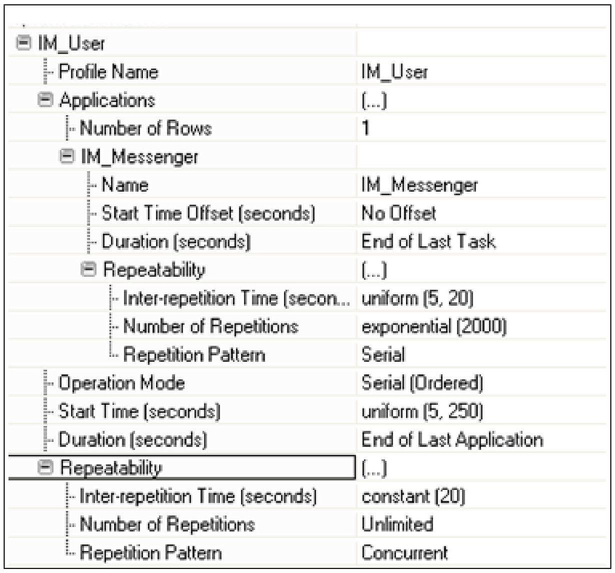


Figure 15: Simulation Configuration Setup [10]

interval. It might be argued whether application instances can run in parallel within same profile. It is possible to design multiple communication sessions within same application instance. However, we do not see significant impact of it on the desired outcome of the classification time with different machine learning algorithms. We, therefore, keep it simple for the modeling purposes by allowing one session for an application instance and globally manage the traffic flow rate in the LTE architecture during simulation to verify the performance for our proposed module. Figure 16 depicts the OPNET setup to represent the simulation of profiles. We carefully configure repeatability of the applications and profiles between start and end of a simulation run.

4.4.3 Network Component Setup

Table 4 summarizes the parameters for the LTE environmental entities in our simulation. We characterize and implement instant messaging in LTE framework according to the 3GPP IMS messaging session establishment concept for SIMPLE protocol [3GPP TS 23.228 V9.3.0 2010-03]. After the session establishment, we simulate peer-to-peer instant messaging among the UEs through the proxy(s). We choose a small campus network of 2000 UEs, 10 eNodeBs and 1 EPC for the mobile framework along with core IMS components, namely, P-CSCF, S-CSCF, and I-CSCF. We simulate IM application server and authentication server behind



⊟ IM_User	
Profile Name	IM_User
⊟ Applications	(...)
Number of Rows	1
⊟ IM_Messenger	
Name	IM_Messenger
Start Time Offset (seconds)	No Offset
Duration (seconds)	End of Last Task
⊟ Repeatability	(...)
Inter-repetition Time (secon...	uniform (5, 20)
Number of Repetitions	exponential (2000)
Repetition Pattern	Serial
Operation Mode	Serial (Ordered)
Start Time (seconds)	uniform (5, 250)
Duration (seconds)	End of Last Application
⊟ Repeatability	(...)
Inter-repetition Time (seconds)	constant (20)
Number of Repetitions	Unlimited
Repetition Pattern	Concurrent

Figure 16: OPNET Simulation Setup for a User Profile

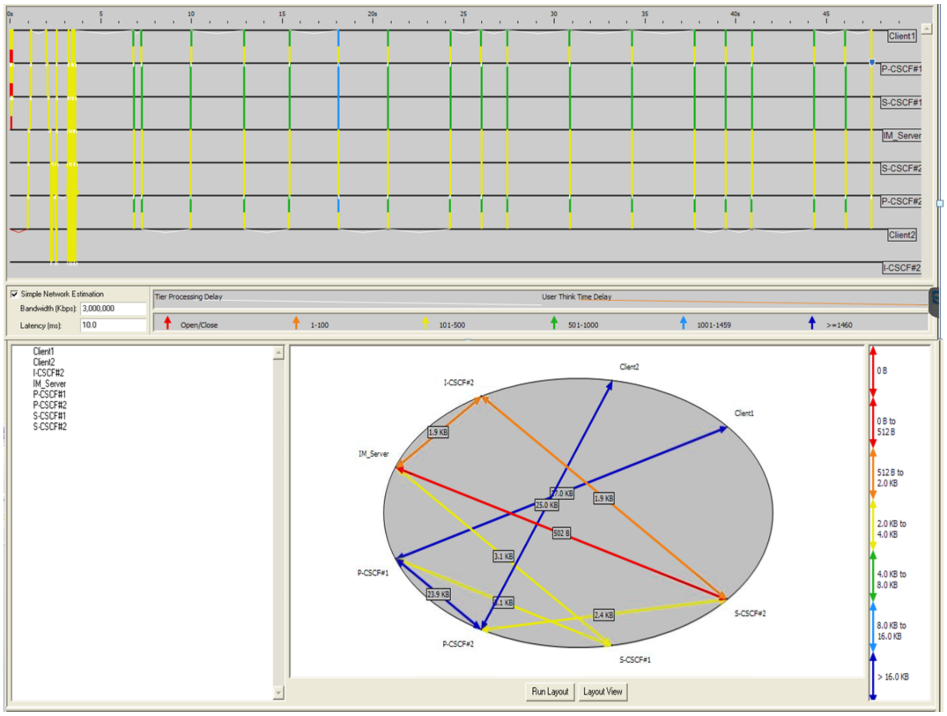
Table 4: LTE Simulation Model Configuration

Name	**Count**	**Link**	**Link Type**	**Bandwidth**
UE	2000	$UE < - > enodeB$	Radio	10 Mbps
eNodeB	10	$enodeB < - > Gateway1$	Ethernet	10 Gbps
Gateway1	4	$Gateway1 < - > EPC$	Ethernet	10 Gbps
EPC	1	$EPC < - > Gateway2$	Ethernet	10 Gbps
Gateway2	1	$Gateway2 < - > I - CSCF$	Ethernet	10 Gbps
Gateway2	1	$Gateway2 < - > P - CSCF$	Ethernet	10 Gbps
Gateway2	1	$Gateway2 < - > S - CSCF$	Ethernet	10 Gbps
Gateway2	1	$Gateway2 < - > Internet$	PPP_Sonet_OC48	2500 Mbps
Internet	1	$Internet < - > Gateway3$	PPP_Sonet_OC48	2500 Mbps
Gateway3	1	$Gateway3 < - > IM - Server$	Ethernet	10 Gbps
IM-Server	1	$IM - Server < - > Auth.Server$	Ethernet	10 Gbps

IP cloud as depicted in Fig. 4 in Section 2.3.2.

