UNIFIED FRAMEWORK FOR SPAM DETECTION AND RISK ASSESSMENT IN SHORT MESSAGE COMMUNICATION MEDIA

ADEWOLE KAYODE SAKARIYAH

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2018

UNIFIED FRAMEWORK FOR SPAM DETECTION AND RISK ASSESSMENT IN SHORT MESSAGE COMMUNICATION MEDIA

ADEWOLE KAYODE SAKARIYAH

THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2018

UNIVERSITY OF MALAYA
ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Adewole Kayode Sakariyah

Registration/Matric No: WHA140057

Name of Degree: Doctor of Philosophy

Title of Thesis: UNIFIED FRAMEWORK FOR SPAM DETECTION AND RISK ASSESSMENT IN SHORT MESSAGE COMMUNICATION MEDIA

Field of Study: Network Security

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date:

Subscribed and solemnly declared before,

Witness's Signature Date:

Name: Nor Badrul Anuar Bin Juma'at

Designation: Associate Professor

Witness's Signature Date:

Name: Amirrudin Kamsin

Designation: Senior Lecturer

UNIFIED FRAMEWORK FOR SPAM DETECTION AND RISK ASSESSMENT IN SHORT MESSAGE COMMUNICATION MEDIA

ABSTRACT

Short message communication media (SMCM), such as mobile and microblogging social networks, have become essential part of many people daily routine. Despite the benefits offered by these communication media, they have become the popular platforms for distributing spam contents. Research in spam message and spam account detection in SMCM has received growing interests in the recent years, mainly focusing on introducing separate frameworks that can identify spam message or spam account. There are hundreds of published works related to spam message and spam account detection that aim to identify effective detection methods. While spam message and spam account studies have recently advanced, there are still areas available to explore, mostly with respect to introduction of unified method that can detect spam message and spam account within a single framework as well as identifying risk levels of spam accounts. Existing content-based methods for spam detection degraded in performance due to many factors. For instance, unlike contents posted on social networks like Facebook and Renren, SMS and microblogging messages have limited size composed using many domain-specific words such as idioms and abbreviations. In addition, microblogging messages are unstructured and noisy. These distinguished characteristics posed challenges to existing approaches for spam message detection. The state-of-the-art solutions for spam accounts detection have faced different evasion tactics in the hands of intelligent spammers. Thus, the need to investigate features, which can be used to identify spam message and spam account in SMCM. This study is concerned with introduction of a unified framework that can detect spam message and spam account as well as assessing account risk level. To achieve this aim, this study proposed a novel

framework, which combines three models: Spam Account Detection Model (SADM), Spam Message Detection Model (SMDM), and Spam Risk Assessment Model (SRAM). Sixty-nine (69) set of features were identified from five main categories to develop the SADM. Additionally, eighteen (18) features were introduced to build the SMDM. The performance of ten (10) machine learning algorithms were evaluated to select the best classifier for both SADM and SMDM. Bio-inspired evolutionary search method was studied to identify the discriminating features for spam account detection. A model to estimate the levels of risk of spam accounts is established using Fuzzy Analytic Hierarchy Process. Four levels of risk were employed with their corresponding response strategies used to map risk levels into different types of response. To assess the performance of the proposed framework, an evaluation study with four stages was undertaken. With promising results being gathered, a proof-of-concept study was conducted using an online assessment mode to demonstrate the applicability of the proposed framework. Based on the results gathered, this study has demonstrated that the proposed framework can be used to detect spam message and spam account as well as assess the risk level of spam accounts in SMCM.

Keywords: online social network; microblog; spam account; machine learning; graph mining.

RANGKA KERJA YANG DIPERLUKAN UNTUK PENELITIAN SPAM DAN PENILAIAN RISIKO DALAM MEDIA KOMUNIKASI PESAN PELANGGAN

ABSTRAK

Media komunikasi mesej ringkas (SMCM), seperti rangkaian sosial mudah alih dan microblogging, telah menjadi sebahagian penting dari banyak orang rutin harian. Walaupun manfaat yang ditawarkan oleh media komunikasi ini, mereka telah menjadi platform popular untuk mengedarkan kandungan spam. Penyelidikan dalam mesej spam dan pengesanan akaun spam di SMCM telah menerima minat yang semakin meningkat pada tahun-tahun kebelakangan ini, terutamanya memberi tumpuan kepada memperkenalkan kerangka berasingan yang dapat mengenal pasti mesej spam atau akaun spam. Terdapat beratus-ratus karya yang diterbitkan berkaitan dengan mesej spam dan pengesanan akaun spam yang bertujuan untuk mengenal pasti kaedah pengesanan yang berkesan. Walaupun kajian spam dan akaun spam baru-baru ini telah maju, masih ada kawasan yang tersedia untuk diterokai, kebanyakannya berkenaan dengan pengenalan kaedah bersatu yang dapat mengesan mesej spam dan akaun spam dalam satu kerangka serta mengenal pasti tahap risiko akaun spam. Kaedah berasaskan kandungan sedia ada untuk pengesanan spam terdegradasi dalam prestasi kerana banyak faktor. Contohnya, tidak seperti kandungan yang dipaparkan di rangkaian sosial seperti Facebook dan Renren, mesej SMS dan microblogging mempunyai saiz terhad yang terdiri daripada banyak kata kata domain tertentu seperti idiom dan singkatan. Di samping itu, mesej microblogging tidak berstruktur dan berisik. Ciri-ciri terkenal ini menimbulkan cabaran kepada pendekatan sedia ada untuk pengesanan mesej spam. Penyelesaian yang paling canggih untuk pengesanan akaun spam telah menghadapi taktik mengelak yang berbeza di tangan spammer pintar. Oleh itu, keperluan untuk menyiasat ciri, yang boleh digunakan untuk mengenal pasti mesej spam dan akaun spam

di SMCM. Kajian ini berkenaan dengan pengenalan rangka kerja terpadu yang dapat mengesan mesej spam dan spam serta menilai tahap risiko akaun. Untuk mencapai matlamat ini, kajian ini mencadangkan rangka kerja baru, yang menggabungkan tiga model: Model Pengesanan Akaun Spam (SADM), Model Pengesanan Mesej Spam (SMDM), dan Model Penilaian Risiko Spam (SRAM). Sebanyak enam puluh sembilan (69) ciri telah dikenalpasti dari lima kategori utama untuk membangunkan SADM. Tambahan pula, lapan belas (18) ciri telah diperkenalkan untuk membina SMDM. Prestasi sepuluh (10) algoritma pembelajaran mesin dinilai untuk memilih pengelas terbaik untuk kedua-dua SADM dan SMDM. Kaedah carian evolusi yang diilhami oleh bio dikaji untuk mengenal pasti ciri mendiskriminasi pengesanan akaun spam. Model untuk menganggarkan tahap risiko akaun spam ditubuhkan menggunakan Proses Hierarki Analitik Analisis Fuzzy. Empat tahap risiko digunakan dengan strategi tindak balas yang sama yang digunakan untuk memetakan tahap risiko ke dalam pelbagai jenis tindak balas. Untuk menilai prestasi rangka kerja yang dicadangkan, kajian penilaian dengan empat peringkat dilaksanakan. Dengan hasil yang menjanjikan dikumpulkan, satu kajian bukti-konsep dijalankan menggunakan mod penilaian dalam talian untuk menunjukkan kebolehgunaan rangka kerja yang dicadangkan. Berdasarkan hasil yang dikumpulkan, kajian ini menunjukkan bahawa rangka kerja yang dicadangkan dapat digunakan untuk mengesan pesan spam dan spam serta menilai tingkat risiko akun spam di SMCM.

Kata kunci: rangkaian sosial dalam talian; microblog; akaun spam; pembelajaran mesin; perlombongan graf.

ACKNOWLEDGEMENTS

It is a well-known fact that where kindness cannot be returned, it needs to be appreciated and passed unto others. Throughout my studies, Allah has been my provider and protector; to HIM I give all praises.

The past three years of my Ph.D. programme at University of Malaya, Malaysia, have been wonderful, challenging, interesting, and indeed opportunistic to meet with great-minded people who have been the source of the successful journey.

First, I am very much grateful to my supervisor, Associate Professor Dr. Nor Badrul Anuar Jumaat, for his mentorship, supervision, and constructive comments from the inception of this study through to the completion of my thesis. The thoughts he has offered have enriched my study without which this work would not have materialized in the present form.

Second, I would like to thank my second supervisor, Dr. Amirrudin Kamsin, for his constructive comments, personal guidance, mentorship, and excellent advice throughout this study.

I would like to specially thank the entire family of Adewole, my wife, children, friends, and members of the Security Research Group at University of Malaya for their prayers, supports, and advices. May Allah continue to provide for your needs.

I would not also forget to appreciate the efforts of my colleagues at University of Ilorin, Nigeria and the management of UNILORIN for their great supports towards ensuring that this study is achievable. Your financial assistance, for which I am greatly indebted, is really appreciated.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

AHP	Analytic Hierarchy Process
ANN	Artificial Neural Networks
API	Application Programming Interface
APT	Advanced Persistent Threats
AUC-ROC	Area Under Curve Receiver Operating Characteristic
BMF	Bell Membership Function
CDF	Cumulative Distribution Function
CI	Consistency Index
CR	Consistency Ratio
DAG	Directed Acyclic Graph
DCA	Dendritic Cell Algorithm
DDOS	Distributed Denial of Service
DM	Direct Message
DOM	Document Object Model
DR	Detection Rate
EA	Evolutionary Algorithm
EC	Eigenvector Centrality
FAHP	Fuzzy Analytic Hierarchy Process
FNR	False Negative Rate
FPR	False Positive Rate
FW	Filter Wall
GD	Gradient Descent
GMF	Gaussian Membership Function
GM	Graphical Model

GMM	Gaussian Mixture Model
GUI	Graphical User Interface
HC	Hierarchical Clustering
HTML	Hypertext Markup Language
IREP	Incremental Reduced Error Pruning
KNN	K-Nearest Neighbor
MCDM	Multiple-Criteria Decision Making
MCL	Markov Clustering
ML	Machine Learning
MLP	Multilayer Perceptron
MOSN	Microblogging Online Social Networks
NIST	National Institute of Standards and Technology
NLTK	Natural Language Toolkit
OSN	Online Social Network
PCA	Principal Component Analysis
PR	PageRank
QEA	Quantum-inspired Evolutionary Algorithm
RAM	Risk Assessment Matrix
RBF	Radial Basis Function
RDBMS	Relational Database Management System
REST	Representational State Transfer
RI	Random Index
RIPPER	Repeated Incremental Pruning to Produce Error Reduction
SADM	Spam Account Detection Model
SMCM	Short Message Communication Media
SMDM	Spam Message Detection Model

SMF	Sigmoidal Membership Function
SMS	Short Message Service
SMSADM	Spam Message and Spam Account Detection Model
SQL	Structured Query Language
SRAM	Spam Risk Assessment Model
SVM	Support Vector Machine
TF-IDF	Term Frequency Inverse Document Frequency
TFN	Triangular Fuzzy Numbers
TMF	Triangular Membership Function
TNR	True Negative Rate
TPR	True Positive Rate
TRMF	Trapezoidal Membership Function
TSVM	Transductive Support Vector Machines
UML	Unified Modeling Language
URL	Uniform Resource Locator
VSM	Vector Space Model
WEKA	Waikato Environment for Knowledge Analysis
WWW	World Wide Web

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

1.1 Overview

In the past few years, short message communication media (SMCM), such as mobile and microblogging online social networks (MOSNs) have become essential part of many people daily routine. Mobile devices offer a plethora of textual communication and convenient platforms for users to perform operations, such as accessing resources on the Internet, e-banking transactions, entertainments, instant messaging and Short Message Service (SMS). The number of mobile users has dramatically increased with an estimate of over 7 billion subscriptions globally (El-Alfy & AlHasan, 2016). The common form of textual communication between mobile devices is the use of SMS, which utilizes standardized communication protocols to enable mobile phones exchange short text messages with 160 character long (Almeida et al., 2013). On the other hand, MOSNs, such as Twitter, has counted an active monthly users of over 320 million as at April 2016, posting more than 500 tweets per day (DMR, 2015; Statista, 2016). Twitter has been utilized for a range of social activities including sharing of interesting contents about past experiences, locating long-lost friends, posting photos and videos, building communities joined by families, acquaintances, and friends (Shyni et al., 2016).

MOSNs have been in existence for almost a decade. For instance, the launch of Twitter in 2006 witnessed a rise in the number of microblogging platforms (Adewole et al., 2016; Yang et al., 2013). The common characteristic of microblogging networks is that they allow users to share short messages usually called microposts or tweets with a maximum of 140 characters. These distinguished characteristics of SMS and microblogging messages forced users to introduce many domain-specific words. As a consequence, the traditional semantic analysis approach for spam detection degraded in performance (Almeida et al., 2013; Hu et al., 2013). The increasing popularity of mobile and MOSNs has attracted the spammers who utilize the platforms to spread bogus

contents (Ab Razak et al., 2016; Almeida et al., 2013; Chen et al., 2014). Despite the various benefits offered by mobile and microblogging platforms, they have become the popular media for distributing spam messages (Lee & Kim, 2014; Zainal & Jali, 2015).

Spamming is a method of spreading bulk unsolicited contents usually for the purpose of advertisements, promoting pornographic websites, fake weight loss, bogus donations, fake news, job scams, and a host of other malicious intents, which are perpetrated by spammers. The rise in spamming activities on various communication media has long been investigated. For instance, between the year 2009 and 2012, Akismet identified over 25 billion comment spams in Wordpress blogs and the proportion of email spam traffic generated in 2013 was about 69.6% (Chan et al., 2014). The problem of spam distribution on communication media has spanned beyond email and blog communication platforms. The increasing rate of mobile SMS spam messages was analyzed in Cloudmark report (Almeida et al., 2013). This report revealed that the distribution rate of mobile SMS spam varies according to regions. For instance, in the part of Asia, about 30% of mobile messages were represented by spam. An estimate of 400% increase in unique SMS spam campaigns was witnessed in the U.S during the first half of the year 2012 (Almeida et al., 2013). According to the Nexgate report in 2013, social spam has grown for almost 355% and for every seven new social media accounts created, there are at least five spammers' accounts (Nguyen, 2013). As a result, mobile and online social networks (OSNs) are becoming the target for spam distribution. Indeed, the traditional definition of spamming does not capture spam activities on OSNs. For instance, aside the use of MOSNs for spreading bulk unsolicited spam contents, spammer also creates fake profiles to mislead legitimate users. They engage in underground market services where spammers can purchase fake followers to boost their profile reputation. This illegal behavior hinders the reliance on the information generated on microblogging social network and negatively affects the

systems that utilize followers and friends' connections to predict user's influence (Khan et al., 2016).

1.2 Spam detection systems

Research in spam message and spam account detection in communication media has received growing interests in the recent years. Spam message detection systems studies the textual information posted by spammer using techniques such as natural language processing with machine learning (Balakrishnan et al., 2016; Chan et al., 2014; Martinez-Romo & Araujo, 2013). Majority of the studies in spam message detection focus on content-based analysis and treat textual contents as collection of documents where individual message is preprocessed and represented using vector space model (VSM). VSM is a widely used method for text representation. Each vector identify by VSM is described using bag of words model where a document is represented based on the words it contains, neglecting grammar and words order. Individual document can further be represented using Boolean occurrence of each word in the document or by counting the frequency of occurrence of each word (Cui, 2016; Zhang & Wang, 2009). A more sophisticated scheme using Term Frequency Inverse Document Frequency (TF-IDF) has been studied to establish the importance of a word in the document (Schütze, 2008). Bozan et al. (2015) and Zhang and Wang (2009) have proposed Bayesian model for SMS spam classification using content analysis techniques. Yoon et al. (2010) combined content analysis and challenge-response to provide hybrid model for mobile spam detection. The content-based spam filter first classifies message as spam, legitimate or unknown. The unknown message is further authenticated using a challenge-response protocol to determine if the message is sent by human or automated program. El-Alfy and AlHasan (2016) introduced a Dendritic Cell Algorithm (DCA) to improve the performance of anti-spam filters using email and SMS data. Chan et al. (2014) investigated the capability of existing spam filters in defending against an

adversarial attack in SMCM. The authors introduced a reweight method with a new rescaling function to prevent an adversarial attack on linear Support Vector Machine (SVM) classifier. Although the proposed model increases the security level of short message spam filter, its classification accuracy on untainted samples drops significantly.

Exiting studies on spam accounts detection have utilized different detection approaches (Ghosh et al., 2012; Grier et al., 2010; Lee & Kim, 2014). The first of its kind blacklist-based analysis on Twitter was investigated by Grier et al. (2010). The authors demonstrated that 8% of 25 million links shared on Twitter point users to phishing, malware, and different scams websites, which are listed on the most popular blacklists. They also found that a large proportions of accounts used for spamming on Twitter were hijacked from legitimate users. A further analysis of the clickstream data of users' activities confirmed that Twitter is a successful platform for distributing spam messages. Grier et al. (2010) investigated the effectiveness of using blacklist approach to reduce spamming activities. However, they discovered that blacklists method is slow in detecting new social threats, exposing more than 90% of legitimate users to spam risk. In addition, blacklist based approach is sometimes platform-dependent. For instance, a malicious link caught by Google Safe Browsing blacklist may go undetected by URIBL blacklist, making spam account detection filter depends on many external resources.

Ghosh et al. (2012) applied social network analysis to distribute trust values using both known spammers and legitimate accounts as initial seeds. The algorithm, Collusionrank, assigned trust and untrust values to the neighbor of the selected seeds. The value assigned to each account depicts the strength of trust and for identifying other spammers on the network. However, since the number of seeds is very limited taking into consideration the overall size of Twitter microblogging network, the initial score of

the original seeds can dilute easily. This may propagate imprecise scores to many accounts on the network, which are less efficient to rank unknown users as spammers or legitimates (Liu et al., 2015). Ahmed and Abulaish (2012) applied Markov clustering (MCL) algorithm to group a set of accounts as spam and non-spam. The MCL algorithm takes a weighted graph as input and uses random walk approach to assign probabilities to each node on the network. Based on the assigned probabilities, the algorithm is able to cluster set of accounts using Frobenius norm.

Another line of research focused on identifying features for spammer detection, which can be utilized to train machine learning algorithms. For instance, Lee and Kim (2014) proposed five name-based features from Twitter account group. The problem with this approach is the evasion of the name-based features introduced in the study. For example, spammers can break this detection method using different character combinations to generate account names that mimic the characteristics of the legitimate account. In addition, the use of underground markets that allows spammers to purchase fake information, such as tweets and followers, has further limited the capability of existing solutions that rely on the number of followers and tweets for spam account detection in social networks. Hence, it is important to investigate the different features that can be used to identify spam message and spam account.

While hundreds of published works are available related to spam message (Abu-Nimeh et al., 2011; Almeida et al., 2013; Chan et al., 2014) and spam account detection (Ahmed & Abulaish, 2012; Bhat et al., 2014; Lee et al., 2010a) in SMCM. However, there are still areas to explore particularly in the process of investigating more features or behavioral characteristics that can be useful to efficiently identify spam message and spam accounts. Previous studies have so far concentrated on developing a single framework for either spam message or spam account detection in SMCM (Alsaleh et al.,

2014; Chu et al., 2012a). However, very few studies addressed the need to introduce a co-detector (Wu et al., 2016). The performance accuracy of this study on spam detection also needs to be improved. To date, research focusing on spam risk assessment of microblogging social network accounts has been limited despite the evidence of the rise in spam activities. Thus, this study proposed a unified framework for co-detect spam message and spam account in SMCM and at the same time access the risk level of accounts on Twitter microblogging network.

1.3 Research motivation

According to a report from Nexgate digital security organization, social spam has grown at the rate of 355 percent during the year 2013 (Nguyen, 2013). The rise in spam activities is common on most popular social media platforms more than any other networks. Spam may appear inform of text or links with the text adds used for phishing attacks, while the links usually direct legitimate users to malware contents, pornographic websites or both (Morrison, 2013). There are spam mechanisms like social bots, fake accounts, spammy applications, and like-jacking (Chen et al., 2014; Echeverría & Zhou, 2017). The design of social networks has made it easier for spam contents to propagate easily. Unlike email spam distribution, social spam can witness huge spread in seconds. According to Nexgate report, one-in-four comments on social media included spam contents and one-in-eleven contents contained aggressive speech (Nguyen, 2013).

Several classical methodologies focused on investigating the network structure of spammers without much emphasis on other behavioral characteristics such as assessing some features outside the network structure to model spammer's behavior. These systems have been used for Sybil detection (Danezis & Mittal, 2009; Gong et al., 2014; Zhi Yang et al., 2015). Sybil connotes fake account created by attacker to perpetrate

malicious activities, which may include distribution of spam contents. However, several evidences have shown the limitations of these systems for effective malicious users detection in social networks, some of which center on the evasion strategies used by spammers to avoid detection (Viswanath et al., 2011).

To evade existing spam filters, spammers have devised strategies to prevent the system from identifying their malicious intents. These may include the purchase of fake followers and tweets to improve accounts visibilities and reputations (Yang et al., 2013). For instance, a platform such as Intertwitter (http://intertwitter.com/) offered 10,000 fake followers accounts at the rate of $79, giving spammers the opportunity to embed themselves seamlessly within the network of legitimate users. Fake accounts are now offered in large volumes, varying from thousands to millions (Zhang & Lu, 2016). These bogus accounts and their fake links are infringing on the normal social network trusts and disrupting the social media for effective social communication. Thus, there is a need to investigate more features for spam detection in SMCM and go beyond the structural analysis for developing effective classification framework. Indeed, at present, research on co-detection is still in its infancy. Although it is acceptable that a lot of works have been done on structural analysis and machine learning modeling for spam detection, none have assessed the possibility of classifying social network accounts according to risk level.

1.4 Problem statement

Unlike social networks like Facebook and Renren, mobile and microblogging messages are short, unstructured, and contain many domain specific words, such as abbreviations, idioms, bad punctuations, short URLs, and emoticon symbols. These characteristics posed challenges to the traditional content-based analysis (Hu et al., 2013). Thus, existing studies that focus on content analysis degraded in performance

(Cui, 2016; Lee et al., 2010a; Wang, 2010c). To address the problem with content-based analysis, a hybrid method that integrates content, behavioral and network information have been studied (Hu et al., 2013; Wu et al., 2016; Yang et al., 2013). However, the performance of these hybrid models still needs to be improved by investigating more features to effectively identify spammers.

Studies on mobile and social spam detection addressed spam account (Lee & Kim, 2014; Zheng et al., 2015) and spam message detection (Chan et al., 2014; El-Alfy & AlHasan, 2016; Martinez-Romo & Araujo, 2013) as two separate tasks utilizing different frameworks. However, these approaches faced the problem of evasion (Chan et al., 2014; Yang et al., 2013). Thus, combining both spam message and spam account detection within a single framework could provide better performance against evasion on spam filter.

In addition, research has shown that spammers utilize different malicious accounts and strategies to engage in spamming activities (Viswanath et al., 2014; Zhang & Lu, 2016). Therefore, assessing the spam risk level of accounts on microblogging social network is important to provide comprehensive analysis to legitimate users (Echeverría & Zhou, 2017).

1.5 Aim and objectives

The aim of this thesis is to propose a novel unified framework to co-detect spam message and spam account in SMCM and at the same time assess the risk level of microblogging social network accounts. To achieve this aim, the following issues need to be addressed:

(a) To investigate features for spam account detection in microblogging social network using hybrid method.

(b) To design a unified framework for spam message and spam account detection in short message communication media.

(c) To design a ranking scheme for spam risk assessment model in microblogging social network.

(d) To evaluate the performance of the proposed unified framework by validating it using evaluation studies at different stages.

(e) To implement a novel prototype of the proposed framework for practical evaluation in an online environment.

1.6 Research questions

This study addressed the following research questions:

(a) What are the features to identify spammers on microblogging social network using hybrid method?

(b) How can a unified framework be developed to detect both spam message and spam account in short message communication media?

(c) How can a spam risk assessment model for microblogging social network be developed?

(d) What is the effect on performance accuracy when the proposed unified framework is compare with existing approaches?

(e) Can a prototype of the proposed framework achieve promising results when deploy in an online environment?

1.7 Scope of the study

Although there are several malicious activities on the social network, this study addressed social spam detection with specific focus on microblogging social network. Twitter microblogging social network was utilized as test bed for spam account detection due to its openness and comprehensive APIs for data collection. In addition, a

hybrid feature analysis was studied to provide better spam account detection framework using five categories of features: User profile, content, network, timing, and automation. This study also investigated mobile SMS spam detection by proposing unique features to develop a compact model for spam message detection on both mobile and microblogging platforms. During the cause of this research, several techniques have been adopted to achieve the aim of this study. These include natural language processing (NLP), graph analytic, machine learning, evolutionary computation, and Fuzzy Analytic Hierarchy Process.

1.8 Research methodology

This section presents the research methodology adopted in this thesis. The different stages of the research methods are shown in Figure 1.1. The author adopted systematic approach to address each of the phases presented. For instance, the literature review and problem extraction are discussed in Phase 1. Phase 2 elaborates the research objectives. Phase 3 deals with the framework design. Phase 4 focuses on data collection and framework implementation. In Phase 5, results gathering, analyses, and evaluation are critically considered. Finally, Phase 6 addresses the development of prototype for the system evaluation.

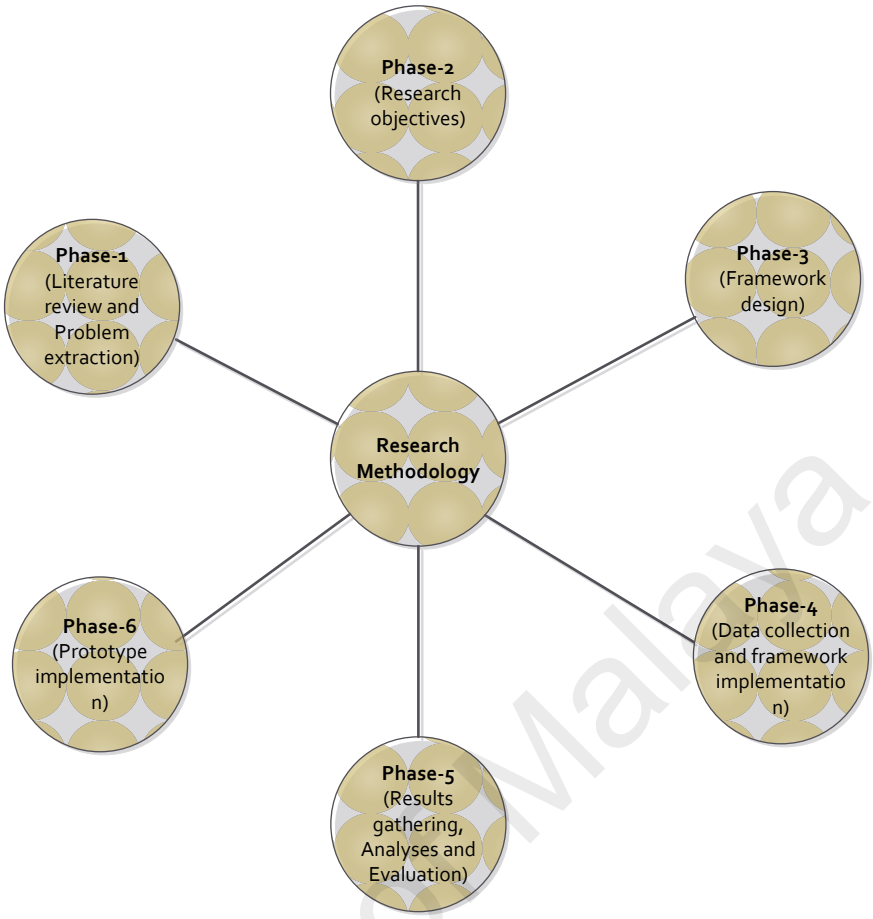


Figure 1.1: Research methodology

Phase 1: Literature Review and Problem Extraction

The focus of this thesis is to develop a unified framework for both spam message and spam account detection in SMCM and to access the risk level of microblogging social network accounts. Therefore, the systematic literature review undertaken is centered on the following related works:

1) First, the existing spam message and spam account detection studies in social networks are categorized according to the features employed for detection;
2) Second, through a thorough analysis of the existing related works, the taxonomy of the various methods used for spam detection in social networks are presented;

3) Third, the existing studies on SMS spam detection are reviewed in order to understand the current state-of-the-art approaches used for SMS spam message detection as well as their performances;
4) Finally, issues to be addressed are identified from the literature review.

Phase 2: Research objectives

After identifying the critical issues to be addressed, the objectives for this study are formulated as discussed in the previous section. The central goal of this thesis is to investigate the possibility of developing a co-detector that will benefit both mobile and microblogging communication media due to their inherent characteristics. The objectives formulated are systematically followed until the final objective is achieved.

Phase 3: Framework design

This phase deals with the overall design of the proposed unified framework, which incorporates two unique models: Spam Message and Spam Account Detection Model (SMSADM) and Spam Risk Assessment Model (SRAM). The SMSADM contains two sub-models namely Spam Account Detection Model (SADM) and Spam Message Detection Model (SMDM). The details of these models are discussed in Chapter 4 of this thesis.

Phase 4: Data Collection and Framework Implementation

In this phase, both mobile SMS and microblogging data are collected in order to test the proposed unified framework. Twitter microblogging data are specifically used to evaluate the capability of the proposed framework for spam account detection due to its openness and comprehensive API for data collection. The framework is implemented using different tools that are discussed in Chapter 5.

Phase 5: Results gathering, Analyses, and Evaluation

The results obtained from the experiments conducted are collated and analyzed in Chapter 5 to ascertain the performance of the proposed unified framework. Evaluation of the different models is carried out by considering the standard evaluation metrics from the literatures, such as Accuracy, Precision, Recall, F-measure, False Positive Rate (FPR), and Area under Curve Receiver Operating Characteristics (AUC-ROC).

Phase 6: Prototype Implementation

This phase focuses on the prototype development of the proposed unified framework. It demonstrates the various modules incorporated to achieve the overall aim of this thesis. The details discussions on prototype implementation are highlighted in Chapter 6. The main purpose of the prototype is to ascertain the performance of the proposed framework when deploy in an online environment.

1.9 Thesis structure

Chapter 2 introduces spam detection systems in both OSNs and mobile communication media. The chapter first discusses OSNs categorization and datasets before proceeding to the taxonomy of the different features for malicious accounts detection in OSNs. The chapter also highlights the taxonomy of the various methods for malicious account detection in OSNs and then presents mobile SMS spam message detection systems. A discussion on risk assessment studies is further explored in this chapter.

Chapter 3 focuses on investigating the rise of spam bots in SMCM. It introduces the various studies on social bots detection and critically assesses the negative impact of social bots in microblogging network. Several social engineering threats are discussed to understand the strategies used by spammers to perpetrate malicious intents. The

chapter also discusses the evasion issues with content analysis systems. Finally, the limitation of Analytic Hierarchy Process (AHP) for risk assessment is presented.

Chapter 4 explains the main contribution of this study, which introduces the novel unified framework for spam detection and risk assessment in SMCM. It describes the compositional details of SMSADM and SRAM models utilized to rate, rank, and categorized microblogging accounts based on their risk level. The chapter starts by introducing the rationale behind the framework and subsequently presents its operational details. The various components of the proposed unified framework are presented. The bio-inspired evolutionary computational approach employed to identify discriminating features for spam account detection is summarized in this chapter.

Chapter 5 focuses on the results of the experiments conducted to validate and evaluate the proposed unified framework. In order to show the progress of the results, the experimental results are presented in four stages. The first stage highlights the results obtained based on the spam account detection. The second stage focuses on the experimental results of the spam message detection. The third stage addresses the experimental results of the SRAM model. The fourth stage discusses the performance comparison of the different proposed models in this thesis with existing related studies as well as the verification of the SRAM model.

Chapter 6 addresses the prototype implementation, which highlights the key components of the proposed unified framework as well as the relationship among them. The chapter starts by presenting an outline of the system development method, the prototype functionalities, and the various modules, such as spam message detection module, spam account detection module, and the risk assessment module. Furthermore, sample demonstration of the operational details of each module is presented to elaborate how the prototype can be used to assist security analysts in making an informed

decision. This chapter also provides a discussion on the underlining advantages and limitations of the proposed approach.

In conclusion, Chapter 7 discusses how the research questions were addressed, the achievements of the study, limitations as well as providing suggestions for future enhancements.

A number of appendices are also included in this thesis, which presents an array of information to support the main discussion. These include source codes, sample spam messages, sample spam accounts, and a list of peer-reviewed publications from this study.

CHAPTER 2: ONLINE SOCIAL NETWORK AND MOBILE SPAM DETECTION SYSTEMS

The trends in security studies related to attacks, threats and vulnerabilities on OSNs and mobile communication media have received substantial interests from academic community in the recent years. To understand the domain of spam detection in both OSNs and mobile communication media, this chapter presents an introduction of the different taxonomies used to categorize related studies. The chapter begins by providing an overview of social network, its definition and categorization, and discussing the taxonomy based on features and methods for detecting malicious users. It continues by presenting related studies on mobile SMS spam message detection as well as risk assessment studies.

2.1 Online Social Networks (OSNs)

OSNs have emerged as important platforms for people to communicate across the globe. Since the introduction of the first OSN, SixDegree, in 1997 several social networking platforms (see Figure 2.1), such as Twitter, Facebook, and LinkedIn have gained popularity (Heidemann et al., 2012). Social networks emerged from different interdisciplinary fields of studies. The term social network is also used in the fields of social psychology, sociology, statistics, and graph theory to represent a social structure that consists of a set of individuals or organizations with various interactions or relationships among them. With the emergence of World Wide Web (WWW) and the advancement of technologies, social networks have become widely used communication media (Ahmad & Sarkar, 2016; Heidemann et al., 2012).

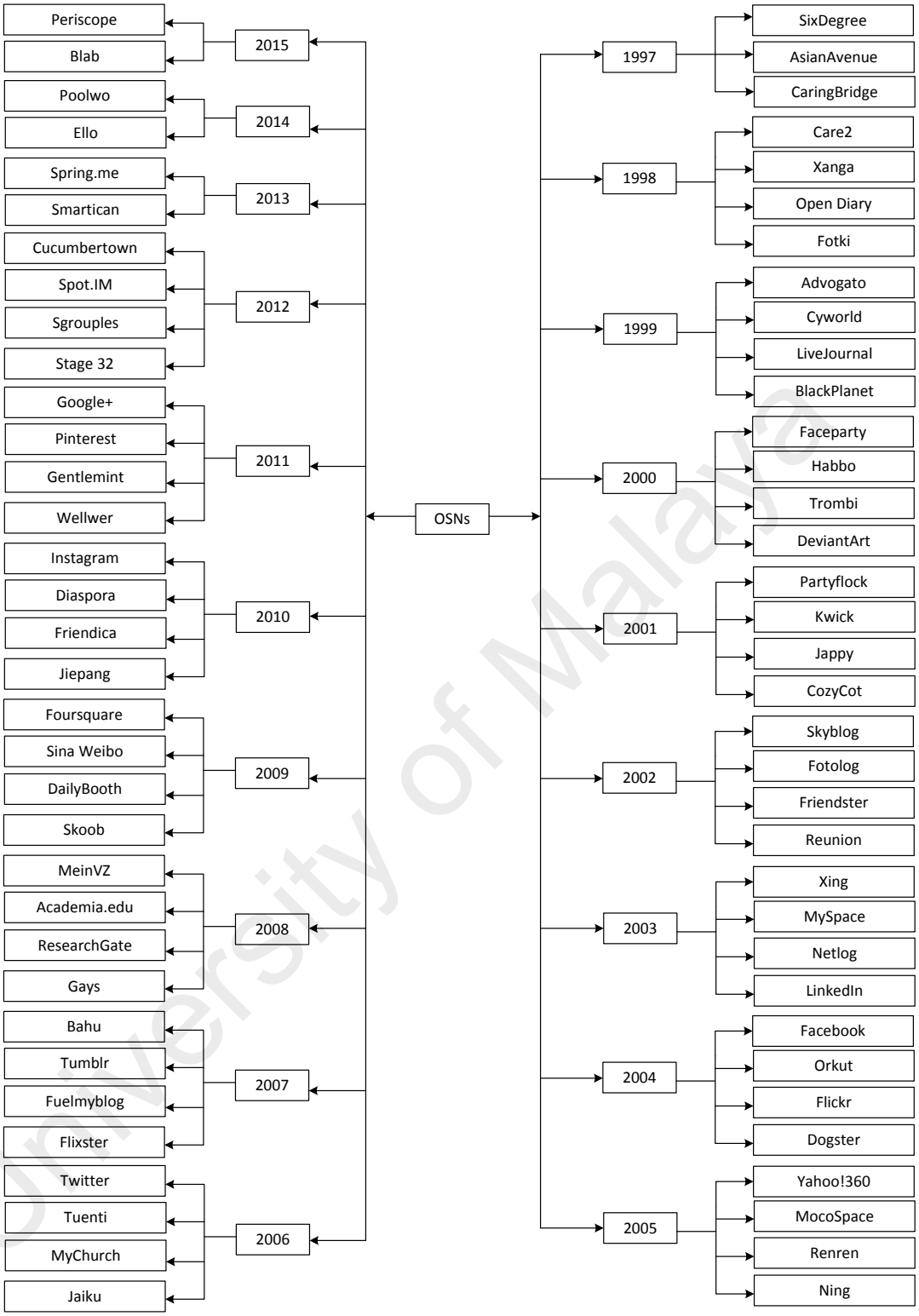


Figure 2.1: Some examples of OSNs since 1997

According to Statista in April 2016, social network data has grown tremendous. For instance, market leader Facebook was the first social networking site whose registered

users have surpassed 1 billion and the number of its monthly active users is currently estimated at 1.59 billion (Statista, 2016). The eighth-ranked photo-sharing app, Instagram, had more than 400 million monthly active users. Meanwhile, Twitter microblogging social network released in 2006 has attracted more than 320 million monthly active users (Statista, 2016).

The popularity of OSNs attracts a great deal of attentions among social network users. For instance, organizations leverage social platforms to promote their products and reach out to customers directly on their networks. Celebrities utilize OSN to communicate with their fans. Academia takes advantage of them to improve their citations, and news media distribute their breaking news on these platforms (Cresci et al., 2015; Igawa et al., 2016). Individual also uses social networks to connect with long-lost friends, create text-based contents, publish contents, browse friends' profiles, post photos, share multimedia files, and engage in other numerous social activities. As a consequence, the rapid growth of social networks has triggered a dramatic increase in malicious activities (Fire et al., 2014).

2.1.1 Definition and categorization

Following the launch of SixDegree.com, several notable definitions have been used to describe social network. For example, Boyd and Ellison (2007) defined online social networking site as "*web-based service which allows individuals to construct a public or semi-public profile within a bounded system, articulate a list of other users with whom they share a connection and traverse their list of connections and those made by others within the system*". According to Adamic and Adar (2005), social network is viewed as a service that gather information on users' social contacts, construct a large interconnected social network, and reveal to users how they are connected to others in

the network. Schneider et al. (2009) defined OSN as a form of online communities among people with common interests, activities, backgrounds, and/or friendships.

Table 2.1: Types of social networks based on purpose

Type	Description	Example
Private social networks	These social networks are specifically developed for private use.	Facebook, MySpace
Business social networks	Introduced for business purpose.	Xing, LinkedIn
Academic social networks	These are developed for academic researchers.	Academia.edu, ResearchGate
Microblogging and News update	They provide a platform for sharing latest updates about what people are doing.	Twitter, Tumblr.
Video sharing social networks	These are developed for sharing different kinds of videos including tutorials and news.	YouTube, Flickr
Instant messaging social networks	These are cross-platform messaging applications developed for sharing video, text, images and audio contents.	Skype, WhatsApp
Event social network	The OSNs connect customers with events, entertainments, and movies.	Eventful, Zvents
Location-based social networks	These OSNs help people to look around for perfect places to go with their friends.	Foursquare, Apontador

Social networks can be categorized according to the purpose they are developed. For instance, Table 2.1 presents social networks based on their primary objectives. Each of these networks particularly targets a diverse group of users with a specific focus on rendering unique services to the registered users. For example, a social network like Facebook was developed with the prime purpose of providing a private network where users can share their experiences. A network, such as LinkedIn was launched for business purpose. If a researcher wants to make his articles and research activities

available to the research community, he may choose to use ResearchGate. This shows that each OSN has unique purpose and functionalities for which the platform is developed to serve the registered users.

2.1.2 OSNs Datasets

The datasets used in the previous studies that deal with detection of malicious accounts on OSNs can be broadly grouped into two main categories: graph and non-graph. The first category modeled social network as a graph represented as nodes and edges. The second category contains different features extracted from social network data, which are used to build a detection system. It is important to state that there are several publicly available graph datasets for social network research as shown in Table 2.2. The most prominent are those compiled by Stanford University social network research community (Leskovec, 2015). The datasets contain many social network graphs including Facebook, Twitter, and LiveJournal. However, some researchers have developed web crawlers to collect private graph data from a social network of interest.

In some cases, researchers evaluated their proposed models using synthetic social graph generated by applying Barabasi-Albert preferential attachment model. This model assumes that social network is scale-free and it follows a power law distribution. For instance, SybilRank (Cao et al., 2012) and community detection algorithms (Viswanath et al., 2011) were evaluated using synthetic datasets in addition to publicly available real-world graph data.

Table 2.2: Public datasets

Type	Category	Name	Description	Web address
Public	Graph	Stanford large network dataset	The dataset is organized into different social graphs, such as ego-Facebook, ego-Twitter, wiki-Vote, com-DBLP, com-Youtube etc.	https://snap.stanford.edu/data/
Public	Non-graph	Deceptive Opinion Spam Corpus	Contains 400 truthful positive reviews from TripAdvisor, 400 deceptive positive reviews from Mechanical Turk, 400 truthful negative reviews from Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor and Yelp, and 400 deceptive negative reviews from Mechanical Turk. These datasets consist of 20 reviews for the most popular Chicago hotels.	http://myleott.com/op_spam/
Public	Non-graph	BibSonomy	This dataset is part of ECML/PKDD Discovery Challenge 2008. The dataset is organized into seven files: tas, tas spam, bookmark, bookmark spam, bibtex, bibtex spam, and user.	www.kde.cs.uni-kassel.de/ws/rsdc08
Public	Non-graph	Tweets2011 Corpus	This dataset is part of the TREC 2011 microblog track. It contains 16 million tweets sampled between Jan. 23rd to Feb. 8th, 2011. It is available for individual download at NIST website. However, this dataset cannot be distributed to other researchers due to privacy issue.	http://trec.nist.gov/data/tweets
Public	Graph and Non-graph	FakeProject	The dataset was released by MIB project hosted at Institute of Informatics and Telematics (IIT) of the Italian National Research Council (CNR). The dataset is organized into five groups: TFP, E13, INT, FSF, and TWT. The dataset requires password due to users' privacy.	http://mib.projects.iit.cnr.it/dataset.html

Due to user's privacy issue, some of the public non-graph datasets have been secured with passwords. In addition, they contain limited numbers of users' attributes released for research purpose. This constraint forces researcher to develop web crawlers to collect private data using the API from social network providers (Alsaleh et al., 2014; Yang et al., 2013). Twitter provides REST API and Streaming API to collect tweets, network data, and other information from its platform. Facebook also provides Facebook Graph API to get data in and out of the Facebook social network.

2.2 Malicious Accounts Detection in OSNs

As shown in Figure 2.2, accounts used for malicious activities on the social networks can be categorized into two types: Fraudulent/career-spamming and compromised accounts, which are categorized under abuse of credentials. An adversary creates fraudulent accounts to engage in malicious contents distribution, such as embedding malicious links to phishing web pages in order to obtain sensitive information from the victim. By collecting a large number of legitimate users information as well as information about friends of friends on the network, an adversary can create Sybil or fake accounts that mimic existing users' identities. The fake identity is used to exploit legitimate users and undermine the trust relationship on the social network in order to perform different malicious activities.

These malicious activities may include social spamming, drive-by-download, and private data harvesting (Chen et al., 2014). To ensure that these fraudulent accounts stay for a longer period on the network, attackers sometimes equipped them with automated characteristics, which provide them with the ability to post contents that resemble real users. Fake account on the social network has turned into a multimillion-dollar business, where several fake accounts are advertised on the Internet for those who want to boost their account reputation. Previous report indicates that accounts of celebrities, politicians, and popular organizations featured a suspicious increase in fake accounts (Cresci et al., 2015) and another study has shown that most of the celebrities accounts on the social network are fake (Wüest, 2010).

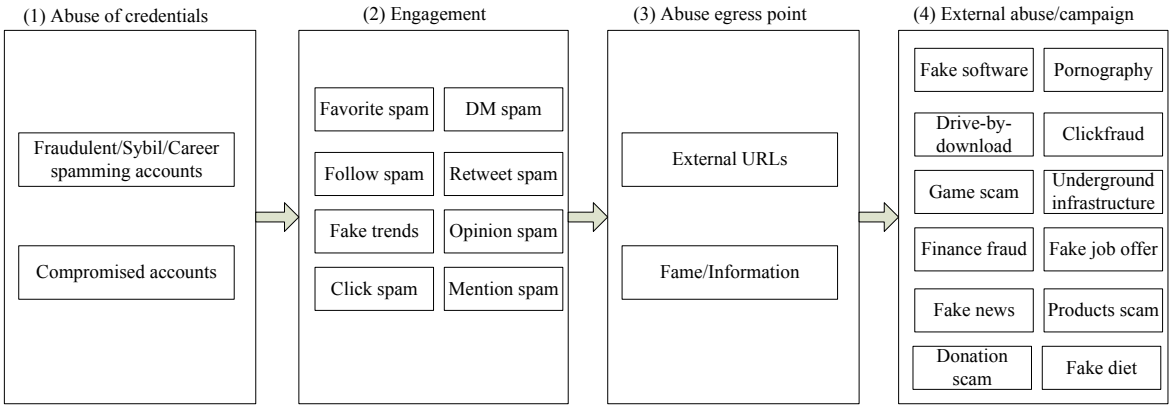


Figure 2.2: Abuse of social network accounts for spamming, phishing, and their campaigns

Conversely, compromised account is an account hijacked from legitimate users using a strategy, such as social engineering attack to deceive legitimate users by clicking on links to phishing web pages. Study has shown that compromised accounts are more useful to spammers than career-spamming since they enable spammers to leverage the existing trust relationship already established by the accounts (Egele et al., 2015). Account hijacked from legitimate user will suddenly experienced a change in the posting patterns because it will be difficult for spammers to completely maintain the normal posting behaviors of the real owners (Ruan et al., 2016). An example of the slight changes in posting patterns may include the use of compromised accounts to spread spam messages, such as favorite, direct message (DM), and click spam that contains malicious links. The malicious link embedded in the post usually serves as an egress point through which the victim can be taken to external pages. These web pages exploit the victim using different malicious campaigns, such as pornography, fake news, and donation scam.