4.4.4 Traffic Configuration

OPNET offers ACE Whiteboard utility to configure new application with visualization of model design and application analysis. It helps to configure traffic rate, bandwidth and other necessary details to setup IM activities. Figure 17 elaborates a composed view of two diagrams where the top diagram depicts the sequences of traffic generation and the bottom image represents the system view of such a packet data communication. We



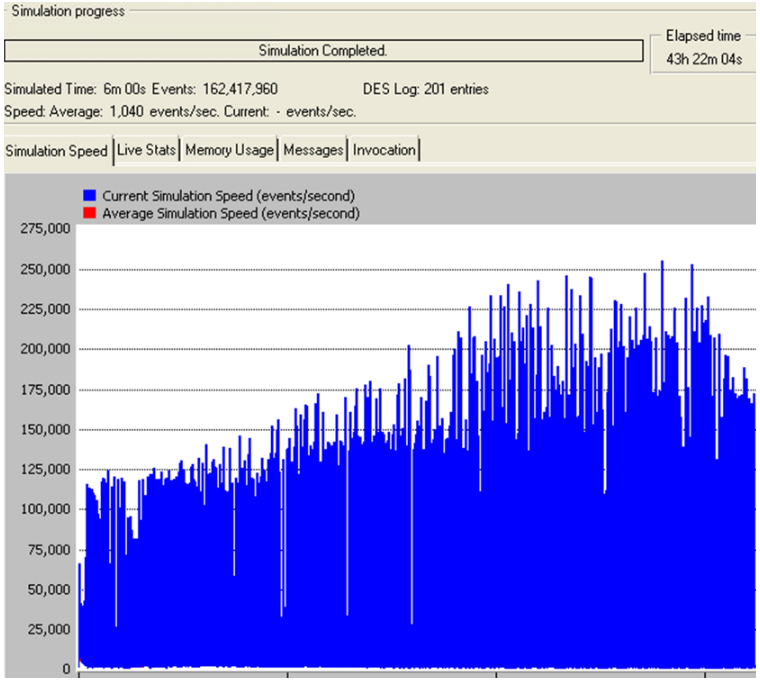
☑ Simple Network Estimation
Bandwidth (Kbps): 3,000,000
Latency (ms): 10.0

Figure 17: Simulation Traffic Setup Using ACE-White Board

present modeling of register, invite, and messaging functions in this figure. Furthermore, an ensured traffic rate is also guaranteed by the EPS bearer in the discrete simulation engine. A Guaranteed Bit-Rate (GBR) of 10 Mbps at uplink and downlink will always offer a minimum traffic rate of 10 Mbps between client and higher layer of EPS during the simulation run. The overall traffic distribution among the IM application instances in LTE is kept exponential in the setup.

4.5 Experiments on Mobile Network

Figure 18 presents an user interface of the OPNET simulation process during an experiment run. The graph provides an approximation of the events generated per second (speed) along with the other information of “Elapse Time”, “Simulated Time”, “Number of Total Events” etc. It is designed such that the event generation at each node of our designed network follows an exponential distribution of traffic generation during the allotted simulation time.



Simulation progress

Simulation Completed.

Elapsed time
43h 22m 04s

Simulated Time: 6m 00s Events: 162,417,960 DES Log: 201 entries

Speed: Average: 1,040 events/sec. Current: - events/sec.

Figure 18: Example of OPNET Visualization for Simulation Events

More than 162 million events are generated with an average speed of 1040 events/second in our experiment.

4.5.1 Simulation Results

Figure 19 depicts a comparative study of average TCP delay on Instant Messaging server with and without use of SPIM detection module. These tests are performed while simulating an overall exponential traffic flow in the LTE network. The average TCP delay represents an average time for packets received by the TCP layers in the complete network for all connections. It is measured as the time difference between data packet sent from the source TCP layer to the time it is received by the TCP layer at the destination node. The red line denotes the TCP delay (in seconds) occurred without applying any spam detection module during the simulation run. The blue and green lines refer to the TCP delay for SPIM detector using Naive Bayes and Support Vector Machine, algorithms respectively. Though NB and SVM follow the red line, it is clear that both SVM and NB incur high TCP delay according to our traffic configuration.

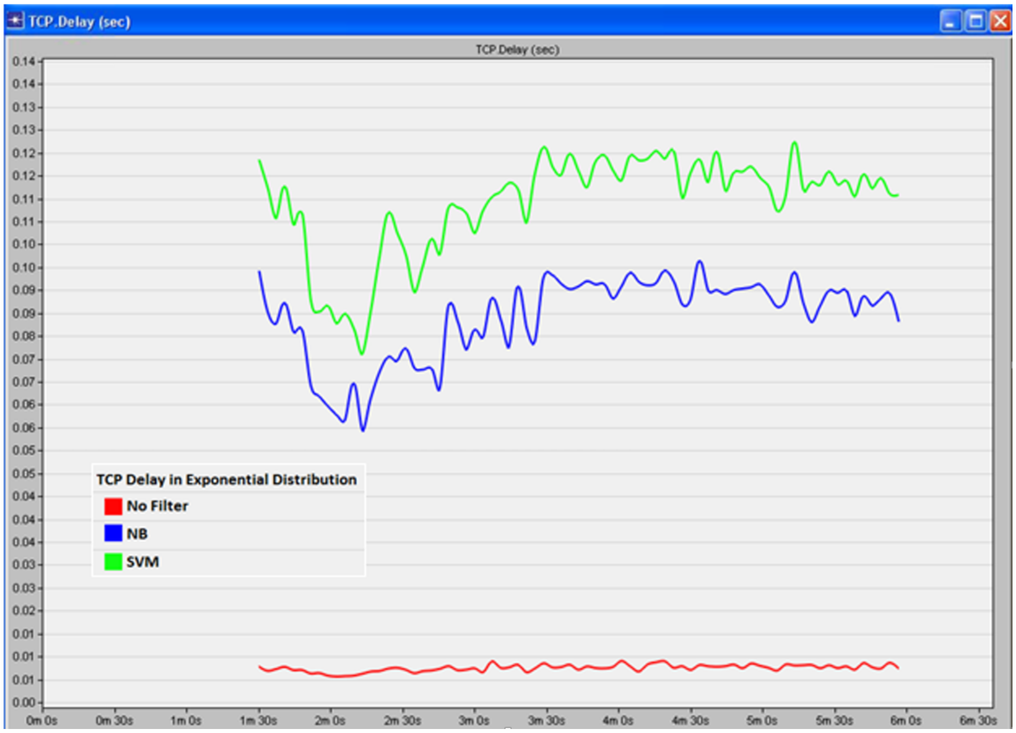


Figure 19: Comparison - Testing Time for NB and SVM Algorithms

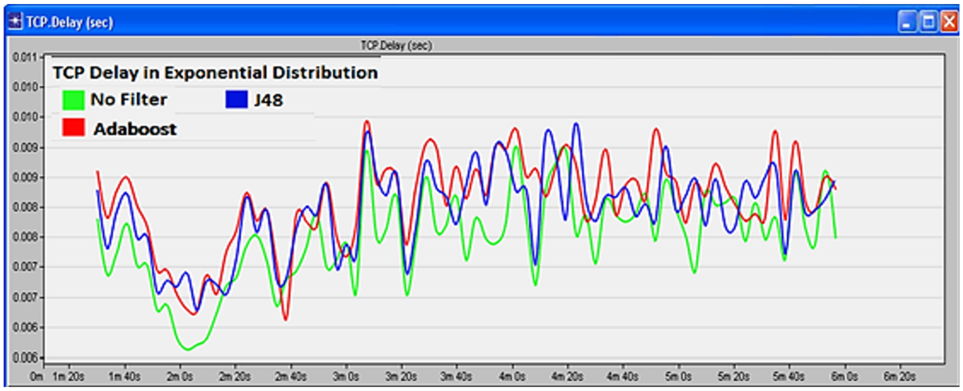


Figure 20: Comparison - Testing Time for J48 and AdaBoost Algorithms

Figure 20 demonstrates another comparison of average TCP delay while using other two algorithms, namely J48 and AdaBoost, with respect to no spam detection filter. The figure clearly depicts that both algorithms perform fairly well with very low TCP delay in compare to SVM and NB algorithms. In the next section, we analyze our experience to verify the test performance while searching values of other key execution parameters in using J48 and AdaBoost algorithms within the SPIM detector.