2.2.1 Taxonomy of features for malicious accounts detection

Generally, malicious accounts detection on social networks can be conceptualized using the generic framework as shown in Figure 2.3. The inputs to the detector originate

from the previously discussed datasets, which represent adjacency matrix or set of features. The input may require data preprocessing, such as removal of accounts with a small number of connections or stringent preprocessing like extraction of N-gram features from the messages posted by the accounts under consideration. The preprocessed data is passed to malicious account detector, which produces output in form of class and rank. The class may be viewed as spammer or legitimate, compromised or normal, Sybil or non-Sybil nodes, malicious or legitimate and so on. The rank indicates the probability that a given account belong to the final class label. After producing the final class label, the detector is evaluated using the popular evaluation metrics, such as precision, recall, F-measure, and accuracy. During the detection, different features and methods have been considered with the goal of identifying the class of a given account or the class of the message sent by this account owner.

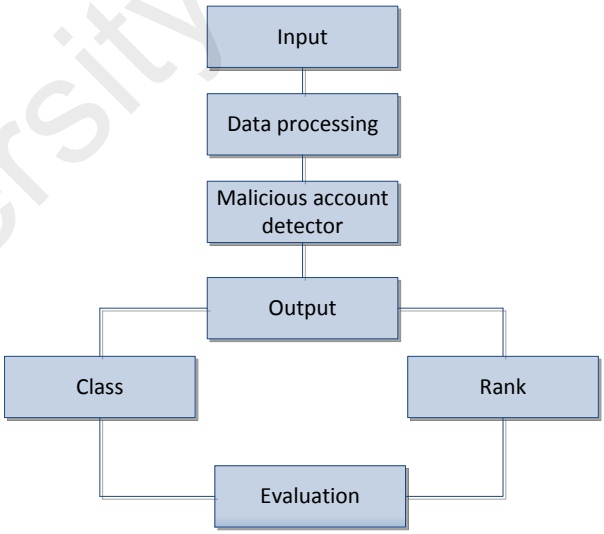


Figure 2.3: Generic framework for malicious account detection

Features used in the previous studies for detecting malicious accounts in social networks can be broadly merged under three main analyses: social network analysis,

content/behavioral analysis, and hybrid analysis as shown in Figure 2.4. This section discusses each of these categories and highlights the different features proposed in the literature to identify malicious accounts and their behaviors in OSNs.

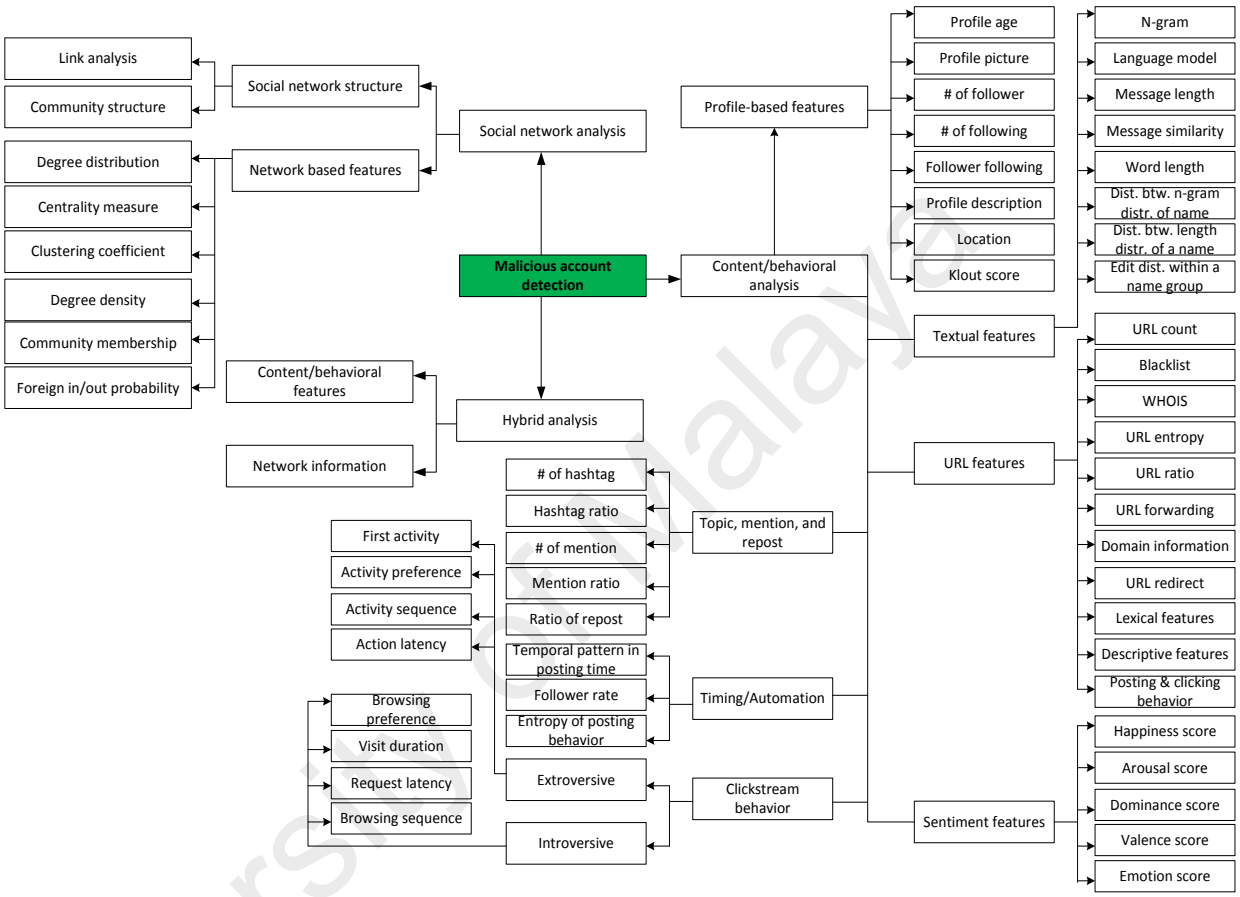


Figure 2.4: Taxonomy of malicious accounts detection features

(1) Social network analysis

Social network analysis involves analyzing the topological social structure of the accounts within the network or extracting discriminative network features to detect malicious users. Some studies focused on analyzing the structure of the users on the network (Cao et al., 2012; Mulamba et al., 2016) while others concentrated on identifying network features rather than studying the topological structure in details. Such network features include community-based features (Bhat & Abulaish, 2013; Bhat

et al., 2014) or features based on neighborhood and centrality (Almaatouq et al., 2016; Yang et al., 2013).

(a) Social network structure

An example of social structural analysis is presented by (Stein et al., 2011) in Facebook Immune System, which makes the assumption that although malicious users may create a large number of fake identities on social networks, however, it will be difficult to establish arbitrarily large number of social relationships to legitimate accounts. This makes them poorly connected to the network when compared with legitimate accounts. This assumption was adopted to develop many Sybil defense algorithms, such as SybilGuard (Yu et al., 2006), SybilLimit (Yu et al., 2008), SybilRank (Cao et al., 2012), and GateKeeper (Tran et al., 2011) believing that fake account will require significant trustworthy social ties to appear legitimate within the network. This feature is analyzed to identify densely connected Sybil regions (Viswanath et al., 2011). For instance, Tran et al. (2011) leveraged random expander graph property and improved ticket distribution technique to study the structural connection of social network users. The proposed GateKeeper algorithm applied different randomly chosen points to run the ticket distribution and merges the outcomes to perform a decentralized node admission control. During node admission, a node that acts as a ticket source distributes a certain number of tickets on the network until a considerable proportion of the honest nodes receives some tickets. The ticket distribution follows a breadth first search approach where each node is placed at a breadth first search level according to its shortest path distance from the ticket source. The ticket source splits the tickets and distributes them to its neighbors. A single ticket is kept by each node on the network and the remaining tickets are propagated to the nodes in its neighbors at the next level. If a node does not have outgoing connections to

the next level, the node simply destroys the remaining tickets. This process continues until no ticket remains to distribute.

However, the effectiveness of this detection approach is limited by this assumption and several experiments have proven the weakness of this approach (Elyashar et al., 2013; Viswanath et al., 2011; Zhi Yang et al., 2015). For instance, a social network like Twitter allows a unidirectional user binding, which permits an account to follow anyone on the network without the followee prior consent. Although the followee may decide to block the follower, however, in reality, majority of them follow back for the sake of courtesy. This behavior allows malicious accounts to add more legitimate users to his network (Hu et al., 2013). In addition, fake accounts are now sold in thousands on the Internet allowing Sybil to embed seamlessly within the network and appear as legitimate accounts. Viswanath et al. (2011) analyzed various Sybil detection algorithms by decomposing Sybil defense approaches. The study revealed that Sybil defense algorithms operated by implicitly ranking nodes (i.e accounts) based on how well they connect to a trusted node. Accounts with a strong connection to the legitimate users are given a higher rank score and are deem to be more trustworthy.

Since the assumption used by the early social network structural Sybil detection algorithms is very weak in providing efficient malicious account detection systems, algorithms like SybilRadar (Mulamba et al., 2016) and VoteTrust (Zhi Yang et al., 2015) deviated from this assumption. For example, SybilRadar leveraged the assumption that attackers can create as many malicious accounts as possible and they can establish significant large links to legitimate accounts. VoteTrust exploited friendship invitation relationship between fake accounts and legitimate users using request invitation graph in order to identify malicious users. However, a small temporal change in Sybil behavior will break the detection approaches in these studies. This

accounts for why some researchers explored the possibility of extracting network features and content/behavioral based features to train machine learning classifiers. The authors of SybilRadar algorithm also admitted that the use of some account attributes might improve SybilRadar performance.

In addition to studying the link structure of the users on the network, few studies also analyzed the community structure of the social network to rank accounts and identify the community of malicious users (Liu et al., 2015; Mulamba et al., 2016; Viswanath et al., 2011). However, an intelligent adversary may forge the connectivity of the controlled malicious accounts to imitate the network community structure of the portion of the social network exhibited by normal users. This tactic would make it difficult for methods relying on community analysis to effectively detect malicious accounts.

(b) Network based features

Another line of research using network analysis involves the identification of effective network features to detect malicious accounts. Features such as degree distribution, centrality measures, clustering coefficient, degree density, and community membership have been widely studied (Almaatouq et al., 2016; Bhat & Abulaish, 2013). For instance, Bhat and Abulaish (2013) extracted different community-based features from a user's social connections to train machine learning classifiers to detect social spammers. The study reveals that community features, such as total in/out ratio, core node, community membership, foreign out-degree, foreign in/our ratio, foreign out-link probability, reciprocity, and foreign out-link grouping are effective for social spammer detection. Centrality measures like betweenness and closeness, as well as degree computation such as weighted in and out degree, degree density, weighted bidegree, and density of relative edges of both mention and followers networks have played a key role in spammer detection (Almaatouq et al., 2016; Yang et al., 2013). Fire

et al. (2014) studied the connection strength between the pair of accounts on Facebook to detect malicious accounts, which may pose a risk to the legitimate user and then restrict these set of accounts from accessing the private information of the legitimate users. The authors proposed a number of network-based features, such as mutual friend and the number of common group between a pair of accounts. They further defined a global connection strength heuristic function capable of identifying fake accounts on the Facebook network.

Apart from the fact that an adversary can evade some network-based features, such as bidirectional links, and bidirectional link ratio (Mulamba et al., 2016; Yang et al., 2013), other challenges have centered on how to deal with the computational complexity when extracting network based features for large social networks (Fire et al., 2014). Several studies revealed that some network features are expensive for attackers to evade. For example, features such clustering coefficient, betweenness centrality, and following to media neighbors' followers are more robust for identifying social spammers (Almaatouq et al., 2016; Yang et al., 2013).

(2) Content/behavioral analysis

Content/behavioral analysis involves identification of effective features outside the social connections of users. The studies in this category assumed that the content generation pattern of malicious accounts is different from legitimate users. Thus, extracting many features around this behavior could distinguish malicious accounts from legitimate ones. One of the advantages of using content/behavioral analysis is that it can be easily encoded into features. These features are provided as input to machine learning algorithms, which can learn the signature of malicious and legitimate activities. Thus, it allows classification of accounts based on the observed behaviors. In this domain, many classes of features are commonly employed to represent users' behaviors

as shown in Figure 2.4 including the use of profile-based, textual, URL, timing/automation, topic, mention, and reposting behaviors, sentiment as well as clickstream features like extroversive and introversive behaviors.

(a) Profile-based features

Profile-based features involve studying the basic profile information of an account on the network. Studies that utilize profile-based features established that by analyzing profile information of an account, such as profile age, profile picture, number of follower, number of following, follower-following ratio, profile description, geolocation, Klout score, account verification status, listed count, and total number of tweets posted; it is possible to distinguish malicious accounts from legitimate ones (Alsaleh et al., 2014; Chu et al., 2012a; Main & Shekokhar, 2015; Miller et al., 2014).

The study conducted by Benevenuto et al. (2009) shows that the use of user behavioral attributes like the total view of tag videos, total ratings of tag videos, and user rank can identify social spam accounts and content promoters on social networks that support video sharing, such as YouTube. Chu et al. (2012a) studied profile-based features to distinguish human accounts from the accounts controlled by an automated program called social bot. Studies that utilized profile-based features also combined these features with other categories, such as textual, URL and so on. For instance, Stringhini et al. (2010) created honeypot accounts on three social networks: Facebook, MySpace, and Twitter. They logged every user's activities through the honeypot accounts and extracted profile-based and URL features to identify spammers. The issue with content/behavioral features has centered on how to deal with the evasion tactics of malicious users. For instance, profile age was identified as a discriminative feature based on the fact that account of malicious users usually exhibits short profile age (Almaatouq et al., 2016; Lee & Kim, 2014). However, Egele et al. (2015) highlighted

that majority of accounts used for spamming on social networks are compromised accounts which are more valuable to spammers due to the pre-established trust relationship. This is similar to the findings in (Gao et al., 2010; Grier et al., 2010) on Facebook and Twitter networks.

(b) Textual features

The use of textual or content features, such N-gram, language model, message length, message similarity, and word length has been studied (Balakrishnan et al., 2016; Gani et al., 2012; Harsule & Nighot, 2016; Martinez-Romo & Araujo, 2013). N-gram based system called Filter Wall (FW) capable of filtering messages in a user's timeline was developed to build user's N-gram profile (Harsule & Nighot, 2016). From this N-gram profile, a similarity distance metrics is applied to categorize wall posts as spam and non-spam messages. Martinez-Romo and Araujo (2013) introduced features based on language model from the textual content of a user's tweets. The feature studied the divergent of textual information of malicious and normal tweets.

A language model is a statistical model for text analysis, which is based on a probability distribution over pieces of text, indicating the likelihood of observing these pieces in a language. Usually, the real model of a language is unknown and is estimated using a sample of text representative of that language. Different texts can be compared by estimating models for each of them, and analyzing the models using well-known methods for comparing probability distributions (Martinez-Romo & Araujo, 2013). The authors examined the use of language in different entities, such as a topic, a tweet, and the external page linked from the tweet. It was established that the language model of a legitimate tweet is more likely to be different from the spam tweet. They applied Kullback–Leibler divergence, which is an asymmetric divergence measure adopted from information theory, between respective language models of the entities considered

to measure how bad the probability distribution of one language model deviate from other. Based on this assumption, the authors exploit the divergence between the language models to classify tweet as spam or non-spam. This approach has been shown to work well for detecting malicious tweets in trending topics. However, it requires the knowledge of some external contents that may introduce other computational costs. Gani et al. (2012) extracted several features from user's messages including average words length, average message length, average number of words per message, the ratio of uppercase letters, the ratio of short words per message, average number of short words, standard deviation and variance of special characters to detect social spam.

Studies that examined the content/behavioral characteristics of malicious accounts applied off-the-shelf machine learning algorithms to check the effectiveness of the extracted features in distinguishing malicious from legitimate accounts (Lee & Kim, 2014; Martinez-Romo & Araujo, 2013). For instance, Lee and Kim (2014) applied different name-based features, such as distance between unigram/bigram distribution of a name group, edit distance within a name group, distance between length distribution for a name group, and distance between position-wise unigram distribution to train a SVM classification algorithm. One of the identifiable key issues with textual features lies in the computational complexity. For example, feature such as N-gram analysis may require several preprocessing steps, which can introduce more computational costs.

(c) URL features

A large body of studies examined URL features to analyze the URL posting patterns between malicious and legitimate users. A URL is a link embedded within a user's post with an attempt to redirect users to an external malicious page. Malicious users can use this strategy to distribute malicious links and engage victim with fake advertisements. For instance, some studies found that forwarding patterns of URLs, domain, and lexical

features are effective for detecting malicious URLs, which provide the opportunity to mine URL posting patterns of malicious and legitimate accounts (Cao et al., 2016; Chen et al., 2014; Lin et al., 2013). While some social bots are created to post malicious contents on social networks, others mimic the posting patterns of legitimate users. This strategy has been witnessed in a democratic setting where malicious social bots artificially inflate support for political candidate and abuse the outcome of elections (Cresci et al., 2017; Ratkiewicz et al., 2011).

The use of URL features, such as dash count in the hostname, longest domain name, domain rank, URL domain age, and URL count was explored in (Chen et al., 2014) to identify malicious links. Aggarwal et al. (2012) combined many content/behavioral features including URL features based on WHOIS and URL redirection status. However, the use of WHOIS and URL redirection require the need to query some contents or Internet host-based information, which limit their application in real-time detection of millions of URLs encountered on the social network on a daily basis (Cao & Caverlee, 2015; Lin et al., 2013). To address the problem of querying host-based information, lexical and descriptive URL features were adopted (Lin et al., 2013). The first feature describes the lexical information of the links while the second feature represents some statistical attributes. This study further shows that lexical features are more efficient than descriptive features, but they can only work within a short period. In addition, the descriptive features are less effective but they can be used for a longer period. However, malicious users may continue to change their posting behaviors and try to act like normal users, which can lead to an increase in false alarm (Wu et al., 2016).

(d) Topic, mention, and retweet

The majority of social networks allow users to use varieties of tools for communication, such as grouping of messages using topic, sending messages directly to a specific target user, and reposting a user's message. For instance, in Twitter, a user can use "#" symbol to indicate topic in a post. Similarly, "@" symbol can be used to forward message directly to a target user on Twitter (e.g @obama). The retweet function allows users to repost messages that appear on their timeline or through specific search keywords. It is evident that malicious users can hide behind trending topics or bypass any requirement for social connection with legitimate accounts by simply use mention function to reach their target victims (Almaatouq et al., 2016). Studies have established that some malicious accounts are equipped with automation capability to repost messages from legitimate users in order to make their account appear legitimate (Cao et al., 2016). Therefore, features such as the number of hashtag, the number of mention, the number of post retweeted as well as their ratios have been considered (Alsaleh et al., 2014; Chu et al., 2012b; Gupta & Kaushal, 2015).

(e) Timing/automation

Due to the automation capability of some malicious accounts, researchers have studied the temporal posting patterns of malicious and legitimate accounts. For instance, the use of entropy component, which employs tweeting interval as a measure of behavioral complexity has been studied (Chu et al., 2012a). Features like following rate, tweeting rate, API ratio, API URL ratio, and API tweet similarity, which considered the posting time and tweets posting source can be used to identify malicious accounts. Because of the relatively high cost of manually operating a large number of spam accounts, some spammers designed a custom program using API to spread spam

messages. Therefore, it is possible to identify malicious accounts by studying the source of messages.

(f) Sentiment features

Sentiment analysis deals with the process of categorizing opinions expressed within a piece of text to determine the attitude or opinion of the writer towards a specific topic or product. It has been shown that malicious accounts used for cyberbullying can concentrate on specific keywords to spread aggressive spam messages (Ferrara et al., 2014; Galán-García et al., 2014). For instance, a message such as "*if you don't follow me you will die, follow me now*" has been used by spammers to spread cyberbullying contents as a strategy to lure the target victims to accept their friendship requests (Galán-García et al., 2014). Thus, extracting sentiment features, such as happiness, arousal, dominance, valence, and emotion scores can identify spam accounts used for cyberbullying in social networks (Ferrara et al., 2014).

(g) Clickstream behavior

To detect accounts hijacked from legitimate users for spamming, researchers have presented behavioral based features, which analyzed the clickstream characteristics of social network accounts (Ruan et al., 2016). A model of the normal user is developed by considering some of the user's posting patterns over a specific period. This model is then compared with subsequent user behavior to ascertain if the account has been compromised. To effectively build this behavioral profile, clickstream features such as extroversive and introversive behaviors have been studied (Ruan et al., 2016).

Extroversive behaviors consider characteristics, such as the first activity the account engages in. While many users may start their social activity by randomly accessing their friends' timelines, others start by liking the posts that appear on their own timeline.

Extroversive behavior also includes activity preference, activity sequence, and action latency. Conversely, introversive clickstream behaviors include browsing preference, visit duration, request latency, and browsing sequence. Since clickstream features require extensive study of the user's behavioral patterns, it is difficult to efficiently capture all the normal user's clickstream behaviors.

(3) Hybrid analysis

The presence of many platforms for underground markets where it is possible for malicious users to purchase a large number of followers to boost their fake accounts has hindered the effectiveness of relying on content/behavioral analysis. For example, underground markets, such as *BuyTwitterFriends.com* or *TweetSourcer.com* provides cheap services to purchase fake followers allowing malicious account to appear legitimate (Yang et al., 2013). Since malicious users on the social network have devised strategies to make their accounts appear normal, some studies combined both content/behavioral and network information to detect misbehaving users. For example, Yang et al. (2013) analyzed the effectiveness of combining network and content information to detect spammers on Twitter. Wu et al. (2016) combined content and network information to develop a classification algorithm based on L1 and L2 regularization, which can identify spammers and spam message simultaneously. Similarly, Hu et al. (2013) developed a classification model to combine textual features with adjacency matrix represented by the users network connections. The main challenges with these unified approaches centers on the optimal performance of the classification systems, coupled with the need to identify the most discriminating features that can be combined for better performance.

2.2.2 Taxonomy of methods for malicious accounts detection

This section presents a taxonomy of the different methods used in the literature to detect malicious accounts. Figure 2.5 shows the taxonomy of the different methods for detecting malicious accounts and their behaviors specifically in social networks. The methods include crowdsourcing, graph-based and machine learning (ML).

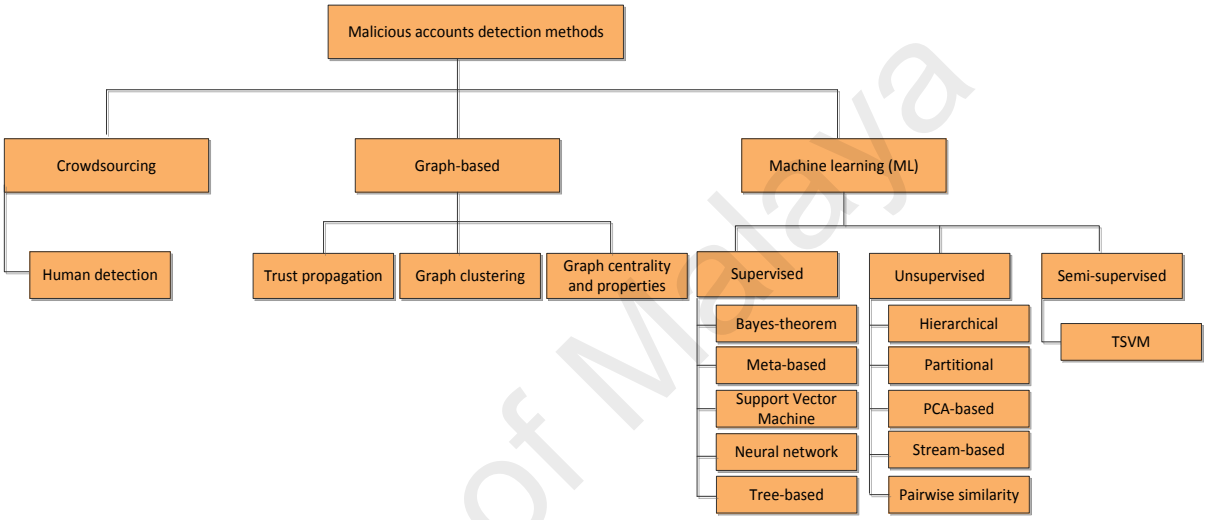


Figure 2.5: Taxonomy of malicious accounts detection methods

Table 2.3 shows the studies that deal with malicious behavior detection on social networks using the three aforementioned methods. The table provides a general overview of the current state-of-the-art literature in malicious accounts detection with a specific focus on social networks.

Table 2.3: Summary of studies on malicious accounts detection

Year	Ref.	Objectives	Database source	Detection category	Method
2006	Yu et al. (2006)	Proposed SybilGuard based on fast-mixing assumption and random walks	ACM	Fake account	Graph-based
2009	Markines et al. (2009)	Proposed framework for spam detection in social tagging system	ACM	Spam account	Machine learning
2009	Benevenuto et al. (2009)	Developed ML model to detect spammer on YouTube	ACM	Spam account	Machine learning
2010	Viswanath et al. (2011)	Analyzed Sybil defense schemes and developed a community-based Sybil detection approach	ACM	Fake account	Graph-based
2010	Gao et al. (2010)	Proposed a clustering approach to group spam into campaigns	ACM	Spam account	Graph-based
2010	Wang	Applied profile-based and content-based features to	Springer	Spam account	Machine

Table 2.3, continued.

	(2010b)	detect spammer			learning
2010	Lee et al. (2010b)	Deployed honeypots and ML system to detect spammer on Twitter and MySpace	ACM	Spam account	Machine learning
2010	Stringhini et al. (2010)	Analyzed the impact of spamming on OSN and developed ML classifier to detect spammers	ACM	Spam account	Machine learning
2011	Sadan and Schwartz (2011)	Developed graph-based model using betweenness centrality metric	ScienceDirect	Phishing detection	Graph-based
2011	Tran et al. (2011)	Proposed a decentralized node admission control protocol algorithm called GateKeeper to separate Sybil accounts from normal accounts	IEEE	Fake account	Graph-based
2011	Yang et al. (2011)	Combined network and content-based features to detect spammer	Springer	Spam account	Machine learning
2011	Mccord and Chuah (2011)	Analyzed content features for spam accounts detection on Twitter	Springer	Spam account	Machine learning
2011	Kontaxis et al. (2011)	Developed tool to detect fake account on LinkedIn	IEEE	Fake account	Machine learning
2011	Jin et al. (2011)	Proposed framework to identify suspicious identities on Facebook	ACM	Fake account	Machine learning
2011	Stein et al. (2011)	Presented the underlying design of Facebook Immune System	ACM	Fake account	Graph-based
2012	Wang et al. (2012b)	Analyzed clickstream data to detect existence of malicious crowdsourcing platforms	ACM	Fake account	Crowdsourcing
2012	Wang et al. (2012a)	Proposed crowdsourcing platform to detect fake accounts	ACM	Fake account	Crowdsourcing
2012	Ahmed and Abulaish (2012)	Applied MCL algorithm to cluster social network accounts into spam and non-spam	IEEE	Spam account	Graph-based
2012	Cao et al. (2012)	Developed SybilRank algorithm using power iteration approach	ACM	Fake account	Graph-based
2012	Ghosh et al. (2012)	Analyzed link farming activities used by accounts on Twitter	ACM	Spam account	Graph-based
2012	Conti et al. (2012)	Studied time evolution of social graph to detect fake accounts on social network	IEEE	Fake account	Graph-based
2012	Chu et al. (2012b)	Developed ML model to detect spam campaigns on Twitter	Springer	Spam account	Machine learning
2012	Aggarwal et al. (2012)	Proposed a tool called PhishAri for real-time detection of malicious tweet	IEEE	Phishing	Machine learning
2012	Chu et al. (2012a)	Focused more on automated account detection approach to identify malicious socialbots, human, and cyborg accounts	IEEE	Spam account	Machine learning
2012	Jiang et al. (2012)	Proposed Sybil group detector on Renren network	IEEE	Fake account	Machine learning
2012	Gani et al. (2012)	Proposed framework that relies on ML model, social interaction and authorship analysis for fake account detection	ACM	Fake account	Machine learning
2013	Lin et al. (2013)	Focused on introducing lightweight features for phishing detection	IEEE	Phishing	Machine learning
2013	Yang et al. (2013)	Combined network and content/behavioral analysis to detect spammers	IEEE	Spam account	Machine learning
2013	Tan et al. (2013)	Designed unsupervised Sybil defense scheme to identify spam accounts in OSN	ACM	Spam account	Graph-based
2013	Martinez-Romo and Araujo (2013)	Combined language model and tweet content approaches to detect spammer	ScienceDirect	Spam account	Machine learning
2013	Lin and Huang (2013)	Studied features for detecting long-surviving spammers on Twitter	IEEE	Spam account	Machine learning
2013	Ahmed and Abulaish (2013)	Proposed 14 generic features for spam detection on Twitter and Facebook	ScienceDirect	Spam account	Machine learning
2013	Bhat and Abulaish (2013)	Developed spam account detection system using community-based features	IEEE	Spam account	Machine learning
2013	Li et al. (2013)	Proposed semi-supervised approach to detect phishing attack	ScienceDirect	Phishing	Machine learning
2014	Chen et al. (2014)	Proposed different features for phishing detection on social network	ScienceDirect	Phishing	Machine learning
2014	Alsaleh et al. (2014)	Classified accounts on Twitter as human, bots, and Sybil using ML models	IEEE	Fake account	Machine learning
2014	Galán-García et al. (2014)	Detected spammers account used for cyberbullying on Twitter	Springer	Fake account	Machine learning
2014	Yang et al.	Developed real-time Sybil detector on Renren	ACM	Fake account	Machine

Table 2.3, continued.

	(2014)				learning
2014	Chan et al. (2014)	Proposed re-weight method in adversarial learning for spam filtering in OSN	ScienceDirect	Spam account	Machine learning
2014	Bhat et al. (2014)	Trained ensemble of classifiers using community-based features	IEEE	Spam account	Machine learning
2014	Singh et al. (2014)	Developed ML model for malicious account detection on Twitter	ACM	Spam account	Machine learning
2014	Fire et al. (2014)	Developed social privacy protector system for fake account detection on Facebook	Springer	Fake account	Machine learning
2014	Lee and Kim (2014)	Developed model using name-based features to detect malicious account	ScienceDirect	Fake account	Machine learning
2014	Kiruthiga et al. (2014)	Introduced extended clone spotter algorithm that employed classification and clustering techniques	IEEE	Fake account	Machine learning
2014	Miller et al. (2014)	Modified stream clustering algorithms to detect spammers on Twitter	ScienceDirect	Spam account	Machine learning
2015	Zhi Yang et al. (2015)	Developed VoteTrust algorithm to detect Sybil accounts using signed graph	IEEE	Fake account	Graph-based
2015	Gupta and Kaushal (2015)	Combined different learning algorithms to detect spam accounts on Twitter	IEEE	Spam account	Machine learning
2015	Zheng et al. (2015)	Developed tool to detect Sina Weibo spammers	ScienceDirect	Spam account	Machine learning
2015	Egele et al. (2015)	Analyzed and proposed compromised accounts detection framework in OSNs	IEEE	Compromised account	Machine learning
2015	Cao and Caverlee (2015)	Proposed framework based on posting and clicking behaviors of posters and clickers of URLs to identify phishing links	Springer	Phishing	Machine learning
2015	Devineni et al. (2015)	Proposed PowerWall algorithm based on modified power law property of a social graph	ACM	Fake account	Graph-based
2015	Ezpeleta et al. (2015)`	Analyzed spam vulnerability with public profile information on OSN	Springer	Spam account	Crowdsourcing
2015	Cresci et al. (2015)	Introduced new baseline dataset for fake follower detection in OSN	ScienceDirect	Fake account	Machine learning
2015	Liu et al. (2015)	Proposed community-based approach to identify social spammers based on two step-process	Springer	Spam account	Graph-based
2015	Main and Shekokhar (2015)	Proposed five features for spammer detection	ScienceDirect	Spam account	Machine learning
2016	Wu et al. (2016)	Proposed a unified framework based on network and content information	ScienceDirect	Spam account	Machine learning
2016	Igawa et al. (2016)	Developed a wavelet-based approach for account classification that detects textual dissemination of spam accounts	ScienceDirect	Spam account	Machine learning
2016	Ruan et al. (2016)	Introduced extroversive and introversive features based on clickstream to detect compromised accounts	IEEE	Compromised account	Machine learning
2016	Zhang and Lu (2016)	Proposed approach for detecting near-duplicate accounts on Weibo	Springer	Fake account	Graph-based
2016	Zuo et al. (2016)	Leveraged friends-of-friends relationship to detect misbehaving users	ScienceDirect	Spam account	Graph-based
2016	Mulamba et al. (2016)	Proposed SybilRadar, an algorithm that improves over SybilRank	Springer	Fake account	Graph-based
2016	Pérez-Rosés et al. (2016)	Studied endorsement relationship between accounts based on some selected skills	ScienceDirect	Spam account	Graph-based
2016	Almaatouq et al. (2016)	Applied Gaussian mixture model (GMM) to identified two categories of spammers and proposed network and content features	Springer	Spam account	Machine learning
2016	Harsule and Nighot (2016)	Developed a system called Filter Wall (FW) based on N-gram analysis	Springer	Spam account	Machine learning
2016	Cao et al. (2016)	Proposed forwarding-based and graph-based features for phishing detection	Springer	Phishing	Machine learning

(1) Crowdsourcing

Wang et al. (2012a) suggested the method of crowdsourcing for detecting malicious accounts. This method leverages human detection, which distributes intelligent tasks to

the Internet users who can identify a pattern of anomalies exhibited by social network accounts. Crowdsourcing involves the use of large and distributed group of workers known as crowd workers to identify suspicious behaviors. The crowd workers analyze social network accounts by checking the information on their profiles and decide whether the accounts are Sybil or legitimate. For instance, Tuenti, the largest social network in Spain, employed 14 full-time employees to detect fake accounts on its network (Cao et al., 2012). By applying crowdsourcing method on two popular OSNs platforms: Facebook and Renren, Wang et al. (2012a) observed that the performance of the hired crowd workers reduces over time, although this method brings about a concession where majority votes can be used to reach the final judgment. This strategy is found suitable for social network providers since it demonstrates a near-zero false alarm. However, a number of drawbacks hinder the applicability of this method when used to detect malicious accounts.

First, Wang et al. (2012a) stated that crowdsourcing method is effective if adopted by social network providers at the early stage. This shows that crowdsourcing will incur a high cost when used on social networks with a large number of pre-existing users, such as Twitter. Second, this method still requires the knowledge of experts to guarantee reliable annotation. However, not all crowd workers possess the expert knowledge needed to produce zero false alarms. Third, exposing personal data of social network users to external crowd workers may raise the issue of privacy and this can encourage even the crowd workers to exploit the concerned users (Wang et al., 2012b). Finally, several malicious crowdsourcing platforms are in existence, which negatively used their platforms to control a large number of accounts and make a huge financial gain (Wang et al., 2012b).

(2) Graph-based

The possibility of modeling social network as a graph has played a key role in identifying malicious behavior in OSNs. A graph is formally represented as $G = (V, E)$, where V is a set of vertices and E is a set of edges (Al Hasan et al., 2006; Nettleton, 2013). The interpretation of nodes and edges in social graph varies according to the problem under consideration and the modeling technique. While an edge may represent friendship invitation (Zhi Yang et al., 2015), in some cases it may denote URL links between a pair of nodes (Tan et al., 2013). The social graph can be unipartite, bipartite or tripartite. A unipartite social graph has one type of node as shown in Figure 2.6, while a bipartite or tripartite social graph considered a graph with its nodes partitioned into multiple types (Savage et al., 2014; Vlasselaer et al., 2013). This section presents the different graph-based methods used to detect malicious accounts, which include Trust propagation, graph clustering, and graph metrics and properties.

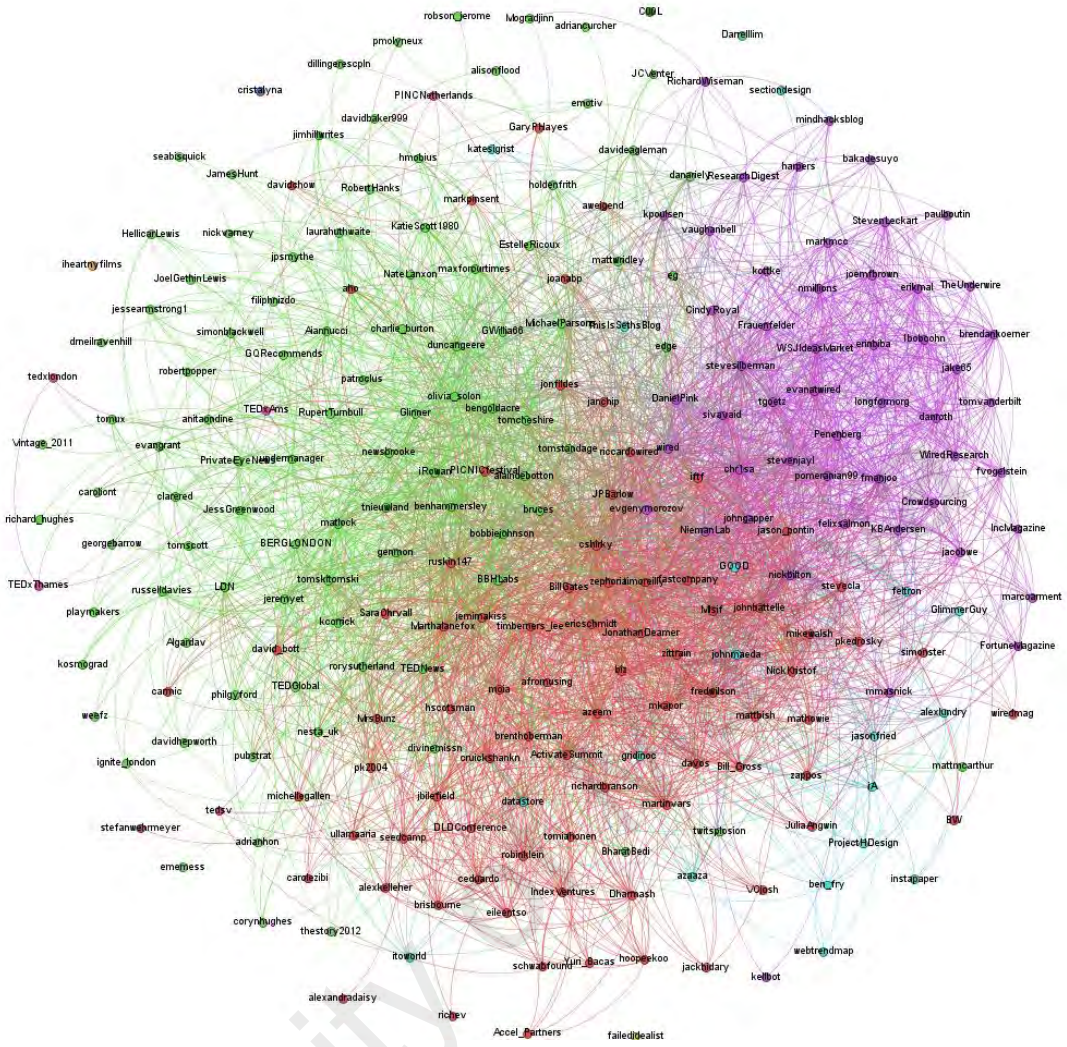


Figure 2.6: Visualization of social graph using Gephi - an open-source software for visualizing and analyzing large network graphs

(a) Trust propagation

Social graph can have two trust relationships: strong or weak trust. OSN graphs with strong trust are those that possess the property of fast-mixing (Mulamba et al., 2016; Yu et al., 2006). In malicious accounts detection problem, this can be viewed as a social network with a small cut, which represents a set of edges that when remove will partition the graph into two regions of honest and Sybil. For the sake of clarity, OSN with strong trust relationships has a limited number of attack edges between honest and Sybil regions. Conversely, a social graph with weak trust does not possess the fast-

mixing property. Another assumption similar to fast-mixing is the random expander assumption used for developing Gatekeeper algorithm (Tran et al., 2011). Mohaisen et al. (2010) demonstrated that many social networks are not fast-mixing indicating that the number of attack edges on several social networks can be in millions. Attack edges are the links between Sybil and non-Sybil regions. The link prediction problem can be used to predict such attack edges using feature similarity or social structural similarity (Mulamba et al., 2016). The former similarity measure considered the node attributes in the social graph while the latter only studies the structural link that exists between a pair of nodes. Since the goal of malicious account detection system using social graph method is to identify the misbehaving nodes, link prediction problem has been shown to perform poorly in a social network that exhibits weak trust relationship (Mulamba et al., 2016). Thus, with the use of trust propagation method, it is possible to improve detection of Sybil in OSNs.

In trust propagation method, a degree-normalized landing probability is computed and assigned to each node in the social graph. This probability corresponds to the probability of a modified random walk to land on each node. The random walk starts from a know non-Sybil node. This node distributes its trust value to the neighboring nodes. At each step of the random walk, a trust rank is computed, which indicates the strength of the trust connections that exist between the nodes. The step of random walk's probability distribution is a trust propagation process. It is important to note that a random walk can be made to terminate at an early stage; such random walk is called a short walk. A random walk that runs for a long period will produce uniform trust rank values for all the nodes in the social graph. This uniform trust value is known as the convergence value of the random walk. Random walk convergence relies on a number of steps known as the mixing time of the social graph (Cao et al., 2012; Mulamba et al.,

2016). One of the popular algorithms for computing the trust value during the random walk is power iteration (Cao et al., 2012).

In the realm of Sybil detection on social networks, random walk approach has been widely used to separate Sybil from legitimate accounts. For instance, algorithm such as SybilGuard (Yu et al., 2006), Gatekeeper (Tran et al., 2011), SybilLimit (Yu et al., 2008), SybilRank (Cao et al., 2012), and SybilRadar (Mulamba et al., 2016) used random walk technique to identify malicious nodes. It has been shown that the early Sybil detection algorithm drops significantly in performance when the number of attack edges is increased (Mulamba et al., 2016; Viswanath et al., 2011; Zhi Yang et al., 2015). SybilRadar attempts to improve the performance of the early Sybil detection algorithms by introducing a number of stages based on social structural analysis to refine the performance of SybilRadar.

A variation to initial seeds selection using both legitimate and spammers accounts for trust rank computation was demonstrated by CollusionRank algorithm (Ghosh et al., 2012). The algorithm used both known spammers and legitimate accounts as initial seeds and assigned trust and untrust values to the neighbor of these accounts. The value assigned to each account is used to depict the strength of trust and to identify other spammers on the network. However, this approach suffers from the setback of initially selecting the number of known spammers and legitimate accounts that can give a better representation of the entire accounts on the social network. Since the number of seeds is very limited taking into consideration the overall size of OSN, the initial score of the original seeds will quickly get diluted (Liu et al., 2015). This may propagate imprecise scores to many accounts on the network, which are not enough to categorize the unknown spammers or legitimate accounts.

Another algorithm (VoteTrust) that is based on trust propagation leveraged the friendship request acceptance between accounts on social network (Zhi Yang et al., 2015). VoteTrust algorithm applied power iteration to compute the trust probability. VoteTrust is based on the rationale that a Sybil node can be detected by using the friendship request acceptance from a real user. A friend invitation between node pairs is then modeled as a directed signed graph, where an edge between two nodes takes the value of 1 or -1. A value of 1 on the edge indicates that the friendship request is accepted, while -1 indicates non-acceptance. Therefore, a node B is said to cast vote on node A, if B accepts or reject a request from A. One of the advantages of VoteTrust is that the algorithm exhibits high parallelism in processing large social graph. However, in some social networks like Twitter, it is possible to launch an attack without necessarily befriending real users. This limits the capability of VoteTrust to detect some high-level malicious behavior.

(b) Graph clustering

Social graph typically shows clustering characteristics. Graph clustering method attempts to group a set of related nodes on the graph based on their similarity. Two nodes are grouped only if they are within a specific distance to each other. The resulting groups from clustering are called clusters or communities. The goal of the graph-based clustering algorithm is to group nodes into clusters by considering the edge structure of the graph in a way that increases edges within each cluster (Schaeffer, 2007). One of the widely used graphs clustering algorithms is Markov cluster (MCL). MCL accepts transition matrix from a weighted graph. By applying expansion and inflation operations, MCL iteratively clusters nodes on the graph and terminate once a stable matrix is obtained. The resulting clusters can be analyzed to detect malicious accounts (Ahmed & Abulaish, 2012). Ahmed and Abulaish (2012) extracted correlated

information from user's profile, such as the URL shared, list of friends and Facebook fanpage-likes to generate a weighted matrix for MCL algorithm. The result of the MCL clustering algorithm produces three clusters. The first cluster contains accounts classified as spam, the second cluster contains accounts classified as normal, and the third cluster contains accounts classified as both spam and normal. The authors applied a majority vote technique to resolve the third cluster with overlapping classes.

Gao et al. (2010) proposed a clustering algorithm to group wall posts into spam campaigns. The model starts by representing wall post as a pair <description, URL>, where URL is the link embedded within the wall post and description is the content of the wall post. The process connects two wall posts together if they link to the same destination URL. This produces a wall post similarity graph. The connected subgraphs in the wall post similarity graph depict clusters. Applying two widely used properties for identifying spam campaign, distributed and bursty, each cluster classified as malicious or benign. The time complexity of this algorithm limits its applicability to identify spam campaigns in the large social network graph. To address robustness against spam attack, UNIK (Tan et al., 2013) algorithm uses the assumption that the URL non-spam patterns should be identified since they exhibit a more relatively stable pattern than the spam URL. UNIK algorithm is robust to an increasing level of spam attack. However, UNIK suffers from shorten URL attack strategy and attacks coming from compromised accounts on the network.

Another domain of graph clustering focuses on detecting communities that can capture the notation of malicious and legitimate accounts clusters. Detecting community is an important step to identify malicious group, and to study the behavior of this group on the network (Mislove et al., 2010; Viswanath et al., 2011). Liu et al. (2015) proposed a community-based method, which uses a two-step process. The first step clusters

accounts into communities and the second step assigns a label to each account in the community based on the features exhibit by the accounts and the community. However, among the most noticeable challenges of community detection algorithm is lack of scalability and in some cases, community detection rarely provides provable guarantees in detecting malicious accounts (Cao et al., 2012).