4.5.2 Analysis of Results

Figure 19 and Figure 20 indicate similar characteristics to our theoretical evaluation of the classification time in Section 3. It confirms that both J48 and AdaBoost algorithms in SPIM detector have lower SPIM classification time than the other two algorithms. Consequently, we wanted to verify classification accuracy and the training time using these two algorithms. However, the scope of verifying classification accuracy is limited in the simulation environment due to lack of pre-labeled SPIM dataset. From our previous experiments on other dataset (Figure 7) in Chapter 3, it is clear that J48 has higher classification accuracy than AdaBoost (more than 88% classification accuracy) for smaller packet size while the latter possesses shorter training time than J48. However, given the scope and purpose of our SPIM detection, we do not exclude the possibility of using boosting based detection technique as it incurs vary small delay in classification and offers better training time for continuous learning over updated SPIM dataset. AdaBoost classification accuracy is also fairly high although it is least among our selected algorithms.

In order to further analyze the suitability of our experimental setup, we perform verification of few key parameters during the execution. Figure 21 represents the average TCP load (packets/sec) on Instant Messaging Server with and without a SPIM detection mechanism. We test two algorithms: J48 and AdaBoost in the SPIM detector. TCP load represents an average rate of traffic placed at the TCP layer by the application layer in the IM server node, for all connections. Figure 21 indicates that the J48 algorithm creates slightly higher TCP load on the IM server in compare to AdaBoost algorithm during the period of simulation. The preparation and the initialization of the experimental setup take almost 90 seconds to

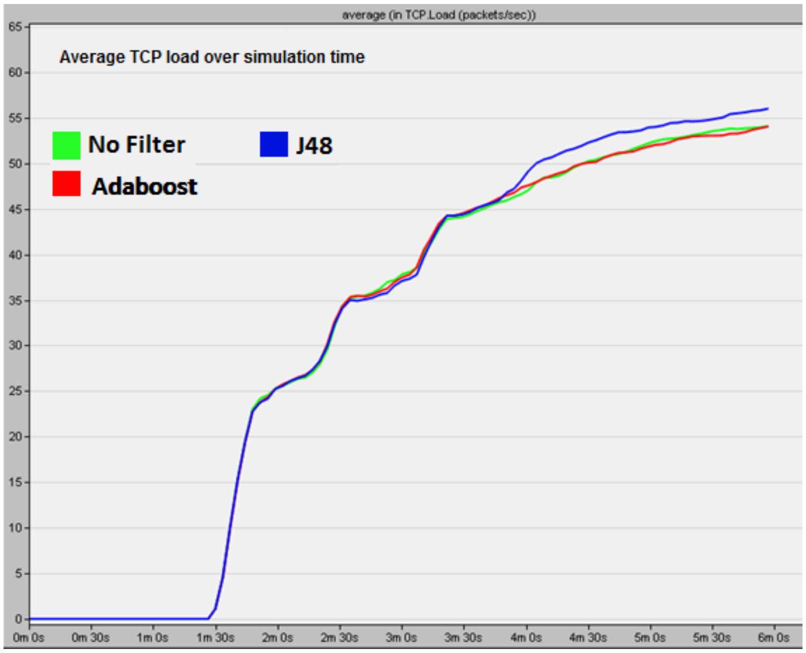


Figure 21: Load Comparison on IM Server over deploying SPIM Using J48 and AdaBoost

initiate the load on the server. Huge flow of traffic requests can be noticed at the IM server at the beginning. However, while the events are continuously generated in an exponential distribution the load get steady after some time.

Figure 22 demonstrates another experiment on the global average response time for the IM applications in our network. This is the average time taken to execute all the tasks in a custom application to complete in every application run. The three different curves denotes response time for two different algorithms (J48 and AdaBoost) with respect to no detection module. The graph shows a sudden pick at the beginning of the simulation which represents larger delay in initial response due to primary initialization at various nodes in the telecommunication networks. Once the initialization is complete, the delay sharply decreases and get steady over simulation time.

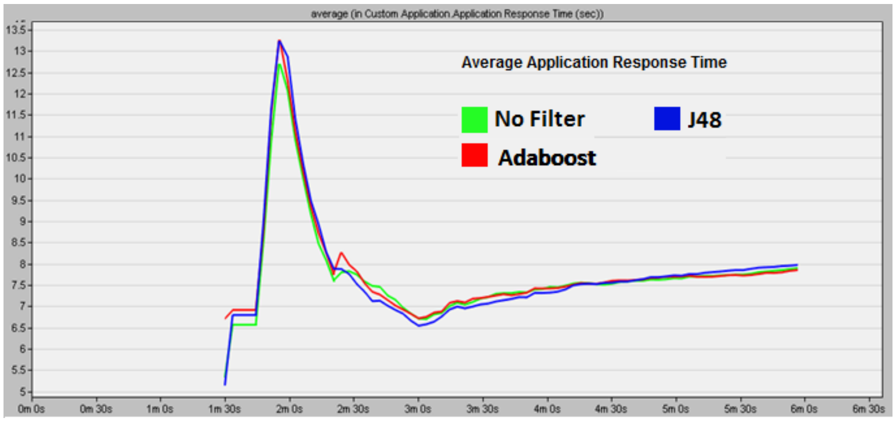


Figure 22: Comparison - Global Application Response Time on Implementing J48 and AdaBoost Algorithms

4.6 Summary

In this chapter, we elaborated our findings on running the proposed SPIM detection module within a simulated LTE architecture designed in OPNET. We configured six core activities of simple peer-to-peer instant messaging and presence services. This step-by-step configuration of IM data flow offers necessary means to run the test on top of a mobile network inside the simulation environment. Thus, we also elaborated the whole simulation setup of the LTE/IMS network and offered a high-level guideline to design such a setup. We tested selected machine-learning algorithms in the SPIM detection module, situated at the PDN-Gateway inside LTE-IMS network. The results of these experiments has been compared to choose suitable algorithms for classifying SPIM from large volume of communication traffic. Our results demonstrate the suitability of specific machine learning techniques over others in terms of classification time. The analysis of the result suggests that the use of tree-based algorithms (J48 in particular) and boosting algorithms (AdaBoost, in particular) help in faster detection of SPIM without significantly impacting the delay and server load for packet data communication.

Chapter 5

Conclusion

It is evident that the enormous potential of the palm-sized cellular devices is generating rapid growth in the telecommunication markets. Their usability, portability and affordable cost are always identified as the prime factors to reach remote end-users around the globe [34, 102]. In this flourishing of advanced mobile telecommunication, tracking spam from real-time message exchange is essential to offer secure messaging services with efficient use of network resources. However, along with the intense growth, the communication technologies are exhibiting crucial limitations in guaranteeing spam-free communication environment. Amid these enormous challenges, we investigated to explore for an appropriate machine-learning based technique in order to detect spam from near real-time IM traffic having short classification time and consistent accuracy inside the current telecommunications infrastructure standard, namely LTE.