(c) Graph centrality and properties

Interesting properties of social graph, such as power law distribution (Xin-fang, 2013), scale-free topological structure, small-world as well as graph centralities, assist in detecting malicious accounts in social networks (Sadan & Schwartz, 2011). Scale-free network is a network having degree distribution that follows a power law (Onnela et al., 2007). This means that the probability distribution of the number of connections between nodes in the network follows a power law distribution. This assumption also holds for clustering coefficient, the vertex connectivity between nodes, and small average path length (Sallaberry et al., 2013). Although some real-world social networks are assumed to be scale-free (e.g web graph and co-authorship), this assumption has not been generalized to all real-world social networks. As a result, the scale-free properties of many social networks are still being debated in the social network research community (Clauset et al., 2009). Graph centrality metric measures the relative importance of each node on the social graph based on position. A node with a high value is assumed to be more relevant. However, the definition of this relevancy depends on the application domain of the problem under consideration. Betweenness is a centrality metric that determines how often a node is located on the shortest path between other nodes in a social graph. The metric represents percentages of all shortest paths in a network that pass through a particular node (Sadan & Schwartz, 2011).

Betweenness centrality metric has been applied in phishing URL detection to reduce false alarm (Sadan & Schwartz, 2011). The study shows that the betweenness centrality value of whitelist domains is notably higher than the blacklist. The strength of this approach is that it provides a powerful metric and effective tool that can complement URL based anti-spam systems as well as a reduction in false positives (Sadan & Schwartz, 2011).

(3) Machine learning

Machine learning (ML) has played significant roles in identifying malicious accounts in social networks. The absorption of machine learning and data mining for data processing and information extraction produced ever-growing research areas from academic communities in the last few years. The goals of these fields focus on the techniques for classifying information and clustering data with similar characteristics. The majority of articles on malicious accounts detection focused on machine learning. ML incorporates a variety of methods, such as supervised, unsupervised, and semi-supervised learning.

(a) Supervised learning

Supervised learning is ML task of inferring a function from labeled training instances that consists of a set of observed examples. In supervised ML, an individual example is a pair consisting of an input typically a vector and the desired output value. Supervised ML analyzed training data to produce a classification model for predicting unseen data (Zheng et al., 2015). The classification model learned from ML during training is used to distinguish malicious and legitimate accounts (Singh et al., 2014). This section presents the various supervised machine learning algorithms that have been employed to detect malicious accounts in OSNs.

(i) Bayes-theorem

Bayes' theorem is a statistical theorem that describes the probability of a hypothesis based on some given conditions. The theorem provides a way to understand how the probability that a given hypothesis is true is affected by a set of evidence. Bayes theorem has applications in a wide variety of domains, ranging from topic modeling (Kharratzadeh et al., 2015) to spam filtering (Chu et al., 2012a) in social networks. NaiveBayes and Bayesian Network algorithms built on top of this theorem have shown good performance in spam account and malicious URL detection in social networks (Chen et al., 2014; Wang, 2010a; Yang et al., 2011).

For instance, Yang et al. (2011) combined network and content features to train a NaiveBayes classifier. The authors trained NaiveBayes classifier with 18 features ten (10) of which were introduced in the study. The NaiveBayes classifier achieved a detection rate of 88.6% when manually evaluated on some samples identified as spammers. Almaatouq et al. (2016) also combined content/behavioral and network features to train NaiveBayes and Bayesian Network algorithms. They applied Gaussian mixture model (GMM) to identified two categories of spammers: compromised and fraudulent accounts. The resultant clusters generated by GMM were used to construct the follower relationship graph used in the analysis. The authors extracted features around the follower relationship and contents posted to train the Bayes algorithms in addition to other five classifiers selected in the study. Chen et al. (2014) applied Bayesian Network to evaluate the discriminative power of some URL-based features. The authors examined seven features based on traditional heuristics and social network attributes of malicious URLs. They investigated the combination of features that can produce an improved classification performance when trained with Bayesian classifier. Cao et al. (2016) analyzed the forwarding patterns of malicious URL on Sina Weibo

social network. They applied URL-based features to train three classifiers with Bayesian Network achieving the highest accuracy.

(ii) Meta-based

The meta-based classifier is a family of supervised learning algorithms aimed at improving the generalization ability of the learned models. Meta-based classifier has no implementation of a classification algorithm on its own; instead, it utilizes other classification algorithms to perform the actual task. In addition, meta-based learning attempts to predict the good classifier for a given task based on the nature of the dataset. Therefore, it enables user in choosing which algorithm is suitable to apply to a given problem (Pappa et al., 2014).

Lee et al. (2010a) and Markines et al. (2009) reported the performance of this classification model on social network data. For instance, Decorate, a meta-learning algorithm for developing various ensembles of classifiers successfully detect spammers who interacted with the social honeypots deployed on Twitter and MySpace networks (Lee et al., 2010a). In this study, the authors extracted profile-based features from the accounts identified by the honeypots approach. They trained ML algorithms based on these features. Out of the ten (10) classification algorithms investigated, Decorate classifier produced the best result. Markines et al. (2009) proposed AdaBoost model to detect spam accounts in a social tagging system. This classifier outperformed LogitBoost and linear SVM with an error rate of 2%. Fire et al. (2014) developed a social privacy protector for Facebook users with Rotation Forest ensemble algorithm achieving the best accuracy among the seven (7) classifiers considered in the study. Gupta and Kaushal (2015) combined NaiveBayes, clustering, and decision tree to detect malicious users. This approach achieved high accuracy with non-spam account

detection, however, the accuracy of spam accounts identified by this meta approach needs to be improved (accuracy is 87.9%).

(iii)Support vector machine (SVM)

With the intention of reducing the error rate in classification task, while maintaining high performance accuracy, SVM is implemented to detect malicious accounts. SVM is a statistical supervised learning model that analyzes data and detects patterns using label samples. The goal of SVM is to separate the boundary between different classes in a dataset by defining a separating plane called hyperplane. This hyperplane separates the classes by maximizing the margin among the closest points known as support vectors from each class to the hyperplane. In the case of a nonlinearly separable problem, SVM uses kernel functions to find an optimal separating hyperplane. Examples of kernel functions used by SVM include linear, Radial Basis Function (RBF), and polynomial kernel.

In the domain of malicious accounts detection, several models based on SVM algorithm have been developed (Galán-García et al., 2014; Lee & Kim, 2014). For instance, Lee and Kim (2014) trained SVM algorithm with different name-based features extracted from the agglomerative clustering stage. The result of the SVM classifier shows that the model can cluster distinguished account names and classify them as benign and suspicious in order to provide a fast filter on which in-depth analysis of potential malicious accounts can be conducted. With the use of authorship identification and SVM model, Galán-García et al. (2014) identified real users behind malicious accounts used for cyberbullying attacks on Twitter. Benevenuto et al. (2009) proposed SVM classifier to identify spammers in video sharing networks (VSNs). Martinez-Romo and Araujo (2013) proposed a framework based on language model and tweet content to train SVM classifier and identify malicious tweets in a trending topic.

(iv)Neural Networks

The applicability of neural network for classifying social network accounts has also been investigated in some studies (Alsaleh et al., 2014; Igawa et al., 2016). Neural network has been used in various application domains, such as pattern recognition, disease diagnoses, image processing and speech processing. However, due to the high computational requirements of neural network, it has limited application in malicious accounts detection in social networks. Neural network, such as multilayer perceptron (MLP) has been used in the work of (Alsaleh et al., 2014). MLP is a class of feedforward artificial neural networks (ANN) that consists of activation units, usually referred to as artificial neurons and weights (Noriega, 2005). MLP modifies the standard linear perceptron by including multiple layers, such as input, hidden, and output layers to solve both linear and non-linear classification problems. The algorithm maps input data to appropriate outputs. During the training stage, MLP applies a learning algorithm, mostly backpropagation, to adjust the weights so that the network can acquire the required knowledge to classify new unseen data.

Alsaleh et al. (2014) introduced a number of content/behavioral features extracted from tweet metadata. The authors trained MLP using gradient descent (GD) method with a learning rate of 0.3. In this study, 50 nodes of neurons were used in the hidden layer with a validation threshold of 20 and a sigmoid activation function.

(v) Tree-based

Algorithms in this category exploit the power of decision tree, where a classifier is learned using a tree structure. In this tree, a node represents the test of an attribute value and a branch denotes the result of the test (Aggarwal et al., 2012; Yang et al., 2011). Decision tree algorithms, such as J48 (C4.5) and Random Forest have shown wide

acceptance in the literature for identifying spam and phishing attacks on social networks. J48 decision tree is based on C4.5 algorithm, a decision tree algorithm introduced by Quinlan in 1993 (Quinlan, 2014). This algorithm is an extension of Iterative Dichotomiser 3 (ID3). C4.5 uses information gain to select the best attribute at each node of the tree. This attribute represents the best candidate to make a decision about the splitting of the tree. Conversely, Random Forest creates an ensemble of classifiers by constructing different decision trees using random feature selection and bagging approach at training time (Chu et al., 2012a; Narudin et al., 2014).

Random Forest algorithm has improved detection accuracy of spam accounts detection system (Igawa et al., 2016; Singh et al., 2014). For instance, Aggarwal et al. (2012) used Random Forest to identify malicious tweets on Twitter network. The authors trained Random Forest using four categories of features based on profile, URL, WHOIS, and tweet contents to distinguish phishing attacks from safe links. Lin and Huang (2013) applied J48 algorithm to detect spammers on Twitter.

(b) Unsupervised learning

Unlike supervised ML approach (i.e classification), unsupervised learning used unlabeled data to build a model. As such, no specific attack behavior is known apriori. The unsupervised method groups data into different classes according to their similar characteristics. Unsupervised learning is very useful in pattern analysis and for grouping social spam into campaigns (Lee & Kim, 2014). The different unsupervised learning methods used in the literature can be categorized into five groups: Hierarchical, Partitional, PCA-based, Stream-based, and Pairwise similarity.

(i) Hierarchical

Hierarchical clustering (HC) groups data over a variety of scales using a tree structure. This tree is a multilevel hierarchy, where clusters at one level are merged or split to obtain clusters at the next level. HC is either bottom-up (i.e agglomerative) or top-down (i.e divisive). Agglomerative clustering builds hierarchy using bottom-up approach by assuming that each instance should initially form its own cluster. The algorithm then iteratively merges pairs of clusters as one move up the tree. Divisive type operates in the opposite way and assumes that all instances are initially in one cluster. The algorithm recursively splits the cluster as it goes down the tree (Kaufman & Rousseeuw, 2009).

Studies have shown that attackers collude to establish malicious group and control a large number of accounts on the network (Ahmed & Abulaish, 2012; Jiang et al., 2012). Jiang et al. (2012) developed an algorithm similar to agglomerative hierarchical clustering to detect Sybil group on Renren network. The algorithm first identifies suspicious users using popularity and social degree property. Users on the suspicious list were merged into Sybil groups based on their IP address similarity. Lee and Kim (2014) applied agglomerative hierarchical clustering to cluster users on Twitter based on their account names. To compare two names, the algorithm measures the likelihood that the names are generated from a Markov chain model. This approach detects malicious accounts at the time of creation without having to wait for the initiation of malicious behavior. One of its limitations is a lack of efficiency in providing a defense mechanism against an intelligent adversary who can launch complex attack strategy to generate valid account names on the network.

(ii) Partitional

Partitional clustering divides set of instances into non-overlapping clusters such that each instance is in exactly one cluster. K-means is an example of a prototype-based partitional clustering algorithm with many application areas (Gani et al., 2012). K-means is a heuristics-clustering algorithm that clusters dataset into user-defined clusters *K* by minimizing the sum of squared distance in each cluster. In order to use K-means algorithm, there is a need to calculate the distance between a point to its centroid, for this reason, Euclidian distance is commonly used (Yang et al., 2015).

Gani et al. (2012) proposed a framework that relies on unsupervised ML model, social interaction, and authorship analysis to identify multiple fake accounts on Twitter. Using K-means and Kohonen map algorithms, they cluster multiple groups of similar identities and perform manual verification to identify fake accounts. Kiruthiga et al. (2014) introduced an extended clone spotter algorithm that leverage clustering technique to detect a group of fake accounts using K-means algorithm. During the execution of clone spotter algorithm, K-means redistribute the identified cluster from which the closest center distance is computed and update the mean of each cluster accordingly. The authors employed two similarity distance metrics: Cosine and Jaccard to find accounts with similar characteristics based on a set of features such as age, the number of visiting friends, the total number of friends, user click patterns, and user action time period.

(iii) Principal Component Analysis

Principal component analysis (PCA) is a statistical tool for identifying patterns in high dimensional data. PCA is suitable for detecting variation in a dataset, suggesting

that it is a good candidate for malicious behavior detection in social networks (Viswanath et al., 2014).

Motivated by the need to develop a malicious account detection system without relying on apriori knowledge of attackers' strategies, Viswanath et al. (2014) proposed a PCA-based detection system. The system captures normal user behavior within three to five principal components. Any behavior that deviates from this pattern is considered as anomalous. While this approach is very promising toward identifying malicious behavior without relying on labeled data, the PCA algorithm implemented takes $O(n^3 + n^2m)$ time complexity during eigenvalue decomposition of the covariance matrix. Where *"n"* is the number of input dimensions and *"m"* is a total number of accounts considered. This computational complexity is on the high side when considering large data involved in social networks.

(iv) Stream-based

The basic idea behind the stream-based approach is motivated by the development of stream clustering algorithms to separate spam accounts from legitimate ones. Miller et al. (2014) adapted two stream-based clustering algorithms, DenStream, and StreamKM++ to detect spam accounts on Twitter. DenStream is a stream-based clustering algorithm that extends the traditional batch learning DBSCAN algorithm by defining core-micro-clusters rather than the core objects concept used in DBSCAN (Cao et al., 2016). StreamKM++ algorithm extends the K-means++ with the use of a weighted point (i.e coreset) to address the streaming data.

Miller et al. (2014) applied content features to train DenStream and StreamKM++ and achieve good performance accuracy. This approach treats spam account detection as anomaly problem, and with the use of labeled training data divided into 1500 normal

and 100 spam, the algorithms separate malicious accounts from legitimate users. However, this approach needs to be improved on a large dataset to ascertain it scalability in categorizing spammers from legitimate users.

(v) Pairwise similarity

Pairwise similarity is the method of comparing two accounts based on their activities to determine which account exhibit sudden malicious characteristics. By building a profile of legitimate behavior, it is possible to compare this behavior with incoming user's activities to ascertain whether the new user's behavior conform to the initial profile (Kontaxis et al., 2011). This method is effective for identifying anomalies in social networks. For instance, Ruan et al. (2016), studied the social behavior of users in OSNs to detect compromised account. To determine if a specific account is compromised, the authors studied the behavioral history of the legitimate owner over a specific period. They explored the clickstream activities using both extroversive and introversive user's social behavioral patterns to build behavioral model. This approach starts by applying Euclidean distance to measure the differences between two profiles. Given two profiles *P* and *Q*, which contains both extroversive and introversive feature vectors for each profile. Let $A = (a1, a2, ..., a_n)$ and $B = (b1, b2, ..., b_n)$ denote a feature vector for both *P* and *Q*. Euclidean distance between vector *A* and *B* is calculated as shown in Eq. 2.1 and Eq. 2.2 shows the computation of Euclidean norm between profiles *P* and *Q* based on the Euclidean distance for each feature vector. The higher the value of *Dist*, the more significant the two profiles differ. In Eq. 2.2, *m* denotes the number of features vectors. The authors considered eight extroversive and introversive behaviors. They further defined the concept of self-variance based on the mean differences between the pair of profiles as well as the standard deviation of the self-variance to refine the distance metric. Based on the self-variance and standard deviation,

the behavioral differences between two profiles can be determined to detect if a profile is compromised.

$$E(A,B) = \sqrt{\sum_{k=1}^{n} (a_k - b_k)^2} \quad \textbf{2.1}$$

$$Dist(P,Q) = \sqrt{\sum_{i=1}^{m} (E_i)^2} \quad \textbf{2.2}$$

Egele et al. (2015) also developed a behavioral based model using pairwise similarity method to identify compromised accounts on Facebook and Twitter. The authors extract content features from user's messages to build a user's normal behavioral profile. Any significant deviation from this behavioral profile is considered as a form of anomaly and can be used to identify compromised accounts. Using message features, such as time sent, message source, message text, message topic, link in the message, direct user interaction, and proximity, a global thresholding value is computed, which combined all the feature models. This global threshold is used to determine if a profile is compromised or not. The threshold indicates the percentage of violation of the normal user behavioral profile. Kontaxis et al. (2011) defined a similarity score based on common fields between a pair of profiles to detect fake accounts on LinkedIn network. Jin et al. (2011) proposed two statistical similarity measures using attribute and friend network similarity to cluster fake accounts on Facebook.

While this approach is promising towards identifying behavioral violation, the definition of what constitute normal user behavior is complex in the real world, especially on the social network with a diverse set of functions, such as Facebook. A slight deviation in normal user activities may create a problem for a model that relies only on pairwise similarity. As an evidence of this limitation, Egele et al. (2015) confirmed that an adversary can break their similarity measure by sending messages to

evade detection. An approach proposed in the work of (Jin et al., 2011; Kontaxis et al., 2011) relies on exact matching of fields before detecting similar identities. Therefore, it is important to fine-tune models based on pairwise similarity in order to reduce the increase in a false alarm.

(c) Semi-supervised learning

Semi-supervised learning algorithm attempts to identify a suitable classification model by combining both labeled and unlabeled data. Because of the difficulty in obtaining labeled data in most application domains, such as in the case of social networks, the semi-supervised algorithm tries to learn a suitable model by permitting a small quantity of labeled data with a large amount of unlabeled data. (Kondratovich et al., 2013; Li et al., 2013) demonstrated the applicability of this learning approach. Some popular semi-supervised learning algorithms include expectation maximization, self-training, transductive support vector machines (TSVM), and co-training (Zhu & Goldberg, 2009).

Li et al. (2013) applied TSVM algorithm to detect phishing attack. They used both image and document object model (DOM) features to train TSVM algorithm. The authors introduced quantum-inspired evolutionary algorithm (QEA) to deal with the local convergence problem of TSVM. However, TSVM suffers from a number of drawbacks, such as its difficult non-convex optimization problem and the need to estimate the ratio of positive or negative examples from the dataset.

2.3 Mobile Spam Message Detection

A number of studies have presented the current trends in spam filtering including methods that applied machine learning and those outside machine learning approaches (Carpinter & Hunt, 2006). Guzella and Caminhas (2009) focused on discussion of

machine learning approaches for spam filtering. As an example of machine leaning approach and content-based SMS spam analysis, Bozan et al. (2015), presented SMS spam filtering technique that is based upon text classification using Bayesian, SVM, K-Nearest Neighbour (KNN) algorithms. In Karami and Zhou (2014) and Delany et al. (2012), content-based SMS spam filtering have been presented, which is an active research area. Authors have proposed Bayesian model for SMS spam classification using content analysis techniques (Bozan et al., 2015; Zhang & Wang, 2009). Bayesian classification method for SMS spam filtering was also investigated in (Zhang & Wang, 2009). Yoon et al. (2010) combined content analysis and challenge-response to provide hybrid model for mobile spam detection. The content-based spam filter first classify message as spam, legitimate or unknown. The unknown message is further authenticated using a challenge-response protocol to determine if the message is sent by human or automated program. Li and Li (2007) also presented SMS spam filtering using SVM classification method. Chen et al. (2015) proposed SMS spam detection system based upon trust evaluation by analyzing spam detection behaviors and SMS traffic data. A behavioral based SMS spam filtering has also been studied in (Wang et al., 2010).

Almeida et al. (2013) introduced raw non-encoded SMS spam collection corpus known to be the largest public SMS spam dataset in the literature. The authors proposed several classification models to benchmark the dataset and found that SVM outperformed other classifiers investigated in the study. El-Alfy and AlHasan (2016) proposed a Dendritic Cell Algorithm (DCA), inspired by the danger theory and immune based systems to detect email and SMS spam messages.

While most of the existing studies on SMS spam message detection have focused on traditional content-based analysis and complex processes for feature extraction, this

research proposed a slightly different approach from the traditional bag-of-word models to investigate a number of lightweight features that can improve the performance of the proposed SMS spam detection model. A sentiment analysis approach that addressed the polarity of messages has been presented in (Ezpeleta et al., 2016). However, the performance of this sentiment analysis model still needs to be improved. Thus, motivated by the feature extraction method in (El-Alfy & AlHasan, 2016), this thesis presents a novel approach that benefits both mobile and microblogging social network when addressing the problem of spam message filtering in SMCM.

2.4 Risk assessment

Cyber-attacks on information technology infrastructure have prompted the prioritization of critical events. As many attacks, such as social media threats, have surfaced recently, the trend of cyber-attacks has begun to experience the worse scenarios and can result in very real physical damages. The growth of these categories of damages is a concern for social network users that has resulted in the re-imagination of defensive mechanisms and processes globally (Karchefsky & Rao, 2017). Unfortunately, the efforts to reduce the negative effects of risk taking-behavior in social media have not yet been able to turn the tide, as there has been a steadily increase in the amount of cyber-security incidents according to a study on risk-based security (Wallen, 2015). Monitoring cyber-security incidents through risk assessment have been recently adopted (Anuar, 2012; Karchefsky & Rao, 2017).

Risk is defined as the effect of uncertainty on objectives. It is often expressed in terms of a combination of the consequences of an event and the associated likelihood of occurrence (ISO, 2009). Risks are events with potential hazard, having some probability of occurrence and an impact. The impacts may include financial, reputational harm, and many more. Therefore, risk management in the cyber-security context refers to

identification of cyber threats and planning of controls to mitigate the effects of those threats. One of the cyber threats that require risk assessment is social spam account, which can be used to exploit legitimate users and undermine trust relationship (Echeverría & Zhou, 2017). The goal of using risk management in the cyber-security critical infrastructure is to provide cyber resiliency for those systems. National Institute of Standards and Technology (NIST) and International Organization for Standardization (ISO) presented comprehensive definitions of risk management as follows:

(a) NIST defines risk management as a comprehensive process that requires organizations to establish the context for risk-based decision, assess risk, respond to risk once determined, and monitor risk on an ongoing basis using effective organizational communications and a feedback loop for continuous improvement in the risk-related activities of organizations (NIST, 2013).

(b) ISO defines risk management as "the systematic application of management policies, procedures and practices to the tasks of establishing the context, identifying, analyzing, assessing, treating, monitoring and communicating" (ISO 31000:2009).

Risk assessment is a risk estimation process based upon a combination of the likelihood and consequences of an event, as well as the relationship between risk and uncertainty (Anuar, 2012). Several risk assessment models have been successfully studied in different domains, such as incidence prioritization, intrusion response system, business management, supply chain risk management, and risk management in engineering constructions. Risk assessment helps an analyst to identify, evaluate, quantify, and mitigate the consequences and impacts of risk. According to Haimes (2015), risk assessment facilitates the decision-making process of a security analyst.

In the domain of computer security, risk management is viewed as a systematic process aids to identify, mitigate and control cyber-security risks (Wallace, 2016). Looking at this from social spam account issues, however, risk is more particularly the likelihood that an attacker threat will endanger or affect some legitimate users. Information security management relies so much on risk assessment as its core competence and several standards and frameworks have been previously introduced to provide flexible and adaptable layouts for risk management systems. Some of the popular standards include the NIST SP 800-53 (NIST, 2013), the ENISA Evaluation Framework for National Cyber Security Strategies (ENISA, 2014), and the Robust ICS Planning and Evaluation framework (Karchefsky & Rao, 2017). For instance, Figure 2.7 shows the ISO 31000:2009 risk management process, which comprises of different stages of risk management. The context of the risk is first established and the risk is then identified, analyzed, evaluated, and treated. In analysis stage, the nature of risk is understood and the level of risk is determined. Meanwhile, in evaluation stage, the result of risk analysis is compared with risk criteria to determine whether the risk is acceptable (Rijal, 2016). Other frameworks for risk management include NIST Special Publication 800-53 for managing information security risk, ISO 27001, and COBIT 5 (Karchefsky & Rao, 2017).

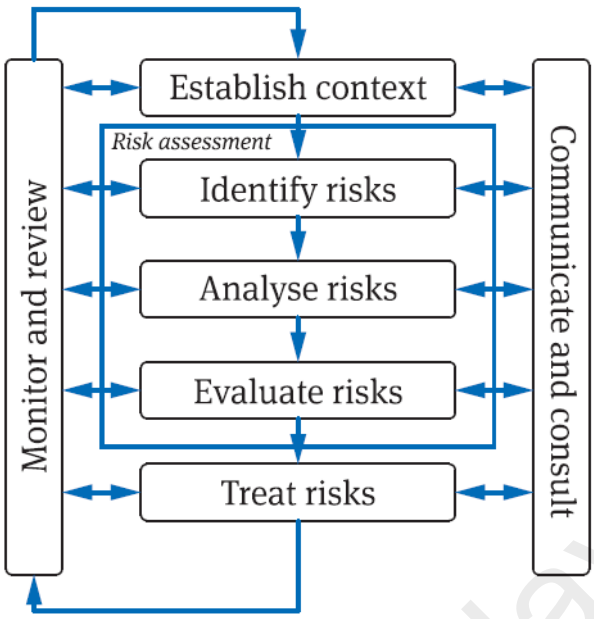


Figure 2.7: ISO 31000:2009 risk management process

According to Anuar (2012), risk assessment offers several advantages in monitoring cyber-security incidents, such as:

(i) *Systematic techniques*: Risk assessment follows systematic procedures in identification of risk incidence as well as the consequences and how to efficiently manage them.

(ii) *Availability of different factors*: Risk assessment has different factors to assist in effective decision-making process, such as assets and values analysis. It provides means to identify threats and vulnerabilities as well as offering management control and cost-benefit estimation.

(iii) *Easy to adopt*: With the availability of various risk assessment standards and frameworks, risk assessment has been considered easy to adopt in any organization. Furthermore, it provides flexible means of prioritizing incident. Risk assessment is easy to understand by different levels of management.

(iv) *Appropriate responses to risks*: Risk assessment helps decision makers to apply qualitative or quantitative approaches to identifying the risk of incidents. It allows the discovery of high risk incidents by considering their priority, urgency and importance. Thus, to counter these incidents, different responses can be provided.

(v) *Usability of results*: Risk assessment always produces consistent results based on the same decision factors employed for risk incident evaluation. This in turn allows sharing of information between network and organizations. It also ensures that the incident prioritization process covers a wide range of networks.

(vi) *Easy to understand results*: While incident prioritization process provides systematic techniques for ranking incident, sometimes the results produced by such prioritization procedures are not user-friendly and can be difficult to comprehend. However, the use of risk assessment provides platforms for sharing assessment results with other third parties. Thus, the diversity in results presentation offers a straightforward and realistic way to aid different levels of management in decision-making.

A number of articles have attempted to study risk in different context. For instance, in mobile SMS spam domain, Zainal and Jali (2015) proposed a perception based model to assess the risk associated with mobile messages using danger theory of artificial immune systems. Email spam risk assessment system has been studied in (Bates & Illg, 2011). Rijal (2016) presented a study on the risk assessment matrix (RAM). RAM is a widely used method for risk assessment, which defines various levels of risk as the product of the harm probability and harm severity. RAM enables observers to understand the consequences of the risk and as well as the probability of likelihood that the risk will occur. Consequences of the risk are placed in row of the RAM while the likelihood probabilities are placed in column as shown in Figure 2.8. The probability or likelihood of occurrence of risk is categorized as very likely, likely, moderate, unlikely

and rare. Based on the effect or damage caused by risk, consequences can be categorized as trivial, minor, moderate, major or extreme. Trivial and minor consequences can refer to risks that cause negligible amount of damage to the overall system. Moderate refers to risks that result in few damages, which has less significant impact and does not impose great threat. Major or critical consequence refers to the threat that can results in significantly large damages to the system and can result in significant amount of losses. An extreme consequence refers to risks that can totally destroy the system. In risk management, much attention is giving to major and extreme risks that could cause significant damages to the overall system. Rijal (2016) further presented risk assessment in cybercrime as well as solutions to some possible risk consequences.

		Impact				
		Trivial	Minor	Moderate	Major	Extreme
Probability	Rare	Low	Low	Low	Medium	Medium
	Unlikely	Low	Low	Medium	Medium	Medium
	Moderate	Low	Medium	Medium	Medium	High
	Likely	Medium	Medium	Medium	High	High
	Very likely	Medium	Medium	High	High	High

Figure 2.8: Risk Assessment Matrix (RAM)

In OSNs where people shared emotions and thoughts by sending messages, photos, and short video clips, it is very likely that such behaviors are deeply involved with risks. For instance, clicking on unidentified link or multimedia file could cause severe impact, such as system infection, leakage of personal information, spam propagation, and even financial loss. One of the most critical challenges is how to mitigate those risks on these

networks (Echeverría & Zhou, 2017; Yoon & Lee, 2016). However, from privacy point of view, the security risks of social networking sites have expanded greatly. Yoon and Lee (2016) presented the effect of providing information of the unknown's trustworthiness, such as clicking unknown URL and possibility of hidden relationship on this risk-taking behavior. The study shows how trust can be transferred on social network to change user's attitude towards responding to unknown requests that can lead to malicious attacks.

As opposed the previous related studies, part of the objectives of this thesis is to introduce spam risk assessment model (SRAM) that incorporates method capable of accommodating the uncertainty in risk modeling. The proposed approach in this thesis specifically focuses on risk assessment in microblogging social network using Twitter microblog as a test bed. The goal is to assess individual accounts on Twitter based on their likelihood to endanger legitimate users through their patterns of interactions. A behavioral model is studied along with an approach to rate, rank, and categorizes accounts on Twitter.

2.5 Summary

This chapter first presents OSNs including the definition, categorizations and social network datasets. A discussion on the social network datasets that have been utilized to detect malicious users and their activities was underlined. It then presents the current state-of-the-art features for malicious accounts detection in OSNs, which include social structural analysis, content/behavioral analysis, as well as hybrid analysis. Extensive discussions on the related studies have been highlighted in each category. The goal is to identify areas for improvement through the exploration of hybrid feature learning approach to achieve the study objectives. It is evident that studies on hybrid feature analysis still need improvement through identification of more relevant features to

counter evasion. The chapter further provides the taxonomy of the different methods that have been used to finally build a spam detection model. The taxonomy by methods was grouped under three main headings: crowdsourcing, graph-based, and machine learning. Similarly, the chapter highlighted the related state-of-the-art studies that explored each category. Based on the result of an investigative survey, it is evident that majority of the studies on malicious account detection in OSNs focused on machine learning method.

Furthermore, the chapter also provides related studies on mobile SMS spam detection. Over the past few years, research in SMS spam detection has concentrated on content-based analysis using techniques like VSM, bag-of-words model, and TF-IDF. Different classifiers have been deployed to provide effective system for mobile spam message detection. Although other areas such as hybrid analysis have also been studied. However, there is a little effort in identifying lightweight features to detect spam messages, particularly with a focus on mobile and microblogging social network.

As part of the objectives of this thesis is to rate, rank, and categorizes microblogging social network accounts based on their risk level. This chapter highlights the current efforts on risk assessment. As far as the author is aware, no comprehensive work has been done on risk assessment for microblogging social network. Therefore, in order to propose a framework to satisfy the study objectives, there are many other aspects that need to be explored. In the next chapter, more issues will be discussed to understand the various ways that spammer can launch different attacks on SMCM.

CHAPTER 3: SPAM DETECTION AND RISK ASSESSMENT IN SMCM - THE ISSUES

Incident investigation reports have indicated that cyber-attacks, such as targeted attacks or advanced persistent threats (APTs) often use SMCM, such as Twitter microblog, to collect personal information and launch social engineering attacks (Echeverría & Zhou, 2017; Miller et al., 2014; Varol et al., 2017). In other words, the convenience of Twitter microblog facilitates potential cyber-threats. For instance, a social network based worm spreads by attempting to steal account information and infect additional users using a social engineering trick, which sends malicious links in spam messages. Because OSN users typically trust their friends, they sometimes responded by clicking malicious links that rapidly spread worms through the friendship networks of victims. Malware applications often leverage short URL to mask original destination and evade security systems, such as blacklist filtering inspections. URL shortening service providers often find themselves blacklisted due to the abuse of short URL by malicious users. According to Zhang et al. (2012), frequently used URLs are either of high value or are spam. Several evidences have shown that spammers used automated tools such as social bots to automatically post spam messages (Chu et al., 2012a; Ferrara et al., 2014).

Therefore, unlike in email spam distribution, it is impossible to spam an individual unless a spammer posses a valid email address for that individual, and as a result, spammers expand considerable efforts developing mailing lists of valid email addresses (Bates & Illg, 2011). However, in social network deception may be used to obtain addresses or other private information due to the structural connection of social network users. Thus, spam posts containing malicious URLs are faster and more effective to use

on SMCM like Twitter. This is possible as far as the content of the post presents hot topics, it can catch the attention of many victims.

In many instances, tools for spam detection rely on filtering that attempt to identify and block potential spam messages or accounts. Such filters typically are based upon analysis of the contents of the communications. In this instance, spammers have developed strategies such as good word attacks that make it difficult to detect spam. For example, spammers often purposely misspell words that might trigger spam detection. Therefore, existing spam filters faced a continual cat and mouse game with spammers. Before presenting the proposed framework for spam detection and risk assessment in this study, it is imperative to analyze the existing strategies used by spammers to evade detection. This chapter provides a review of issues on the impact of the rise in social bots on Twitter SMCM as well as social engineering attacks and feature evasion. The chapter concludes by identifying issues with existing approach used for risk assessment, which is based on Analytic Hierarchy Process (AHP).

3.1 The rise of spam bots in SMCM

Twitter is a popular SMCM and social networking service released in 2006. Twitter enables users to post and read short messages usually known as tweets. The possibility of embedding several entities such as hashtag, mention, and short URLs, has greatly improved communication on this platform (Chu et al., 2012a). Users on Twitter microblog utilize hashtag to group tweets according to topics, such as the case of #RioOlympics2016, a popular topic discussed on Twitter during the 2016 Rio Olympic Games. A topic can be categorized as trending, if it receives many attentions from the users on the network. For example, #JustinBieber is one of the popular topics in 2011 on Twitter. Mention feature uses the "@" symbol to indicate the users who can receive tweet directly on their timelines. Studies have shown that spammers employ mention

tool for target attack since the Twitter microblog featured a unidirectional user binding (Hu et al., 2013). Although Twitter has introduced features to deactivate unsolicited mention, a majority of the users on Twitter still utilize default account settings. The visibility of a tweet on the network is increased through a process of re-tweeting. Re-tweeting a user's tweet has been identified as another strategy used by spammers to keep their accounts running (Lee et al., 2010a). In addition, spam accounts exhibit automated posting patterns since there is a need for spammers to get across to a large number of users on the network (Chu et al., 2012a). Twitter microblogging social network has become an important platform for real-time communication (Al-garadi et al., 2016), however, it has gone through several cases of abuses in the hands of social spammers. To identify malicious users, Twitter introduced a number of rules to suspend accounts with abusive behaviors. A comprehensive list of Twitter rules can be found in (Twitter, 2016). Even though Twitter has published a number of rules to suspend accounts on its network, the rise of social bots for posting malicious contents is still on the high side. In addition, Twitter suspension algorithm is slow in identifying malicious users and social bots (Chu et al., 2012a; Cresci et al., 2017; Echeverría & Zhou, 2017; Ferrara et al., 2014).

The history of bot is dated as far back as 1999 during the evolution of Pretty Park, a worm that can listen to malicious commands (Atluri & Tran, 2017). In 2003, Spybot was created, which introduced many new functionalities such as keylogging, data mining, and instant messaging spam. In the same year, Rbot that introduced Distributed Denial of Service (DDOS) and information stealing tools surfaced. Rbot employed compression and encryption to evade detection system. The year 2004 witnessed the rise of Bagle and Bobax, the first spam botnets. In 2009, the first and most influential social bot, Koobface, attacked Twitter social network. Koobface attacks by spreading messages that contained links to malicious websites, leveraging social network

information sharing, as well as social network applications as the means of spreading malware (Grier et al., 2010). Koobface forces users to download fake plug-in, which is the Koobface downloader that attempted to detect the type of OSN the user is using and immediately carry out the infection. According to Atluri and Tran (2017) bots are implemented using different topologies:

(i) *Star*: This topology allows bot to interact directly with its master. This method facilitates bot management and ensures that interactions between bot and its master are fast and accurate. However, the problem with this topology is the single point failure, which allows system administrators to block the connection of the bot to its master.

(ii) *Multi-server*: This architecture is a more robust than Star topology. The topology addresses the issue of single point of failure in Star topology. It ensures that the bots can easily get across to its closest geographical master. However, this architecture requires significant effort to set up.

(iii) *Hierarchical*: The hierarchical architecture permits a bot to function as a supervisor for a group of other bots. The supervisor bot can directly connect to the master and update instructions/code base. This approach prevents the bot master from being visible on the network and makes tracing back to the master more complicated. However, the real-time attack is harder in this topology compared with other architectures due to the additional level of latency added during updates between bots.

(iv) *Random or Peer-to-Peer*: This topology is the most advanced architecture in bot implementation. It allows individual bot agent to send or forward commands to the next bot in the network. This makes it difficult to detect the bot master, because communication between bots would be difficult to trace. Nevertheless, the design

enables researchers to track down the infected hosts easily by analyzing the communication of individual bot with others.

The openness of Twitter and increase in the number of accounts created on the network has made it an ideal platform for exploitation from automated software called social bots. Using Twitter API, a social bot can perform virtually most human tasks. Legitimate bots produce a large volume of legitimate tweets, such as news and blog updates, which complies with the Twitter's objective of functioning as a news and information network. However, malicious bots have been greatly exploited by social spammers to distribute spam contents. Malicious bots arbitrarily add users as their friends with the expectation that some of them will follow back. If legitimate users on Twitter are surrounded by malicious bots and spam tweets, the activities of these social bots will eventually hurt the entire Twitter community.

Chu et al. (2012a) shows that between human and social bots are cyborgs, which refer to either human-assisted bots or bot-assisted humans. These bots categories have become popular on Twitter with the goal of distributing malicious contents on the network. Some of the criteria for identifying a social bot are listed as follows:

(a) *Lack of intelligent or original content*: One of the characteristics of a bot is retweeting other users' tweets or posting contents that lack intelligent or originality, such as adages.

(b) *Automation*: Another characteristic of a bot is excessive automation of tweeting, such as RSS feeds and blogs updates.

(c) *Malicious URLs*: Malicious bots usually post excessive unsolicited contents with malicious links, which may be used for phishing or malware distribution.

(d) *Duplicate contents*: Another characteristics of a bot is repeated posting of duplicate tweets. This can be accomplished using Twitter API or other sophisticated tools.

(e) *Unrelated external contents*: A bot can post links to external web contents, which are mostly unrelated to the tweets descriptions.

(f) *Aggressive following*: Bots engage in aggressive following by adding more friends to their accounts in order to gain attention from human users. This is usually done within a short period.

Ferrara et al. (2014) presented a review of bot detection studies. The authors discussed various methods that have been used to identify bots in OSN. They further introduced a system for bot detection that achieved AUC of 95%. The system is publicly available online for evaluating Twitter accounts as bot or not bot (Davis et al., 2016). Echeverría and Zhou (2017) established that a large number of Twitter users are social bots, which are designed to send spam messages, manipulate public opinion, and undermine the basic function of Twitter API. The study uncovers more than 350K Star Wars botnets designed using Star topology with their bot master located centrally. It has been shown that Twitter bots contaminate Streaming API by automating their tweets so that they can be included in the API with probability as high as 82%. The authors revealed that Twitter bots often quote from book or online resources, such as the case of Star Wars bots that quoted several sections of Star War novels. Another study by Varol et al. (2017) claimed that about 48 million users on Twitter are not human showing the rise of social bots on Twitter (see Figure 3.1). The investigation suggests that between 9% and 15% of active Twitter users are social bots. Using cluster analysis, the authors revealed different categories of social bots. They further propose a system for bot detection based on supervised machine learning approach using different combinations of features to train machine learning classifiers. The bot detection system produced AUC of 95% based on Random Forest classification algorithm.

Figure 3.1: The rise of social bots (source: www.dailymail.co.uk)

Recent study on Twitter social spam bots detection has argued that there is a paradigm shift in spam bots behaviors, which rendered existing approaches less effective to identify spam bots (Cresci et al., 2017). First, the authors investigated the capability of the current Twitter bots detection method in identifying new social spam bots. They further assess the capability of human in distinguishing between legitimate accounts, social spam bots, and traditional spam bots. Their findings show that neither Twitter, nor humans, nor existing approaches are currently capable of accurately detecting the new social spam bots. This calls for a new approach that is capable of turning the tide in the fight against the rise in social spam bots. Although some social bots distribute benign contents such as news and information on natural disasters, however, the goal of this thesis is to focus on detecting those Twitter accounts that are specifically used for malicious activities.

3.2 Social engineering threats

Social engineering attacks involve the tricks used by malicious users to lure their victims. Attackers have used different strategies to overpower legitimate users. For instance, spammers utilized social engineering tactics to steal the credentials of legitimate users and eventually compromise their accounts (Egele et al., 2015). Information stolen from legitimate users can be used to create fake accounts in order to deceive the friends of the real users (Bilge et al., 2009) or to send customized spam messages (Ezpeleta et al., 2015; Fire et al., 2014). This section presents a number of social engineering tricks that have been successfully used by spammer to abuse Twitter network.

(a) *Phishing*: Despite the efforts of Twitter at notifying users about phishing attacks, however, the nature of the links shared on this platform enables spammers to successfully used phishing trick. Spammers continue to compromise existing accounts by sending messages to fool users into clicking on harmful links that will lure users to external pages where their login credential is hijacked. Phishing attacks in social network come in many different dimensions. For instance, there is a case of message with phishing link claiming to be from the social network service providing certain update or contest (Wüest, 2010). Therefore, the user will have to provide his login credentials to receive the update. At first, user is not aware of the landing page that intends to compromise his credentials due to the nature of the URL shared on Twitter. There are also other techniques of distributing phishing links where the user is presented with an interesting content like "*You look different in this photo. cuts.pX/8*". Grier et al. (2010) established that over 2 million URLs were found on Twitter directing users to scam, malware, and phishing sites, which accounts for about 8% of all links distributed on the network. In August of 2009, about 11% of tweets posted on Twitter were spam. In 2009, a number of legitimate users' accounts

were hacked to distribute advertisement. For instance, in 2010, the Twitter accounts of Press Complaints Commission as well as BBC correspondent Nick Higham, were hijacked to distribute phishing links.

(b) *Advanced fee scam*: Twitter is an interesting platform for advanced fee fraud. Malicious user can easily target potential victim that will usually fall for the scam by exploring the user's private information that are publicly available online. The scammer can then amend the intention that the selected social engineering trick will exploit. This scam usually come with information that will promise the victim some benefits. The scammer will later inform the user of certain problem and request for some amount of money to be paid up front. Once the money has been paid, the scammer disappears or refuses to respond to the victim. Several cases of advanced fee fraud have been cited (Wüest, 2010). This type of fraud shows how user's personal information on social media can be misuse by scammers.

(c) *Fake followers and friends*: Sometimes the popularity of social network user is based on the number of friends and followers the user can attract on his profile. As the benefits of social media continue to grow, the pressure on users to get as many friends and followers as possible became attractive. In some instances, social acceptance is usually based on the number of connections in the user's network. This gave rise to different platforms for underground markets where users can purchase fake followers to boost their social reputations (See Figure 3.2). Other platforms include buytwitterfriends.com, tweetsourcer.com, unlimitedtwitterfollowers.com, twitter1k.com, socialkit.com, usocial.net, tweetcha.com, autotweeter.in, fastfollowerz.com, intertwitter.com, and twittertechnology.com (Cresci et al., 2017; Yang et al., 2013). Some platforms offer free services that required users to supply their usernames and passwords to get many new followers on a daily basis. Through this social engineering trick, a number of accounts have been compromised and

majority of legitimate users have indirectly added spammers on their networks. Most of these fake followers are artificially crafted to post malicious contents (Cresci et al., 2017). Another social trick used by spammer is to automatically send thousands of friends' requests believing that some of them will follow back (Chu et al., 2012a). The sales of fake accounts have become a multimillion-dollar business. In fact, the so-called celebrities, politicians, and popular brands have purchased fake accounts from underground markets to boost their profiles (Cresci et al., 2015). This type of risk-taking behavior further allows malicious accounts to spread across the social networks.

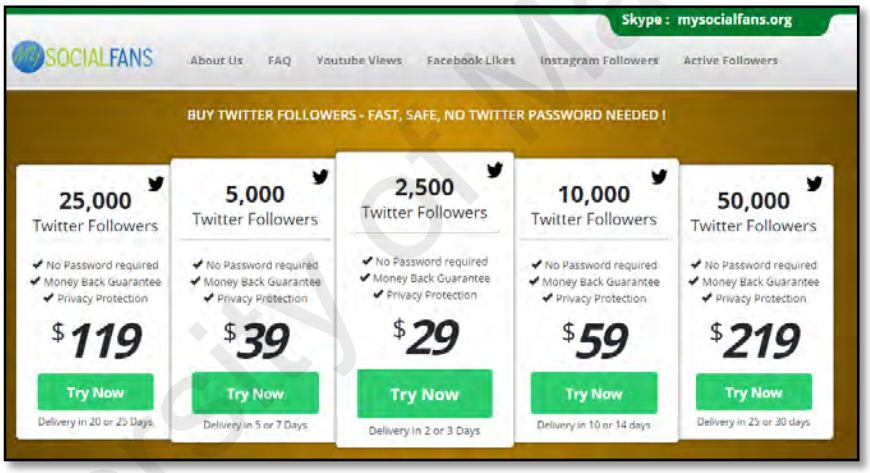


Figure 3.2: Prices of fake Twitter followers from http://www.mysocialfans.org/

(d) *Identity theft and Impersonation*: Identity theft is a social engineering trick that allows malicious users to steal another person's personal information and impersonate as the legitimate user by using his identity. A significant number of users have been victims of identity theft leading to large expenditures of resources to recover their identities (Abeer et al., 2016). According to a recent report from Javelin Strategy and Research, the total number of identity fraud victims has grown to about 13 million per year and around $112 billion has been stolen in the past six

years (Javelin Strategy & Research, 2016). Social spammers make about $200 million every year constituting to a loss in social trust, productivity and profit. There have been reports of some fake profiles of celebrities created on various social networks (Wüest, 2010). Since there is nothing stopping attacker from registering a new account under the name of a celebrity, the public available photos of celebrity is used along with other basic personal information from online resources to create the new fake account and attract followers and friends within a short period. These friends and followers can later be spammed. Attackers have successfully used other fake celebrities' accounts to get in touch with real celebrities, pretending as their friends. Thus, identity theft and impersonation attacks could lead to harmful effects to the real owner of the identity.

(e) *Malware distribution:* The popular social media, such as Twitter, represents the ideal target environment for malicious users to spread their viruses and malware with minima efforts. This is achieved by embedding the virus in applications or redirect users to malicious websites where they can be forced to install the malware. Therefore, thousands of users can be easily affected just by distributing such malicious links on the network. One of the cases of successful malware distribution on Twitter is Koobface worm, which spreads through propagation of malicious links (Thomas & Nicol, 2010). We have also seen the cases of malware distribution through malicious campaigns such as game, ringtone, and fake music downloads (Gao et al., 2010).