In this dissertation, we introduced an overview of the concept, mentioned the potential dangers of spam message exchange in instant messaging and discussed the underlying difficulties in detection. We analyzed an important research problem of message spam detection from a large volume data traffic within the whole scope of our research. We proposed a suitable SPIM detection module in PDN-Gateway of the LTE network, elaborated a detection approach and evaluated a number of machine learning algorithms through a simple and efficient implementation inside the module. Furthermore, we designed a proof-of-concept

LTE/IMS framework setup along with proposed detection module and documented our experimental findings by simulating small-scale scenarios using OPNET simulator. The major contributions of this thesis include finding of an appropriate location for SPIM detection module in LTE, evaluation of various machine learning algorithms for SPIM classification and performance verification for the proposed setup in a simulated environment.

The extracted knowledge from this research and development is important in designing an effective decision support system to manage LTE instant messaging traffic flow among the end-users. Although the scope of our evaluation is limited to instant messaging, the procedure, as discussed in this thesis, is generic enough to be used on any similar online packet traffic with necessary alterations. In our experiments, we found AdaBoost and J48 algorithms are more suitable than others in detecting SPIM using machine learning. However, given the accuracy of AdaBoost and J48 algorithms in SPIM detection and its possibility of detecting false positives, we recommend that fully correct SPIM detection and removal thereof will be inappropriate using this algorithm. We suspect that it may remove few legitimate packets due to false positive alerts. It is also possible that the removal of packets in the lower communication layer may trigger resending of the same packets by the sender that counter intuitively will increase network load. A more appropriate workaround will be to control effective bandwidth for the SPIM senders and the receivers by exploiting this acquired knowledge. Thus, the ISPs can significantly reduce the Internet bandwidth and the speed if a sender is understood as a spammer.

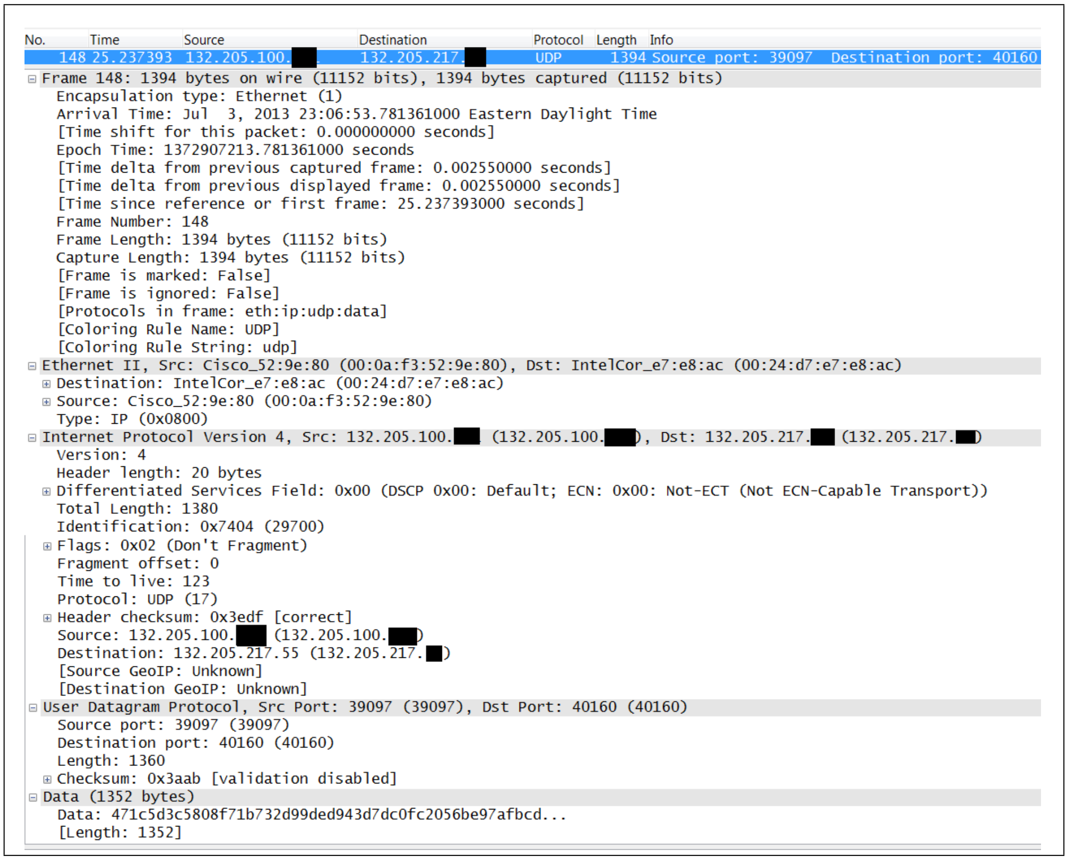
During the period of researching the subject, we faced difficulties in finding appropriately labeled (pre-classified) datasets (with SPIM information) of online IM traffic. To the best of our knowledge, the availability of IM packet data is very limited in the free Internet domain. We also understood that the fitting of this data into an off-the-shelf simulator is also challenging. Therefore, we decided to create a simulation framework to explain the effectiveness of the proposed machine learning algorithms. In case of availability of such a dataset, a proper implementation framework could be developed to perform online SPIM detection using machine learning algorithms on various traffic related attributes. A number of network traffic and flow features have been discussed in Section 3.6. An actual implementation of the packet traffic flow management for instant messaging traffic in LTE is expected to be complex and associated to a number of technical and legal implications. We consciously kept such discussion out of the scope of our thesis.

5.1 Future Work

In this thesis, we laid the foundation of a novel procedure for SPIM detection. In future, we intend to extend this study in two different directions. We wish to test the SPIM classification accuracy and time in presence of combined data traffic of email, voice, video, etc. A possible way of differentiating SPIM traffic can be achieved by identifying different packet types based on their characteristics [84]. For example, signal packets and voice packets are usually smaller in compare to email packets. Each IM packet follows a different protocol and tries to connect some preferred ports. Differences in packet size along with protocol, session, flow and traffic related information will allow to differentiate specific application traffic from others in the incoming packet stream. We briefly worked in Wireshark [35] software application that offers an in-build set of rules to differentiate various packet traffic. Secondly, we like to verify the suitability of SPIM detection algorithms on various speed of incoming packet stream and messaging event distributions (e.g. Poisson distribution, uniform distribution, etc.) to conclude over a particular machine learning technique. A majority of these events reflects user activities in real LTE network. The results can also be compared with other existing techniques apart from machine learning such as rule-based spam detection [28].

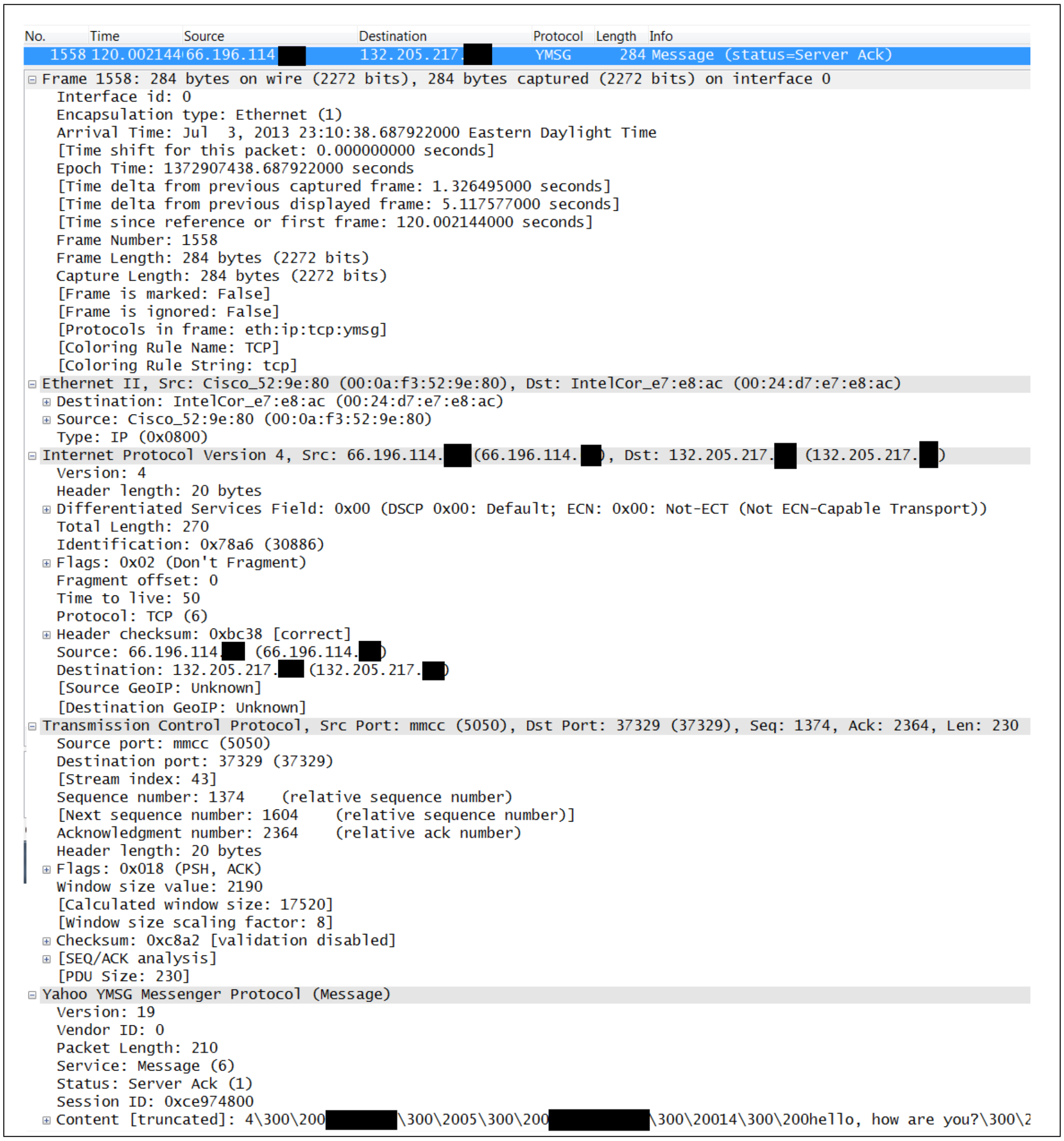
A thorough understanding of the real-time spam detection requirements and techniques for modern telecommunication networks will certainly help to secure national and international communication infrastructures against SPIM and SPIM-based cyber-attacks. We expect that our research and development work will encourage ISPs and other organizations to initiate further research. It will also facilitate large-scale, real-life implementation of such message spam detection system.

Appendix



```
No.     Time        Source                Destination           Protocol Length Info
   148 25.237393  132.205.100.[redacted]    132.205.217.[redacted]     UDP       1394 Source port: 39097  Destination port: 40160
⊟ Frame 148: 1394 bytes on wire (11152 bits), 1394 bytes captured (11152 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Jul  3, 2013 23:06:53.781361000 Eastern Daylight Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1372907213.781361000 seconds
    [Time delta from previous captured frame: 0.002550000 seconds]
    [Time delta from previous displayed frame: 0.002550000 seconds]
    [Time since reference or first frame: 25.237393000 seconds]
    Frame Number: 148
    Frame Length: 1394 bytes (11152 bits)
    Capture Length: 1394 bytes (11152 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:udp:data]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
⊟ Ethernet II, Src: Cisco_52:9e:80 (00:0a:f3:52:9e:80), Dst: IntelCor_e7:e8:ac (00:24:d7:e7:e8:ac)
  ⊞ Destination: IntelCor_e7:e8:ac (00:24:d7:e7:e8:ac)
  ⊞ Source: Cisco_52:9e:80 (00:0a:f3:52:9e:80)
    Type: IP (0x0800)
⊟ Internet Protocol Version 4, Src: 132.205.100.[redacted] (132.205.100.[redacted]), Dst: 132.205.217.[redacted] (132.205.217.[redacted])
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 1380
    Identification: 0x7404 (29700)
  ⊞ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 123
    Protocol: UDP (17)
  ⊞ Header checksum: 0x3edf [correct]
    Source: 132.205.100.[redacted] (132.205.100.[redacted])
    Destination: 132.205.217.55 (132.205.217.[redacted])
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
⊟ User Datagram Protocol, Src Port: 39097 (39097), Dst Port: 40160 (40160)
    Source port: 39097 (39097)
    Destination port: 40160 (40160)
    Length: 1360
  ⊞ Checksum: 0x3aab [validation disabled]
⊟ Data (1352 bytes)
    Data: 471c5d3c5808f71b732d99ded943d7dc0fc2056be97afbcd...
    [Length: 1352]
```