(f) *Fake donation and weight loss*: Another successful social engineering trick on Twitter is fake donations. Users would be asked to donate certain amount of money for a cause. There are also cases of fake weight loss scam that enticed the victims with some photographs posted along with short messages, which contain link to malicious website (see Figure 3.3).

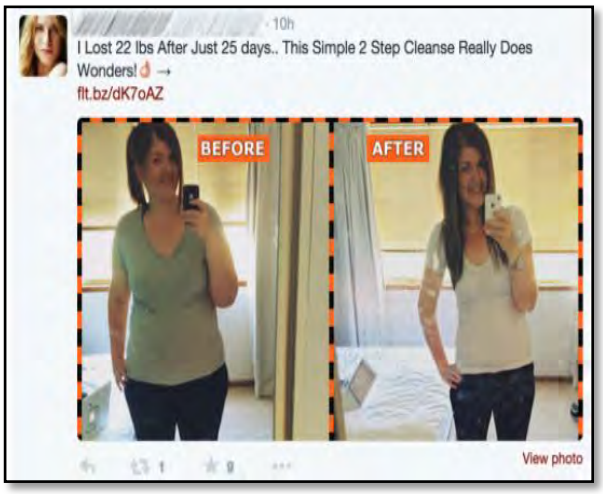


Figure 3.3: Fake weight loss scam

3.3 Content analysis and feature evasion

One of the characteristics of social spammers is that they adopt sophisticated strategies to evade existing detection approaches, such as those models that relied on textual content of tweets posted on the network (Cui, 2016; Lee et al., 2010a). As spammers became more sophisticated to detect using early-detection approaches, researchers tend to explore the use of hybrid analysis to counter feature evasion (Wu et al., 2016; Yang et al., 2013). Yang et al. (2013) presented the different methods used by spammers to evade early-detection systems that relied on textual content and behavioral analysis. These methods are highlighted as follows:

(a) *Profile-based feature evasion*: Profile-based feature are usually extracted from account profile information or meta-data as provided by the Twitter API. Example of profile-based features includes the number of followers and the number of tweets posted by the user. Traditionally, these features indicate the level of popularity of Twitter accounts. For instance, a user with more followers and tweets are deemed to be more influential. The high degree of these features also suggest that more users will trust these accounts and would possible prefer to receive information from

them. Features such as the ratio of followings to follower (FoFo) and the ratio of the number of followers of an account to the sum of the number of followings and followers have been explored (Lee et al., 2010a; Wang, 2010c). However, to evade these features, spammer can buy more followers from underground markets or exchange followers through malicious collaboration. Spammers can also create more fake accounts, which can be used to follow their spam accounts. To evade feature based on the number of tweets, attackers can utilize automated tools, such as AutoTweeter, to increase their posting patterns.

(b) *Content-based feature evasion*: As discussed earlier, spam accounts employs phishing trick to lure legitimate users by including malicious links in their tweets, which can direct users to scam websites. In addition, spam accounts can post duplicate tweets with different short URLs that land the victim to the same malicious website. Based on this evidence, researchers have introduced feature such as tweet similarity to identify such category of spammers (Lee et al., 2010a; Wang, 2010c; Yang et al., 2013). However, spammers designed new technique to evade such detection systems by utilizing automated tools to post heterogeneous tweets. In some cases, words with similar semantic are used to evade content-based detection systems. Cui (2016) applied content-based features to identify spammers on Twitter; however, the low detection accuracy of this system further confirms the effectiveness of evading content-based features using the aforementioned techniques. Based on these evidences, it is imperative to investigate the current-state-of-the-art features that can be used to identify social spammers.

3.4 Analytic Hierarchy Process (AHP) and Risk assessment

Analytic Hierarchy Process (AHP) is one of the most popular prioritization techniques that has been successfully used in both security (Anuar, 2012) and non-security domains (Mustafa & Al-Bahar, 1991; Saaty, 2008). AHP is a mathematical

technique for multi-criteria decision-making. Complex problems or issues involving quantitative and qualitative (i.e subjective) judgments are suitable applications of the AHP method. AHP was proposed by Thomas L. Saaty in 1980 to help solve problems with multiple levels of hierarchies and select the best alternatives. AHP relies on pairwise comparisons and uses expert judgment to derive priority scales (Saaty, 2008). Over the last few years, we have witnessed successful application of AHP is several areas, such as supply chain risk assessment (Schoenherr et al., 2008), project risk assessment (Mustafa & Al-Bahar, 1991), risk assessment in incidence response system (Anuar, 2012), and selection of automobile purchase model (Byun, 2001). AHP has helped decision maker to solve complex problems and also as a method for structuring complexity (Forman & Gass, 2001). AHP has been found as the most suitable method for prioritization based on its comparison with other approaches like spanning tree matrix, bubble sort, binary search tree and priority groups (Anuar, 2012; Karlsson et al., 1998). AHP employs a ratio scale and has the ability to facilitate a synthesized process (Forman & Gass, 2001). During AHP pairwise comparison, the decision maker examines two alternatives by comparing one criterion with another and indicates a preference. The comparison is made using a preference scale, which assigns numerical values to different levels of preference. The standard preference scale used for AHP is a crisp value between 1 to 9 where 1 indicates equal importance and 9 indicates extreme importance. In AHP comparison matrix, the value 9 means that the criterion under consideration is extremely more important than the other while 1/9 indicates extremely less important or preferred (Özdağoğlu & Özdağoğlu, 2007). The crisp scale of judgment used by AHP between criteria is shown in Table 3.1:

Table 3.1: Saaty AHP fundamental scale for judgment

Intensity of Importance	Definition	Explanation
1 2	Equal Importance Weak or slight	Two indicators contribute equally to the objective.
3 4	Moderate Importance Moderate Plus	Experience and judgment slightly favour one indicator over another.
5 6	Strong Importance Strong Plus	Experience and judgment strongly favour one indicator over another.
7 8	Very Strong or demonstrated Importance Very, very strong	An indicator is favoured very strongly over another; its dominance demonstrated in practice.
9	Extreme Importance	The evidence favouring one indicator over another is of the highest possible order of affirmation.
Reciprocals of above	If indicator *i* has one of the above non-zero numbers assigned to it when compared with indicator *j*, then *j* has the reciprocal value when compared with *i*.	A reasonable assumption.
1.1 - 1.9	If indicators are very close	May be difficult to assign the best value but when compared with other contrasting indicators, the size of the small numbers would not be too noticeable, yet they can still indicate the relative importance of the activities.

The basic steps in AHP involve: 1) definition of objective 2) structuring of elements into criteria, sub-criteria, and alternatives 3) making pairwise comparison of elements in each group 4) calculating weighting and consistency ratio 5) evaluating the alternative according to weighting. Figure 3.4 shows an example of hierarchy in AHP. The first level defines the goal or objective of the overall project analysis. The second level comprises of the different criteria used for judgment based on the project goal. The third level defines the various alternatives to be finally weighted. At level two, AHP allows more sub-criteria to be defined to further model the complexity of the task.

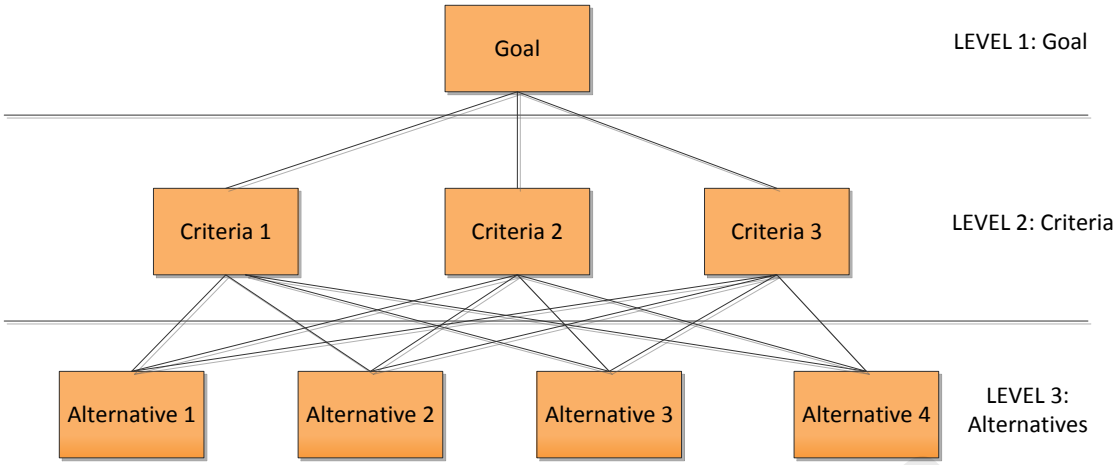


Figure 3.4: AHP hierarchy

Formally, AHP pairwise comparison matrix is given as:

$$A = \begin{bmatrix} 1 & a12 & \dots & a1n \\ a21 & 1 & \dots & a2n \\ \dots & \dots & 1 & \dots \\ an1 & an2 & \dots & 1 \end{bmatrix} \qquad \textbf{3.1}$$

Where $a_{ij} = 1$, if i is equal to j, and $a_{ij} = 3, 5, 7, 9$ or $1/3, 1/5, 1/7, 1/9$, if i is not equal j. To determine the consistency of the judgment matrix, Saaty defined a Consistency Ratio (CR) as follows:

$$CR = \frac{CI}{RI} \qquad \textbf{3.2}$$

$$\text{where, } CI = \frac{\lambda_{\max} - n}{n-1} \qquad \textbf{3.3}$$

CI is called Consistency Index, RI is Random Index, n is the number of indicators, and $\lambda_{\max}$ is obtained from the largest Eigen value of the pairwise comparison matrix. Table 3.2 is proposed by Saaty to obtain the value of RI based on the number of indicators considered in the AHP analysis. Therefore, according to Saaty (2008), if the CR value is less than 10%, then the value can be considered as a reasonable and acceptable judgment, otherwise, the judgment matrix is not consistent and thus needs to be re-modified.

Table 3.2: Random Index (RI)

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RI	0	0	0.52	0.89	1.11	1.25	1.35	1.40	1.45	1.49	1.52	1.54	1.56	1.58	1.59

Table 3.3: AHP judgment matrix with three factors/criteria

Criteria	C1	C2	C3
C1	1	3	2
C2	1/3	1	1/5
C3	1/2	5	1

The most commonly used methods in AHP for computing the weights of criteria as well as the alternatives are eigenvector and row geometric mean methods. Table 3.3 shows an example of AHP pairwise judgment matrix with three criteria. AHP as a multi-criteria decision making method could help security analysts rate and rank incident according to the degree of importance or severity. Some of the advantages of AHP to security analysts are:

(i) It allows multi-criteria decision making where multiple criteria or factors can be utilized to make a choice in a multiple-criteria environment. The decomposition ability of AHP method permits a complicated problem to be organized into a hierarchy of criteria or sub-criteria.

(ii) AHP allows different weightings of criteria and sub-criteria during prioritization process.

(iii) The ability to use homogenous clusters of criteria in AHP allows complex problem to be easily modeled.

(iv) AHP provides easy way to measure both objective and subjective factors.

(v) The ratio scale produces as a result of estimation process allows decision makers to distinguish between results in a statistical manner as well as their consistency. This

ratio scale has been shown to be more powerful than other theories, which rely on ordinal or internal scales (Anuar, 2012; Forman & Gass, 2001).

(vi) AHP allows decision criteria to be combined to generate a more complex result as well as facilitate analysis of the decision goals.

Although AHP has been used in many domains to solve complex multi-criteria decision-making problems, it suffers from certain drawbacks, most especially for risk assessment. Since risk management involves dealing with a lot of uncertainties, AHP method does not permit modeling of expert uncertainty or impreciseness in judging the criteria. In addition, AHP is used in nearly crisp decision applications and deals with a very unbalanced scale of judgment. Therefore, existing studies on risk management deviated from this important requirement. Thus, this study proposed a modified version of AHP called Fuzzy Analytic Hierarchy Process (FAHP) that can accommodate uncertainty or impreciseness in human judgment. The SRAM model proposed in this thesis for risk assessment is based on FAHP analysis in order to improve the performance of the spam risk assessment model.

3.5 Summary

This chapter established the main challenges confronting existing spam detection systems specifically in SMCM. The chapter started with an extensive discussion of the rise of spam bots in Twitter SMCM. It presents brief history of bots, bots architecture as well as the characteristics of social spam bots that have contributed to the success of spam bots operations. The chapter discusses issues regarding social engineering tricks utilized by spammers to exploit their victims, some of which have negative impacts on the performance of the existing detection systems. The discussion on social engineering threats covers areas such as phishing, advanced fee fraud, purchase of fake followers

and friends to boost reputations, identity theft and impersonation, malware distribution, fake donations and weight loss scam.

Furthermore, issues with content and behavioral based features for spammer detection were discussed centering on the complex techniques used by spammer to evade those features. In conclusion, the chapter presents AHP multi-criteria decision making method as well AHP theoretical backgrounds, applications, and limitations for risk assessment. Thus, the issues identified are highlighted to serve as guidelines for developing a framework that improves the existing approaches.

CHAPTER 4: UNIFIED FRAMEWORK FOR SPAM DETECTION AND RISK ASSESSMENT

This chapter presents the details architecture of the proposed unified framework in this thesis. The aim of the framework is to investigate the features that can be used to detect spam message and spam account in SMCM as well as categorizing accounts on Twitter microblog based on their risk level. To achieve the goal of introducing a system that can detect both spam message and spam account within a single framework, the proposed framework studied hybrid features analyses, which allow extensive investigation of the different categories of features for spam detection. The discussion of the chapter continues with a detailed description of the models proposed for both spam message and spam account detection as well as the novel spam risk assessment model. The chapter concludes by discussing the risk index computation method, rating threshold as well as the proposed response strategy.

The proposed framework also aimed to improve the limitations of existing content-based detection systems by applying a slightly different approach to content analysis for spam message detection. In addition, this study combined different models within a single framework as opposed the existing studies. The goal is to provide robust framework for spam detection and risk assessment that will prevent evasion of spam filter. As shown in Figure 4.1, the proposed unified framework consists of two main models: Spam Message and Spam Account Detection Model (SMSADM) and Spam Risk Assessment Model (SRAM). The SMSADM contains two sub-models namely Spam Account Detection Model (SADM) and Spam Message Detection Model (SMDM). Both SADM and SMDM occupied the upper layer of the proposed framework. The bottom layer addresses the spam risk assessment for Twitter accounts. The output of the SADM in terms of the investigated features is passed to the SRAM

model to effectively prioritize and categorize Twitter accounts based on their risk level. The information from SADM are used along with other decision criteria to develop the SRAM model. Figure 4.1 also shows the hierarchy of connection established in the proposed framework between the two models.

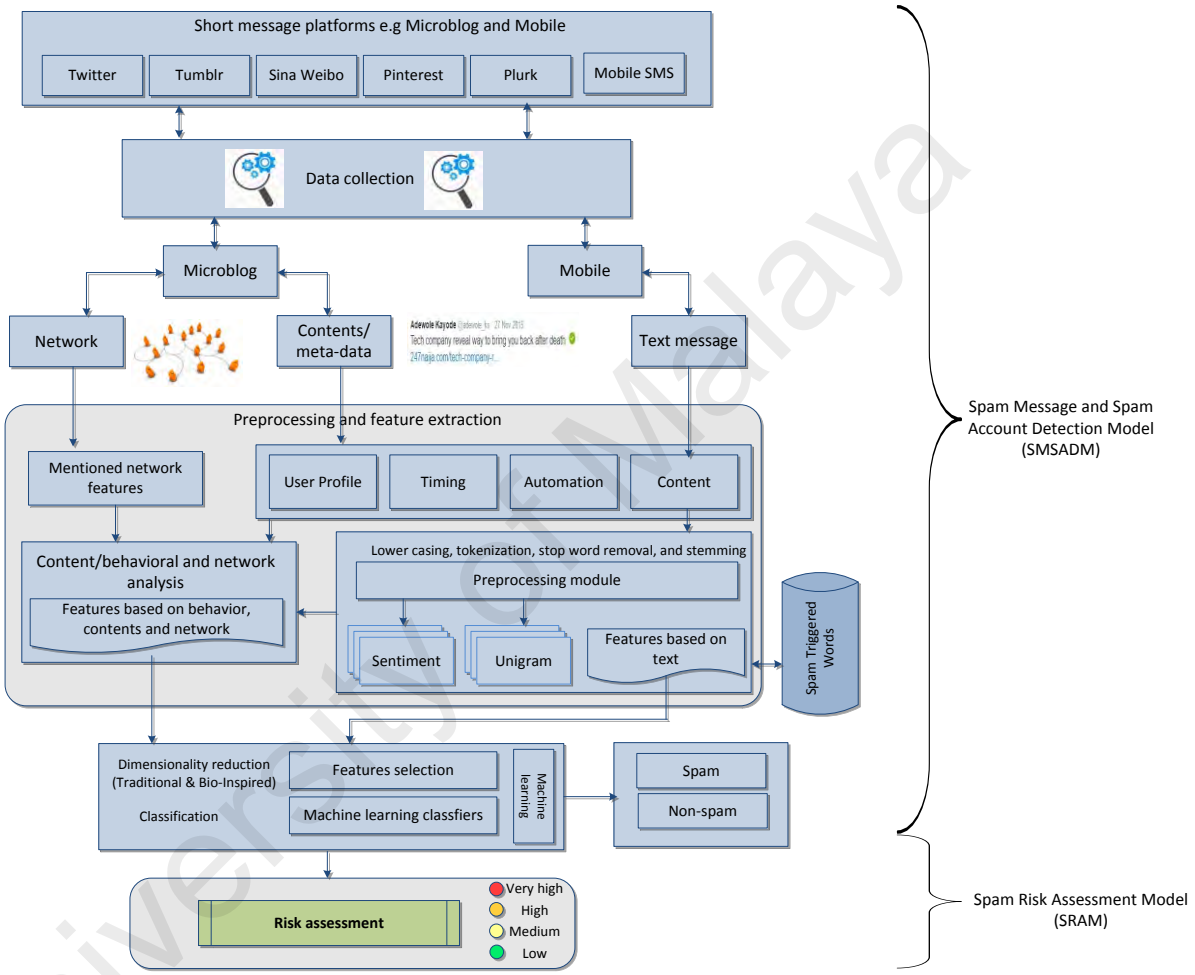


Figure 4.1: The proposed unified framework

The proposed SRAM model is established based on Fuzzy Analytic Hierarchy Process (FAHP), decision criteria and a list of discriminative indicators identified in the top-level model (i.e SADM). Furthermore, SRAM model presents methodical approach to provide appropriate response in the risk assessment based on the account analyzed.

The model maps different types of response options based upon their prioritization procedure.

4.1 Spam Message and Spam Account Detection Model (SMSADM)

To address the problem of detecting both spam account and spam message using a single framework, this study proposed SMSADM, which comprises of two sub-models: SADM and SMDM. The compositions of these models are discussed in details in the subsequent sections.

4.1.1 Spam Account Detection Model (SADM)

SADM explored a unified feature learning approach considering five categories of features: user profile, content, mention network, timing, and automation. The reason for using these features is to explore a hybrid features learning rather than features based only on content analysis. The goal of SADM is to detect spam account on microblogging social network. Twitter microblog is selected as a test bed for evaluating the proposed model due to its openness and robust API for data collection. As an overview of the proposed SADM, the model development starts from data collection from Twitter using the Twitter API (Twitter rate limit, 2015). Before extracting the necessary features for SADM, data collected from Twitter are passed to a pre-processing module. Since features such as Term Frequency-Inverse Document Frequency (TF-IDF) is proposed as well as sentiment features; this stage breaks the textual content into unigram model in order to extract both the TF-IDF and sentiment features. A unigram is an ngram model whose size is 1. For instance, the unigram of the statement "*I hate spammer*" is *'I', 'hate', 'spammer'*. Unigram model is employed in this study due to the small size of the textual contents posted on SMCM. TF-IDF is a numeric weighting approach that is applied to score the importance of a word in a document based on its frequency of occurrence in that document as well as the given

collection of documents. Documents in this domain refer to the bunch of tweets texts or mobile SMS messages that are treated as textual contents. The main idea behind TF-IDF measure is that if a word appears more frequent in any specific category of documents, this implies that such word should be important and should as well be giving a high score. However, if a word appears in many categories of documents, it is possibly not a unique identifier, thus such a word should be assigned a lower score. This can assist in modeling word usage pattern between spammers and legitimate users. The TF-IDF score is calculated as follows:

$$tf - idf(t,d,D) = tf(t,d) * idf(t,D) \qquad \textbf{4.1}$$

Where t represents the terms, d denotes each document, and D represents the collection of documents. The first part of the equation *tf(t,d)* is computed as the number of times each term (i.e. word) appeared in each document. Terms such as stop words are removed and all words are converted to lower cases before computing TF-IDF score. The second part *idf(t,D)* is a global value and it is calculated as follows:

$$idf(t,D) = \log \frac{|D|}{1 + |\{d \in D : t \in d\}|} \qquad \textbf{4.2}$$

Where $|D|$ is the size of the document space, the denominator $|\{d \in D : t \in d\}|$ is the total number of times the term t appeared in all the documents and the plus 1 at the denominator is used to avoid divide-by-zero error. After the pre-processing and features extraction stages, a suitable machine learning classifier is identified for the proposed SADM by exploring different machine learning algorithms.

(1) Feature analysis

A critical stage in developing effective classification model is the identification of features that can separate one class from another. The use of machine learning approach to identify spammers on social networks depends on many factors. The most important

factor is the identification of features that can distinguish spammers from legitimate accounts. In this study, the focus is on five main categories of features as earlier stated: user profile, content-based, network, timing and automation, which amount to 69 features in total. Therefore, this section provides the detailed descriptions of each feature category used to build the SADM model.

(1) *User profile features:* The user profile features have been considered for spam account detection in the work of Yang et al. (2013). The features captured the basic profile information of an account, such as the number of followers, the number of friends, and so on. The values of these features are extracted from the meta-data returned from Twitter microblog. The user profile features capture the behavioral changes of an account based on its profile contents. For instance, Lee and Kim (2014) established that the length of the screen name of spammers is usually longer than legitimate users. Table 4.1 shows the user profile features used in this study with the additional features introduced to complement the existing ones.

Table 4.1: Description of user profile features

Feature name	Description	Reference
Screen name length	The length of the screen name based on characters.	Lee and Kim (2014)
User location	The presence or absence of profile location.	Proposed
Profile URL	Whether the user includes URL or not in his profile.	Proposed
Age in days	Age of the account in days.	Zheng et al. (2015)
Followers count	Number of followers of the user.	Yang et al. (2013)
Friends count	Number of friends/followees of the user.	Miller et al. (2014)
Statuses count	Total statuses of the account.	Proposed
Favourites count	Number of tweets the user has favorited.	Miller et al. (2014)
User description	Indicating presence or absence of profile description.	Aggarwal et al. (2012)

Table 4.1, continued.

Default profile	When true, indicates that the user has not modified the theme of their profile.	Proposed
User Time zone	Indicates presence or absence of time zone.	Proposed
Account verified	Indicates whether the account has been verified or not.	Chu et al. (2012a)
Default profile image	When true, indicates that the user has not changed the default profile egg avatar.	Alsaleh et al. (2014)
Listed count	The number of the public lists the user is a member.	Miller et al. (2014)
Geo-enabled	Indicates whether or not the user has enabled the possibility of geotagging their tweets.	Proposed
Account reputation	Normalized ratio of followers to friends.	Shyni et al. (2016)
Follower following ratio	Ration of the number of follower to friends.	Yang et al. (2013)
Following follower ratio	Ratio of the number of friends to followers.	Zheng et al. (2015)

(2) *Content-based features:* Content-based features study the behavioral patterns of Twitter accounts around the tweets posted by the users. Studies have shown that spammers lure their victims to click malicious links embedded within the tweets. Thus, the accounts of the victims are compromised upon visiting the malicious website (Grier et al., 2010; Yang et al., 2013). Many social spammers dedicate their efforts posting duplicate tweets. In addition, they employed automated tools to post tweets with very similar semantic (Yang et al., 2013). Based on this evidence, a set of statistical features is designed as shown in Table 4.2 to evaluate the classification results of the selected classifiers.

Table 4.2: Description of content-based features

Feature name	Description	Reference
Total tweets	Total tweets sent by the user.	Yang et al. (2013)
Total hashtag	Total number of hashtag used.	Shyni et al. (2016)
Total link	Total number of link posted.	Miller et al. (2014)
Total mention	Total number of users mentioned.	Shyni et al. (2016)
Total retweet	Total number of retweet.	Miller et

Table 4.2, continued.

		al. (2014)
Hashtag ratio	Ration of total hashtags to total tweets	Yang et al. (2013)
Link ratio	Ratio of total links to total tweets.	Yang et al. (2013)
Mention ratio	Ratio of total mention to total tweets.	Yang et al. (2013)
Retweet ratio	Ratio of total re-tweet to total tweets.	Yang et al. (2013)
Total tweet favorite count	The number of time the user's tweets has been favorited.	Proposed
Deviation of hashtag	Population deviation of hashtags.	Proposed
Deviation of link	Population deviation of links.	Proposed
Deviation of mention	Population deviation of mentions.	Proposed
Deviation of re-tweet	Population deviation of retweets.	Proposed
Deviation of tweet length	Population deviation of tweet lengths.	Proposed
Deviation of hashtag position aggregate	Population deviation of hashtag position aggregate.	Proposed
Deviation of link position aggregate	Population deviation of link position aggregate.	Proposed
Deviation of mention position aggregate	Population deviation of mention position aggregate.	Proposed
Average daily tweet	Ratio of the total tweet to the number of days between first and last tweets posted.	Proposed
Average tweet length	Mean of tweet length.	Proposed
Average sentiment polarity	Mean of sentiment polarity for each tweet posted.	Proposed
Average sentiment subjectivity	Mean of sentiment subjectivity for each tweet posted.	Proposed
Average TF-IDF score	Mean of TF-IDF weight of the tweets.	Proposed
Popularity ratio	Ratio of the sum of total tweets favorite and total re-tweet to the number of tweets posted.	Proposed
Tweet similarity	Similarity of the tweets text using cosine similarity.	Yang et al. (2013)
Unique URL ratio	Ratio of unique URLs posted to total tweets.	Yang et al. (2013)
Duplicate tweet count	Number of duplicate tweets posted.	Yang et al. (2013)
Unique hashtag	Total number of unique hashtags used.	Shyni et al. (2016)
Unique mention	Total number of unique mentions.	Shyni et al. (2016)
Maximum frequency of hashtag	Maximum value of hashtag frequency.	Shyni et al. (2016)
Average frequency of hashtag	Mean of hashtag used.	Shyni et al. (2016)
Average frequency of mention	Mean of mentions used.	Proposed
Average frequency of URLs	Mean of URLs posted.	Shyni et al. (2016)

(3) *Network-based features:* This study investigates the mention network of users on Twitter as opposed the followers' network used in Yang et al., (2013), which can be easily compromised by purchasing fake followers from underground market. This network helps to understand the mention patterns of users on Twitter social network. As shown in Figure 4.2, mention patterns of Twitter accounts is categorized into four: malicious mention collaboration, random target attack, reflexive reciprocity, and legitimate collaboration. The mention network captures the connections or interactions among users on the Twitter microblog. We modeled users' mentions as a graph $G = (V, E)$, where V represents the vertexes and E the edges corresponding to the mention links between users. If a user u mentions user v in his tweet, we construct an edge $u => v$, which indicates a direct link between u and v. Thus, the graph G is a directed graph that modeled users' mention patterns. A set of graph-based network features are extracted from graph G and some network features based on the neighborhood as defined in the work of (Yang et al., 2013). Table 4.3 shows the network features used in this study, some of which are described as follows:

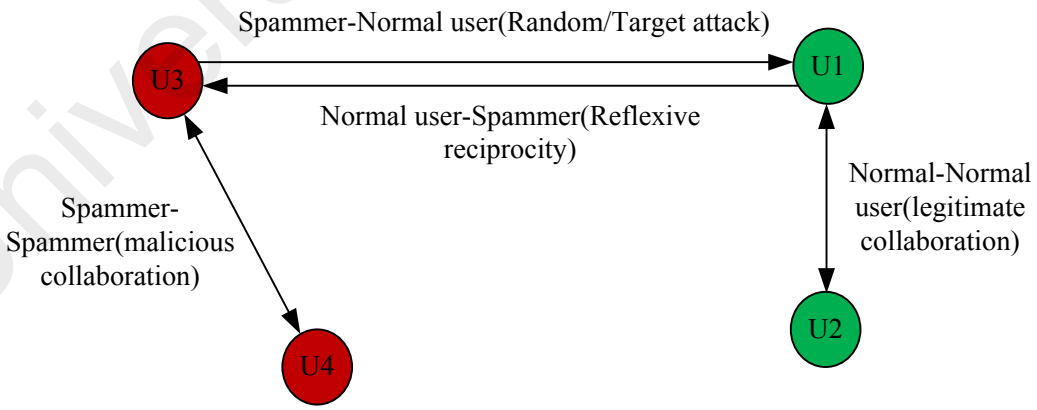


Figure 4.2: Accounts mention patterns

Table 4.3: Description of network features

Feature name	Description	Reference
Average neighborhood followers	Ratio of sum of the followers of a user's friends to the number of friends of the user.	Yang et al. (2013)
Average neighbor tweets	Ratio of the sum of tweets of a user's friend to the number of friends of a user.	Yang et al. (2013)
Local clustering coefficient of mention	User's local clustering coefficient based on mention network.	Proposed
Betweenness centrality of mention	Betweenness centrality of user based on mention network.	Proposed
Bidirectional link of mention	Bidirectional link of user based on mention network.	Proposed
Bidirectional link ratio of mention	User's bidirectional link ratio from mention network.	Proposed
In-degree of mention	User's In-degree from mention graph.	Proposed
Out-degree of mention	User's Out-degree from mention graph.	Proposed
Degree reputation of mention	Degree reputation based on mention network.	Proposed
Degree centrality of mention	Degree centrality of user from mention graph.	Proposed
Closeness centrality of mention	User's closeness centrality based on mention network.	Proposed
Eigenvector centrality of mention	Eigenvector centrality of user mention network.	Proposed
Pagerank of mention	User's Pagerank from mention graph.	Proposed

(i) *Local clustering coefficient of mention:* This is a useful metric to determine how close a vertex's neighbors are to being a clique. A clique is a small group of accounts with shared interests. As opposed the work of (Yang et al., 2013), this study focuses on extracting the graph-based features around the mentioned network, which enables the author to study the mention relationships among Twitter accounts. For each vertex in the mentioned graph G, its local clustering score can be computed with Eqn. 4.3, where K_u is the sum of the in-degree and out-degree of the vertex, and e^u is the total number of edges built by all u's neighbors. It was noticed that the local clustering coefficient of spammer based on mentioned network is

smaller compared to legitimate users. The reason may be that spammer mentions target users randomly and these accounts may not know each other in reality.

$$LCC(u) = \frac{2|e^u|}{K_u(K_u - 1)} \qquad \textbf{4.3}$$

(ii) *Betweenness centrality of mention:* This is a centrality measure that uses shortest paths to compute the strength of a vertex in the graph. The metric is obtained using Eqn. 4.4, where σ_{st} is the total number of shortest paths from node *s* to *t* and $\sigma_{st}(u)$ is the number of those paths that pass through the vertex *u*. *n* is the total number of nodes in graph *G*. Similar to the behavior of spammer as identified in the local clustering coefficient of mentioned network, it was also noticed that betweenness centrality of spammer is smaller than the legitimate users.

$$C_B(u) = \sum_{s \neq u \neq t \in V(G)} \frac{\sigma_{st}(u)}{\sigma_{st}} \qquad \textbf{4.4}$$

(iii) *Bidirectional link of mention:* This feature defines the total number of links reciprocated by those users mentioned in the tweets. Because spammers randomly mention users in their tweets to launch target attacks, they tend to receive low bidirectional links from the account mentioned as compared to legitimate accounts.

(iv) *Bidirectional link ratio of mention:* Defines the ratio of the number of bidirectional link of a vertex to the total number of out-degree of the vertex. The value is usually low for spammers and high for legitimate users.

(v) *In-degree of mention:* Defines the total number of edges that enters a node. It is computed using Eqn. 4.5. The value is low for spammers and high for legitimate users.

$$d^{in}{}_u = \sum_{[v,u]} G(v,u) \qquad \textbf{4.5}$$

(vi) *Out-degree of mention:* Represents the total number of edges that leaves a node. It is computed using Eqn. 4.6. The value is high for spammers and low for legitimate accounts. The reason is that spammers tend to mention more users for target attacks than legitimate users.

$$d^{out}{}_u = \sum_{[u,v]} G(u,v) \qquad \textbf{4.6}$$

(vii) *Degree reputation of mention:* This is the normalized ratio of the In-degree to the Out-degree of a vertex. The value of degree reputation of mention for spammers is low compared to the degree reputation of legitimate users. The feature is computed as shown in Eqn. 4.7.

$$dr(u) = \frac{\left| d^{in}{}_u \cup d^{out}{}_u \right|}{\left| d^{in}{}_u \right|} \qquad \textbf{4.7}$$

(viii) *Degree centrality of mention:* Defines the sum of the total In-degree and Out-degree of a vertex. The degree centrality of spammers based on the mention network is low compared to legitimate accounts as observed. Eqn. 4.8 shows how to compute degree centrality for a vertex.

$$\deg(u) = d^{in}{}_u \cup d^{out}{}_u \qquad \textbf{4.8}$$

(ix) *Closeness centrality of mention:* Measures the importance of a vertex based on how close a given vertex is to the other vertices in the graph. The most center vertices are important as they can reach the whole network more quickly than non-central vertices. This can be utilized to measure the quality of the connection of a node within the network. Closeness centrality metric can be obtained using Eqn. 4.9, where *d(v,u)* is the distance between vertices *v* and *u*. It was noticed that the closeness centrality of spammers based on the mention network is low compared to legitimate accounts.

$$C(u) = \frac{1}{n-1} \sum_{v \in V(G)} d(v,u) \qquad \textbf{4.9}$$

(x) *Eigenvector centrality of mention:* This is useful for measuring how the centrality of a node depends on its neighbors' centralities. The metric does not only measure how the vertex is position within the network, but also the quality of the links built with the vertex neighbors. Eqn. 4.10 shows how the eigenvector centrality is computed from the mentioned graph, where $EC(v_j)$ is the eigenvector of the vertex v_j connected to u, $A=[a_{ij}]$ is the adjacency matrix, and λ is a constant. The EC of one vertex relies on the *EC* of another vertex it is connected to. The *EC(v)* is calculated by finding the eigenvector associated with the highest eigenvalue according to Perron-Frobenious theorem (Ferrara & Fiumara, 2012). The i^{th} entry of the vector corresponds to the eigenvector centrality score of i^{th} vertex. The value of eigenvector of spammers is low compared to legitimate users.

$$EC(u_i) = \frac{1}{\lambda} \sum_{j=1}^{N} a_{ij} EC(v_j) \qquad \textbf{4.10}$$

(xi) *PageRank of mention:* The Google PageRank is a modified version of eigenvector centrality metric. PageRank of a vertex u relates the PageRank of the vertex it is connected to in the graph. Eqn. 4.11 shows how PageRank score is obtained from the graph G, where $d = 0.85$ is the damping factor. N is the total number of vertices considered in the mentioned graph, $PR(v_j)$ is the PageRank of the vertex v_j, $M(u)$ is the set of vertices that link to vertex u. $L(v_j)$ is the number of outbound links of vertex v_j. Similarly, it was found that spammers have low PageRank score compared to legitimate accounts.

$$PR(u) = \frac{1-d}{N} + d \sum_{v_j \in M(u)} \frac{PR(v_j)}{L(v_j)} \qquad \textbf{4.11}$$

(4) *Timing-based features:* This deals with the tweeting rate and following rate of an account. The features examine the posting and following patterns of users on the

Twitter microblog. These features have been studied in the work of (Shyni et al., 2016; Yang et al., 2013). Table 4.4 shows the description of the two features in this category. Spammer follows a large number of users and generates more tweets than the legitimate users.

Table 4.4: Description of timing-based features

Feature name	Description	Reference
Following rate	Ratio of the number of friends to the age of an account.	Yang et al. (2013)
Tweeting rate	Ratio of the total number of tweets to the age of the account.	Yang et al. (2013)

(5) *Automation-based features:* Similar to the timing-based features, this study adopted automation features utilized in (Yang et al., 2013). Yang et al. (2013) established that spammers resolved to using automation technique for posting tweets due to the high cost of manually maintaining many spam accounts. The technique relies on the use of API to post a large number of spam tweets on the network, thus, spammers' accounts exhibit a high rate of automation. In this regards, a higher API ratio implies automation behavior, which provides an indicator to flag the account as suspicious. Table 4.5 shows the description of automation-based features adopted in our study.

Table 4.5: Description of automation-based features

Feature name	Description	Reference
API ratio	Ratio of the number of tweets sent using API to total number of tweets.	Yang et al. (2013)
API URL ratio	Ratio of the number of tweets sent using API that contains URL to the total number of tweets sent using API.	Yang et al. (2013)
API tweet similarity	Number of similar tweets sent using API.	Yang et al. (2013)

(2) Bio-inspired features identification

This study explored the possibility of identifying the most discriminating features among the several features proposed for SADM model. To assess the discriminating power of the features, this study employed bio-inspired evolutionary computation paradigm using evolutionary algorithm (EA) search method. This algorithm is employed due to its wide acceptance and ability to identify good solution within the search space (Bhattacharya et al., 2016). Initially, to achieve this goal, a combination of traditional search approach using Chi-square test statistics is utilized alongside the EA search method. The purpose of identifying the most discriminating features for SADM is to assist during the SRAM model development using the proposed FAHP approach. This is well-explained in section 4.2 during the spam risk assessment model development.

Chi-squared test feature selection evaluator is implemented using a ranker search method. Chi-squared statistics (χ^2) tests the independence of two events, *A* and *B* where the independence is defined as $P(AB) = P(A)P(B)$ or $P(A|B) = P(A)$ and $P(B|A) = P(B)$. In the case of feature selection, the algorithm assumes that the two events are the occurrence of feature and class. The features are ranked using Eqn. 4.12:

$$\chi^2 (D,f,c) = \sum_{e_f \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_f e_c} - E_{e_f e_c})^2}{E_{e_f e_c}} \qquad \textbf{4.12}$$

where *N* is the observed frequency in *D* and *E* is the expected frequency (Stanford University, 2008). After executing the Chi-squared test method, EA is used to finally identify the most discriminating features. EA is a generic meta-heuristic optimization search approach that concurrently explores numerous points in a search space, and navigates the search space stochastically in order to prevent the search exploration from being trapped at the local maxima (Manurung, 2004). EA utilizes biologically inspired

evolution mechanisms, such as recombination, mutation, fitness, and selection. The detail operations of EA, according to Oliveira (2014), are represented in Figure 4.3. The basic generic structure of EA algorithm is described as follows:

Step 1: **Initialization**

For time $t=0$, initialize a population $P(t)$ such that $P(t) = (x_1^t, x_2^t, ...x_n^t)$. These are the initial points, which the EA will use to explore the search space. In the case of feature selection, the population corresponds to the different features subsets selected from the original features.

Step 2: **Evaluation**

At this stage, each solution in the initial population is evaluated by measuring its fitness.

Step 3: **Selection**

This step creates a new population by stochastically selecting individuals from $P(t)$.

Step 4: **Evolution**

At this stage, the algorithm transforms some members of the new population created in Step 3 using genetic operators, such as crossover and mutation, to form new solutions.

Step 5: **Testing for termination**

Steps 2 to 4 are repeated until the termination condition is satisfied. The EA algorithm may terminate if a given number of iterations is reached, a particular fitness value has been achieved, or when the algorithm converges to a near-optimal solution.

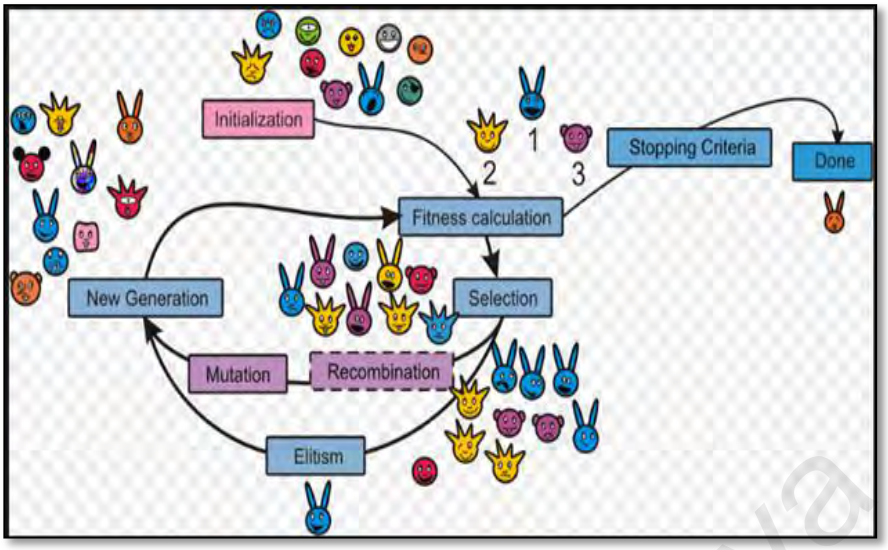


Figure 4.3: Operation of EA

4.1.2 Spam Message Detection Model (SMDM)

This study adopts a slightly different approach to extract features for the proposed SMDM model. Motivated by the feature extraction process in (El-Alfy & AlHasan, 2016), in total, this research extracts eighteen (18) features for spam message detection in SMCM as shown in Table 4.6. This feature extraction provides a compact representation of each of the collections utilized for spam message detection. Unlike VSM and bag of words models where features for spam message detection are represented using the words in each corpus either by adopting a unigram, bigram or ngram approaches, this study proposes a different method that provides compact representation of the text messages. The proposed SMDM is capable of detecting spam message on mobile and Twitter microblogging platforms. The subsequent section discusses the features used for spam message detection in details.

(1) SMS spam detection features

To extract features for spam message detection model, SMDM, each instance of the message in the corpus used in this study is passed through a preprocessing module.

Table 4.6: List of features extracted for spam message detection

Feature name
Frequency of Comm100 spam words
Frequency of ultimate spam words
Frequency of 438 spam words
Frequency of 100 worst spam words
Frequency of combined spam words
Message length in character
Number of words
Frequency of money words
Frequency of money symbols
Number of words in capital
Frequency of function words
Number of special character
Number of emoticon symbol
Number of links
Frequency of phone number
Average number of words
Number of sentence
Sentence ratio

The preprocessing module first converts the message to lower case and then proceeds to tokenization phase in order to separate the message by the words it contains using the unigram approach. The tokenized collection is processed by removing stop words that will not provide any significance contribution to the final representation. The stop words removal stage is followed with stemming process, which allows us to generate the base or root form of each word in the corpus. For instance, the word *"buying"* is reduced to *"buy"* and the words *"credited"* and *"crediting"* are both reduced to *"credit"*. The stemming stage was implemented using the Porter stemming algorithm embedded in NLTK package in python. After completing the preprocessing steps, 18 features are extracted. These features enable a compact model to be developed for spam message detection that will benefit both mobile and microblogging platforms and address the limitations of the existing traditional bag-of-words models. These features are discussed as follows:

- *Frequency of spam triggered words:* 257 list of spam triggered words and phrases were collected from Comm100 website at (https://emailmarketing.comm100.com/email-marketing-ebook/spam-words.aspx), such as *urgent*, *call now*, and *free access*. Comm100 provides global enterprise-level customer service and communication solutions. It has been shown that spammers tend to use more spam words and phrases when composing spam message (El-Alfy & AlHasan, 2016). Similarly, a list of 393 spam words and phrases described at HubSpot blog (http://blog.hubspot.com/blog/tabid/6307/bid/30684/The-Ultimate-List-of-Email-SPAM-Trigger-Words.aspx#sm.00000h35svjkfxez7rh42q7pa3mpp), a list of 438 spam words and phrases at Automational blog (http://blog.automational.com/2016/03/08/spam-trigger-words-to-avoid/), and a list of 100 spam triggered words and phrases at Benchmark blog (http://www.benchmarkemail.com/blogs/detail/the-100-worst-spam-words-and-phrases) were also collected. The frequency of spam words that appear in each message is computed as a feature. For instance, the frequency of comm100 spam words, frequency of ultimate spam words, frequency of 438 spam words, and frequency of 100 worst spam words presented in Table 4.6 are calculated from the spam words and phrases collected from Comm100, HubSpot blog, Automational blog, and Benchmark blog respectively. In addition to these spam words and phrases, a list of spam words from each spam message that appear in the different corpus used in this study is selected. Thus, the frequency of combined spam words is calculated from this list.
- *Message length in character:* This is the length of each message based on the number of characters present in the message.

- *Number of words:* This feature represents the total number of words in the message. For instance, the message *"Act now to win cash price"* contains six (6) words.
- *Frequency of money words:* In some situations, spammer tries to overpower legitimate users by sending unsolicited messages that request for money. For this reason, a list of money words such as thousand, million, and trillion is collected. The frequency of money words that appear in each message is computed as feature.
- *Frequency of money symbols:* The value of this feature is calculated using regular expression. The regular expression identifies the occurrence of money symbol in each message and then computes the total number of time the money symbol is used in the message.
- *Number of words in capital:* Regular expression is applied to compute the number of words that appear in capital letter from each message.
- *Frequency of function words:* Similar to the approach used in (El-Alfy & AlHasan, 2016), the frequency of function words that appear in each message is computed and used as a feature. These are words with little or ambiguous lexical meaning, which are used to express structural relationship with other words in a sentence. A comprehensive list of function words can be found at (www2.fs.u-bunkyo.ac.jp/~gilner/wordlists.html#functionwords).
- *Number of special character:* The number of special character in each message is computed using regular expression and this value is utilized as a feature.
- *Number of emoticon symbol:* Emoticon symbols like sad, sigh, and happy are mostly used by legitimate users to express mood in a message. Similarly, this feature is extracted using regular expression to find the number of emoticon symbol that appear in each message.
- *Number of links:* Studies have shown that spammers can redirect their victims to phishing website where their sensitive information can be collected and

subsequently used for malicious purpose (Chen et al., 2014). For this reason, the number of links that appear in each message is computed using a regular expression.