Figure 23: Packet Excerpt from Skype Insiant Messenger Collected Using Wireshark



```
No.     Time        Source               Destination          Protocol Length Info
   1558 120.002144(66.196.114[redacted]   132.205.217.[redacted]  YMSG       284 Message (status=Server Ack)
⊟ Frame 1558: 284 bytes on wire (2272 bits), 284 bytes captured (2272 bits) on interface 0
    Interface id: 0
    Encapsulation type: Ethernet (1)
    Arrival Time: Jul  3, 2013 23:10:38.687922000 Eastern Daylight Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1372907438.687922000 seconds
    [Time delta from previous captured frame: 1.326495000 seconds]
    [Time delta from previous displayed frame: 5.117577000 seconds]
    [Time since reference or first frame: 120.002144000 seconds]
    Frame Number: 1558
    Frame Length: 284 bytes (2272 bits)
    Capture Length: 284 bytes (2272 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:tcp:ymsg]
    [Coloring Rule Name: TCP]
    [Coloring Rule String: tcp]
⊟ Ethernet II, Src: Cisco_52:9e:80 (00:0a:f3:52:9e:80), Dst: IntelCor_e7:e8:ac (00:24:d7:e7:e8:ac)
  ⊞ Destination: IntelCor_e7:e8:ac (00:24:d7:e7:e8:ac)
  ⊞ Source: Cisco_52:9e:80 (00:0a:f3:52:9e:80)
    Type: IP (0x0800)
⊟ Internet Protocol Version 4, Src: 66.196.114.[redacted](66.196.114.[redacted]), Dst: 132.205.217.[redacted] (132.205.217.[redacted])
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 270
    Identification: 0x78a6 (30886)
  ⊞ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 50
    Protocol: TCP (6)
  ⊞ Header checksum: 0xbc38 [correct]
    Source: 66.196.114[redacted] (66.196.114.[redacted])
    Destination: 132.205.217.[redacted] (132.205.217.[redacted])
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
⊟ Transmission Control Protocol, Src Port: mmcc (5050), Dst Port: 37329 (37329), Seq: 1374, Ack: 2364, Len: 230
    Source port: mmcc (5050)
    Destination port: 37329 (37329)
    [Stream index: 43]
    Sequence number: 1374    (relative sequence number)
    [Next sequence number: 1604    (relative sequence number)]
    Acknowledgment number: 2364    (relative ack number)
    Header length: 20 bytes
  ⊞ Flags: 0x018 (PSH, ACK)
    Window size value: 2190
    [Calculated window size: 17520]
    [Window size scaling factor: 8]
  ⊞ Checksum: 0xc8a2 [validation disabled]
  ⊞ [SEQ/ACK analysis]
    [PDU Size: 230]
⊟ Yahoo YMSG Messenger Protocol (Message)
    Version: 19
    Vendor ID: 0
    Packet Length: 210
    Service: Message (6)
    Status: Server Ack (1)
    Session ID: 0xce974800
  ⊞ Content [truncated]: 4\300\200[redacted]\300\2005\300\200[redacted]\300\20014\300\200hello, how are you?\300\2
```

Figure 24: Packet Excerpt from Yahoo Messenger Collected Using Wireshark

Bibliography

[1] Bill C-28: Canada's Anti-Spam Legislation. http://www.ic.gc.ca/eic/site/ecic-ceac.nsf/eng/h_gv00567.html. Accessed: 01/02/2013.

[2] Coda Recent Reports. http://www.codaresearch.co.uk/reports.htm. Accessed: 01/02/2013.

[3] Email Statistics Report, 2011-2015. http://www.radicati.com/wp/wp-content/uploads/2011/05/Email-Statistics-Report-2011-2015-Executive-Summary.pdf. Accessed: 01/02/2013.

[4] Giz Explains: Whats the Difference Between GSM and CDMA? http://gizmodo.com/5637136/giz-explains-gsm-vs-cdma. Accessed: 01/02/2013.

[5] Global Mobile Information Systems Simulation. http://pcl.cs.ucla.edu/projects/glomosim. Accessed: 01/02/2013.

[6] Jabber.org. http://www.jabber.org/. Accessed: 01/02/2013.

[7] LTE Deployments and Commitments. http://ltemaps.org/. Accessed: 01/02/2013.

[8] Manually Detecting Maximum Transmission Unit. online, http://ampledata.org/manually_detecting_maximum_transmission_unit.html. Accessed: 01/02/2013.

[9] NetSim Network Simulator. http://www.boson.com/netsim-cisco-network-simulator. Accessed: 01/02/2013.

[10] Network Simulation. http://www.opnet.com/solutions/network_rd/modeler.html. Accessed: 01/02/2013.

[11] Network Simulator. http://www.isi.edu/nsnam/ns. Accessed: 01/02/2013.

[12] QualNet Simulation. http://web.scalable-networks.com/content/qualnet. Accessed: 01/02/2013.

[13] SMS Spam Collection V.1. http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/. Accessed: 01/02/2013.

[14] Spam (electronic). http://en.wikipedia.org/wiki/Spam_(electronic). Accessed: 01/02/2013.

[15] TETCOS. http://www.tetcos.com. Accessed: 01/02/2013.

[16] Weka 3: Data Mining Software in Java. http://www.cs.waikato.ac.nz/ml/weka/. Accessed: 01/02/2013.

[17] WiMAX. http://en.wikipedia.org/wiki/WiMax. Accessed: 01/02/2013.

[18] XMPP Technologies: Jingle. http://xmpp.org/about-xmpp/technology-overview/jingle/. Accessed: 01/02/2013.

[19] Protect Your Business from Instant Messaging Threats. online, http://www.symantec.com/es/es/library/article.jsp?aid=instant_messaging_threats, July 2006.

[20] About Detecting SPIM. online, http://www.symantec.com/business/support/index?page=content&id=HOWTO54058, June 2011.

[21] Shivani Agarwal. Ranking Methods in Machine Learning: A Tutorial Introduction. online, http://drona.csa.iisc.ernet.in/~shivani/Events/SDM-10-Tutorial/sdm10-tutorial.pdf, May 2010.

[22] Hasan Shojaa Alkahtani, Paul Gardener-Stephen, and Robert Goodwin. A Taxonomy of Email Spam Filters. In *ACIT*, 2011.

[23] Dmitri Alperovitch, Paul Judge, and Sven Krasser. Taxonomy of Email Reputation Systems. In *Proceedings of the 27th International Conference on Distributed Computing Systems Workshops*, ICDCSW '07, 2007.

[24] Aston Blake. Smart Phones: How do They Affect Us Really. `http://www.slideshare.net/ashtonblake/smart-phones-how-do-they-affect-us-really`.

[25] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When Is "Nearest Neighbor" Meaningful? In *In Int. Conf. on Database Theory*, pages 217–235, 1999.

[26] Enrico Blanzieri and Anton Bryl. A Survey of Learning-Based Techniques of Email Spam Filtering. *Artif. Intell. Rev.*, 29(1):63–92, 2008.

[27] Dario Bonfiglio, Marco Mellia, Michela Meo, Dario Rossi, and Paolo Tofanelli. Revealing Skype Traffic: When Randomness Plays with You. *SIGCOMM Comput. Commun. Rev.*, 37(4):37–48, October 2007.

[28] Elias Bou-Harb, Mourad Debbabi, and Chadi Assi. A First Look on the Effects and Mitigation of VoIP SPIT Flooding in 4G Mobile Networks. In *IEEE-ICC*, 2012.

[29] Andrea Buonerba. Skype Traffic Detection and Characterization. Master's thesis, HELSINKI UNIVERSITY OF TECHNOLOGY, September 2007.

[30] G. Camarillo and M.A. García-Martín. *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds.* Wiley, 2007.

[31] Gonzalo Camarillo and Miguel A. Garcia-Martin. *The 3G IP Multimedia Subsystem IMS - Merging the Internet and the Cellular Worlds (2. ed.).* Wiley, 2006.

[32] B. Campbell, J. Rosenberg, H. Schulzrinne, and C. Huitema. Session Initiation Protocol (SIP) Extension for Instant Messaging. Technical report, IETF: Request for Comments: 3428, 2002.

[33] Sujata Chavan. Understanding Instant Messaging (IM) and Its Security Risks. SANS Institute, August 2003.

[34] Thomas Claburn. SPIM, Like Spam, Is on the Rise. Information week, March 2004.

[35] Gerald Combs. Wireshark. online, `http://www.wireshark.org/`.

[36] Gordon V. Cormack, José María Gómez Hidalgo, and Enrique Puertas Sanz. Feature engineering for mobile (SMS) spam filtering. In *SIGIR*, pages 871–872, 2007.

[37] Gordon V. Cormack, José María Gómez Hidalgo, and Enrique Puertas Sanz. Spam filtering for short messages. In *CIKM*, pages 313–320, 2007.

[38] Lingling Cui, Sharanya Eswaran, Wei Hu, and Xinyu Liu. Secure Instant Messaging. Project Report, 2006. `www.cs.virginia.edu/~wh5a/personal/psi.doc`.

[39] Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati. P2P-Based Collaborative Spam Detection and Filtering. *IEEE International Conference on Peer-to-Peer Computing*, pages 176–183, 2004.

[40] J. N. Darroch and D. Ratcliff. Generalized Iterative Scaling for Log-linear Models. In *The Annals of Mathematical Statistics*, volume 43, pages 1470–1480, 1972.

[41] Swagata Das, Mourad Debbabi, and Makan Pourzandi. SPIM Detection in LTE Networks. In *25th Canadian Conference on Electrical and Computer Engineering*, May 2012.

[42] M. Debbabi and M. Rahman. The War of Presence and Instant Messaging: Right Protocols and APIs. In *IEEE Consumer Communications and Networking Conference*, pages 341–346, USA, January 2004. IEEE Press.

[43] Dialogic White Papers. The Architecture and Benefits of IMS. online, `http://www.dialogic.com/~/media/products/docs/whitepapers/11297-ims-arch-benefits-wp.pdf`, 2009.

[44] James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and Unsupervised Discretization of Continuous Features. In *Machine Learning: Proceedings of the Twelfth International Conference*, pages 194–202. Morgan Kaufmann, 1995.

[45] Mark Dredze. Machine Learning Finding Patterns in the World. online, `http://www.docstoc.com/docs/108806134/Machine-Learning-Tutorial`, 2009.

[46] Sinem Coleri Ergen. ZigBee/IEEE 802.15.4 Summary. online, `http://pages.cs.wisc.edu/~suman/courses/838/papers/zigbee.pdf`, September 2004.

[47] Eventhelix. Presence IMS Feature Successful Subscription (IMS Presence Subscription, Publication and Notification). `http://www.eventhelix.com/ims/presence/ims-presence-subscribe-notify-flow.pdf`. Accessed: 01/02/2013.

[48] Paul Festa. Spammers target IM accounts. online, `http://news.cnet.com/2100-1023-857637.html`, March 2002.

[49] L. Firte, C. Lemnaru, and R. Potolea. Spam Detection Filter Using KNN Algorithm and Re-sampling. In *Intelligent Computer Communication and Processing (ICCP), 2010 IEEE International Conference on*, pages 27–33, August 2010.

[50] Yoav Freund and Robert E. Schapire. A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, EuroCOLT '95, pages 23–37. Springer-Verlag, 1995.

[51] Steven Gianvecchio, Mengjun Xie, Zhenyu Wu, and Haining Wang. Measurement and Classification of Humans and Bots in Internet Chat. In *Proceedings of the 17th conference on Security symposium*, pages 155–169. USENIX Association, 2008.

[52] J. Goodman. Sequential Conditional Generalized Iterative Scaling. [online] `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.3035`, 2002.

[53] Dimitris Gritzalis and Yannis Mallios. A SIP-oriented SPIT Management Framework. *Computers & Security*, 27(5-6):136–153, 2008.

[54] Anglica Garca Gutirrez. Instant Messaging and Presence Services: Analysis of the Standards and Example implementation. Master's thesis, Technische Universitt Hamburg-Harburg, 2004.

[55] Zoltn Gyngyi and Hector Garcia-Molina. Web Spam Taxonomy. In *AIRWeb'05*, pages 39–47, 2005.

[56] Shuang Hao, Nadeem Ahmed Syed, Nick Feamster, Alexander G. Gray, and Sven Krasser. Detecting spammers with SNARE: spatio-temporal network-level automatic reputation engine. In *Proceedings of the 18th conference on USENIX security symposium*, SSYM'09, pages 101–118. USENIX Association, 2009.

[57] Neal Hindocha and Eric Chien. Malicious Threats and Vulnerabilities in Instant Messaging. White paper, Symantec Security Response, Symantec Corporation, September 2003.

[58] Matt Holliday. Facebook Chat Surpasses 1 Billion Messages Sent Per Day. `http://www.insidefacebook.com/2009/06/17/facebook-chat-surpasses-1-billion-messages-per-day/`.

[59] E. B. Hunt. *Concept Learning: an Information Processing problem.* Wiley, 1962.

[60] International Telecommunication Union. ITU Study on the Financial Aspects of Network Security: Malware and Spam. `http://www.itu.int/ITU-D/cyb/cybersecurity/docs/itu-study-financial-aspects-of-malware-and-spam.pdf`, 2008.

[61] C. Jennings, R. Mahy, and A. B. Roach. Relay Extensions for the Message Session Relay Protocol (MSRP). Technical report, IETF: Request for Comments: 4976, 2007.

[62] Thorsten Joachims. Training Linear SVM in Linear Time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 217–226, New York, NY, USA, 2006. ACM.

[63] KI Lakhtaria. Study and Modeling Instant Messaging and Presence over IMS. online, `http://shodhganga.inflibnet.ac.in/bitstream/10603/734/12/12_chapter7a.pdf`, 2010.

[64] Maciej Korczynski. *Classifying Application Flows and Intrusion Detection in Internet Traffic.* PhD thesis, UNIVERSIT DE GRENOBLE, November 2012.

[65] Madhuri Kulkarni. 4G Wireless and International Mobile Telecommunication (IMT) Advanced. online, `http://www.cse.wustl.edu/~jain/cse574-08/ftp/imta.pdf`, April 2008.

[66] Abdelkader Lahmadi and Olivier Festor. SecSip: A Stateful Firewall for SIP-based Networks. *CoRR*, abs/0907.3045, 2009.

[67] Wei Li, Marco Canini, Andrew W. Moore, and Raffaele Bolla. Efficient application identification and the temporal and spatial stability of classification schema. *Comput. Netw.*, 53:790–809, April 2009.

[68] Zhijun Liu, Weili Lin, Na Li, and David Lee. Detecting and filtering instant messaging spam: a global and personalized approach. In *Proceedings of the First international conference on Secure network protocols*, NPSEC'05, pages 19–24, Washington, DC, USA, 2005. IEEE Computer Society.

[69] M. Day and J. Rosenberg and H. Sugano. A model for presence and instant messaging. online, `http://tools.ietf.org/html/rfc2778`, 2000.

[70] M. Mannan and P.C. van Oorschot. Secure Public Instant Messaging: A Survey. *Proceedings of Privacy, Security and Trust*, 2004.

[71] Mohammad Mannan and Paul C. van Oorschot. On Instant Messaging Worms, Analysis and Countermeasures. In *WORM*, pages 2–11, 2005.

[72] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval.* Cambridge University Press, 2008.

[73] U. Maroof. Analysis and detection of SPIM using message statistics. In *2010 6th International Conference on Emerging Technologies (ICET)*, pages 246–249, 2010.

[74] Joseph Menn. N.Y. Man Arrested Over Instant-Message Spam. Los Angeles Times, February 2005.