- *Frequency of phone number:* This feature represents the number of time a phone number appear in each message. Almeida et al. (2013) have shown that a large proportion of SMS spam messages contain phone numbers, which are intentionally added by spammer to lure their victims. This feature is extracted using regular expression.
- *Average number of words:* The average number of words in each message is calculated as the ratio of the number of words to the message length in character.
- *Number of sentence:* This feature represents the total number of sentences present in each message. The sentence tokenizer in python NLTK package is used for this purpose.
- *Sentence ratio:* This is the ratio of the number of sentences to the message length in character.

4.1.3 Machine learning algorithms

The absorption of machine learning and data mining for data processing and information extraction produced ever-growing research areas from academic communities in the last few years. The goals of these fields focus on the techniques for classifying information and clustering data with similar characteristics. Such techniques have been applied in many areas for practical problem solving including predicting the possibility of stock market fluctuation, detecting oil spills in satellite radar images, clustering text documents for sentiment analysis and users opinion mining on the web. The literature is vast in machine learning and data mining application domains with the introduction of many algorithms for data processing and information extraction. To investigate the best machine learning classification algorithm for both SADM and SMDM, this study explored ten (10) machine learning classifiers namely Random

Forest, J48, ADTree, SVM, Multilayer perceptron (MLP), AdaBoost, Decorate, LogitBoost, Bayes Network, and Random committee. The aim is to find the best performing classifiers that can provide better performance across the datasets used in this study.

(1) *Random Forest:* This is a class of decision tree algorithms based on ensemble approach. Decision tree algorithms classify instances using a tree structure. In this tree, a node represents the test of an attribute value and a branch denotes the result of the test. Random Forest decision creates an ensemble of classifiers by constructing different decision trees using random feature selection and bagging approach at training stage. The decision tree produces two types of nodes: the leaf node labeled as a class and the interior node associated with a feature. A different subset of training data is selected with a replacement to train each tree (Chu et al., 2012a; Narudin et al., 2014). Entropy is applied to compute the information gain contributed by each feature. Let *D* represents the dataset with the labeled instances and *C* as the class such that $C = \{C_1, C_2, C_3, ..., C_j\}$, where *j* is the number of classes considered. In this study, the value of *j* is set to 2, which represents spam or non-spam. Thus, the information needed to identify the class of an instance in the dataset *D* is denoted as *Info(D) = Entropy(P)*, where *P* is the class probability distribution such that:

$$P = \left\{ \frac{|C1|}{|D|}, \frac{|C2|}{|D|}, \frac{|C3|}{|D|}, ..., \frac{|Cj|}{|D|} \right\} \qquad \textbf{4.13}$$

By partitioning *D* based on the value of a feature *F* according to subsets $\{D_1, D_2, D_3, ..., D_n\}$, *Info(F,D)* according to *F* is computed as:

$$Info(F, D) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} Info(D_i) \qquad \textbf{4.14}$$

The corresponding information gain after obtaining the value of F is computed as:

$$Gain(F,D) = Info(D) - Info(F,D) \qquad \textbf{4.15}$$

Then the *GainRatio* is defined as:

$$GainRatio(F,D) = \frac{Gain(F,D)}{SplitInfo(F,D)} \qquad \textbf{4.16}$$

where *SplitInfo(F,D)* represents the information due to the splitting of *D* according to the feature *F*. Random Forest uses the majority voting of all the individual decision to obtain the final decision (Chu et al., 2012a).

(2) *J48:* Decision tree algorithm generates the rules that will be used to predict the target variable. Decision tree algorithm helps to easily understand the distribution of the data. J48 is a decision tree algorithm that extends Iterative Dichotomiser 3 (ID3). The algorithm was developed by the WEKA project team (WEKA, 2016). The WEKA data mining software implanted J48 as an open source Java implementation of C4.5 decision tree algorithm introduced by Quinlan in 1993. J48 supports additional features such as handling missing value in the training data, rules derivation, decision tree pruning and many more. Pruning of the decision tree algorithm helps to reduce classification errors during training stage and enables generalization ability of the tree produced. The algorithm calculates gain in information that results from a test of the attribute using the entropy measure. Then the best attribute is found and selected on the basis of the present selection criterion (Kaur & Chhabra, 2014).

(3) *ADTree:* This algorithm is called an alternating decision tree (ADTree) and it is a machine learning algorithm for classification tasks. ADTree generalizes decision trees with a support for boosting. This algorithm consists of an alternation of

decision trees representing decision and prediction nodes. Decision node represents a predicate condition while prediction node contains a single number. The prediction node is usually presented on both root and the leaves of the decision tree. An instance is classified by an ADTree by following all paths for which all decision nodes are true and summing any prediction nodes that are traversed. This is different from binary classification trees such as C4.5 in which an instance follows only one path through the tree. Figure 4.4 shows an example of ADTree for classifying spam and non-spam accounts based on some selected features.

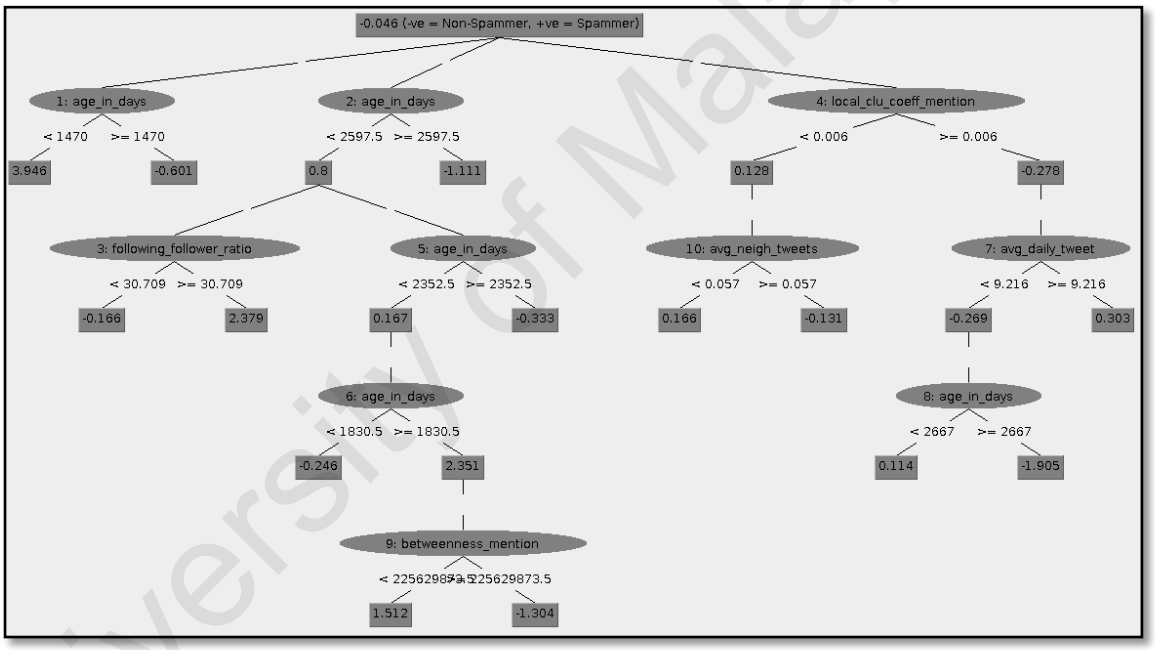


Figure 4.4: Tree generated by ADTree algorithm

(4) *Support Vector Machine (SVM):* This is a statistical supervised learning classification algorithm for data analysis and pattern recognition based on labeled samples. SVM was developed by Vapnik and co-workers (Smola & Schölkopf, 2004). The algorithm can serve the purpose of both classification and regression tasks. The aim of SVM is to define a hyperplane that separates the boundary between different classes in a dataset. The hyperplane separates the classes by

maximizing the margin among the closest points known as support vectors from each class to the hyperplane. To address non-linear separable problem, SVM employs kernel functions to find an optimal separating hyperplane by projecting the training data from low to high dimensional space. SVM uses kernel functions such as linear, Radial Basis Function (RBF), and polynomial kernel.

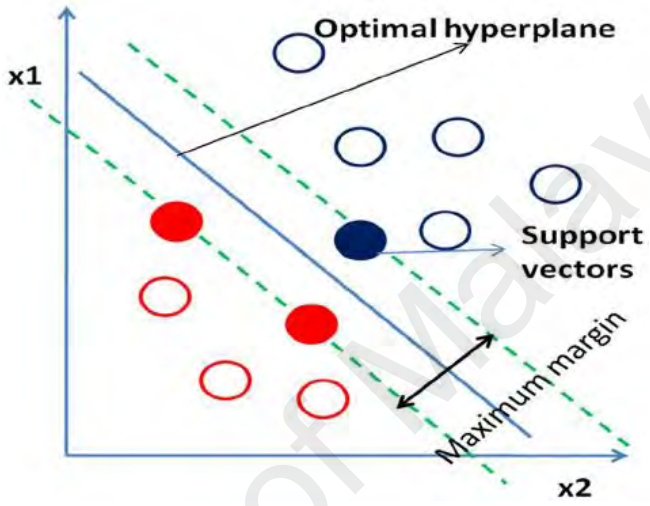


Figure 4.5: Operation of SVM algorithm

For instance, considering the task of separating the blue circle from the red ones as shown in Figure 4.5, the blue line that divides the two objects is called the optimal hyperplane. The blue and red circles on the dotted green lines are called the support vectors. SVM defines a maximum margin in order to ensure that the two classes are at a wider distance from each other. This allows reduction in misclassification during the operation of the algorithm.

(5) *Multilayer perceptron (MLP):* This is a class of feedforward artificial neural networks, which consists of activation units, usually referred to as artificial neurons and weights (Noriega, 2005). MLP modifies the standard linear perceptron by including multiple layers, such as input, hidden, and output layers to solve both

linear and non-linear classification problems. The algorithm maps input data to appropriate outputs. During the training stage, MLP applies a learning algorithm, mostly backpropagation, to adjust the weights so that the network can acquire the required knowledge to classify new unseen data. The process of finding the correct values of weights is known as the learning rule, and this involves initializing the weight matrix to a set of random numbers between −1 and +1. As the network learns, these values are changed until it has been shown that the network has solved the problem under consideration. This is done to minimize the error generated by the output unit when compared with the expected result.

(6) *AdaBoost:* This algorithm usually called *Adaptive Boosting*, is a machine learning meta-learner, which can be used alongside many other individual machine learning algorithms to get better classification performance. The other individual learning algorithms are called the *weak learners* and their result is merged into a weighted sum that corresponds to the final classification result of the boosted classifier. The algorithm is adaptive in the sense that instances misclassified by previous classifiers are used to improve the performance of the subsequent weak learners. Despite the adaptive nature of the algorithm, it is sensitive to noisy data and outliers. However, AdaBoost can be less prone to the over-fitting issue than other learning algorithms (Kégl, 2013).

(7) *Decorate:* Decorate algorithm is a meta-learner for building diverse ensembles of classifiers by using specially constructed artificial training instances. The artificial training data is constructed by randomly generating examples using an approximation of the training data distribution, such as a Gaussian distribution determined by estimating the mean and standard deviation of the training set. Thus, an ensemble is generated iteratively, learning one class at each iteration and adding it to the current ensemble. The ensemble is initialized with the classifier trained on

the given data. At each iteration, a specified number of artificial training examples are generated based on a simple model of the data distribution (Melville & Mooney, 2003; Roli et al., 2004).

(8) *LogitBoost:* Similar to AdaBoost algorithm, LogitBoost is a boosting classification algorithm, which optimizes the logloss instead of exponential function. The major difference between LogitBoost and AdaBoost is that LogitBoost minimizes the logistic loss whereas AdaBoost minimizes the exponential loss. Both algorithms perform additive logistic regression. The purpose of using logistic loss in LogitBoost algorithm is to reduce the sensitivity of the algorithm to outliers, thus, it is expected to perform more than AdaBoost in this area.

(9) *Bayesian networks:* This classifier also known as belief networks, or Bayes Nets belong to the family of a probabilistic graphical model (GM) that represents a set of random variables and their conditional dependencies using the concept of a directed acyclic graph (DAG). According to the graphical structure of Bayesian Net, the nodes in the graph represent random variables and the edges between the nodes represent probabilistic dependencies among the corresponding random variables. The conditional dependency in the Bayes Nets DAG can be estimated using a known statistical and computational method. Similar to graph theory, the structure of a DAG is represented using two main sets namely, the set of nodes and the set of directed edges. The nodes in the graph are labeled using the variable names and are usually represented with circles. On the other hand, the arrows are used to represent dependencies between nodes. For instance, an edge between nodes X_i to X_j represents a statistical dependence between the corresponding random variables. Therefore, the arrow indicates that variable X_j depends on the value of variable X_i or in other word, variable X_i influences X_j. Thus, node X_i is called the parent of X_j and similarly, X_j is referred to as the child of X_i. Bayesian Network has been used in

many domains, such as bioinformatics, document classification, information retrieval, semantic analysis, image processing, and decision support systems (Ben-Gal et al., 2007).

(10) *Random committee:* The basic operation of Random committee classifier is that the algorithm simply creates an ensemble of different randomizable base learners or classifiers, each of which is built using different random number seed on the same dataset. The final prediction result of a random committee classifier is basically an average of the predictions produced by the individual base classifiers.

4.2 Spam Risk Assessment Model (SRAM)

To address the objective of developing a new risk assessment model that assesses the spam risk level of Twitter accounts, this study proposes SRAM model as shown in Figure 4.6, which is based on Fuzzy Analytic Hierarchy Process (FAHP) analysis. The SRAM model computes the risk index for all the instances of Twitter accounts in the ground truth dataset based upon the different risk assessment indicators established in this study. Using the value of the risks index computed, the Twitter accounts are ranked quantitatively and the corresponding risk categories produced. As an overview of the proposed SRAM model, the flow is partitioned into three main phases: problem definition and data entry, FAHP calculation and risk assessment. The first phase handled goal definition, criteria and alternatives identification. It also involves inputting the different fuzzy decision matrices based upon pairwise comparisons using the standard triangular membership function scale defined to replace the Saaty crisp scale of judgment. This allows uncertainty and human impreciseness to be modeled. The second phase first computes the consistency ratios for all the fuzzy-based judgment matrices based upon centroid defuzzification method and Saaty consistency ratio. The weight of each criterion is then obtained using the Ramik FAHP method, and finally, the global weight for each alternative is calculated. Using the global weights of the alternatives,

the third phase computes the risk index from the normalized data for each account. Finally, an account is classified based on its risk score using the standard risk membership function. The subsequent sections detailed the procedures followed to execute each phase.

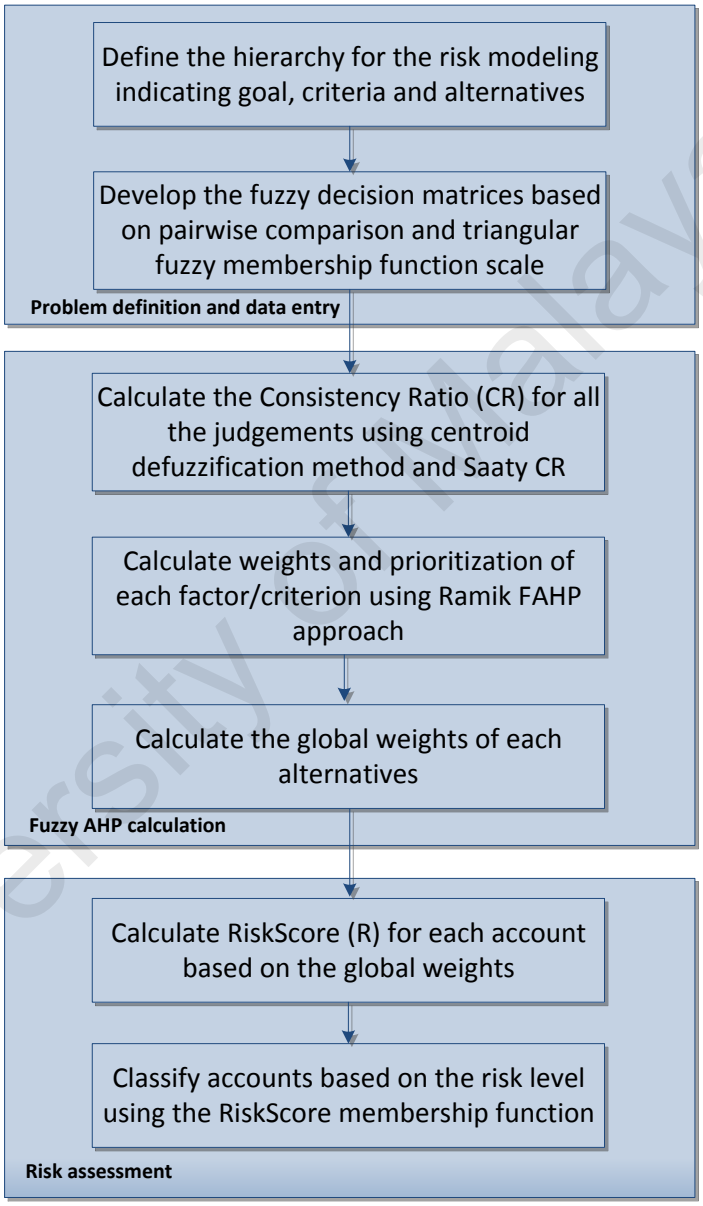


Figure 4.6: Proposed SRAM model

4.2.1 Decision criteria for spam risk assessment

Risk assessment is often expressed in terms of the likelihood or probability of occurrence and the risk impact. The risk impact may include financial, reputational harm, and many more. To satisfy this requirement in modeling the proposed SRAM, two important criteria are considered, which are likelihood and impact. Although there are other indicators, which are involved in the overall risk assessment model, in this study, these indicators are limited to only ten. The rationale behind this decision was to allow the proposed SRAM model to function effectively in facilitating the process of rating and categorizing each Twitter account and to improve the response mode of the risk assessment model. The selection of these indicators is based upon the discriminating features assessment using the evolutionary algorithm. This process enables the author to have an underlying pre-information about each feature selected as indicator.

4.2.2 Hierarchy for spam risk assessment

Following the structural connection of AHP method as earlier discussed, Figure 4.7 shows the hierarchy defined for the SRAM model. This hierarchy comprises of three levels: the goal, criteria, and alternatives.

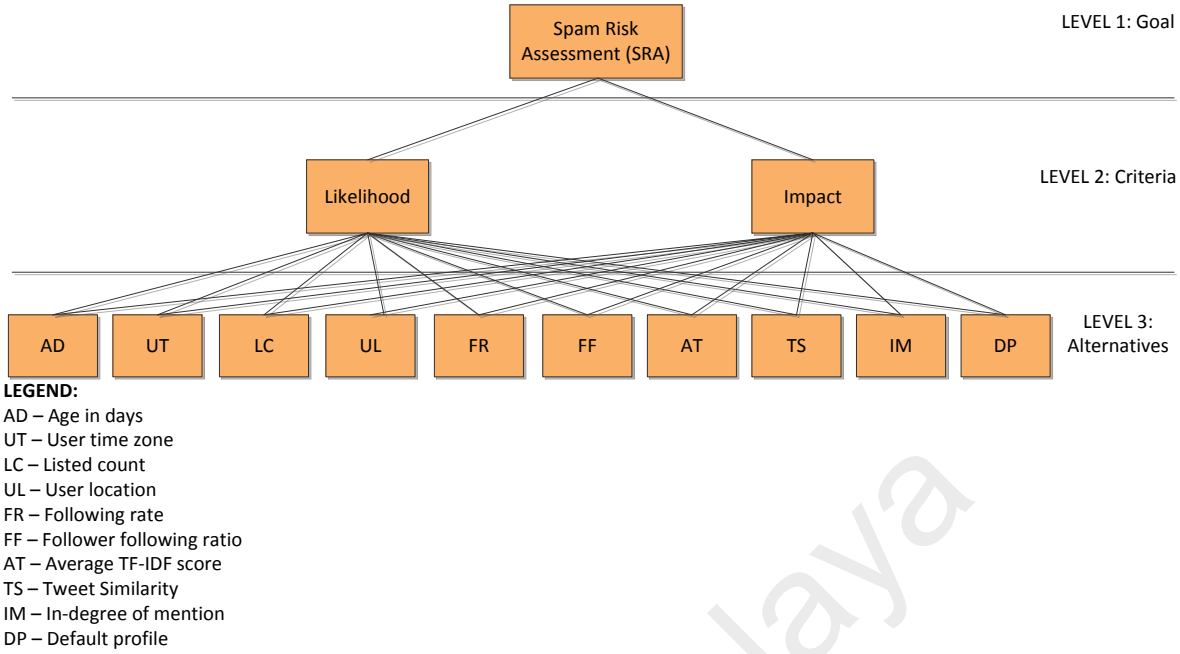


Figure 4.7: Hierarchy of SRAM

1) Level 1 addresses the goal of the model. As part of the objectives of this thesis, this goal centers on assessing the spam risk level of Twitter account using the method of FAHP. Therefore, in this context, the SRAM aims to rate, quantify and estimate the risk score for each Twitter account.
2) Level 2 presents the two most important criteria used by the SRAM for decision making on the subject domain based on the goal of the model. The decision criteria are the likelihood and impact factors that influence the goal of the model. In the FAHP pairwise comparison matrices, as will be seen in the subsequent chapter, the likelihood of risk to occur on Twitter microblog is rate higher so as to produce a rating scheme to counter earlier threats and vulnerabilities.
3) Level 3 details the decision alternatives or indicators, which are critically judged subject to both the likelihood and impact criteria. According to the adopted AHP structure, the proposed ten indicators have full connection to the likelihood and impact criteria and they are judged according to the respective criterion. The

alternatives used as indicators for the SRAM are selected based on the pre-knowledge of the variables from evolutionary computation section. To rank the proposed alternatives, this study uses a ranker search algorithm for this purpose. The final goal is to obtain the global weights for each alternative. This will aid in the computation of the risk index for each account.

4.2.3 Fuzzy Logic and Fuzzy Membership Function

Fuzzy logic is a computing approach that is based on the degree of truth rather than the usual true or false (1 or 0) as in the case of Boolean logic on which the modern computer is designed. Lotfi Zadeh of the University of California at Berkeley proposed the idea of fuzzy logic in the 1960s when he was working on the problem of making computer to understand natural language. In the real world, natural language, like most other activities the universe, is not easily translated into the absolute values of 1 and 0. Fuzzy logic has emerged as an important method to develop systems control and address complex industrial projects, as well as for building diagnostic and expert systems. Different from the traditional way of reasoning based on the Boolean logic, in real life, many abundance of knowledge exists that are known to be imprecise and very ambiguous. These imprecise information are handled conveniently by human reasoning, thus, fuzzy logic was designed to imitate human behaviour at dealing with vagueness. Fuzzy logic is also popular for its flexibility and simplicity. This logic is capable of dealing with vagueness and incomplete data and to model non-linear functions of arbitrary complexity. Fuzzy logic has spanned a wide range of applications, such as pattern recognition, earthquake prediction, computer vision, robotics, decision support systems, scheduling optimization, as well as various kinds of control systems. As oppose the classical logic, fuzzy logic introduces a degree of imprecision when evaluating elements of the fuzzy set. Intuitively, the degree of membership introduced by the fuzzy logic represents the extent to which an expert's judgment places an element

in the set. With fuzzy logic, it is possible that an element will belong to more than one set with varying degrees of membership. This condition allows a gradual transition among adjacent sets. Therefore, it allows us to view the concepts of possibility and vagueness as separate entities from probabilistic or random uncertainty (Precup & Hellendoorn, 2011).

The basic idea of classical sets based on Boolean logic arises from the need for humans to classify objects and concepts. These sets can be described as well-defined sets of elements or a membership function μ that can take a value of 1 or 0 from a universe of discourse for all elements that can belong (or not) to the concerned set. Formally, suppose X is the universe of discourse and x represent the elements contained in X. Then, suppose A is a set that contains some elements in the universe of discourse X. Using classical sets theory, it can be said that x belongs or does not belong to set A, based on the following membership function $\mu_{A(x)}$:

$$\mu_{A(x)} = \begin{cases} 1, & if\ x \in X \\ 0, & if\ x \notin X \end{cases} \qquad \textbf{4.17}$$

On the other hand, since in real life, some concepts have unclear boundaries in their definition, this necessitates the need for fuzzy sets to model the vagueness. While classical set theory categorizes items into crisp sets using well-defined boundaries between the values of the elements, fuzzy set theory classifies elements into continuous sets using an underlying theory that an element belongs to the set based on the degree of membership. In other word, membership functions contain a value ranging from 0 to 1. This implies that fuzzy set allows a continuum grade of objects based on its degree of membership. Formally, let X represent a space of points in the universe of discourse, with a generic element X denoted by x, then a fuzzy set B in the universe X is

characterized by a membership function which associates with each point in X a real number in the interval [0, 1] such that:

$$B = \{x, \mu_{B(x)} \mid x \in X\} \qquad \textbf{4.18}$$

Where $\mu_{B(x)}$ at x represent the "*grade of membership*" in a fuzzy set B. If $\mu_{B(x)} = 1$, the element is said to have full membership or partial membership when $0 < \mu_{B(x)} < 1$. If $\mu_{B(x)} = 0$, the element has no membership. Similar to the crisp sets in classical logic, fuzzy sets also defines some operations. Particularly, some of these operations are the complement, relation, convexity, equality, containment, union, and intersection of fuzzy sets. For instance, two fuzzy sets A and B are said to be equal, if $\mu_{A(x)} = \mu_{B(x)}$ for all x in X. The complement of a fuzzy set A is denoted by $\overline{A}$ and this can be defined as:

$$\mu_{\overline{A}(x)} = 1 - \mu_{A(x)} \qquad \textbf{4.19}$$

The union of two fuzzy sets A and B is a fuzzy set C, written as $C=A \cup B$. This can be obtained using the following equation:

$$\mu_{C(x)} = \max\left\{\mu_{A(x)}, \mu_{B(x)}\right\} \qquad x \in X \qquad \textbf{4.20}$$

Similarly, the intersection operation is defined using the *min* function as follows:

$$\mu_{C(x)} = \min\left\{\mu_{A(x)}, \mu_{B(x)}\right\} \qquad x \in X \qquad \textbf{4.21}$$

The membership function for a particular fuzzy set can be of any shape; however, the experts in the field usually determine this shape. Some of the fuzzy membership functions are discussed as follows:

(a) *Triangular Membership Function (TMF):* Fuzzy TMF contains three parameters $\{a, b, c\}$ as follows:

$$triangle(x:a,b,c)=\mu_{A(x)}=\begin{cases}0, & if\ x\le a\\ \dfrac{x-a}{b-a}, & if\ a<x\le b\\ \dfrac{c-x}{c-b}, & if\ b<x<c\\ 0, & if\ x\ge c\end{cases} \qquad \textbf{4.22}$$

Using the *max* and *min* function of fuzzy set, Eqn. 4.23 can alternatively be stated as follows:

$$triangle(x:a,b,c)=\mu_{A(x)}=\max\left\{\min\left\{\frac{x-a}{b-a},\frac{c-x}{c-b}\right\},0\right\} \qquad \textbf{4.23}$$

These parameters $\{a,b,c\}$ where $a < b < c$ gives the x coordinates of the three corners of the triangle.

(b) *Trapezoidal Membership Function (TRMF):* TRMF is specified using four parameters $\{a,b,c,d\}$ where $a < b < c < d$ gives the x coordinates of the four corners of the underlying trapezoid. TRMF is expressed as follows:

$$trapezoid(x:a,b,c,d)=\mu_{A(x)}=\begin{cases}0, & if\ x<a\\ \dfrac{x-a}{b-a}, & if\ a\le x\le b\\ 1, & if\ b\le x\le c\\ \dfrac{d-x}{d-c}, & if\ c\le x\le d\\ 0, & if\ x>d\end{cases} \qquad \textbf{4.24}$$

Using the *max* and *min* function of fuzzy set, Eqn. 4.25 can alternatively be stated as follows:

$$trapezoid(x:a,b,c,d)=\mu_{A(x)}=\max\left\{\min\left\{\frac{x-a}{b-a},\frac{d-x}{d-c}\right\},0\right\} \qquad \textbf{4.25}$$

Both TMF and TRMF are widely used because of their simplicity and computational efficiency (Jang et al., 1997).

(c) *Gaussian Membership Function (GMF):* Fuzzy GMF is specified using two parameters $\{c,\sigma\}$, where c is the membership center and σ represents the membership width. GMF is defines as follows:

$$gaussian(x:c,\sigma)=\mu_{A(x)}=\ell^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \qquad \textbf{4.26}$$

(d) *Bell Membership Function (BMF):* BMF also called generalized BMF is specified using three parameters $\{a,b,c\}$ as follows:

$$bell(x:a,b,c)=\mu A(x)=\frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}} \qquad \textbf{4.27}$$

The parameter *b* is usually positive in most cases, however, if *b* is negative, the shape of the membership function will become an upside-down bell. GMF and BMF are also becoming increasingly popular for specifying fuzzy sets due to their smoothness and concise notations.

(e) *Sigmoidal Membership Function (SMF):* Fuzzy SMF is defined using two parameters $\{a,c\}$, where the parameter *a* controls the crossover point *x*=*c*. SMF is defined as follows:

$$sig(x:a,c)=\mu_{A(x)}=\frac{1}{1+\exp[-a(x-c)]} \qquad \textbf{4.28}$$

According to Jang et al. (1997), SMF is inherently open left or right depending on the sign of the parameter *a*. Thus, the membership function is most appropriate for representing concept such as "*very large*" or "*very negative*" to build a fuzzy control system.

Fuzzy controller provides a formal approach for representing, manipulating and implementing a human's heuristic knowledge about a specific control system. Contrary

to the operation of the classical controllers, fuzzy controller are capable of utilizing knowledge obtain from human operators. Therefore, fuzzy controllers are special expert systems with each employs a knowledge base expressed in terms of relevant fuzzy inference rules, and an appropriate inference engine to solve the given control problem. Fuzzy controllers are of different types according to the problems they are developed to solve. According to Klir and Yuan (1996), a general fuzzy controller consists of four modules: a fuzzy rule base, a fuzzy inference engine, a fuzzification interface, and a defuzzification interface as shown in Figure 4.8.

A controller starts its operation by first taking measurements of all variables that represent relevant conditions of the controlled process. These measurements are converted into appropriate fuzzy sets to express measurement uncertainties. This step of converting to appropriate fuzzy set is called a fuzzification. The fuzzified measurements are then used by the inference engine to evaluate the control rules stored in the fuzzy rule base. The result of this evaluation is a fuzzy set (or several fuzzy sets) defined on the universe of possible actions. The fuzzy set resulted as output is then converted into a single (crisp) value or a vector of values that best represent the fuzzy set. The process of converting this fuzzy set to a crisp value is called defuzzification. The defuzzified crisp value represents the action taken by the fuzzy controller.

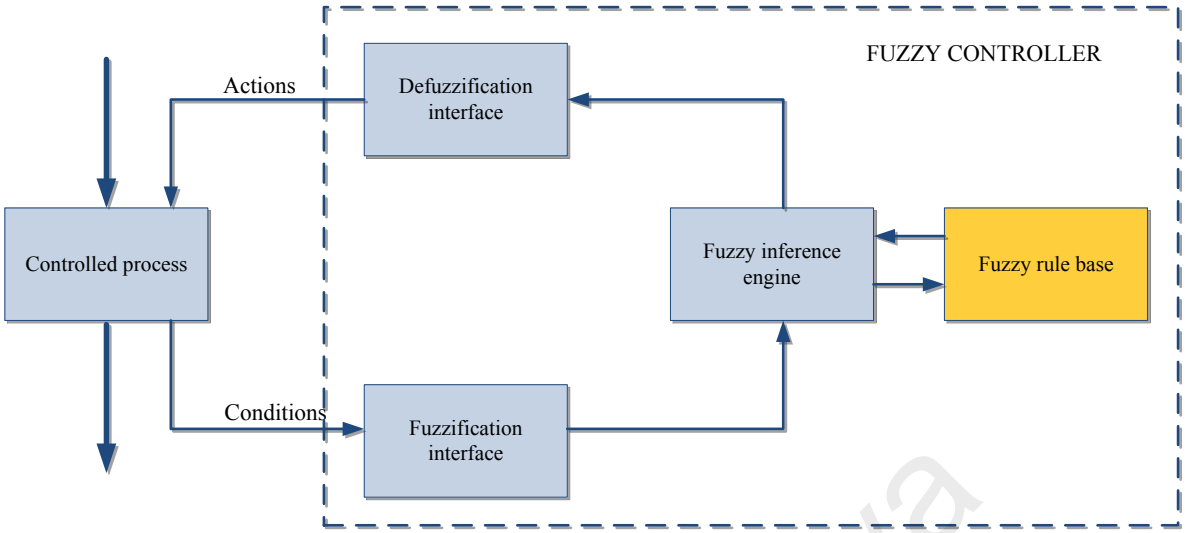


Figure 4.8: Fuzzy controller general scheme

As stated earlier, TMFs are widely used because of their simplicity and computational efficiency. In FAHP, a special version of TMF called Triangular fuzzy numbers (TFNs) are usually used to present linguistic terms of an individual's pairwise comparisons. This TFN represents the fuzzified version of the Saaty scale of preference discussed in Chapter 3 under section 3.4. For instance, Saaty scale of preference *"3"* may be fuzzified as *"*$\tilde{3}$*"*, which represents the fuzzy membership function *(2,3,4)*. The reciprocal of this fuzzy number is *(1/4,1/3,1/2)*. A TFN can be defined as follows (Başaran, 2012):

$$\mu_{\tilde{A}(x)} = \begin{cases} 0, & if\ x < l \\ \dfrac{x-l}{m-l}, & if\ l \le x < m \\ 1, & if\ x = m \\ \dfrac{u-x}{u-m}, & if\ m < x \le u \\ 0, & if\ u < x \end{cases} \qquad \textbf{4.29}$$

Where l and u represent the lower and the upper bounds of the fuzzy number $\tilde{A}$, respectively, m represents the midpoint, and the TFN is represented as $\tilde{A} = (l, m, u)$ as

shown in Figure 4.9. This process allows the FAHP to accommodate impreciseness in human judgments as opposed the classical AHP method.

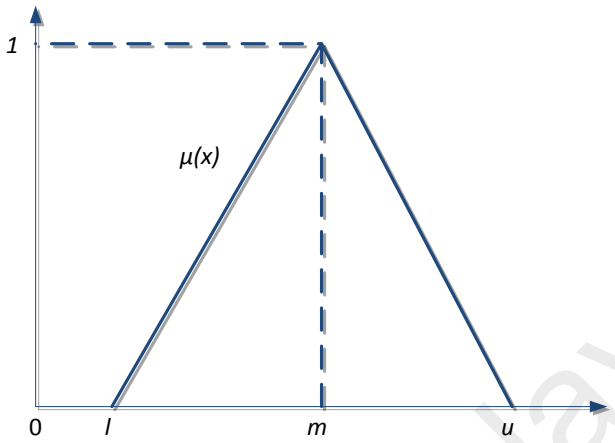


Figure 4.9: Membership function of a triangular number

4.2.4 Fuzzification and defuzzification

As the name implies, fuzzification is the process of making a crisp measurement fuzzy. This is done by converting many of the quantities or measurements that are considered crisp and deterministic to undeterministic measurements. These measurements carry considerable uncertainty and this form of uncertainty arises because of imprecision, ambiguity, or vagueness. This variable is then considered as fuzzy and can be represented using a membership function. In the real world, pairwise comparisons based on crisp judgments are subject to certain error or bias. This biasness can be handled using fuzzy sets. For instance, rather than concluding that the scale of preference 3 is moderately important, as in the case of the classical AHP, fuzzy AHP can be introduced to handle the uncertainty in Saaty's scale by fuzzifying scale of preference 3 as *(2,3,4)* using a TFN. This approach allows modeling human vagueness that may arise when passing judgment on the variables or quantities of interest.

On the other hand, defuzzification is the process of converting the degrees of membership of output linguistic variables into numerical values. Defuzzification involves producing a quantifiable result in crisp logic, given fuzzy sets and corresponding membership degrees, it maps a fuzzy set to a crisp set. Defuzzification is needed in fuzzy control systems. It interprets the membership degrees of the fuzzy sets into a specific decision or real value. There are several methods for defuzzifying a TFN as follows:

(a) *Weighted Mean Method:* This method is mostly appropriate for TFNs with equilateral triangle shapes. However, the reciprocal of the TFNs utilized in fuzzy pairwise comparisons do not produce an equilateral triangle in most cases. It is computed as follows:

$$crisp(\widetilde{A}) = \frac{(l + 4m + u)}{6} \qquad \textbf{4.30}$$

(b) *Centroid Method:* This is the most commonly used method for defuzzifying a TFN (Başaran, 2012). Centroid defuzzification method is also known as center of gravity as it returns the center of area under the curved. The result of a centroid method of a TFN is given as $crisp(\widetilde{A}) = (l + m + u)/3$. The following equation is used for centroid defuzzification:

$$crisp(\widetilde{A}) = \frac{\int \mu_{\widetilde{A}}(x).xdx}{\int \mu_{\widetilde{A}}(x)dx} \qquad \textbf{4.31}$$

(c) *Bisector Method:* This is a vertical line used to divide the region into two sub-regions of equal area. It is usually equal to *m* for equilateral TFNs.

(d) *Middle, Smallest, and Largest of Maximum Methods:* Produce the same results since any TFN used in the pairwise comparison matrix has a unique maximum.

(e) *Mean of Maximum:* In this method, the fuzzy logic controller first determines the scaled membership function with the largest degree of membership. The fuzzy logic controller then identifies the typical numerical value for that membership function. This numerical value is the mean of the values corresponding to the degree of membership at which the membership function was scaled. Other defuzzification methods include constraint decision defuzzification, fuzzy clustering defuzzification, fuzzy mean, first of maximum, generalized level set defuzzification, and weighted fuzzy mean among others. This study utilized centroid defuzzification method due to its wide acceptance and simplicity.

4.2.5 The aid of Fuzzy Analytic Hierarchy Process (FAHP)

In the real world, decision making may involves a multitude of objectives and decision criteria that are in conflict with one another. In this situation, decision analysis examines the situation in which an individual decision maker faces a choice of action in an uncertain environment. Since decision analysis helps the individual to make a choice among a set of pre-specified alternatives, it turns out that decision making process will rely on information about these alternatives. However, the nature of information in a decision making environment can vary based on the objectives of the decision makers. While some decisions are made from scientifically-derived hard data, some involve subjective interpretations, such as a deterministic decision outcomes (certainty) or uncertain outcomes represented by probabilities and fuzzy numbers. Therefore, the variability in the type and nature of information about a decision problem requires methods and techniques, such as Multiple-Criteria Decision Making (MCDM), that can assist in efficient information processing. The aim of MCDM is to identify conflicting areas, compare and evaluate the different alternatives involves in decision making problem according to their diverse criteria, and derive an approach to give the best compromise solution in a transparent manner (Zhang, 2010).

Different from methods that assume the availability of measurements, MCDM derives or interprets measurements subjectively as indicators of the strength of various preferences. In the real world situation, decision preferences varies from decision maker to decision maker, therefore the outcome depends on what the goals and preferences represents (Saaty, 2005). MCDM introduces an element of subjectiveness whose accuracy and fairness depends largely on the decision maker's ethics and understanding of the domain knowledge. A number of methods for MCDM have been studied in the literature, however, AHP remains one of the most widely used of these methods. AHP method was introduced by Saaty to address the great challenges of decision situations that are brought by multiple or even conflicting criteria (Chen & Hwang, 2012).

AHP method is very useful especially where human subjectivity is involved in judgments whose resolutions have long-term repercussions. AHP has unique advantages when important elements of the decision making process are complex to quantify or compare. It is a form of MCDM methods aims to rank decision alternatives and select the best alternative for a complex multi-criteria decision-making problem based on pairwise comparison of those criteria involved in the decision making. Some of the decision situations where AHP is very useful are as follows (Zhang, 2010):

(a) *Choice:* This application area involves selection of a single alternative from a given set of alternatives based on multiple decision criteria.
(b) *Ranking:* This involves ordering the set of alternatives usually from most to least desirable.
(c) *Prioritization:* This involves determination of the relative merit of members of a set of alternatives in contrast to selection of a single alternative or merely ranking of alternatives.

(d) *Resource allocation:* This application involves allocating resources among a set of alternatives.

(e) *Benchmarking:* This handles comparison of the processes between organizations.

(f) *Quality management:* This involves the multidimensional aspects of quality and quality improvement.

AHP provides a comprehensive and rational method for organizing a decision problem. It represents and quantifies decision elements and for relating the elements to the overall goals, which eventually assist in deriving alternative solutions. In AHP process, a hierarchy tree based on the decomposition of the complex problem is created having the goal at the top of the tree and criteria and/or sub-criteria at the next levels. The decision alternatives are placed at the bottom of the tree. The decision elements are then compared in pairs to determine their relative preference and to make decision according to the pairwise comparison and calculation. Despite the applicability of AHP in handling both quantitative and qualitative criteria during the decision making process based on decision makers' judgments, it lacks the ability to handle the fuzziness and vagueness that may exist in many decision making problems. Therefore, fuzzy AHP or in short FAHP was introduced to reduce or even eliminate the vagueness, which may contribute to imprecise judgments of decision makers in the conventional AHP method.

FAHP technique is developed as an advanced analytical method that extends the classical AHP. The earliest study in FAHP appeared in a van Laarhoven and Pedrycz paper, where the author proposed a fuzzy logarithmic least squares method to determine the triangular fuzzy weights based on a triangular fuzzy comparison matrix. To derive local fuzzy priorities, Lootsma's logarithmic least square has been employed (Zhang, 2010). Later, Buckley utilized the geometric mean method to calculate fuzzy weights (Zhang, 2010). Chang (1996) introduced extent analysis to obtain the weights of the

alternatives in MCDM problems. Other methods used in FAHP include fuzzy alpha cut and interval arithmetic, eigenvector method, fuzzy preference programming, Ramik fuzzy AHP and so on. This study adopted Ramik FAHP method due to its solid mathematical foundation, minimal computational complexity, and it encompasses the classical AHP method, thus, it is expected to perform better than the classical AHP.

To model uncertainty in risk management, FAHP is proposed in this study. FAHP converts the crisp pairwise comparison in the conventional AHP, which seems to be insufficient and imprecise to capture the right judgments of decision-maker, to fuzzy pairwise comparison using TFN. To avoid these uncertainties, FAHP is applied under fuzzy circumstances, with respect to possible pairwise comparison values, instead of restricted comparison value. FAHP allows decision makers to present their preferences within a reasonable interval if they are not sure about them. These intervals result in fuzzy judgment matrix, corresponding to the constant value judgment matrix of classical AHP. Fuzzy AHP commonly used TFN to define level of importance. Fuzzy fundamental scale of FAHP is shown in Table 4.7:

Table 4.7: Definition and membership function of fuzzy number with respect to Saaty scale

Intensity of Importance	Fuzzy number	Definition	Membership function	Reciprocal
1	$\tilde{1}$	Equally important/preferred	(1,1,2)	(1/2,1,1)
3	$\tilde{3}$	Moderately more important/preferred	(2,3,4)	(1/4,1/3,1/2)
5	$\tilde{5}$	Strongly more important/preferred	(4,5,6)	(1/6,1/5,1/4)
7	$\tilde{7}$	Very strongly more important/preferred	(6,7,8)	(1/8,1/7,1/6)
9	$\tilde{9}$	Extremely more important/preferred	(8,9,10)	(1/10,1/9,1/8)

Formally, fuzzy AHP pairwise comparison matrix is given as:

$$\tilde{A} = \begin{bmatrix} 1 & \tilde{a}12 & \dots & \tilde{a}1n \\ \tilde{a}21 & 1 & \dots & \tilde{a}2n \\ \dots & \dots & 1 & \dots \\ \tilde{a}n1 & \tilde{a}n2 & \dots & 1 \end{bmatrix} \qquad \textbf{4.32}$$

where $\tilde{a}_{ij} = (1,1,1)$, if i is just equal to j, and $\tilde{a}_{ij} = \tilde{1}, \tilde{3}, \tilde{5}, \tilde{7}, \tilde{9}$ or $\tilde{1}^{-1}, \tilde{3}^{-1}, \tilde{5}^{-1}, \tilde{7}^{-1}, \tilde{9}^{-1}$, if i is not equal j. A fuzzy judgment vector is obtained for each criterion using fuzzy numbers to indicate the relative contribution or impact of each alternative on a criterion. This fuzzy judgment matrix is developed by using all the fuzzy judgment vectors. The weight vector W is used to represent the decision maker's opinion of the relative importance of each criterion during the decision making process. In fuzzy AHP, the final ranks produced for the alternatives are also represented using fuzzy numbers. These fuzzy numbers are then defuzzified to obtain the crisp global weights for each alternative. By ranking the fuzzy number using special algebraic operators, the optimum alternatives are obtained.

4.2.6 The Ramik FAHP

Ramik FAHP was introduced in 2010 in the work of (Ramík & Korviny, 2010). This method provides a convenient approach to obtain the global weight of each alternative using the fuzzy pairwise judgment matrix. Let a represent a triangular fuzzy number as follows:

$$a = (a^L; a^M; a^U)$$

Where a^L is the lower number, a^M is the middle number, and a^U is the upper number, such that, $a^L \le a^M \le a^U$. If $a^L = a^M = a^U$, then a is called a non-fuzzy number (crisp number). To distinguish fuzzy number from crisp number, a tilde symbol is used

above the fuzzy number such as $\tilde{a} = (a^L; a^M; a^U)$. As discussed earlier, the arithmetic operations +, - , *, and / can also be extended to fuzzy numbers (Ramík & Korviny, 2010). For instance, let $\tilde{a} = (a^L; a^M; a^U)$ and $\tilde{b} = (b^L; b^M; b^U)$ represent two triangular fuzzy numbers, then the following arithmetic operations hold on $\tilde{a}$ and $\tilde{b}$:

$$\tilde{a} + \tilde{b} = (a^L + b^L; a^M + b^M; a^U + b^U) \quad \textbf{4.33}$$

$$\tilde{a} - \tilde{b} = (a^L - b^U; a^M - b^M; a^U - b^L) \quad \textbf{4.34}$$

$$\tilde{a} * \tilde{b} = (a^L * b^L; a^M * b^M; a^U * b^U) \quad \textbf{4.35}$$

$$\tilde{a} / \tilde{b} = (a^L / b^U; a^M / b^M; a^U / b^L) \quad \textbf{4.36}$$

In Ramik FAHP, the fuzzy pairwise comparison matrix $\tilde{A}$ comprised of triangular fuzzy elements as follows:

$$\tilde{A} = \begin{bmatrix} (a_{11}{}^L, a_{11}{}^M, a_{11}{}^U) & \dots & \dots & (a_{1n}{}^L, a_{1n}{}^M, a_{1n}{}^U) \\ (a_{21}{}^L, a_{21}{}^M, a_{21}{}^U) & \dots & \dots & (a_{2n}{}^L, a_{2n}{}^M, a_{2n}{}^U) \\ \dots & \dots & \dots & \dots \\ (a_{n1}{}^L, a_{n1}{}^M, a_{n1}{}^U) & \dots & \dots & (a_{nn}{}^L, a_{nn}{}^M, a_{nn}{}^U) \end{bmatrix}, \quad \textbf{4.37}$$

Where for all *i,j = 1,2,...,n*, we have:

- $a_{ij}{}^L, a_{ij}{}^M, a_{ij}{}^U$ are real number such that $1/\sigma \le a_{ij}{}^L \le a_{ij}{}^M \le a_{ij}{}^U \le \sigma$ for $\sigma > 1$. Thus, the preference intensity provided by the expert are not limited to the interval [1/9,9] as in classical AHP, but can take general form $[\frac{1}{\sigma}, \sigma]$ for a chosen value of $\sigma > 1$.
- If $\tilde{a}_{ij} = (a_{ij}{}^L, a_{ij}{}^M, a_{ij}{}^U)$, then the reciprocal of $\tilde{a}_{ij}$ is $\tilde{a}_{ji} = (\frac{1}{a_{ij}{}^U}, \frac{1}{a_{ij}{}^M}, \frac{1}{a_{ij}{}^L})$.

Then, the fuzzy weights $\tilde{w}_k = (w_k{}^L, w_k{}^M, w_k{}^U)$, for *k=1,2,...,n*, are computed as follows:

$$w_k^{\,L} = c_{\min} \cdot \frac{(\prod_{j=1}^{n} a_{kj}^{\,L})^{1/n}}{\sum_{i=1}^{n}(\prod_{j=1}^{n} a_{ij}^{\,M})^{1/n}}, \qquad \textbf{4.38}$$

$$\text{where } c_{\min} = \min_{i=1,2,\ldots n}\left\{\frac{(\prod_{j=1}^{n} a_{ij}^{\,M})^{1/n}}{(\prod_{j=1}^{n} a_{ij}^{\,L})^{1/n}}\right\} \qquad \textbf{4.39}$$

$$w_k^{\,M} = \frac{(\prod_{j=1}^{n} a_{kj}^{\,M})^{1/n}}{\sum_{i=1}^{n}(\prod_{j=1}^{n} a_{ij}^{\,M})^{1/n}} \qquad \textbf{4.40}$$

$$w_k^{\,U} = c_{\max} \cdot \frac{(\prod_{j=1}^{n} a_{kj}^{\,U})^{1/n}}{\sum_{i=1}^{n}(\prod_{j=1}^{n} a_{ij}^{\,M})^{1/n}}, \qquad \textbf{4.41}$$

$$\text{where } c_{\max} = \max_{i=1,2,\ldots n}\left\{\frac{(\prod_{j=1}^{n} a_{ij}^{\,M})^{1/n}}{(\prod_{j=1}^{n} a_{ij}^{\,U})^{1/n})}\right\} \qquad \textbf{4.42}$$

To measure the consistency of the fuzzy pairwise comparison matrix with the triangular fuzzy elements, Ramik proposed inconsistency index (*NI*) as follows:

$$NI_n^{\,\sigma}(\tilde{A}) = \gamma_n^{\,\sigma} \max_{i,j}\left\{\max\left\{\left|\frac{w_i^{\,L}}{w_j^{\,U}} - a_{ij}^{\,L}\right|, \left|\frac{w_i^{\,M}}{w_j^{\,M}} - a_{ij}^{\,M}\right|, \left|\frac{w_i^{\,U}}{w_j^{\,L}} - a_{ij}^{\,U}\right|\right\}\right\} \qquad \textbf{4.43}$$

$$\text{where } \gamma_n^{\,\sigma} = \begin{cases} \dfrac{1}{\max\{\sigma - \sigma^{(2-2n/n)}, \sigma^2((2/n)^{2/(n-2)} - (2/n)^{n/(n-2)})\}} \; if \sigma < (n/2)^{n/(n-2)}, \\ \dfrac{1}{\max\{\sigma - \sigma^{(2-2n/n)}, \sigma^{(2n-2/n)} - \sigma\}} \; otherwise. \end{cases} \qquad \textbf{4.44}$$

The value of *NI* ranges from 0 to 1, where 0 means that the matrix is fully consistent. However, it should be noted that the value of *CR* and *NI* are not the same for the same fuzzy comparison matrix. Although *NI* has been shown to provide information about inconsistency of a fuzzy pairwise comparison matrix, however, it is not to be confused with the popular CR method. In this study, Saaty CR formula (see Eqn. 3.2) is adopted

to measure the consistency of the pairwise comparison matrices due to its wide acceptance. Centroid defuzzification method is first applied to convert the fuzzy pairwise matrix to Saaty crisp matrix and then the value of CR is determined before proceeding to applying Ramik FAHP method to compute global weights.

4.2.7 Rule induction for data normalization

After obtaining the global weight for each alternative, the next step is to apply the global weights on the ground truth data to model the risk assessment system. As shown in the list of alternatives used (see Figure 4.7), the data type for each alternative is not uniform. This call for data normalization process where the value of each data instance is converted to a real number ranges from 0 to 1. This process reduces bias in the SRAM development. To obtain a normalized data for the ground truth, rule induction method is employed. Rule induction belongs to machine learning domain where formal rules are induced or extracted from a set of data instances. These rules represent patterns in the data or a scientific model of the data. One of the most important techniques in data mining and machine learning is rule induction, which can be useful in extracting hidden patterns and relationships in a dataset. Basically, rules can be expressed in the form:

IF (attribute-1, value-1) and (attribute-2, value-2) and ⋯

and (attribute-n, value-n) THEN (decision, value).

In supervised rule induction learning method, an expert assigns the decision value for each observation in the dataset. The attributes used in rule induction are independent variables while the decision is a dependent variable. Formally, let x represent an observation from the dataset, then a case x is covered by a rule r if and only if all the conditions in r based on attribute-value pair is satisfied by the corresponding attribute value for x. Let C be a concept (i.e decision) defined by the right hand side of rule r,

then it can be said that a concept *C* is completely covered by a rule set *R* if and only if for every case *x* from *C* there exists a rule *r* from *R* such that *r* covers *x*. A rule set *R* is complete if and only if every concept from the dataset is completely covered by *R* (Grzymala-Busse, 2010). Generally, rule induction algorithms belong to two major categories: global and local. The global rule induction algorithms used the set of all attribute values as the search space, while the local rule induction algorithms used the set of attribute-value pairs as the search space. Many rule induction algorithms have been introduced over the years, which include Learning from Examples Module, version 1 (LEM1), LEM2, AQ, CN2, JRip etc. However, this study employs JRip rule induction machine learning algorithm due to its simplicity.

JRip is a rule induction algorithm introduced by William W. Cohen in (Cohen, 1995). JRip implements a propositional rule learner based on a Repeated Incremental Pruning to Produce Error Reduction (RIPPER). The algorithm is an optimized version of Incremental Reduced Error Pruning (IREP). This rule induction algorithm directly extracts rules from the dataset based on propositional rule learning approach. The algorithm executes four main phases: growth, pruning, optimization and selection. The algorithm is described using the following pseudocode (Veeralakshmi & Ramyachitra, 2015):

JRip rule induction algorithm

Input: P -> positive instances, N -> negative instances
Output: RuleSet -> set of rules

```
Procedure BUILDRULESET (P,N)
P=positive instances
N=negative instances
RuleSet= { }
DL=DescriptionLength (RuleSet, P, N)
    WHILE P not equal to { }
    //Grow and prune a new rule
    split (P,N) into (GrowPos, GrowNeg) and (PrunePos, PruneNeg)
    Rule := GrowRule (GrowPos, GrowNeg)
    Rule := PruneRule (Rule, PrunePos, PruneNeg)
    add Rule to RuleSet
```

Figure 4.10, continued.

```
    IF DescriptionLength (RuleSet, P, N) > DL+64 THEN
        // Prune the whole rule set and exit
        FOR each rule R in RuleSet
            IF DescriptionLength (RuleSet -> R}, P, N) < DL THEN
                delete R from RuleSet
                DL := DescriptionLength (RuleSet, P, N)
            ENDIF
        ENDFOR
        return (RuleSet)
    ENDIF
    DL := DescriptionLength (RuleSet, P, N)
    delete from P and N all examples covered by Rule
    ENDWHILE
End BUILDRULESET

Procedure OPTIMIZERULESET (RuleSet, P, N)
    FOR each rule R in RuleSet
        delete R from RuleSet
        U Pos := instances in P not covered by RuleSet
        U Neg := instances in N not covered by RuleSet
        spilt (U Pos, U Neg) into (GrowPos, GrowNeg) and (PrunePos, PruneNeg)
        RepRule := GrowRule (GrowPos, GrowNeg)
        RepRule := PruneRule (RepRule, PrunePos, PruneNeg)
        RevRule := GrowRule (GrowPos, GrowNeg, R)
        RevRule := PruneRule (RevRule, PrunePos, PruneNeg)
        choose better of RepRule and RevRule and add to RuleSet
    ENDFOR
End OPTIMIZERULESET

Procedure RIPPER (P,N, k)
    RuleSet := BUILDRULESET (P,N)
    repeat k times RuleSet := OPTIMIZERULESET (RuleSet, P, N)
    return (RuleSet)
End RIPPER
```

Figure 4.10: Pseudocode for JRip rule induction algorithm

4.2.8 Risk index computation

This section highlights the final procedure to compute risk score for categorizing Twitter accounts based on their risk level. To compute risk index which is to be used for ranking Twitter accounts and subsequently prioritizing them based on their risk level, Eqn. 4.45 is proposed in this study. This computation is introduced after the JRip rule induction algorithm has been applied to normalize the ground truth data for risk modeling. After studying the distribution of the computed risk score for each account category (i.e spam and non-spam), a membership function in Eqn. 4.46 is defined to categorize each Twitter account based on their risk level.

$$RiskScore(R) = \sum_{i=1}^{n} (x\_norm_i * w_i) \qquad \textbf{4.45}$$

Where *x_norm* is the normalized data after applying the rule induction algorithm, w_i is the global weight of alternative *i*, and *n* is the total number of alternatives considered in the Fuzzy AHP modeling. *R* takes value in the interval of 0 to 1, with 1 indicates that the account risk is "*very high*". The Risk level is determined using the following membership function:

$$Risk\ level = \begin{cases} Low, & if\ 0 \le R \le 0.5 \\ Medium, & if\ 0.5 < R \le 0.7 \\ High, & if\ 0.7 < R \le 0.8 \\ Very\ high, & if\ 0.8 < R \le 1 \end{cases} \qquad \textbf{4.46}$$

(a) *Rating threshold:* To map risk score onto appropriate risk level, this study adopted the rating threshold as shown in Figure 4.11. This threshold as seen in Eqn. 4.46 can be used to distinguish between the risk level of each Twitter account based on the result of the Fuzzy AHP analysis. The threshold value is based upon the standard method of obtaining threshold after critically studying the distribution of each account category as regards their risk score. Each risk zone is mapped onto an appropriate risk quadrant as shown in Figure 4.12 in order to understand the relationship between risk impact and likelihood as well as the implication on the computed risk index. The interpretation of each quadrant is as follows:

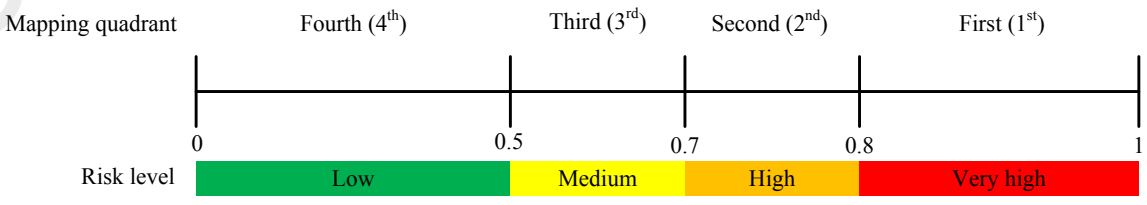


Figure 4.11: Rating threshold

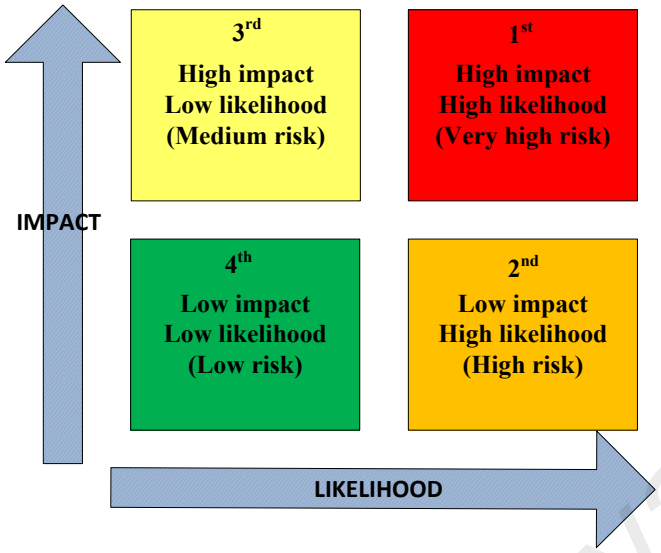


Figure 4.12: Risk quadrant

1) *First (1st) Quadrant:* Addresses risk with high impact and high likelihood. This is for top and high priority social risk account and it demands urgent action to minimize the level of impact of this account's various activities on the legitimate users. For instance, a Twitter account used for sending spam messages, unsolicited mention, and at the same time engaging in massive following may pose serious threat to legitimate users on the network.
2) *Second (2nd) Quadrant:* Focuses on a situation where the risk impact is low but there is high likelihood that the risk will occur. It demands an appropriate response action in order to reduce the risk impact on the legitimate users. For instance, an account used for random following of legitimate users has high likelihood of engaging in spamming activities. However, in some situation, an intelligent user may confidently deny such account from his friendship network, which in turn will reduce the impact on the users.
3) *Third (3rd) Quadrant:* Addresses the situation where there is low likelihood but with high risk impact on the legitimate users. Since the likelihood that the risk will occur

from an account on the network is very low, this quadrant is considered a low priority quadrant. However, it still demands an appropriate response option to mitigate the risk impact.

4) *Fourth (4^{th}) Quadrant:* Presents a scenario where both the risk likelihood and impact are very low. The quadrant addresses the lowest priority in risk management and for a non-critical event. The account whose risk level falls into this quadrant has been critically assessed by the risk management system based on its various social activities and interactions. This activity, according to the system, has very minimal risk with low impact on the legitimate users. Accounts categorized into this quadrant can be accepted if their invitation for friendship is received without the need for further criticism.

(b) *Response strategy:* To map appropriate response action to each risk level according to the risk quadrant, Figure 4.13 shows the adopted risk response used in this study. This response strategy contains four different options: avoidance, mitigation, transference, and acceptance. Hillson (1999) initially proposed this response strategy. In this study, the risk response strategy is adopted because it is an important stage in risk assessment to map appropriate response option to each identified risk level. While there are other methods for defining risk response strategy, such as an approach proposed in (Baker et al., 1999). However, Hillson (1999) is more suitable to this study as it is limited to four response strategic options corresponding to the four different levels of the risk quadrant earlier presented.

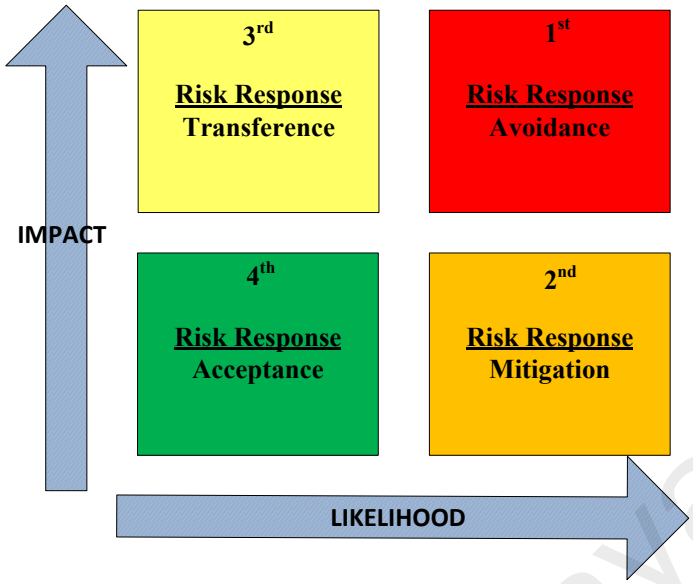


Figure 4.13: Risk response strategy

The description of each response option is as follows:

1) *Avoidance:* This option eliminates uncertainties that may be involved in risk management. Avoidance response strategy provides an opportunity for legitimate users to know beforehand the possibility of facing huge risk impacts by establishing connections to untrusted accounts. For instance, the response options such as blocking various connections established from the unknown account, deleting the account from user's friendship network, as well as reporting the account to social network administrator are ideal options. This will minimize the risk impact on future interactions. Account identified under this category possesses a very high likelihood of engaging in spamming which may eventually lead to high risk impact on legitimate users.
2) *Mitigation:* This option is an alternative strategy that lies between avoidance and transfer options. This strategy deals with risk level classified as high risk. Mitigation strategy involves rating threshold that are above the transfer rate but below the

avoidance threshold. It addresses risk with high likelihood but low risk impact. For example, accounts detected in this category can be removed from the user's friendship network and subsequent communication avoided.

3) *Transference:* This option aims to transfer liability of any possible risk from one account to another party (i.e social network administrator). This will reduce the impact on the users involved. For instance, accounts detected in this category can be reported to social network administrator for further analysis of malicious activities and subsequent serious communication with such unknown accounts is pending. The users may not likely block the account in question until further report from social network administrator is available.
4) *Acceptance:* The option addresses risk events with both low likelihood and impact. The communication from account detected in this category is considered acceptable and safe after analyzing the account social reputation through the risk assessment system. The risk impact is therefore considered very minimal.

4.3 Summary

This chapter described the conceptual unified framework for spam detection and risk assessment in SMCM. It specifically identified two main important sections of the proposed framework: spam detection and risk assessment. The spam detection section of the proposed framework is composed of two main models: SADM and SMDM. The SADM provides effective method for detecting spam accounts on Twitter using machine learning method. SMDM focuses on detecting both Twitter spam messages and mobile SMS spam messages. To achieve the objectives of developing the models for spam message and spam account detection, different features have been proposed and ten (10) machine learning algorithms were studied in details. In addition, bio-inspired evolutionary computational method has been applied to identify discriminating features for spam account detection which, in turn assists with the SRAM development.

The second section of the proposed framework deals with SRAM modeling for spam risk assessment. The proposed SRAM identifies the risk level of each Twitter account using an enhanced method that is based on fuzzy AHP. The fuzzy AHP allows uncertainty and impreciseness involved in human judgments to be incorporated in the risk assessment stage. Furthermore, this section provides the operational characteristics as well as the different composition and rationale behind the proposed SRAM model. The section was concluded with a discussion on the rating threshold and risk response strategies employed in this study.

Having established the composition of the proposed unified framework through identification of the different models involved, the next chapter focuses on the various experimental results and evaluations of the proposed framework.

CHAPTER 5: EVALUATION OF UNIFIED FRAMEWORK FOR SPAM DETECTION AND RISK ASSESSMENT

As stated in the previous chapters, the novelty of this study is the introduction of a unified framework, which is capable of detecting spam message and spam account in SMCM and at the same time rank microblogging social network accounts based on their risk level. This evaluation section presents the achievement of the proposed unified framework. The evaluation phase highlights the performance of the proposed framework based on the reports of the various experiments conducted.

Four evaluation stages are presented with the aim of highlighting the effectiveness of the proposed unified framework in relation to the performance of various models incorporated in the proposed framework. The first phase of this evaluation investigates the performance of the proposed SADM model. The second phase presents the performance of the SMDM model based on mobile and Twitter datasets. The third phase highlights the performance of the SRAM model, which is used for risk assessment with a specific target on Twitter microblog. The fourth phase of the evaluation presents the performance comparison of the proposed models by comparing the results obtained with related studies. This chapter ends with a summary.

5.1 General description

This chapter consists of four evaluation stages and each of the stages has unique objectives. Each stage is presented based on results and discussion to provide underlying performance of the proposed models. The four stages in this evaluation section share some similar requirement in terms of experimental procedures, thus, this section provides the similarities in order to avoid repetition in the introductory part of each stage. These similarities include datasets, ground truth data, general tools, and evaluation metrics.

All experiments were conducted on a computer system running Ubuntu 14.04 operating system. The system has a random access memory (RAM) of 20GB and 3.40GHz Intel Core i7 CPU. At each stage of the experiment, one or two type of dataset is used to ascertain the performance of the proposed models. The dataset is either the Twitter or mobile SMS datasets and each of them has a unique characteristic. The descriptions of the datasets used in this evaluation stage are as follows:

5.1.1 Dataset 1: Twitter Dataset

To the best of the author's knowledge, no public dataset is available for Twitter spam message and spam account detection as at the time of conducting the experimental stage of this study. This is due to the fear of violating user's privacy and the Terms of Use of Twitter API that prevents researchers from sharing tweets data. Although Twitter privacy policy prevents researchers from sharing tweets data, Twitter provides API that can be used by academic community to collect data for experimental purpose. Therefore, to collect data from Twitter microblog, a python crawler is developed, which takes advantage of the Twitter API (Twitter rate limit, 2015). The crawling process covers a period of 24 days starting from 20th March to 12th April 2016. The crawlers collected all the tweets posted by the users in the dataset. The statistics of the tweets collected as well as the number of spam accounts identified from the dataset is shown in Table 5.1. This dataset is hereafter referred to as *Dset1* and it was used to evaluate the performance of the proposed SADM as well as the SRAM model. Section 5.1.4 discusses how the spam accounts were identified.

Table 5.1: Summary of the data collected from Twitter

Dataset item	Number of samples
Total Tweets	3,755,367
Total accounts	52,998
Total Hashtag	1,652,405
Total Mention	3,351,656
Total URLs	1,297,288
Mention edges	1,833,086
Total features	69
Spammers identified	3,648
Non-Spammers	4,000
Total labeled samples	7,648

In addition, to build a collection of Twitter spam messages, a random selection of 8,000 spam tweets posted by spammers in the dataset were selected. In addition, 10,000 tweets were randomly selected from the accounts identified as legitimate making a total of 18,000 tweets used to test the proposed SMDM for spam message detection on Twitter microblog. The spam message is selected in such a way that any tweet whose character length is less than 100 is excluded from the analysis. This spam corpus is hereafter referred to as *Dset2*.

5.1.2 Dataset 2: SMS Collection V.1

To evaluate the proposed SMDM for mobile SMS spam message detection, a public dataset called *SMS Collection V.1* was collected. This corpus of spam and ham messages is publicly available as raw messages at http://www.comp.nus.edu.sg/~rpnlpir/downloads/corpora/smsCorpus/. The corpus contains 747 spam messages and 4,827 ham or legitimate messages making a total of 5,574 ham and spam messages. This dataset is freely available for research purpose. All the 5,574 messages are composed using English language. There are 425 SMS spam messages extracted manually from the Grumbletext website, which is a UK forum at http://www.grumbletext.co.uk/, where mobile phone users can make public claims about SMS spam messages received. A list of 450 SMS legitimate messages collected

from Caroline Tag's PhD Theses at http://etheses.bham.ac.uk/253/1/Tagg09PhD.pdf. A collection of 3,375 SMS ham messages from the total of 10,000 legitimate messages obtained from the National University of Singapore (NUS). In addition, the corpus contains 1,002 SMS ham messages and 322 spam messages extracted from the SMS spam corpus collected by José María Gómez Hidalgo. This corpus is also freely available and can be downloaded at http://www.esp.uem.es/jmgomez/smsspamcorpus/. The summary of the collection is shown in Table 5.2. The dataset is referred to as *Dset3*.

Table 5.2: Summary of SMS Collection V.1 (Dset3)

Dataset	SMS Collection V.1
Type	Mobile SMS
Number of spam messages	747
Number of legitimate messages	4827
Total samples	5574

5.1.3 Dataset 3: SMS Corpus V.0.1 Big

Similarly, the dataset *SMS Corpus V.0.1 Big*, referred to as *Dset4*, is a collection of 1,002 ham messages and 322 SMS spam corpus in English language, which are collected by José María Gómez Hidalgo and Enrique Puertas Sanz. The corpus is freely available at http://www.esp.uem.es/jmgomez/smsspamcorpus/. This dataset contains a list of 202 legitimate messages from Jon Stevenson and a randomly selected ham messages from NUS SMS corpus, which is a corpus of about 10,000 legitimate messages collected at NUS in Singapore. The raw messages were collected from volunteers who have agreed that the corpus be made available publicly. In addition to the number of legitimate messages, the dataset also contains a collection of 322 SMS spam messages extracted manually from the Grumbletext website.

Table 5.3: Summary of SMS Corpus V.0.1 Big (Dset4)

Dataset	SMS Corpus V.0.1 Big
Type	Mobile SMS
Number of spam messages	322
Number of legitimate messages	1002
Total samples	1324

5.1.4 Ground Truth Identification

One of the most important stages in developing a reliable classification model is the identification of labeled samples that are to be used for both training and validation. Ground truth referred to the labeled data, which are used for testing the performance of a classification system. For Twitter spam account detection, several techniques have been employed in the related studies to identify ground truth data. Some of these techniques include honeypot, blacklist, and the Twitter suspension algorithm.

The honeypot approach was originally proposed by Lee et al. (2010a). This approach uses some social honeypots to harvest deceptive spam accounts from Twitter and MySpace. The social honeypot logs users' activities, such as content posting patterns, friendship requests, and profile information in the database. All accounts that send unsolicited friend requests are analyzed to find evidence of spamming before they are added to the spammer's list. The goal of this approach is to reduce the challenges of manually identifying spam accounts on social networks. Yang et al. (2013) adopted this approach to identify spammers in their dataset. One of the issues with honeypot approach is that the honeypot needs to collect a large number of data for behavioral analysis before the suspected accounts can be categorized as spammers or legitimate. In addition, this approach requires a longer period to acquire a significant proportion of spam accounts. To obtain more spammers for developing a classification model, Yang et al. (2013) combined honeypot and blacklist approaches to detect 2,000 spam accounts from their dataset.

The second approach involves the use of the popular blacklists APIs, such as PhishTank, Google Safe Browsing, and URIBL (Google, 2015; PhishTank, 2015; URIBL, 2015). The goal of blacklist-based approach is to identify accounts that include malicious links in their tweets as flagged by the blacklist APIs. These accounts are marked as spammers and added to the list of labeled samples. This approach was employed to identify spammers in the work of Aggarwal et al. (2012) and Yang et al. (2013).

The third approach involves reliance on Twitter suspension algorithm (Twitter, 2016). Twitter suspends accounts once it detects abnormal behaviors in the accounts posting patterns, such as spreading malware, pornographic contents, harassment, invitation spam, and other abusive behaviors. Thomas et al. (2011) and Hu et al. (2013) applied this technique to identify spammers. Since the suspended accounts come from the target microblogging social network, this study utilized this approach to identify spammers. A batch script in python is run to identify those accounts that have been suspended by Twitter. This approach assists in providing consistent method of ground identification. In total, the script returns 3,648 suspended accounts, which are used as spammers in this study. A number of 4,000 accounts from unsuspended users were selected, totaling 7,648 labeled samples as shown in Table 5.1. Based on the identified spammers and legitimate accounts, this study introduced a set of unique features to detect spam accounts on Twitter.

5.1.5 General Tools

In order to perform the experiments in this chapter, this study applied several open source software, namely WEKA, Python, R, and MySQL database. The software were used due to their openness and public availability, as well as the convenience they

offered. In addition, the applications are freely available for use with no cost incur. The following section presents brief details about these applications:

(a) *WEKA:* Waikato Environment for Knowledge Analysis (WEKA) is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or integrated within a customized Java application. WEKA is free software licensed under the GNU General Public License. It can be used to handle several machine learning tasks such as data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. WEKA was developed at the University of Waikato in New Zealand (WEKA, 2016). In this study, the implementation of the ten (10) selected machine learning algorithms was done on WEKA.

(b) *Python:* Python is a general-purpose high-level programming language introduced by Guido van Rossum in 1991. It is a widely used interpreted programming language with a lot of emphasis on code readability using whitespace indentation to delimit code blocks rather than curly brackets in some programming languages. The language provides constructs intended to enable writing clear programs on both a small and large scale (Kuhlman, 2009). Python has large and comprehensive standard libraries. The Python standard libraries used in this study include NLTK for natural language processing, Pandas, MySQLdb, Tweepy etc. Python was used to extract the user profile, content, automation, and timing features proposed in this study.

(c) *R:* This is an open source high-level programming language and software environment that is developed for statistical computing and graphics supported by the R Foundation for Statistical Computing (Vance, 2009). R programming language is widely used by statisticians and data scientists for data analysis and

statistical software development. The R is freely available under the GNU General Public License. R support both command line interface and graphical user interface (GUI). It supports reach set of standard libraries for data analysis and graphics. This study utilized the igraph library in R for social network analysis. This library has been used to extract the mention network graph-based features proposed in this study.

(d) *MySQL*: This is a freely available open source Relational Database Management System (RDBMS), which utilizes Structured Query Language (SQL) to manage data in a database server. The database software was introduced by David Axmark, Allan Larsson, and Michael Widenius. SQL is the most popular language for storing, retrieving and managing data in a database. SQL is noted for its quick processing, reliability, flexibility, and ease of use. MySQL is quoted as the most popular open source database software with high levels of scalability, security, and reliability (Widenius & Axmark, 2002). This study used MySQL as the database for storing the Twitter dataset.

5.1.6 Evaluation Metrics

This section provides the details of the evaluation metrics employed in this study to evaluate the performance of the proposed models. Performance metrics provide a practical method to check the efficiency of a model. The classification performance of a mode can be measured in machine learning using a confusion matrix, which is a table that gives the classification performance on how well a classifier is able to separate one class from the other. The general structure of confusion matrix for binary class classification problem is shown in Table 5.4. In this table, True Positive (TP) and True Negative (TN) referred to the number of correctly classified spam and legitimate samples respectively. False Positive (FP) represents the number of legitimate instances

classified as spam, while False Negative (FN) represents the number of spam instances classified as legitimate.

Table 5.4: Confusion matrix for a binary class problem (spam and non-spam)

		Predicted Class	
		Class = Spam	Class = Non-spam
Actual Class	Class = Spam	TP	FN
	Class = Non-spam	FP	TN

The parameters TP, TN, FP, and FN in this table can be used to derive some standard metrics, such as True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR), and False Negative Rate (FNR) as shown in Eqns. 5.1, 5.2, 5.3, and 5.4 respectively. TPR is also called detection rate (DR), sensitivity or recall, and can be used to indicate the accuracy of a classification model on the labeled samples. A combined metric known as F-measure or F1-score has been widely used to measure the performance of a classification system. This metric is calculated as the harmonic mean of precision and recall as shown in Eqn. 5.6. In addition, AUC-ROC, which is a metric that plots TPR and FPR on a single graph to obtain another robust evaluation measure, has also been applied.

$$TPR / DR / Sensitivity / \mathrm{Re}call(R) = \frac{TP}{TP + FN} \quad \textbf{5.1}$$

$$TNR / Specificity = \frac{TN}{TN + FP} \quad \textbf{5.2}$$

$$FPR = \frac{FP}{TN + FP} = 1 - \frac{TN}{TN + FP} \quad \textbf{5.3}$$

$$FNR = \frac{FN}{TP + FN} \quad \textbf{5.4}$$

$$\mathrm{Pr}ecision(P) = \frac{TP}{TP + FP} \quad \textbf{5.5}$$

$$F - measure = \frac{2PR}{(P + R)} \quad \textbf{5.6}$$

Furthermore, as discussed in Chapter 4, ten machine learning algorithms were selected in this study and the parameters of each algorithm are shown in Table 5.5. This configuration is utilized to test the performance of each classifier during the evaluation process.

Table 5.5: Parameter configurations of the selected algorithms

Classifier	Category	Parameter
Random Forest	Tree	bagSizePercent=100;batchSize=100;breakTiesRandomly=False;calcOutOfBag=False;debug=False;doNotCheckCapabilities=False;maxDepth=0;numDecimalPlaces=2;numExecutionSlots=1;numFeatures=0;numIterations=300;outputOutOfBagComplexityStatistics=False;printClassifiers=False;seed=1;storeOutOfBagPrediction=False.
J48	Tree	batchSize=100;binarySplits=False;collapseTree=True;confidenceFactor=0.25;debug=False;doNotCheckCapabilities=False;doNotMakeSplitPointActualValue=False;minNumObj=2;numDecimalPlaces=2;numFolds=3;reducedErrorPruning=False;saveInstanceData=False;seed=1;subtreeRaising=True;unpruned=False;useLaplace=False;useMDLcorrection=True.
ADTree	Tree	debug=False;numOfBoostingIterations=20;randomSeed=0;saveInstanceData=False;searchPath=Expand all paths.
SVM	Function	batchSize=100;buildCalibrationModels=False;c=1.0;calibrator=Logistic;checksTurnedOff=False;debug=False;doNotCheckCapabilities=False;epsilon=1.0E-12;filterType=Normalize;kernel=PolyKernel;numDecimalPlaces=2;numFolds=-1;randomSeed=1;toleranceParameter=0.001.
MLP	Function	GUI=False;autoBuild=True;batchSize=100;debug=False;decay=False;doNotCheckCapabilities=False;hiddenLayers=a;learningRate=0.3;momentum=0.2;nominalToBinaryFilter=True;normalizeAttributes=True;normalizeNumericClass=True;numDecimalPlaces=2;reset=True;seed=0;trainingTime=500;validationSetSize=0;validationThreshold=20.
AdaBoost	Meta/ensemble	batchSize=100;classifier=J48;debug=False;doNotCheckCapabilities=False;numDecimalPlaces=2;numIterations=10;seed=1;useResampling=False;weightThreshold=100.
Decorate	Meta/ensemble	artificialSize=2.0;batchSize=100;classifier=J48;debug=False;desiredSize=15;doNotCheckCapabilities=False;numDecimalPlaces=2;numIterations=60;seed=1.
LogitBoost	Meta/ensemble	ZMax=3.0;classifier=RandomTree;debug=False;doNotCheckCapabilities=False;likelihoodThreshold=-1.7977E308;numDecimalPlaces=2;numIterations=10;numThreads=1;poolSize=1;seed=1;shrinkage=1.0;useResampling=False;weightThreshold=100.
BayesNet	Bayes	batchSize=100;debug=False;doNotCheckCapabilities=False;estimator=SimpleEstimator;numDecimalPlaces=2;searchAlgorithm=K2;useADTree=False.
Random committee	Meta/ensemble	batchSize=100;classifier=RandomTree;debug=False;numDecimalPlaces=2;numExecutionSlots=1;numIterations=10;seed=1.

5.2 Spam Account Detection Model (SADM) Evaluation

In this stage, three main experiments were conducted to ascertain the performance of the proposed SADM. The objectives of this evaluation stage are as follows:

(a) To rank the categories of features proposed in this study based on their classification performance;

(b) To investigate a suitable machine learning algorithm for the proposed SADM based on the overall features proposed in this study;

(c) To identify the discriminating features using bio-inspired evolutionary algorithm.

5.2.1 Experiment and Procedure Description

The evaluation results of the three experiments conducted in this stage were based on the outcome of the ten classification algorithms selected in this study. Using the pre-established objectives, the following experiments were performed:

1) *Experiment 1:* The purpose of this experiment is to investigate the contributions of each feature category to the overall classification task. At this stage, the performances of the five categories of features discussed in Chapter 4 were ranked. These categories include user profile, content-based, network-based, timing-based, and automation-based features.
2) *Experiment 2:* The aim of this experiment is to investigate the overall performance of the proposed SADM and to identify the most suitable classification algorithm for the spam account detection using all the 69 features proposed in this study. The result from this experiment will aid in establishing the comparative advantages of the proposed model with existing related studies in spam account detection on Twitter network.
3) *Experiment 3:* This experiment investigates the most discriminating features among the proposed features in this study using bio-inspired evolutionary computation. The outcome of this experiment assists in deciding on the different indicators used in the risk assessment model.

5.2.2 Results and Discussions

Before ranking each feature category based on performance as reported by the selected classification algorithms, this study first investigates the behavioral differences of spammer and legitimate accounts based upon some selected features.

(a) *Behavioral differences:* Empirical cumulative distribution function (CDF) was employed to study the differences in behavior based upon some selected features. Empirical CDF is a non-parametric estimator of the underlying cumulative distribution function of the selected feature. An empirical CDF graph can be used to visualize the probability distribution of spam and legitimate accounts in the dataset. Based on the empirical CDF of profile age of spam and legitimate accounts (see Figure 5.1), it was established that spam accounts usually exhibit low profile age as compared with legitimate accounts. More than 80% of spammers have listed count close to zero (see Figure 5.2). This is because a majority of spam accounts focused more on following users than organizing their public lists on Twitter.

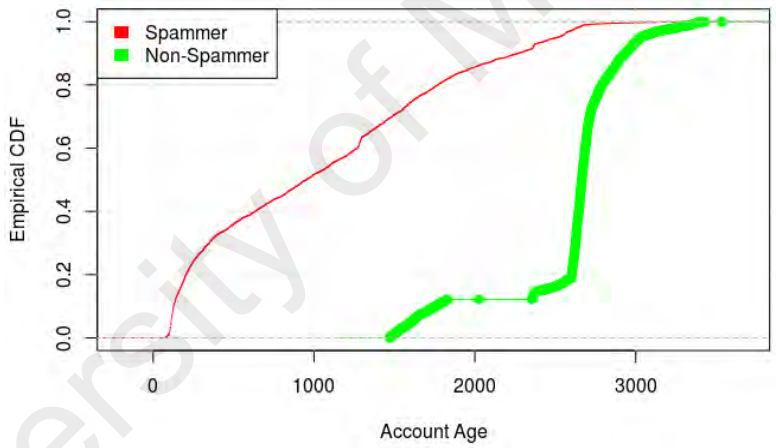


Figure 5.1: Empirical CDF of profile age

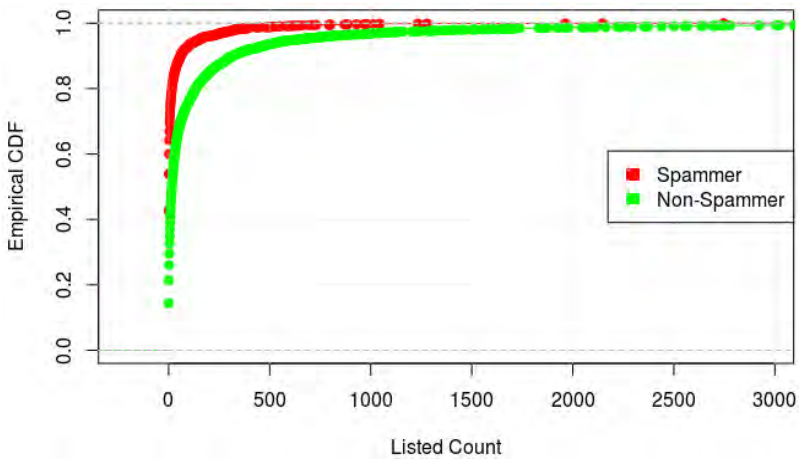


Figure 5.2: Empirical CDF of account listed count

Furthermore, based on 100 most recent tweets of each account, it was established that spammer post more tweets than legitimate users. This finding is shown in Figure 5.3. It was also established that more than 60% of spammers have their account reputations lower than 0.4 as presented in Figure 5.4, showing lack of close relationship between spam accounts and their followers.

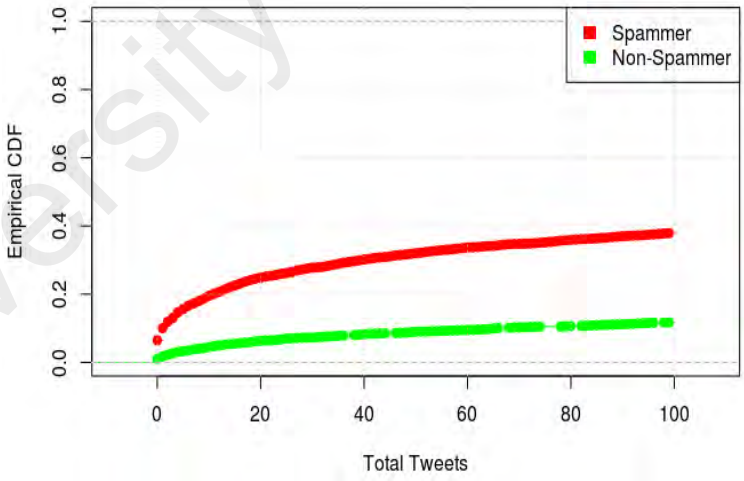


Figure 5.3: Empirical CDF based on 100 most recent tweets

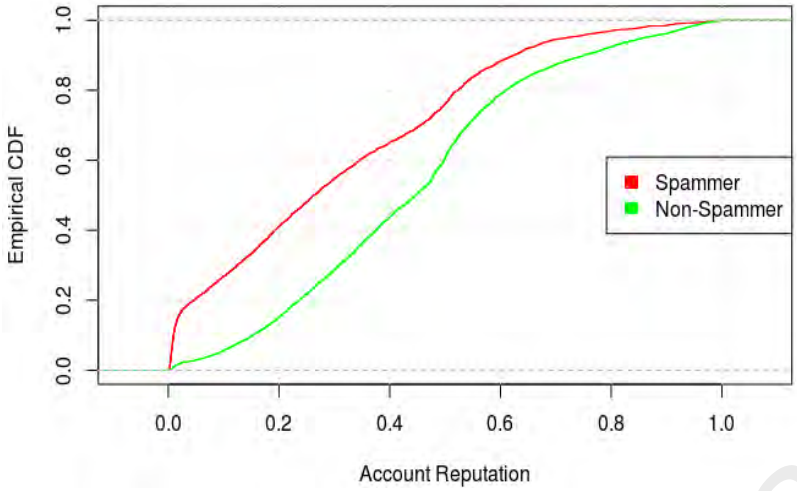


Figure 5.4: Empirical CDF of account reputation

In addition to the use of empirical CDF, this study investigates the behavioral differences between spammers and legitimate accounts along their Boolean features such as location, time zone, profile URL, default profile, and geo-enabled. As shown in Figure 5.5 and Figure 5.6, it was established that the behavior of spammers and legitimate users differ along these features. The investigation revealed that 55.15%, 54.33%, 70.34%, 59.57%, and 76.73% of spammers did not set their profile location, time zone, profile URL, profile theme, and geo-enabled features respectively. Meanwhile, in the case of legitimate users, 18.65%, 14.6%, 48.05%, 24.83%, and 49.68% do not set these features. The implication of this is that spammers focus more on sending messages along the social structure rather than devoting time to beautify their profiles or reveal some identities. In addition, spammers are conscious of disclosing some key information about their accounts in order to avoid detection.

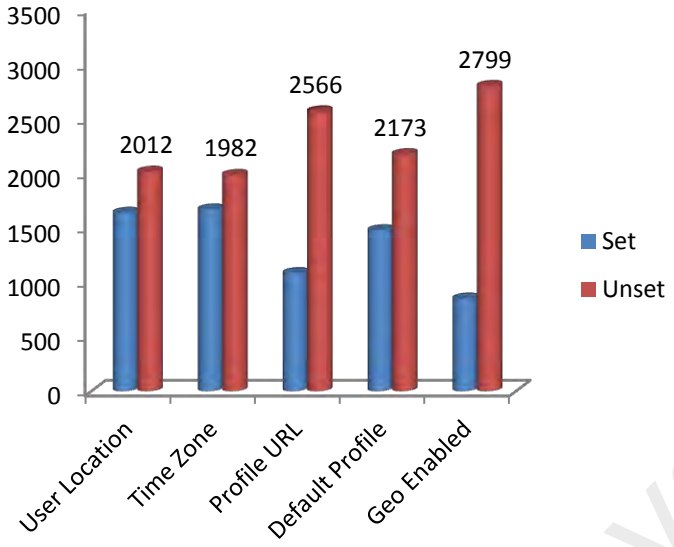


Figure 5.5: Boolean features for spammer

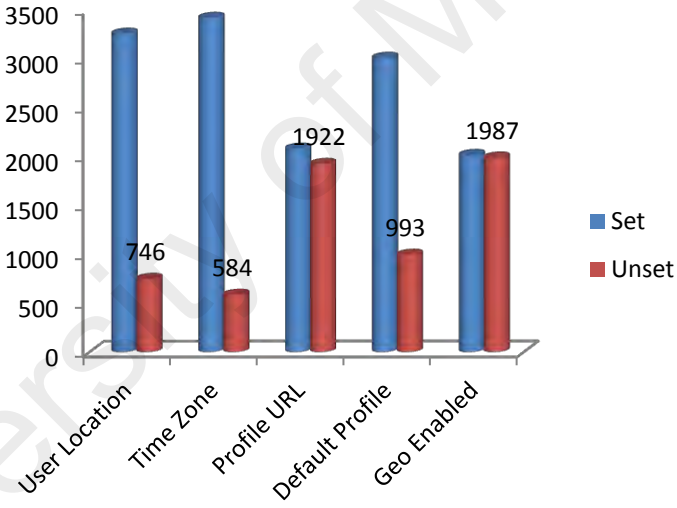


Figure 5.6: Boolean features for legitimate users

(b) *Ranking feature category:* In order to investigate the contribution of each feature category to the classification task and rank them accordingly, this study applied ten machine learning algorithms based on the selected category of feature. Each feature category is selected in isolation to train and validate the classification algorithms. According to the results of this evaluation, it was noticed that the huge contribution to the classification performance was obtained from the user profile based features with LogitBoost ensemble algorithm achieving F-measure of 92.7% and AUC-ROC

of 97% using 10-fold cross-validation (see Table 5.6). In 10-fold cross-validation, the labeled samples are divided into 10 subsets of equal size. In each round of the training, one out of 10 subsets is held as the testing set to validate the classifier, while the remaining nine subsets are used to train the classification algorithm. The implication of this result is that the behavioral profile of accounts taking into consideration the features proposed in this study provides a clear separation between spam and legitimate accounts.

Table 5.6: Classification results based on user profile features

Classifiers	10-fold cross-validation					
	DR	FPR	Precision	Recall	F1	AUC-ROC
Random Forest	0.924	0.086	0.925	0.924	0.923	0.973
J48	0.924	0.086	0.924	0.924	0.923	0.943
ADTree	0.926	0.091	0.931	0.926	0.925	0.972
SVM (SMO)	0.878	0.122	0.879	0.878	0.878	0.878
MLP	0.900	0.112	0.901	0.900	0.899	0.965
AdaBoost	0.923	0.087	0.924	0.923	0.923	0.973
Decorate	0.920	0.090	0.921	0.920	0.920	0.972
LogitBoost	**0.928**	**0.091**	**0.935**	**0.928**	**0.927**	**0.970**
BayesNet	0.865	0.135	0.866	0.865	0.865	0.931
Random committee	0.919	0.091	0.920	0.919	0.919	0.967

The performance of automation-based features follows the user profile features. The result of this classification shows a very close performance with Decorate and Random forest ensemble algorithms based on the evaluation metrics as shown in Table 5.7. For instance, Decorate produces F-measure of 79.5% and AUC-ROC of 83.9% while Random forest produces F-measure of 79.5% and AUC-ROC of 85%.