[75] Messaging Anti-Abuse Working Group. Email Metrics Program: The Network Operators Perspective. http://www.maawg.org/sites/maawg/files/news/MAAWG_2010-Q1Q2_Metrics_Report_13.pdf. Accessed: 01/02/2013.

[76] Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. Spam Filtering with Naive Bayes - Which Naive Bayes? In *CEAS*, 2006.

[77] Motorola. Long Term Evolution (LTE): A Technical Overview. online, 2007.

[78] Ryuei Nishii and Shinto Eguchi. Supervised Image Classification by Contextual AdaBoost Based on Posteriors in Neighborhoods. *IEEE T. Geoscience and Remote Sensing*, 43(11):2547–2554, 2005.

[79] Open Mobile Alliance. Instant Messaging Using SIMPLE Candidate Version 1.0. Release candidate, Open Mobile Alliance Ltd., May 2011.

[80] Tu Ouyang, Soumya Ray, Michael Rabinovich, and Mark Allman. Can network characteristics detect spam effectively in a stand-alone enterprise? In *Proceedings of the 12th international conference on Passive and active measurement*, PAM'11, pages 92–101, Berlin, Heidelberg, 2011. Springer-Verlag.

[81] Yongsuk Park and Taejoon Park. A Survey of Security Threats on 4G Networks. In *Globecom Workshops, 2007 IEEE*, pages 1–6, nov. 2007.

[82] Roland Parviainen and Peter Parnes. Mobile instant messaging, 2003.

[83] John C. Platt. *Advances in Kernel Mmethods*, chapter Fast training of support vector machines using sequential minimal optimization, pages 185–208. MIT Press, 1999.

[84] QoS: Classification Configuration Guide, Cisco IOS XE Release 3S. Classifying Network Traffic. online, http://www.cisco.com/en/US/docs/ios-xml/ios/qos_classn/configuration/xe-3s/qos-classn-ntwk-trfc.html.

[85] J. R. Quinlan. Induction of Decision Trees. *Mach. Learn.*, 1(1):81–106, March 1986.

[86] J. R. Quinlan, P. J. Compton, K. A. Horn, and L. Lazarus. Inductive Knowledge Acquisition: a Case Study. In Ross J. Quinlan, editor, *Applications of Expert Systems*, chapter 9, pages 157–73. Addison-Wesley, 1987.

[87] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[88] Santa Rahman, Nahid Hossain, Nizam Sayeed, and M.L. Palash. Comparative Study Between Wireless Regional Area Network (IEEE Standard 802.22) and WiMAX and Coverage Planning of a Wireless Regional Area Network Using Cognitive Radio Technology. *International Journal of Recent Technology and Engineering (IJRTE)*, 1(6):161–163, January 2013.

[89] Research In Motion Limited. BlackBerry Enterprise Server for Microsoft Exchange. online, `http://docs.blackberry.com/en/admin/deliverables/16574/BlackBerry_Enterprise_Server_for_Microsoft_Exchange-Feature_and_Technical_Overview-T305802-1108946-0615123042-001-5.0.2-US.pdf`, 2010.

[90] Steve Roche. *Protect Your Children from Internet and Mobile Phone Dangers*. Sparkwave, 2004.

[91] Martin Roesch. Snort: Lightweight Intrusion Detection for Networks. In *LISA*, pages 229–238. USENIX, 1999.

[92] J. Rosenberg. Presence Authorization Rules. Technical report, IETF: Request for Comments: 5025, 2007.

[93] J. Rosenberg. The Extensible Markup Language (XML) Configuration Access Protocol (XCAP). Technical report, IETF: Request for Comments: 4825, 2007.

[94] Sourabh Satish. Automatic spim detection. *US-Patent*, 2005.

[95] H. Schulzrinne, H. Tschofenig, J. Morris, J. Cuellar, J. Polk, and J. Rosenberg. Common Policy: A Document Format for Expressing Privacy Preferences. Technical report, IETF: Request for Comments: 4745, 2007.

[96] Stefania Sesia, Issam Toufik, and Matthew Baker. *LTE, The UMTS Long Term Evolution: From Theory to Practice*. Wiley Publishing, 2009.

[97] Asaf Shabtai, Uri Kanonov, Yuval Elovici, Chanan Glezer, and Yael Weiss. "Andromaly": a behavioral malware detection framework for android devices. *Journal of Intelligent Information Systems*, 38(1):161–190, 2012.

[98] Simon Znaty and Jean-Louis Dauphin. IP Multimedia Subsystem : Principles and Architecture. online, `http://www.efort.com/media_pdf/IMS_ENG.pdf`, 2005.

[99] Hardeep Singh and Harish Kumar. Survey of Feature Selection Technique in Internet Traffic Data. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(3):207–210, March 2012.

[100] Kyoungwon Suh, Daniel R. Figueiredo, Jim Kurose, and Don Towsley. Characterizing and detecting skype-relayed traffic. In *In Proceedings of IEEE INFOCOM 06*, 2006.

[101] Tektronix Communications. LTE Networks: Evolution and Technology Overview. online, `http://www.tektronixcommunications.com/sites/tektronixcommunications.com/files/assets/documents/LTE-Network-Evolution-Technology-Whitepaper.pdf`, September 2010.

[102] Today's Net Threat. Instant Messaging Attacks on the Rise. online,`http://www.2010netthreat.com/netthreats/post/2010/03/23/Instant-messaging-attacks-on-the-rise.aspx`, March 2010.

[103] Ulticom. Signaling: Diameter. `http://www.ulticom.com/technologies-signaling/diameter/`. Accessed: 01/02/2013.

[104] UMTS Forum. Towards Global Mobile Broadband: Standardizing the Future of Mobile Communications with LTE (Long Term Evolution). online, February 2008.

[105] A. Wilhelm. Data and Knowledge Mining. In Y. Mori (Hrsg.) W. Hrdle, J. Gentle, editor, *Handbook of Computational Statistics*, pages 787–812. Springer Verlag, 2004.

[106] Nigel Williams, Sebastian Zander, and Grenville Armitage. A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. *SIGCOMM Comput. Commun. Rev.*, 36(5):5–16, October 2006.

[107] Verizon Wireless. LTE: The Future of Mobile Broadband Technology. White Paper, `http://opennetwork.verizonwireless.com/pdfs/VZW_LTE_White_Paper_12-10.pdf`, 2010.

[108] Ian H. Witten and Eibe Frank. Machine Learning in Real World: C4.5. online `http://www.sts.tu-harburg.de/teaching/ss-09/ml-sose-09/03-Decision-Tree-c45.pdf`, 2009.

[109] Jie Xiao, Changcheng Huang, and J. Yan. A Flow-Based Traffic Model for SIP Messages in IMS. In *IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1 –7, 2009.

[110] Yiming Yang and Jan O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML)*, pages 412–420. Morgan Kaufmann Publishers Inc., 1997.

[111] Le Zhang, Jingbo Zhu, and Tianshun Yao. An Evaluation of Statistical Spam Filtering Techniques. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3:243–269, December 2004.

[112] Shenghuo Zhu, Xiang Ji, Wei Xu, and Yihong Gong. Multi-labelled Classification Using Maximum Entropy Method. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 274–281. ACM, 2005.