Table 5.7: Classification results based on automation features

Classifiers	10-fold cross-validation					
	DR	FPR	Precision	Recall	F1	AUC-ROC
Random Forest	**0.797**	**0.225**	**0.799**	**0.797**	**0.795**	**0.850**
J48	0.796	0.236	0.803	0.796	0.791	0.824
ADTree	0.778	0.268	0.804	0.778	0.767	0.806
SVM (SMO)	0.567	0.553	0.694	0.567	0.418	0.507
MLP	0.567	0.500	0.552	0.567	0.531	0.552
AdaBoost	0.794	0.228	0.796	0.794	0.791	0.814
Decorate	**0.800**	**0.230**	**0.807**	**0.800**	**0.795**	**0.839**
LogitBoost	0.773	0.259	0.779	0.773	0.768	0.824
BayesNet	0.729	0.306	0.733	0.729	0.722	0.767
Random committee	0.773	0.248	0.773	0.773	0.771	0.822

The third ranked feature category is content-based. In this case, Random forest and AdaBoost ensemble algorithms produced very close results as shown in Table 5.8. Random forest achieved F-measure of 75.9% and AUC-ROC of 82.5% while AdaBoost produces F-measure of 76% and AUC-ROC of 79.7%. In this experiment, the least perform classifier is Bayesian network.

Table 5.8: Classification results based on content features

Classifiers	10-fold cross-validation					
	DR	FPR	Precision	Recall	F1	AUC-ROC
Random Forest	**0.761**	**0.258**	**0.760**	**0.761**	**0.759**	**0.825**
J48	0.711	0.302	0.710	0.711	0.710	0.701
ADTree	0.720	0.301	0.719	0.720	0.717	0.783
SVM (SMO)	0.705	0.327	0.705	0.705	0.699	0.689
MLP	0.719	0.298	0.718	0.719	0.718	0.766
AdaBoost	**0.762**	**0.258**	**0.762**	**0.762**	**0.760**	**0.797**
Decorate	0.736	0.279	0.735	0.736	0.735	0.795
LogitBoost	0.722	0.299	0.721	0.722	0.719	0.769
BayesNet	0.687	0.344	0.685	0.687	0.681	0.749
Random committee	0.734	0.293	0.734	0.734	0.729	0.790

The forth ranked feature category is timing-based features with decision tree J48 and Decorate ensemble algorithms achieving close results. J48 algorithm produces F-measure of 69.9% and AUC-ROC of 70.4% while Decorate achieves F-measure 69.8% and AUC-ROC of 72.2% as shown in Table 5.9. The implication of this result is that spammers' following and tweeting activities based on their respective profile age deviates from the normal accounts behaviors. In this experiment, MLP achieves the least classification accuracy.

Table 5.9: Classification results based on timing features

Classifiers	10-fold cross-validation					
	DR	FPR	Precision	Recall	F1	AUC-ROC
Random Forest	0.656	0.368	0.653	0.656	0.652	0.687
J48	**0.707**	**0.329**	**0.709**	**0.707**	**0.699**	**0.704**
ADTree	0.701	0.335	0.702	0.701	0.693	0.726
SVM (SMO)	0.567	0.553	0.701	0.567	0.550	0.561
MLP	0.566	0.548	0.579	0.566	0.560	0.555
AdaBoost	0.650	0.373	0.647	0.650	0.647	0.665
Decorate	**0.706**	**0.330**	**0.707**	**0.706**	**0.698**	**0.722**
LogitBoost	0.687	0.321	0.687	0.687	0.687	0.727
BayesNet	0.693	0.329	0.691	0.693	0.690	0.717
Random committee	0.610	0.401	0.610	0.610	0.610	0.604

Finally, the fifth ranked feature based on category is the mentioned graph-based network. In this experiment, Decorate classifier achieves good result. It was observed that J48 decision tree, AdaBoost and Random forest algorithm also produced close results with Decorate ensemble classifier (see Table 5.10). This result indicates that the mention behavior of spammer also deviate from the legitimate accounts.

Table 5.10: Classification results based on mention network features

Classifiers	10-fold cross-validation					
	DR	FPR	Precision	Recall	F1	AUC-ROC
Random Forest	0.678	0.350	0.676	0.678	0.673	0.732
J48	0.677	0.359	0.676	0.677	0.668	0.697
ADTree	0.662	0.383	0.663	0.662	0.649	0.708
SVM (SMO)	0.637	0.375	0.637	0.637	0.637	0.631
MLP	0.653	0.394	0.654	0.653	0.638	0.675
AdaBoost	0.678	0.351	0.676	0.678	0.672	0.687
Decorate	**0.682**	**0.357**	**0.684**	**0.682**	**0.673**	**0.730**
LogitBoost	0.665	0.368	0.663	0.665	0.658	0.705
BayesNet	0.651	0.355	0.653	0.651	0.652	0.704
Random committee	0.651	0.379	0.647	0.651	0.645	0.695

(c) *Results based on 69 features:* In this experiment, the performance of the SADM model is established using the 69 features proposed in this study. Two training methods, 10-fold and percentage split, were utilized to establish the performance of the proposed SADM. The percentage split is based on 80% training and 20% validation method. The result using 10-fold cross-validation shows that Random forest classifier achieves the best result producing F-measure of 93.2% and AUC-ROC of 97.7% (see Figure 5.7). In percentage split as shown in Figure 5.8, Decorate ensemble classifier achieves F-measure of 94% and AUC-ROC of 97.5%. This results show that the proposed features in this study for detecting spam accounts on Twitter are effective and the performances of the various classifiers selected are also promising.

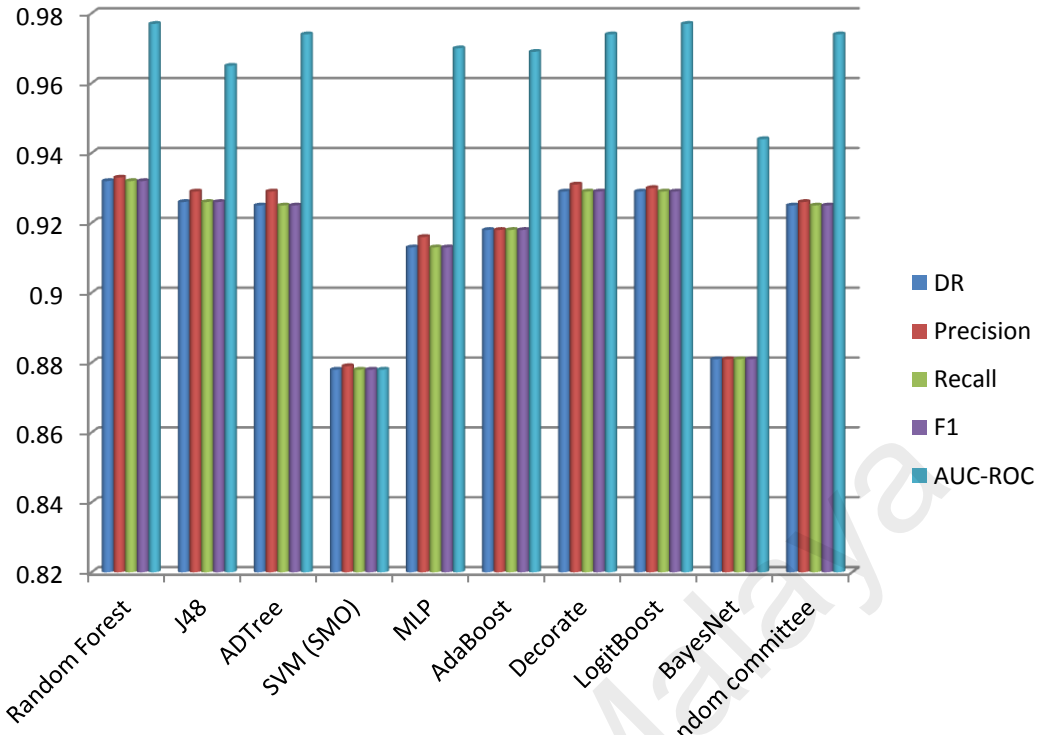


Figure 5.7: Classification results based on 10-fold

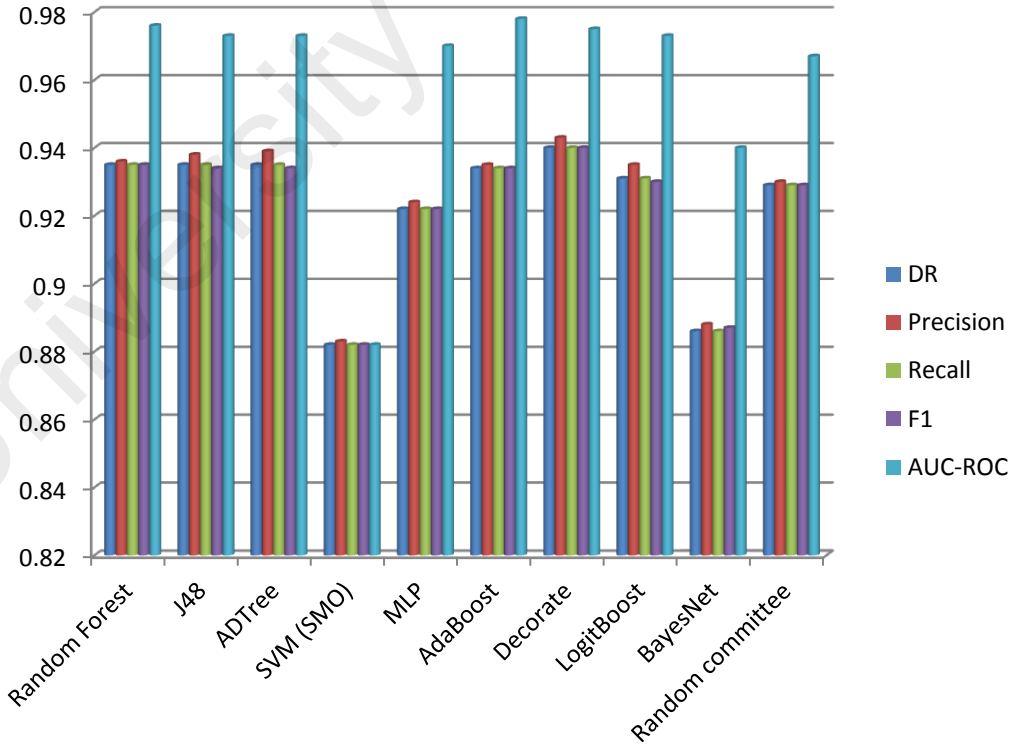


Figure 5.8: Classification results based on percentage split

(c) *Bio-inspired feature identification:* The purpose of this experiment is to investigate the most discriminating features, which in turn will assist in selecting alternative indicators used to model the risk assessment section of the proposed unified framework. EA algorithm is selected among the bio-inspired evolutionary computing algorithms due to its adaptation and wide acceptance in the literature. The implementation of the EA algorithm is done on WEKA machine learning software. Chi-squared test for feature selection was first applied to select 60 features using ranker search method and these features were passed to EA algorithm to identify the reduced features that will enable us make an informed decision. The parameters used for the EA algorithm is shown in Table 5.11. EA produces eighteen (18) features as shown in Table 5.12.

Table 5.11: Parameters configuration for EA algorithm

Parameter	Value
crossoverOperator	spx-crossover
crossoverProbability	0.6
generation	20
initializationOperator	random-init
mutationOperator	bit-flip
mutationProbability	0.01
populationSize	20
replacementOperator	generational
reportFrequency	20
seed	1
selectionOperator	tournament-selection

Table 5.12: Eighteen (18) features produced by EA

Feature name
Age in days
User Time zone
Listed count
User location
Following rate
Average TF-IDF score
Tweet similarity
Follower following ratio
Default profile
Average sentiment subjectivity
In-degree of mention
Total tweet favorite count
Popularity ratio
Profile URL

Table 5.12, continued.

Local clustering coefficient of mention
Deviation of link
Bidirectional link of mention
Favourites count

After obtaining the 18 features from the evolutionary search algorithm, the next step is to evaluate the reduced data on the selected classifiers. The result of this experiment is shown in Figure 5.9. Out of the ten classifiers, the results of seven classifiers improved using only the 18 features identified by the evolutionary search algorithm as compared with all the 69 features. Although the improvement could not reach the accuracies reported for Random forest and Decorate when all the 69 features were applied, however, using the bio-inspired evolutionary search method, LogitBoost ensemble classifier achieves a result close to Random Forest based on the 69 features. LogitBoost produced F-measure of 93% and AUC-ROC of 97.7 based on the 18 features.

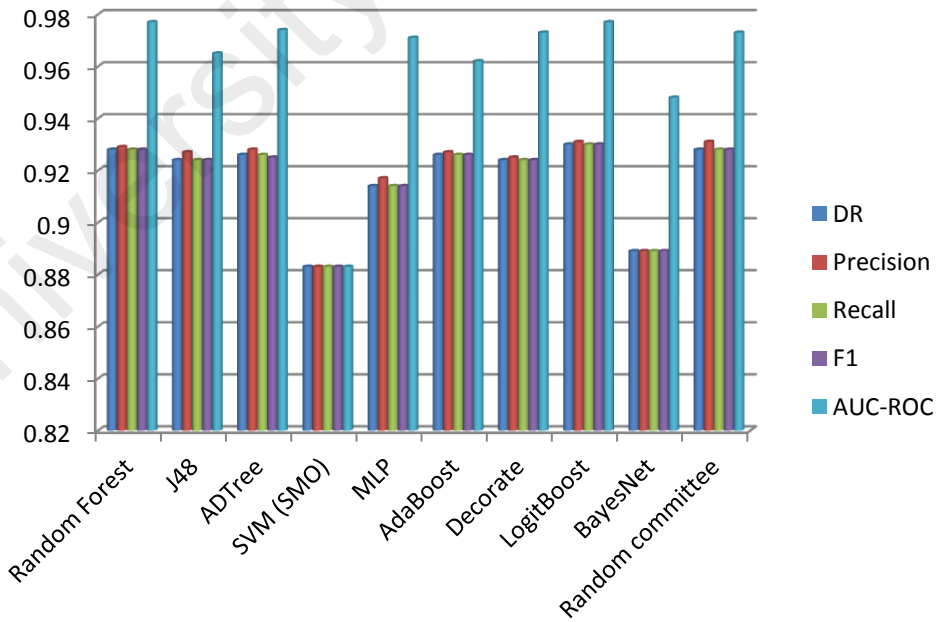


Figure 5.9: Results based on evolutionary algorithm

5.2.3 Conclusion and Limitation

The first stage of this evaluation study has shown the performance of the five categories of features proposed in Chapter 4. The results of this evaluation show that the first rank feature category is user profile followed by automation features. This is followed by content, timing, and network based features respectively. The second stage proposed a suitable classification algorithm for the proposed SADM using all the features identified in this study. The third stage of the evaluation applied evolutionary search method to identify the discriminating features for spam account detection. The overall goal of the experiments in this section is to have a clear understanding of the performance of the proposed SADM.

The results of the experiments conducted in this evaluation section have established that spam and legitimate account behaviors deviate significantly along the proposed features in this study. In the search to identify the most suitable classifier for the proposed SADM for spam accounts detection on Twitter, it was observed that decision tree and ensemble-based classifiers are suitable options for this classification task. Specifically, the performances of the Random forest and Decorate algorithms have been promising across the three experiments. Thus, this study proposed the use of Random forest or Decorate classifier as the suitable algorithm for the SADM.

When conducting the experiments in this section, this study found some limitations concerning the proposed approach. Below are some of the limitations and suggestions to reduce them in the future experiment:

(a) *Performance evaluation:* Although the performance of the proposed SADM has been quite promising, the model is unable to achieve 100% performance accuracy. Therefore, more features can be identified in the future experiments to further improve the performance of the proposed SADM.

(b) *Feature extraction time:* The features used for developing the SADM require a considerable amount of time to be able to extract for all the accounts in the Twitter dataset due to the millions of tweets involved. In the future experiment, a more sophisticated feature extraction module can be developed to improve the complexity of the feature extraction stage.

(c) *Parameter configurations:* The performance of the SADM presented in this study is based on the different configuration settings of the selected classifiers. Therefore, in the future experiment, the parameters of the algorithms can be further tuned to verify the behavior of the selected algorithms.

(d) *Label samples:* This study adopted Twitter suspension algorithm method for identifying labeled samples in order to have a consistent and efficient approach to label the spam accounts in the Twitter dataset. However, this approach requires that the dataset be left for a longer period before more labeled samples can be discovered.

5.3 Spam Message Detection Model (SMDM) Evaluation

The purpose of this evaluation stage is to ascertain the performance of the proposed SMDM on both Twitter and mobile SMS datasets. To achieve this aim, two experiments were conducted with the objectives stated as follows:

(a) To identify the most suitable classification algorithm for spam message detection on Twitter microblogging network;

(b) To identify the most suitable classification algorithm for mobile SMS spam message detection;

5.3.1 Experiment and Procedure Description

The evaluation results of the two experiments conducted in this stage were based on the outcome of the ten classification algorithms selected in this study. Using the pre-established objectives in this section, the following experiments were performed:

1) *Experiment 1:* In this experiment, Twitter spam corpus (i.e *Dset2*) is applied to train and validate the ten selected classifiers in order to investigate the extent to which the 18 features identified in this study for spam message detection can detect spam message on Twitter network.
2) *Experiment 2:* The aim of this experiment is to investigate the performance of the proposed SMDM on mobile SMS datasets using *Dset3* and *Dset4* corpora. At this stage, the two mobile SMS datasets were used to train and validate the ten selected algorithms based on the 18 features discussed in Chapter 4. The result from this experiment will aid in establishing the comparative advantages of the proposed model with existing related studies in spam message detection.

5.3.2 Results and Discussions

Before proceeding with the discussions of the results of the two experiments conducted in this evaluation section, this study first investigates the behavioral differences of spam and legitimate users based on their message contents. The discussion is then followed by the results of the proposed SMDM using Twitter corpus and the two mobile SMS corpora.

(a) *Behavioral differences:* Empirical CDF was employed to study the differences in behavior based upon some selected features. Figure 5.10, Figure 5.11, and Figure 5.12 show the empirical CDF of the length of spam and legitimate messages for each of the corpora respectively. These figures show that the length of spam messages is longer than legitimate messages with over 60% of spammers on Twitter

using more than 130 characters to compose spam messages. Similarly, over 60% of spammers utilized more than 150 characters to compose mobile spam messages based on the two mobile SMS spam corpora.

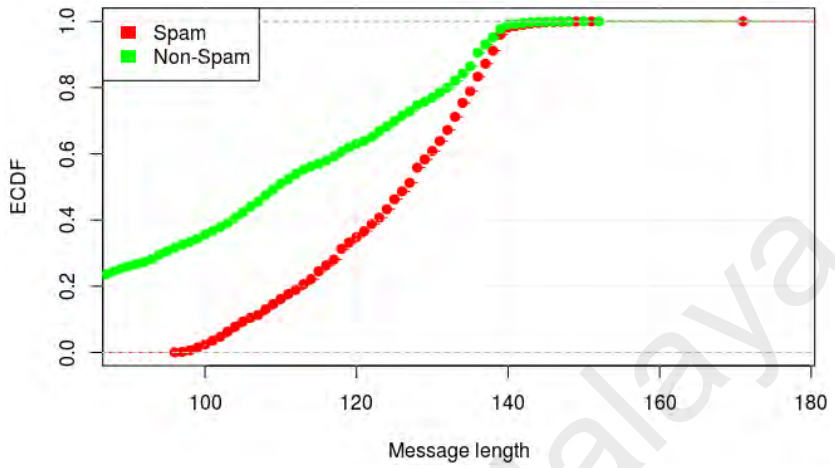


Figure 5.10: Empirical CDF of message length based on Dset2

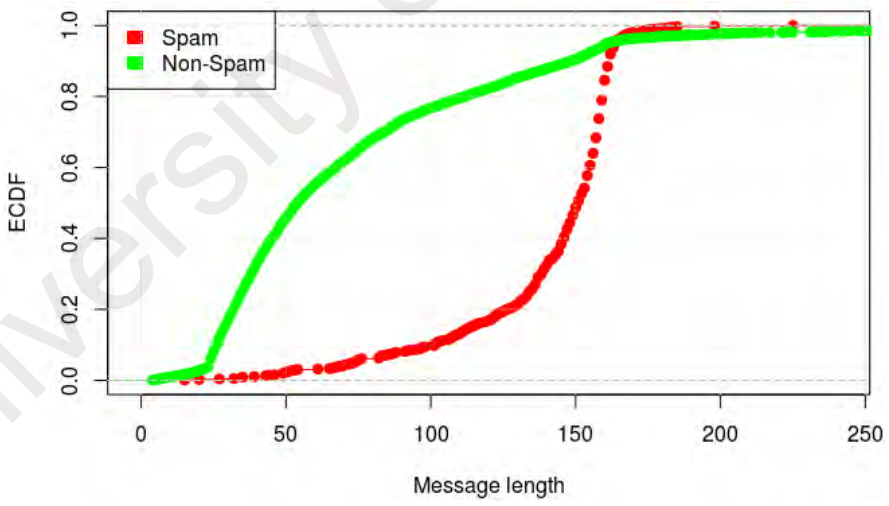


Figure 5.11: Empirical CDF of message length based on Dset3

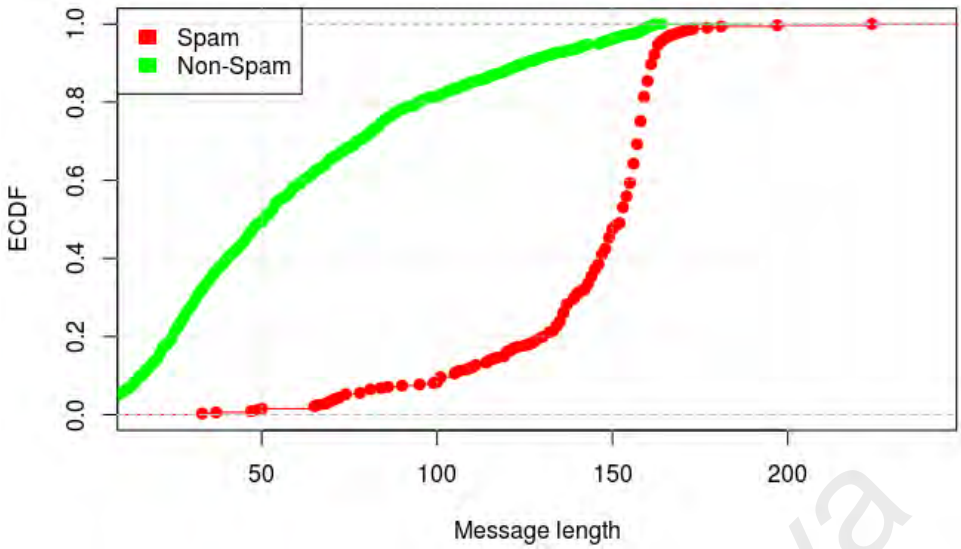


Figure 5.12: Empirical CDF of message length based on Dset4

Figure 5.13, Figure 5.14, and Figure 5.15 show the distribution of words that appear in each class category based on the three datasets respectively. These figures show that the number of words in spam messages is more than the legitimate messages. These findings reveal that a majority of spammers leveraged the maximum character length of spam messages to further deceive their victims. As a result, they tend to use more words during message composition than legitimate users.

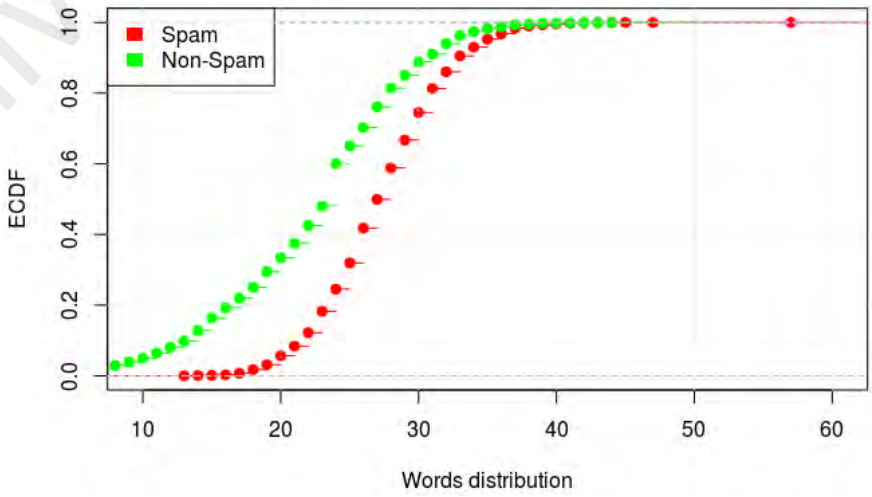


Figure 5.13: Empirical CDF of words distribution for Dset2

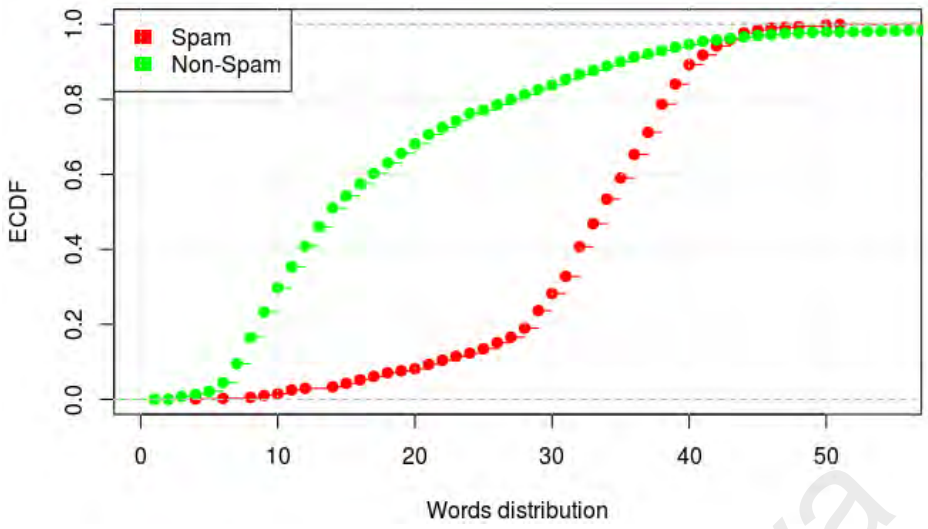


Figure 5.14: Empirical CDF of words distribution for Dset3

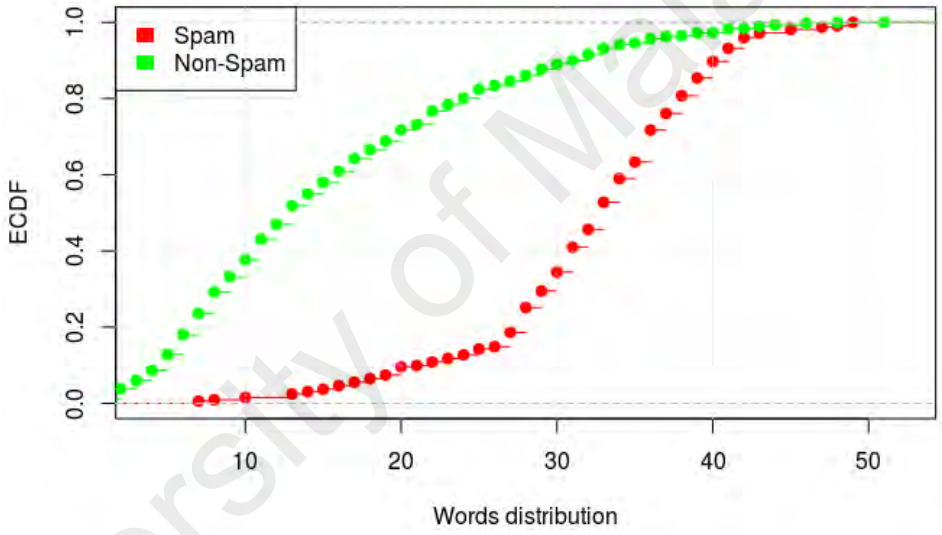


Figure 5.15: Empirical CDF of words distribution for Dset4

(b) *Twitter spam message detection:* This section presents experiment to evaluate the performance of the selected classifiers for Twitter spam message detection. The objective of this evaluation study is to find the most suitable machine learning algorithm for Twitter spam message detection, which can be incorporated in the proposed unified framework. As shown in Figure 5.16 Random forest outperformed other algorithms in this experiment. The result obtained is promising for identifying spam message on Twitter network. Random forest produced F-measure of 93.2%

and AUC-ROC of 98.3%. This shows the applicability of the proposed SMDM for detecting spam message on Twitter.

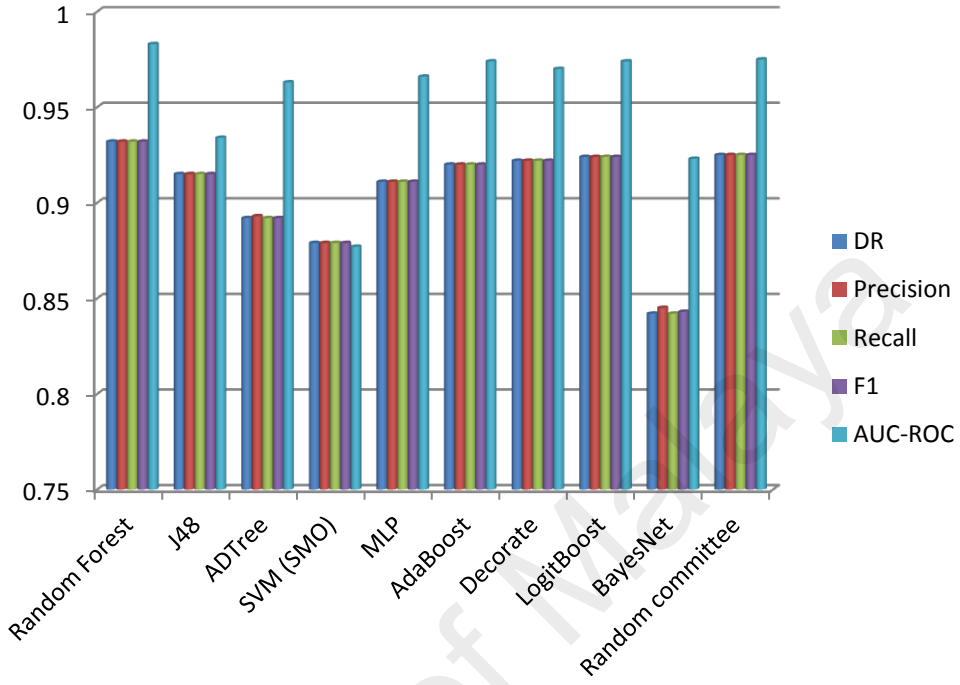


Figure 5.16: Classification results with Dset2

(c) *Mobile SMS spam message detection:* The aim of this evaluation study is to ascertain the performance of the proposed SMDM when used to detect SMS spam messages. The performance of the selected classification algorithms are examined on the two SMS spam datasets, Dset3 and Dset4. This experiment is based on 10-fold cross-validation training method. Random Forest classifier achieves the best results for the two experiments on SMS spam message detection. As shown in Figure 5.17, Random Forest produces the best accuracy, F-measure, and AUC-ROC of 99.2%, 99.1%, and 99.7% respectively. The least performed classifier on Dset3 is Bayesian network.

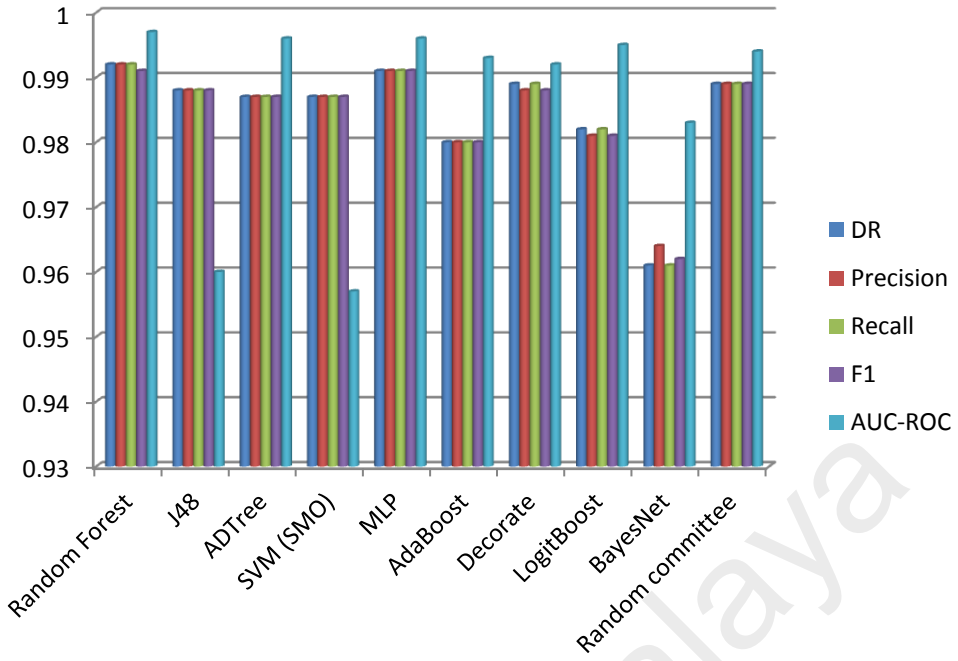


Figure 5.17: Classification results with Dset3

Figure 5.18, Random forest produces accuracy, F-measure, and AUC-ROC of 99.1%, 99.1%, and 99.9% respectively. As observed in the previous results on *Dset3*, Bayesian network also achieves the least accuracy on *Dset4*.

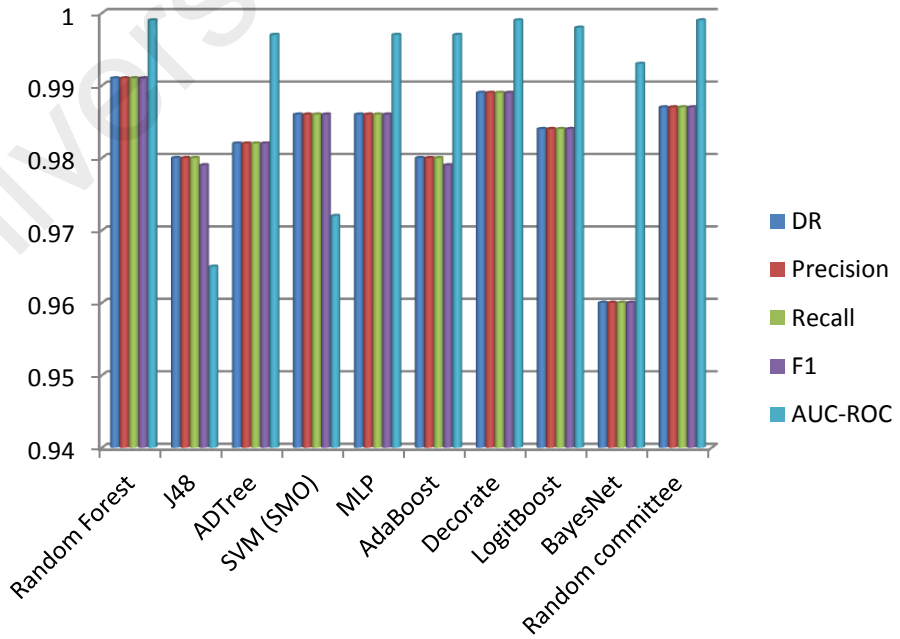


Figure 5.18: Classification results with Dset4

5.3.3 Conclusion and Limitation

The first stage of this evaluation study has shown the performance of the proposed SMDM based on the 18 features identified in this study to detect spam message on Twitter network. The results of this evaluation show that Random forest is the most suitable classification algorithm for the proposed SMDM. Similarly, the results of the second stage of the evaluation also revealed that Random forest algorithm is suitable for detecting mobile SMS spam messages based on the classification results on the two corpora.

The results of the experiments conducted in this evaluation section have established that spam and legitimate users' behaviors also deviate along their message posting pattern. Specifically, the features proposed in this study have revealed that spammer's behavior across different SMCM is similar in terms of the messages shared with legitimate users. In conclusion, it was established from these experiments that the most suitable classifiers for the proposed SMDM in this study is Random forest, which achieved better performances across the different corpora.

In conducting the evaluation experiments presented in this section, some limitations were identified which need to be addressed in the future study. Some of these limitations are highlighted as follows:

(a) *Message length:* Despite the large number of labeled samples used to evaluate the proposed SMDM on Twitter spam message corpus, the domain specific words commonly used on Twitter still have little effect on the classification performance as compared with mobile SMS spam corpora. However, the approach used in this study by combining various spam words used by spammers on Twitter to extract useful feature for the proposed SMDM improves the performance of the classification algorithms. In the future experiment, a sophisticated method can be

developed to expand the domain specific words used on Twitter network before the actual features proposed in this study are extracted.

(b) *Noise data:* During the feature extraction stage on the Twitter corpus, it was discovered that a lot of messages on Twitter are noisy due to the nature of the social network. In this study, the length of the messages used in the evaluation results was set to 100 characters and more. This is to ensure that some of the noisy data are filtered out from the experimental analysis. This challenge can be addressed in the future experiment using a more robust method.

5.4 Spam Risk Assessment Model (SRAM) Evaluation

This evaluation stage is conducted to ascertain the performance of the proposed SRAM for risk assessment on Twitter network. The motivation behind this evaluation study is to categorize Twitter accounts based on their risk level. The overall goal of this evaluation is to build a risk assessment model based on Fuzzy AHP method. The experiment conducted in this stage utilized the Twitter spam account dataset (i.e *Dset1*) using ten indicators as alternatives as discussed in Chapter 4. These indicators were selected from the results of the evolutionary features search approach to provide a model that relies on most discriminating indicators. Therefore, in this stage, two experiments were conducted to ascertain the performance of the proposed SRAM. The objectives of this evaluation stage are as follows:

(a) To compute the global weight for each alternative using Fuzzy AHP method;

(b) To obtain the risk scores and categorize Twitter accounts based on their risk level.

5.4.1 Experiment and Procedure Description

Based upon the objectives established in this evaluation study, the following experiments were performed:

1) *Experiment 1:* In this experiment, Fuzzy AHP judgment matrices were designed based on subjective opinion. The consistency of this opinion was determined using the Saaty consistency ratio formula. The first stage of the experiment involved construction of the judgment matrix based upon risk likelihood and impact, subject to the overall risk assessment objective. The judgment matrices for the alternatives were then constructed subject to both likelihood and risk impact. These pairwise comparison matrices were applied to compute the global weight for each alternative indicator.

2) *Experiment 2:* The aim of this experiment is to apply the global weight in the first experiment to determine the risk score for each Twitter account. To achieve this objective, JRip rule induction algorithm was applied to normalize the dataset. The normalized dataset was used to determine the risk score for each account.

5.4.2 Results and Discussions

The results of the two experiments conducted in this section are discussed as follows:

(a) *Global weight computation:* Table 5.13 presents the fuzzy comparison matrix of the likelihood and impact criteria subject to the overall risk assessment objective. As shown in the table, the likelihood that a risk will occur on Twitter is rated higher than the impact in order to prevent early occurrence of critical incidence. Fuzzy centroid defuzzification method was used to obtain the crisp priority vector, which gives the weights for both likelihood and impact criteria.

Table 5.13: Fuzzy judgment matrix for likelihood and impact criteria subject to spam risk assessment

Criteria	Likelihood	Impact	Fuzzy priority vector	Crisp priority vector
Likelihood	(1,1,1)	(6,7,8)	(0.866,**0.875**,0.875)	0.875
Impact	(1/8,1/7,1/6)	(1,1,1)	(0.125,**0.125**,0.126)	0.125
$\lambda_{max} = 2.0000$, CI = 0.0000, RI = 0.000, CR = undefined				

Table 5.14 and Table 5.15 shows the fuzzy comparison matrices of the alternative indicators subject to both likelihood and impact criteria. These tables show that the two fuzzy judgment matrices are consistent, which indicates that the expert opinion is acceptable. This can be seen according to the values of CR obtained for the two matrices, which are both less than 10% proposed by Saaty. It is important to mention that before computing the CR values for the fuzzy judgment matrices, centroid defuzzification method was applied to obtain crisp judgments. This enables faster computation of Saaty CR in order to evaluate expert opinion on the fuzzy judgments.

Table 5.14: Fuzzy judgment matrix of alternatives subject to likelihood criteria

	AD	UT	LC	UL	FR	FF	AT	TS	IM	DP	Fuzzy priority vector	Crisp priority vector
AD	(1,1,1)	(1,1,2)	(1,1,2)	(2,3,4)	(4,5,6)	(1,1,2)	(2,3,4)	(2,3,4)	(2,3,4)	(1,1,2)	(0.147,**0.149**,0.204)	0.149
UT	(1/2,1,1)	(1,1,1)	(1/4,1/3,1/2)	(1,1,2)	(2,3,4)	(1/4,1/3,1/2)	(1,1,2)	(1,1,2)	(1/6,1/5,1/4)	(1/4,1/3,1/2)	(0.053,**0.056**,0.074)	0.056
LC	(1/2,1,1)	(2,3,4)	(1,1,1)	(1,1,2)	(6,7,8)	(1,1,2)	(1,1,2)	(2,3,4)	(1/6,1/5,1/4)	(1,1,2)	(0.104,**0.106**,0.138)	0.106
UL	(1/4,1/3,1/2)	(1/2,1,1)	(1/2,1,1)	(1,1,1)	(2,3,4)	(1/4,1/3,1/2)	(1/4,1/3,1/2)	(1,1,2)	(1/6,1/5,1/4)	(1/4,1/3,1/2)	(0.043,**0.05**,0.06)	0.05
FR	(1/6,1/5,1/4)	(1/4,1/3,1/2)	(1/8,1/7,1/6)	(1/4,1/3,1/2)	(1,1,1)	(1/10,1/9,1/8)	(1/8,1/7,1/6)	(1/4,1/3,1/2)	(1/10,1/9,1/8)	(1/8,1/7,1/6)	(0.018,**0.018**,0.02)	0.018
FF	(1/2,1,1)	(2,3,4)	(1/2,1,1)	(2,3,4)	(8,9,10)	(1,1,1)	(1,1,2)	(4,5,6)	(1/4,1/3,1/2)	(1,1,2)	(0.119,**0.134**,0.158)	0.134
AT	(1/4,1/3,1/2)	(1/2,1,1)	(1/2,1,1)	(2,3,4)	(6,7,8)	(1/2,1,1)	(1,1,1)	(2,3,4)	(1,1,2)	(1,1,2)	(0.094,**0.111**,0.129)	0.111
TS	(1/4,1/3,1/2)	(1/2,1,1)	(1/4,1/3,1/2)	(1/2,1,1)	(2,3,4)	(1/6,1/5,1/4)	(1/4,1/3,1/2)	(1,1,1)	(1/6,1/5,1/4)	(1/4,1/3,1/2)	(0.036,**0.043**,0.049)	0.043
IM	(1/4,1/3,1/2)	(4,5,6)	(4,5,6)	(4,5,6)	(8,9,10)	(2,3,4)	(1/2,1,1)	(4,5,6)	(1,1,1)	(1,1,2)	(0.181,**0.194**,0.22)	0.194
DP	(1/2,1,1)	(2,3,4)	(1/2,1,1)	(2,3,4)	(6,7,8)	(1/2,1,1)	(1/2,1,1)	(2,3,4)	(1/2,1,1)	(1,1,1)	(0.101,**0.138**,0.138)	0.138

λmax = 10.914 , CI = 0.102, RI = 1.49, CR = 0.068

Table 5.15: Fuzzy judgment matrix of alternatives subject to impact criteria

	AD	UT	LC	UL	FR	FF	AT	TS	IM	DP	Fuzzy priority vector	Crisp priority vector
AD	(1,1,1)	(1/2,1,1)	(1/2,1,1)	(1/4,1/3,1/2)	(1/6,1/5,1/4)	(1/2,1,1)	(1/4,1/3,1/2)	(1/4,1/3,1/2)	(1/4,1/3,1/2)	(1/2,1,1)	(0.03,**0.041**,0.042)	0.041
UT	(1,1,2)	(1,1,1)	(2,3,4)	(1/2,1,1)	(1/4,1/3,1/2)	(2,3,4)	(1/2,1,1)	(1/2,1,1)	(4,5,6)	(2,3,4)	(0.083,**0.11**,0.116)	0.11
LC	(1,1,2)	(1/4,1/3,1/2)	(1,1,1)	(1/2,1,1)	(1/8,1/7,1/6)	(1/2,1,1)	(1/2,1,1)	(1/4,1/3,1/2)	(4,5,6)	(1/2,1,1)	(0.045,**0.059**,0.06)	0.059
UL	(2,3,4)	(1,1,2)	(1,1,2)	(1,1,1)	(1/4,1/3,1/2)	(2,3,4)	(2,3,4)	(1/2,1,1)	(4,5,6)	(2,3,4)	(0.103,**0.123**,0.143)	0.123
FR	(4,5,6)	(2,3,4)	(6,7,8)	(2,3,4)	(1,1,1)	(8,9,10)	(6,7,8)	(2,3,4)	(8,9,10)	(6,7,8)	(0.306,**0.343**,0.343)	0.343
FF	(1,1,2)	(1/4,1/3,1/2)	(1,1,2)	(1/4,1/3,1/2)	(1/10,1/9,1/8)	(1,1,1)	(1/2,1,1)	(1/6,1/5,1/4)	(2,3,4)	(1/2,1,1)	(0.039,**0.046**,0.052)	0.046
AT	(2,3,4)	(1,1,2)	(1,1,2)	(1/4,1/3,1/2)	(1/8,1/7,1/6)	(1,1,2)	(1,1,1)	(1/4,1/3,1/2)	(1/2,1,1)	(1/2,1,1)	(0.048,**0.056**,0.066)	0.056
TS	(2,3,4)	(1,1,2)	(2,3,4)	(1,1,2)	(1/4,1/3,1/2)	(4,5,6)	(2,3,4)	(1,1,1)	(4,5,6)	(2,3,4)	(0.126,**0.145**,0.171)	0.145
IM	(2,3,4)	(1/6,1/5,1/4)	(1/6,1/5,1/4)	(1/6,1/5,1/4)	(1/10,1/9,1/8)	(1/4,1/3,1/2)	(1,1,2)	(1/6,1/5,1/4)	(1,1,1)	(1/2,1,1)	(0.028,**0.032**,0.034)	0.032
DP	(1,1,2)	(1/4,1/3,1/2)	(1,1,2)	(1/4,1/3,1/2)	(1/8,1/7,1/6)	(1,1,2)	(1,1,2)	(1/4,1/3,1/2)	(1,1,2)	(1,1,1)	(0.045,**0.045**,0.061)	0.045

λmax = 10.953 , CI = 0.106, RI = 1.49, CR = 0.071

After establishing the consistency of the fuzzy judgment matrices subject to both likelihood and impact criteria, Table 5.16 shows the final global weights of the alternative indicators. Similarly, centroid defuzzification method was used to obtain the crisp weights, which was applied in the next experiment.

Table 5.16: Global weights of alternative indicators

Alt.	Fuzzy priority vector	Crisp priority vector
AD	(0.131,**0.136**,0.184)	0.136
UT	(0.057,**0.063**,0.08)	0.063
LC	(0.095,**0.10**.129)	0.10
UL	(0.05,**0.059**,0.071)	0.059
FR	(0.054,**0.059**,0.061)	0.059
FF	(0.108,**0.123**,0.145)	0.123
AT	(0.087,**0.104**,0.121)	0.104
TS	(0.047,**0.055**,0.064)	0.055
IM	(0.16,**0.174**,0.197)	0.174
DP	(0.093,**0.127**,0.129)	0.127

(b) *Risk assessment:* In this experiment, the risk level of each Twitter account in the ground truth dataset is computed using the global weights of the alternatives. Before computing the risk score, Table 5.17 shows the rules generated by JRip rule induction algorithm for data normalization. In total, 29 rules were generated using JRip algorithm. The normalized value assigned to each rule is taken from the interval of [0,1] with 1 indicating that the spam characteristics of the accounts covered by this rule is on the high side, while 0 indicates highly normal behavior.

Table 5.17: Rules generated using JRip for data normalization

Features/Alternatives	Generated rules	Normalized value
Age in days	**Rule 1:** (age_in_days <= 2352) and (age_in_days <= 1469) => class=Spam (2553.0/1.0)	1
	Rule 2: (age_in_days <= 2591) and (age_in_days <= 2352) and (age_in_days >= 1831) => class=Spam (348.0/1.0)	0.9
	Rule 3: ELSE class=Non-Spam (4747.0/749.0)	0.2
Time Zone	**Rule 1:** (time_zone <= 0) => class=Spam (2433.0/583.0)	1
	Rule 2: ELSE => class=Non-Spam (5215.0/1798.0)	0.5
Listed Count	**Rule 1:** (listed_count <= 3) => class=Spam (3516.0/1179.0)	1
	Rule 2: ELSE => class=Non-Spam (4132.0/1311.0)	0.5
User location	**Rule 1:** (location <= 0) => class=Spam (2494.0/745.0)	1

Table 5.17, continued.

	Rule 2: ELSE => class=Non-Spam (5154.0/1899.0)	0.5
Following rate	**Rule 1:** (following_rate >= 0.892746) => class=Spam (2855.0/907.0)	1
	Rule 2: ELSE => class=Non-Spam (4793.0/1700.0)	0.5
Follower following ratio	**Rule 1:** (follower_following_ratio <= 0.257962) => class=Spam (2162.0/630.0)	1
	Rule 2: (follower_following_ratio <= 0.869796) and (follower_following_ratio <= 0.452444) and (follower_following_ratio <= 0.304582) and (follower_following_ratio >= 0.276331) => class=Spam (168.0/79.0)	0.9
	Rule 3: (follower_following_ratio <= 0.504747) and (follower_following_ratio <= 0.394249) and (follower_following_ratio >= 0.373124) => class=Spam (110.0/48.0)	0.8
	Rule 4: (follower_following_ratio <= 0.600735) and (follower_following_ratio <= 0.353881) and (follower_following_ratio >= 0.331288) => class=Spam (134.0/65.0)	0.7
	Rule 5: ELSE => class=Non-Spam (5074.0/1896.0)	0.2
Average TF-IDF	**Rule 1:** (avg_tfidf >= 0.1019) => class=Spam (1793.0/549.0)	1
	Rule 2: (avg_tfidf <= 0.073938) => class=Spam (487.0/87.0)	0.9
	Rule 3: (avg_tfidf >= 0.087029) and (avg_tfidf >= 0.09939) => class=Spam (132.0/64.0)	0.8
	Rule 4: => class=Non-Spam (5236.0/1936.0)	0.2
Tweet Similarity	**Rule 1:** (avg_tweet_similarity >= 0.035457) => class=Spam (2300.0/800.0)	1
	Rule 2: (avg_tweet_similarity <= 0.011339) => class=Spam (394.0/85.0)	0.9
	Rule 3: (avg_tweet_similarity >= 0.021221) and (avg_tweet_similarity <= 0.02258) and (avg_tweet_similarity >= 0.022325) => class=Spam (75.0/32.0)	0.8
	Rule 4: ELSE => class=Non-Spam (4879.0/1796.0)	0.2
Degree of mention	**Rule 1:** (deg_mention <= 15) => class=Spam (2188.0/648.0)	1
	Rule 2:	0.9

Table 5.17, continued.

	(deg_mention <= 48) and (deg_mention <= 18) => class=Spam (167.0/77.0)	
	Rule 3: => class=Non-Spam (5293.0/2018.0)	0.2
Default profile	**Rule 1:** (default_profile >= 1) => class=Spam (3038.0/993.0)	1
	Rule 2: ELSE => class=Non-Spam (4610.0/1603.0)	0.5

These rules were used to normalize the ground truth dataset. After applying JRip rule induction algorithm, Eqn. 4.45 was utilized to compute risk scores for all the Twitter accounts in the dataset. The distribution of the risk scores were examined for each account category using the boxplot in Figure 5.19. This distribution is clearly shown in Table 5.18 by analyzing the distribution in percentile. From this distribution, the risk level membership function is defined, which was applied to assign the risk level to each Twitter account.

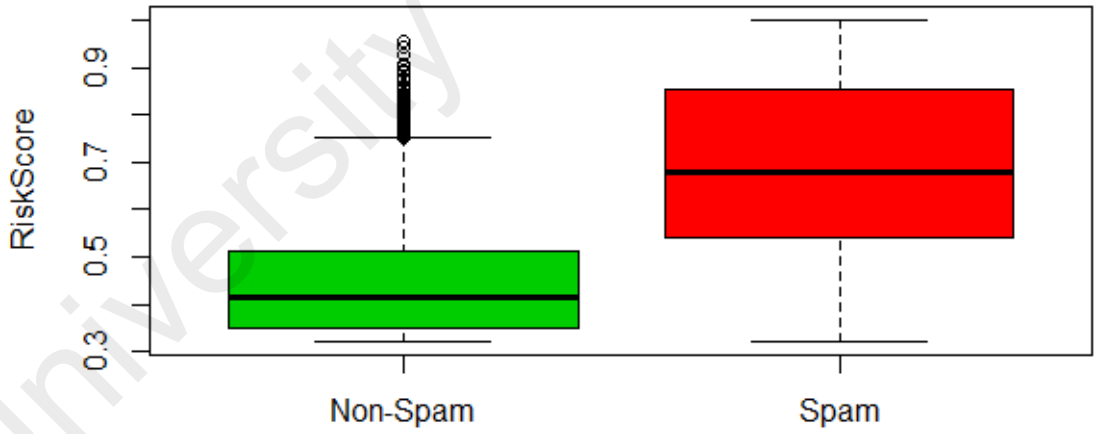


Figure 5.19: Distribution of risk score based on account category

Table 5.18: Distribution in percentile for each account category

Distribution	0%	25%	50%	75%	100%
Spam	0.322400	0.542200	0.679500	0.851675	1.000000
Non-Spam	0.3224	0.3519	0.4154	0.5116	0.9569

Table 5.19 shows the summary of the risk level assigned to each account category. From this table, a majority of legitimate accounts were categorized under low risk level while majorities of spam accounts were categorized under medium and very high risk. Using the summary of the results of the proposed SRAM, the accuracy of the model on legitimate accounts is 93.35% while for spam accounts is 83.20% based on the risk level assessment. These outcomes show the capability of applying the proposed SRAM for risk assessment on Twitter social network.

Table 5.19: Account category and risk level

Range	Risk	Spam	Non-Spam
0 - 0.5	Low	613(16.80%)	2940(73.50%)
0.5 - 0.7	Medium	1343(36.82%)	794(19.85%)
0.7 - 0.8	High	513(14.06%)	164(4.1%)
0.8 - 1	Very high	1179(32.32%)	102(2.55%)

Figure 5.20 and Figure 5.21 clearly show the proportion of both spam and legitimate accounts with their risk levels. Similarly, Figure 5.22 and Figure 5.23 present the percentage distribution of each account category based upon the computed risk levels.

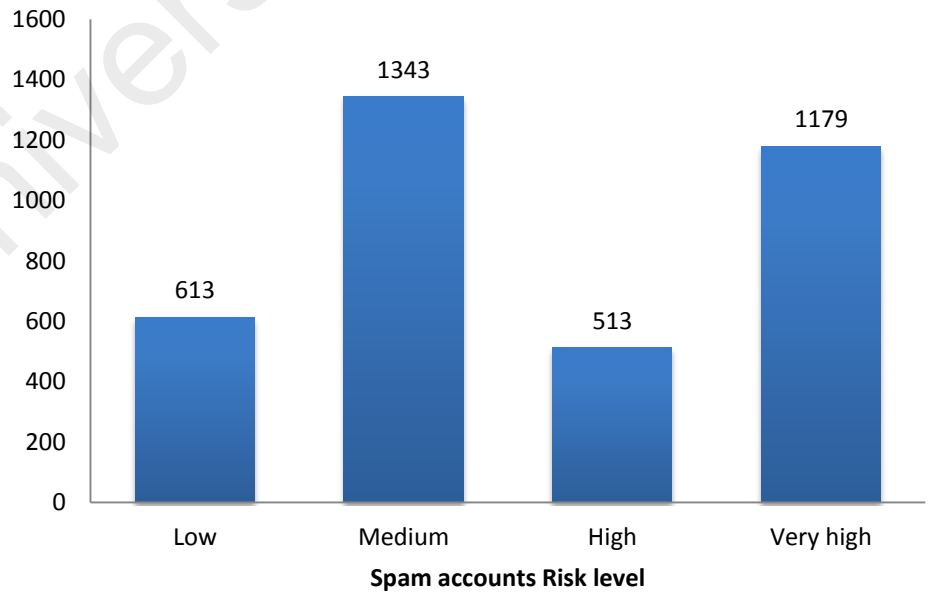


Figure 5.20: Proportion of spam accounts with risk level

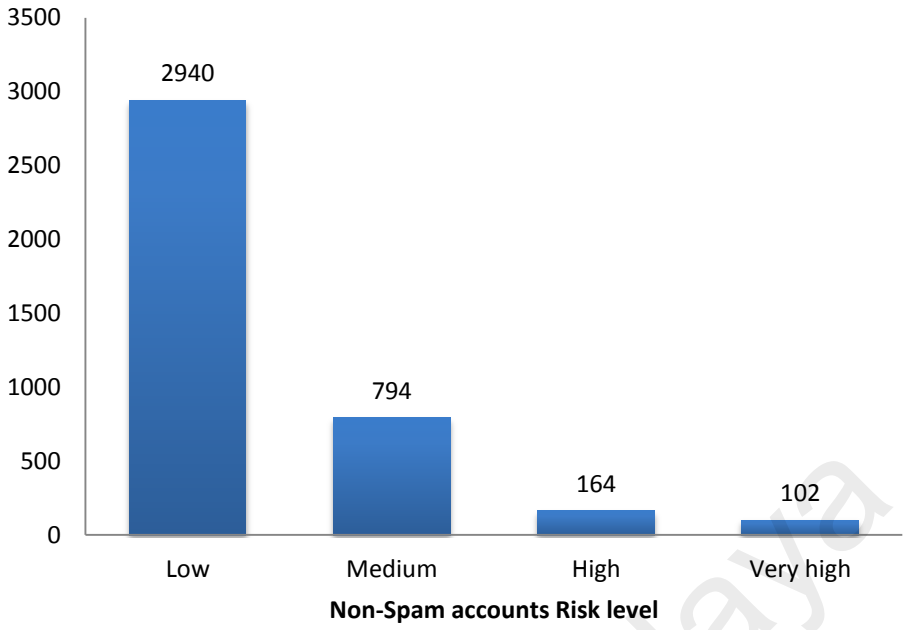


Figure 5.21: Proportion of non-spam accounts with risk level

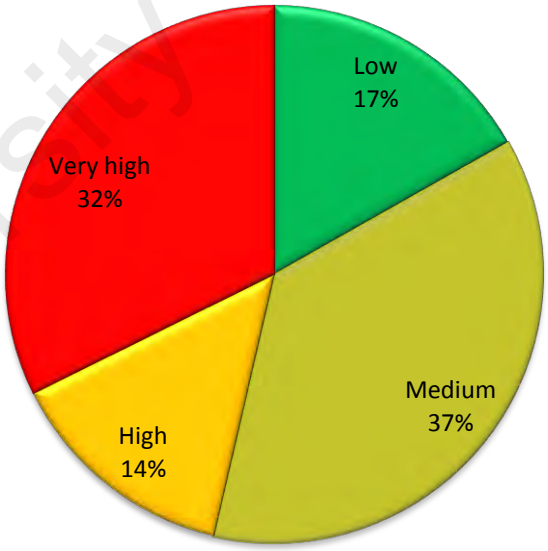


Figure 5.22: Percentage distribution of spam accounts with risk

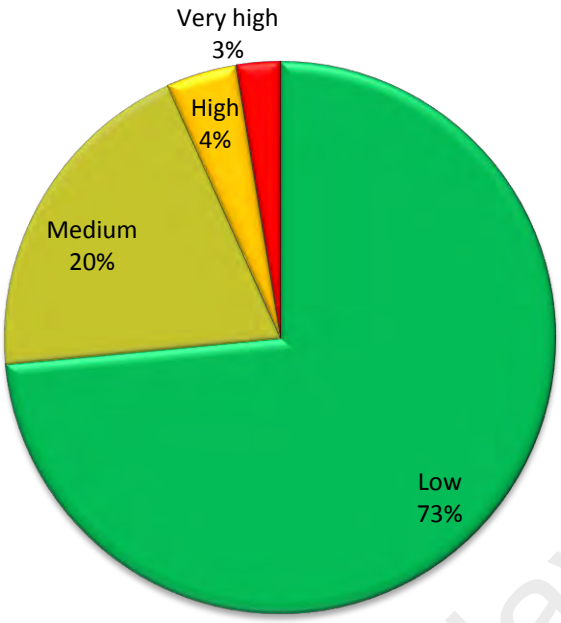


Figure 5.23: Percentage distribution of non-spam accounts with risk

Appendix D1 shows the sample results of legitimate accounts classified as low risk with their risk score and level presented at the last two columns. The sample value for each alternative indicator shows the result of the data normalization step. Similarly, Appendix D2, D3, D4, and D5 show the sample results for spam accounts based on the normalized data respectively.

5.4.3 Conclusion and Limitations

In this evaluation stage, the first phase shows the computation of global weights for the alternative indicators using fuzzy AHP method. Fuzzy comparison matrix based on likelihood and impact criteria was first developed. The alternatives indicators were compared subject to both likelihood and risk impact. One of the advantages of fuzzy AHP in this experiment is the ability to model the subjective opinion of an expert. Since risk assessment deals with a lot of uncertainty, the proposed fuzzy AHP helps to accommodate this characteristic. The second stage of the evaluation applied the

computed global weights to derive risk level of each Twitter account. In this study, four risk linguistic variable were considered namely low, medium, high, and very high risk. The overall goal of this experiment is to provide a clear understanding of the performance of the proposed SRAM.

The results of the experiments conducted in this evaluation section have established that the proposed SRAM can categorize legitimate accounts on Twitter with an accuracy of 93.35% and spam accounts with 83.20%. This result shows the applicability of the proposed SRAM in the unified framework.

In conclusion, when conducting the experiments in this section, this study found some limitations concerning the proposed SRAM for risk assessment. Some of these limitations are highlighted as follows:

(a) *Performance accuracy:* Although the performance of the proposed SRAM model is quite promising, further improvement in the performance accuracy of the proposed model is an important area in the future experiment.

(b) *Expert assumption:* In this evaluation study, assumptions were made as regard the comparison of one criterion or alternative indicator to another based on subjective opinion of the expert. These assumptions involve the strategy used to provide the weighting scales for each judgment matrix. Although AHP approach provides consistency measure to accept or reject the assumptions used to construct the comparison matrices, the future study can critically look into this area to further reduce the value of the consistency measure.

(c) *Method comparison:* As discussed in Chapter 4, this study applied Ramik fuzzy AHP approach; therefore, the results presented in this evaluation study are only based on this methodology. However, it is important to consider a situation where the results of different fuzzy AHP methods such as extent analysis and Ramik

approach are compared in order to provide more findings on the performance of the proposed SRAM based on variation in the fuzzy AHP method applied to compute the global weights.

(d) *Rating threshold:* The rating threshold provides opportunity to establish the risk level membership function used in this evaluation section. This rating threshold was based on the distribution of the risk score computed for both spam and legitimate account categories. This means that the approach used to obtain the rating threshold is based on local computation method in relation to the distribution of spam and legitimate accounts in the ground truth dataset.

5.5 Performance Evaluation

In this section, the proposed models in this study are compared with existing related works in order to gain more insights on their performances when compared with other models in the literature. The evaluation section is divided into three main categories. The first section compares the performance of the proposed SADM with other related studies on spam accounts detection on Twitter network. The second section of the evaluation, compare the performance of the proposed SMDM with other related studies. The third section focuses on SRAM model verification, which provides opportunity to verify the performance of the proposed SRAM when used for risk assessment based upon new set of data that were not utilized during the model training and testing.

5.5.1 Baseline comparison for Spam Account Detection

Several approaches have been studied in the literature for spam accounts detection on Twitter using different datasets. The variation in datasets is due to the Terms of Use of the Twitter API, which disallowed researchers from sharing tweets data. Therefore, to benchmark with the existing approaches selected in the literature, a baseline method is employed. This method provides an unbiased comparison and helps us to ascertain if

the proposed features have indeed improved the model performance. Using baseline method, the features used in the related studies were extracted from the dataset described in this study (see section 5.1.1) in order to generate new datasets used to train and validate the classifier employed in the related studies. Therefore, the comparison results in this section are based on the performance of the proposed SADM in this study with three related works on spam accounts detection. Table 5.20 shows the results of the comparison of the proposed SADM with Shyni et al. (2016), Yang et al. (2013), and Gao et al. (2016).

Table 5.20: Performance comparison of SADM with related studies

Models	Evaluation metrics					
	DR	FPR	Precision	Recall	F1	AUC-ROC
Proposed method	**0.940**	**0.070**	**0.943**	**0.940**	**0.940**	**0.975**
Shyni et al. (2016)	0.866	0.170	0.891	0.866	0.862	0.873
Yang et al. (2013)	0.896	0.109	0.896	0.896	0.896	0.946
Gao et al. (2016)	0.675	0.372	0.679	0.675	0.661	0.713

The results from this table clearly show that using baseline comparison, the proposed SADM model outperformed other approaches in the related studies. These results show that the proposed method with 69 features has improved the classification performance by improving detection rate, error rate, precision, recall, f-measure, and AUC-ROC. This comparison result further confirmed that the proposed unified framework is promising for detecting spam accounts on Twitter network.

5.5.2 Performance comparison of Spam Message Detection

In the case of SMS spam message detection, the proposed SMDM is compared with related works based on the studies that utilized the two public SMS datasets described earlier. For instance, El-Alfy and AlHasan (2016), Almeida et al. (2013), and Ezpeleta et al. (2016) evaluated their models using Dset3 (SMS Collection V.1). The results of this comparison are shown in Table 5.21. On this dataset, the proposed SMDM model

improves in precision and F-measure when compared with El-Alfy and AlHasan (2016), although their method slightly outperformed the proposed approach in terms of detection rate, recall, and AUC-ROC. The performance of the proposed SMDM still provides promising results achieving AUC-ROC of 99.7%. When compared with Almeida et al. (2013) and Ezpeleta et al. (2016) models, the proposed method shows a significant improvement based on the model accuracy. In the case of Dset4 (SMS Corpus V.0.1 Big), the proposed method achieved the same level of performance in F-measure and AUC-ROC with El-Alfy and AlHasan (2016) and improves in precision as shown in Table 5.22.

Table 5.21: Performance comparison of SMDM with related studies on Dset3

Models	Evaluation metrics					
	DR	FPR	Precision	Recall	F1	ROC
Proposed method	**0.992**	**0.048**	**0.992**	**0.992**	**0.991**	**0.997**
El-Alfy and AlHasan (2016)	0.994	N/A	0.980	0.997	0.988	0.999
Almeida et al. (2013) - SVM + tok1	0.9764	N/A	N/A	N/A	N/A	N/A
Ezpeleta et al. (2016)	0.9891	N/A	N/A	N/A	N/A	N/A

Table 5.22: Performance comparison of SMDM with related studies on Dset4

Models	Evaluation metrics					
	DR	FPR	Precision	Recall	F1	ROC
Proposed method	**0.991**	**0.018**	**0.991**	**0.991**	**0.991**	**0.999**
El-Alfy and AlHasan (2016)	0.993	N/A	0.987	0.996	0.991	0.999

Since the Twitter SMS spam corpus is a private dataset, this study benchmarks the performance of the proposed SMDM on Twitter with Bag of words model. The Bag of words model was implemented using *NaiveBayesMultinomialText* classifier in WEKA, which deals specifically with text classification task. The parameters used for *NaiveBayesMultinomialText* classifier are shown in Table 5.23. The results of this performance evaluation are shown in Table 5.24. From this table, the proposed method significantly outperformed the popular Bag of words based on the performance metrics

employed for comparison. This finding shows that the proposed model is able to distinguish spam and legitimate messages more than the popular Bag of words model.

Table 5.23: Parameters configuration of NaiveBayesMultinomialText classifier

Parameter	Value
LNorm	2.0
batchSize	100
debug	False
doNotCheckCapabilities	False
lowercaseTokens	False
minWordFrequency	3.0
norm	1.0
normalizedDocLength	False
numDecimalPlaces	2
periodicPruning	0
stemmer	NullStemmer
tokenizer	WordTokenizer
usewordFrequencies	False

Table 5.24: Performance comparison of SMDM with bag of words model

Models	Evaluation metrics					
	DR	FPR	Precision	Recall	F1	ROC
Proposed method	**0.932**	**0.070**	**0.932**	**0.932**	**0.932**	**0.983**
Bag of words	0.842	0.155	0.845	0.842	0.843	0.923

5.5.3 Spam Risk Assessment Model Verification

This section provides how the performance of the proposed SRAM was verified. Since this study introduced the first approach based on Fuzzy AHP to assess the risk level of social network accounts, a number of significant accounts was selected outside the ground truth earlier identified to verify SRAM performance. Using the Twitter rules as established in (Twitter, 2016), 9070 accounts were manually verified after computing their risk scores by applying the proposed SRAM. Table 5.25 shows the breakdown of this analysis and Figure 5.24 shows the percentage distribution of the selected samples after computing the risk scores. Using Table 5.25, the overall performance of the proposed SRAM model is estimated at 88.50%, considering the multiclass nature of this assessment.

Table 5.25: SRAM verification

Risk level	Total before verification	Correctly classified	Incorrectly classified
Low	5552	5437	115
Medium	1938	1133	805
High	560	514	46
Very high	1020	943	77

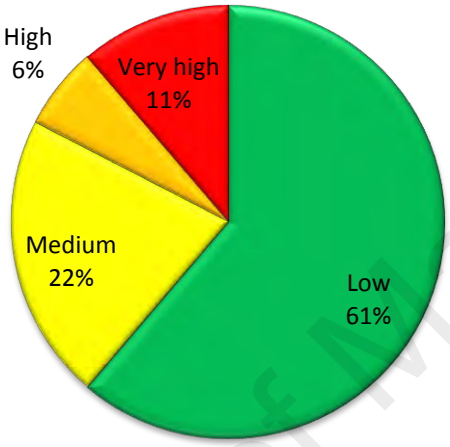


Figure 5.24: Percentage distribution of the selected samples

5.6 Summary

This chapter presented the evaluation studies that were conducted in order to ascertain the performance of the proposed unified framework based on the models incorporated. The evaluation studies have highlighted the results, performances, conclusions, as well as the limitations of the proposed methods. At each stage of the evaluation study, objectives were set to guide the different experiments conducted. The goal of each evaluation stage is to present the uniqueness of the proposed models based upon their performances on the various datasets discussed in this study.

It was evident from the results of the evaluations that the proposed unified framework for spam detection and risk assessment is promising and robust based upon

its operational characteristics. In addition, the comparison studies further strengthen the performance and the suitability of the proposed unified framework for spam detection and risk assessment in SMCM. In conclusion, the analysis presented in this chapter, clearly defined the contributions of each model to the proposed framework as well as revealing their limitations.

Therefore, in order to investigate the usefulness and feasibility of the proposed unified framework, a prototype is presented in the next chapter, which is based on online evaluation of the different models incorporated in the proposed unified framework.

CHAPTER 6: PROTOTYPE IMPLEMENTATION

Having discussed the performance of the proposed unified framework in terms of the results obtained from the proposed models, the next stage is to design and implement a prototype of the proposed framework. This prototype demonstrates the key components of the proposed framework and shows the applicability of the framework in practice. This chapter highlights the implementation of the prototype with specific focus on the three proposed models embedded within the unified framework. The implementation of the three models has been carried out using web based interfaces to demonstrate the framework applicability in real life scenarios. This chapter explains the Use case diagram of the Unified Modeling Language (UML) to visualize the design of the prototype.

6.1 Implementation overview

There are three main models incorporated in the proposed unified framework as discussed in Chapter 4 namely, SADM, SMDM, and SRAM. These three models have been implemented in the prototype. The SADM can categorize Twitter account as either spam or legitimate. After the outcome of SADM, the SRAM model is called to analyze the account based on its risk level. Both SADM and SRAM work hand-in-hand within the prototype to provide a comprehensive analysis of the Twitter account under investigation. The purpose of the SMDM is to categorize both mobile SMS and Twitter messages as either spam or legitimate messages. Each of the three models has been implemented within the web modules of the prototype. The prototype was developed using Flask and MySQL.

Flask is a micro web framework developed using Python. Flask provides support for extensions, which can be used to add application features as if they were implemented in Flask itself. There are different extensions that can be embedded within Flask such as

object-relational mappers, form validation, templates, open authentication technologies and many common framework related tools. During the prototype development, we embedded different standard Python libraries such as MySQLdb, tweepy, pandas, numpy, nltk and so on. These libraries can be seen in the prototype source codes presented in Appendix A. MySQL database was used to store and retrieve tweets and user data from Twitter API server as well as the results of the three web modules used to implement the proposed models.

6.2 Prototype Functionalities

In order to gain more insights on the main functionality of the prototype as well as the proposed unified framework as a whole, this section presents Use case diagram to describe the functionality of a system. This section also presents the three web modules used to implement the proposed SADM, SMDM, and SRAM models.

6.2.1 Use Case Diagram

According to Dan and Neil (2005), Use case diagram shows the functionality of a system in terms of actors, their goals represented as use cases, and the dependencies among those Use cases. Use case diagram consists of named pieces of functionality (Use cases), the persons or things invoking the functionality (actors), and possibly the elements responsible for implementing the Use cases (subjects). Use case diagram has been widely used to portray a graphical representation of a functional description of interaction among external entities and system, as well as their collaborations. This diagram captures the behaviours of the system without having to specify how those behaviours are implemented in details. Figure 6.1 shows the Use case diagram of the prototype components and operations. This diagram shows that there are five major actors: Admin, User, Web Application Server, Twitter API Server, and MySQL Server,

which collaborate to achieve the aim of the system. The roles of each actor are described as follows:

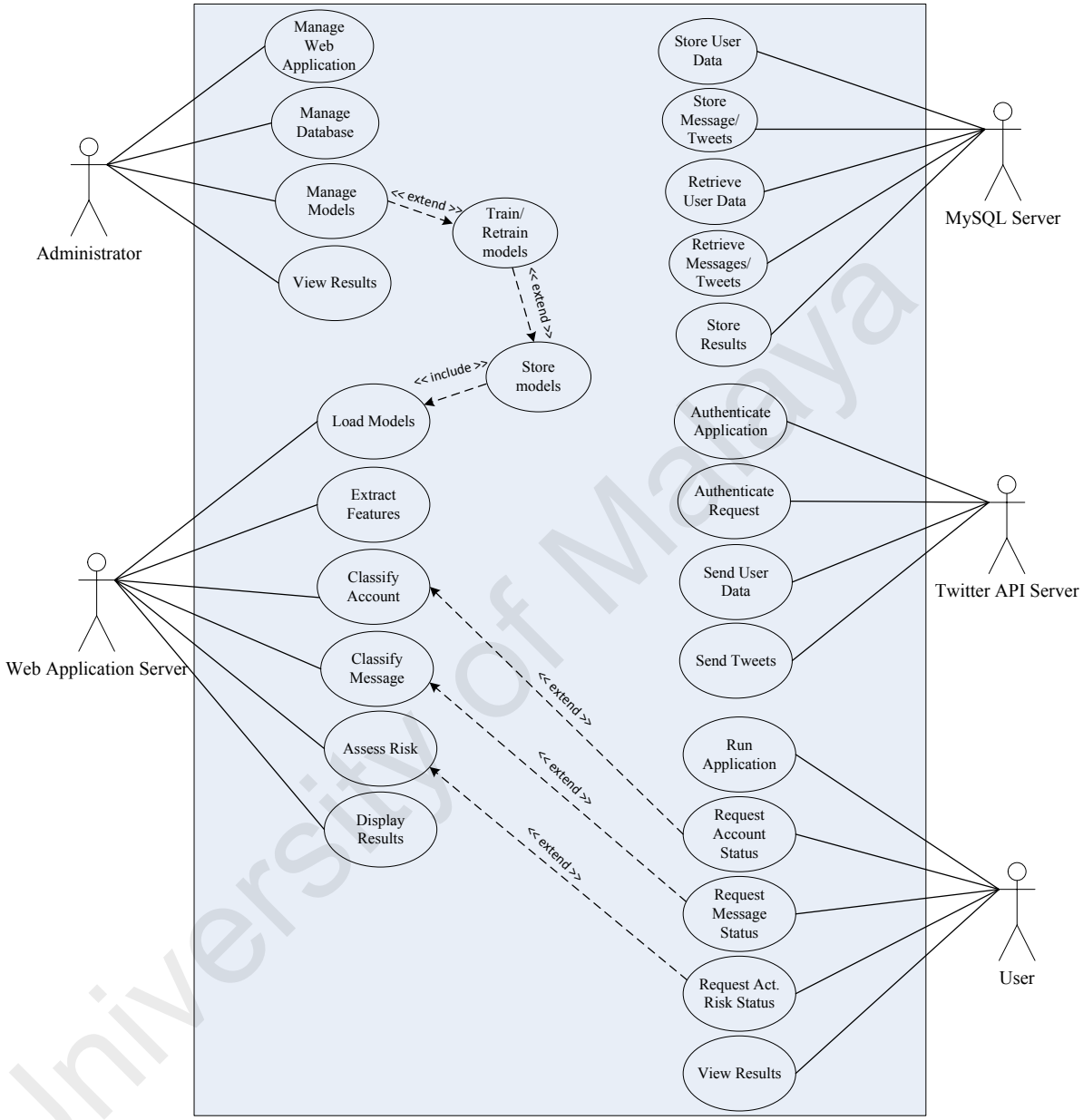


Figure 6.1: Prototype Use case diagram

a) *Administrator:* The function of an administrator is to maintain the web application as well as the database of the proposed system. Administrator is responsible to train or re-train the models and save them on the web server for subsequent use. Administrator also has the opportunity to view the different classification results

generated by the system. He/she must ensure that the application is always available online to serve the users need.

b) *User:* The roles of user are to run the application and make necessary requests from the system, such as requesting for message or account status as well as trying to find out the risk level as an account. In a real life situation, user may be confronted with critical decision or perhaps in a dilemma when a particular friendship request is received from an unknown account on Twitter network. In order to have more information about the behavior of such unknown account, the user will simply supply the account id or screen name to the system. The system will then analyze this account and display the results of the analyses to the user. This will help the user to decide whether to accept or reject the friendship request from the unknown account.

c) *Web Application Server:* Web application server plays important roles such as loading the specific model to be used to answer user's request, extracting features for message or user analysis, classifying message, classifying Twitter account as well as assessing the risk level an account. This server is also responsible for displaying the results of any user's request, which are presented to the user as HTML document.

d) *Twitter API Server:* This server is responsible for authenticating the web application before any request can be granted. It ensures that user's requests are also authenticated to prevent illegal access or violation of user's privacy. For instance, if a user makes a request about an account whose tweets have been protected, Twitter API will deny such request and the web application server will display an error message to the user. However, if the user's request is granted, Twitter API will return both tweets and profile details of the requested account to the web application server, which in turn used the data to classify the account and display results.

e) *MySQL Server:* The server is responsible for storing and retrieving tweets or messages to be classified by the system. It also stores and retrieve user's data returned from Twitter API. The results of the system classifications are stored in MySQL database.

6.2.2 Web Modules

The web modules present graphical user-friendly interfaces, which enable user to interact with the system. The three models proposed in this study have been embedded under the web modules. The web modules provide a convenient and flexible way of implementing the proposed unified framework. The modules give a broader view of the functionality of the developed prototype. The entry point to the three modules embedded within the system is shown in Figure 6.2. These modules are discussed under three main headings as follows:

Figure 6.2: Entry point to the three web modules

1) *SMS Message Classification Module:* This module is responsible for classifying mobile SMS message as either spam or legitimate. Message to be classified is loaded into the text message box as shown in Figure 6.2. The user then clicks on classify command button to activate the module. This module implements SMDM model using Random Forest classification algorithm.
2) *Twitter Message Classification Module:* This module is responsible for classifying tweets as either spam or legitimate message. The module also implements SMDM model using Random Forest classification algorithm.
3) *Account Classification and Risk Assessment Module:* The module classifies Twitter account as either spam or legitimate account as well as providing the risk level of the account under investigation. The web module implements both SADM and SRAM models. SADM was implemented using Random Forest classification algorithm based on the findings discussed in Chapter 5. The SRAM model was implemented using Fuzzy AHP method.

6.3 System demonstration

Having presented the main functionalities of the developed system, this section shows an online demonstration of the prototype of the proposed framework based on selected sample cases. For privacy reason, the user screen name will not be displayed as part of the results during Twitter account classification and risk assessment. This is important to ensure that user's privacy is not violated while demonstrating the capabilities of the proposed system. The first step when executing the prototype is to start the web application server so that Figure 6.2 can be displayed successfully. Figure 6.3 shows the process of starting the web application server using Pycharm IDE for Python project development. After starting the web application server, the three web modules are available for testing.

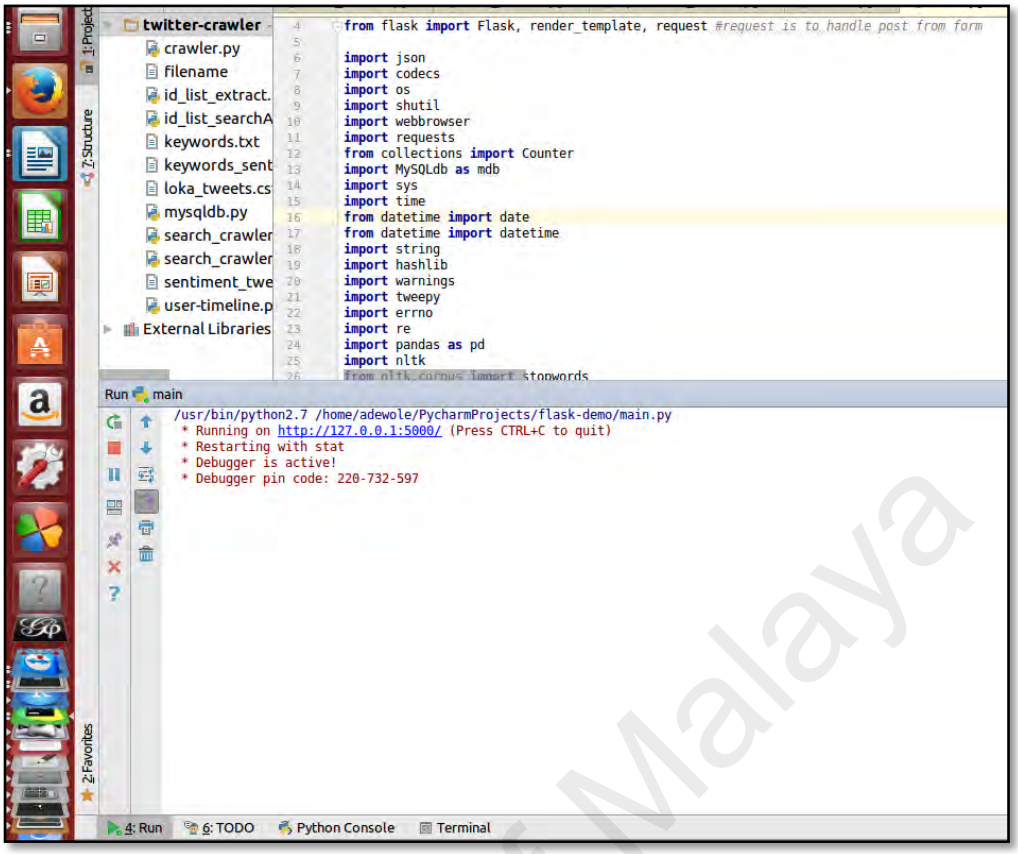
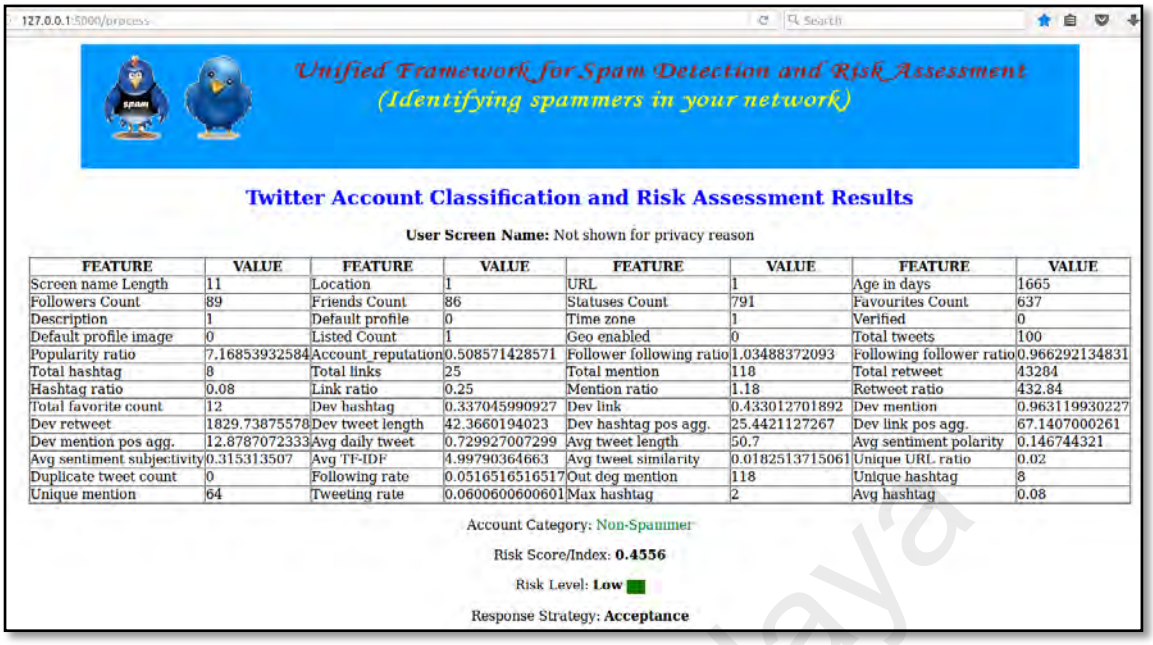


Figure 6.3: Staring the web application server

6.3.1 Spam Account Detection with risk

This section presents the running application using sample user screen names to collect live user data and tweets from Twitter network. The data collected from Twitter were utilized to categorize the account and present risk assessment results. This section is divided into five subsections as follows:

1) *Legitimate Account with Low Risk:* Figure 6.4 shows a sample Twitter account classified as legitimate account with low risk based on the outcome of the risk assessment module. The risk index of the account is estimated at 0.4556. The results of the classification are stored in MySQL database for subsequent preview by the administrator. The results are also displayed to the user with a green colored box beside the risk level.



Unified Framework for Spam Detection and Risk Assessment
(Identifying spammers in your network)

Twitter Account Classification and Risk Assessment Results

User Screen Name: Not shown for privacy reason

FEATURE	VALUE	FEATURE	VALUE	FEATURE	VALUE	FEATURE	VALUE
Screen name Length	11	Location	1	URL	1	Age in days	1665
Followers Count	89	Friends Count	86	Statuses Count	791	Favourites Count	637
Description	1	Default profile	0	Time zone	1	Verified	0
Default profile image	0	Listed Count	1	Geo enabled	0	Total tweets	100
Popularity ratio	7.16853932584	Account_reputation	0.508571428571	Follower following ratio	1.03488372093	Following follower ratio	0.966292134831
Total hashtag	8	Total links	25	Total mention	118	Total retweet	43284
Hashtag ratio	0.08	Link ratio	0.25	Mention ratio	1.18	Retweet ratio	432.84
Total favorite count	12	Dev hashtag	0.337045990927	Dev link	0.433012701892	Dev mention	0.963119930227
Dev retweet	1829.73875578	Dev tweet length	42.3660194023	Dev hashtag pos agg.	25.4421127267	Dev link pos agg.	67.1407000261
Dev mention pos agg.	12.8787072333	Avg daily tweet	0.729927007299	Avg tweet length	50.7	Avg sentiment polarity	0.146744321
Avg sentiment subjectivity	0.315313507	Avg TF-IDF	4.99790364663	Avg tweet similarity	0.0182513715061	Unique URL ratio	0.02
Duplicate tweet count	0	Following rate	0.0516516516517	Out deg mention	118	Unique hashtag	8
Unique mention	64	Tweeting rate	0.0600600600601	Max hashtag	2	Avg hashtag	0.08

Account Category: Non-Spammer

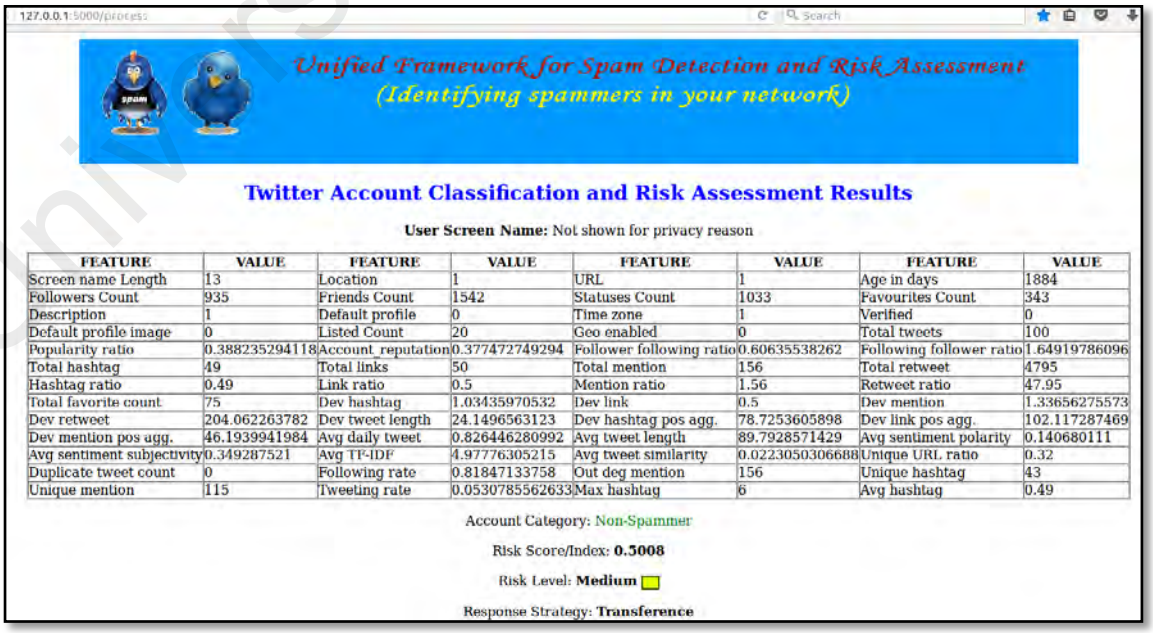
Risk Score/Index: **0.4556**

Risk Level: **Low**

Response Strategy: **Acceptance**

Figure 6.4: Non-Spam account with low risk

2) *Legitimate Account with Medium Risk:* Figure 6.5 shows a sample of legitimate account classified by the system with medium risk. The value of risk index is estimated at 0.5008.



Unified Framework for Spam Detection and Risk Assessment
(Identifying spammers in your network)

Twitter Account Classification and Risk Assessment Results

User Screen Name: Not shown for privacy reason

FEATURE	VALUE	FEATURE	VALUE	FEATURE	VALUE	FEATURE	VALUE
Screen name Length	13	Location	1	URL	1	Age in days	1884
Followers Count	935	Friends Count	1542	Statuses Count	1033	Favourites Count	343
Description	1	Default profile	0	Time zone	1	Verified	0
Default profile image	0	Listed Count	20	Geo enabled	0	Total tweets	100
Popularity ratio	0.388235294118	Account_reputation	0.377472749294	Follower following ratio	0.60635538262	Following follower ratio	1.64919786096
Total hashtag	49	Total links	50	Total mention	156	Total retweet	4795
Hashtag ratio	0.49	Link ratio	0.5	Mention ratio	1.56	Retweet ratio	47.95
Total favorite count	75	Dev hashtag	1.03435970532	Dev link	0.5	Dev mention	1.33656275573
Dev retweet	204.062263782	Dev tweet length	24.1496563123	Dev hashtag pos agg.	78.7253605898	Dev link pos agg.	102.117287469
Dev mention pos agg.	46.1939941984	Avg daily tweet	0.826446280992	Avg tweet length	89.7928571429	Avg sentiment polarity	0.140680111
Avg sentiment subjectivity	0.349287521	Avg TF-IDF	4.97776305215	Avg tweet similarity	0.0223050306688	Unique URL ratio	0.32
Duplicate tweet count	0	Following rate	0.81847133758	Out deg mention	156	Unique hashtag	43
Unique mention	115	Tweeting rate	0.0530785562633	Max hashtag	6	Avg hashtag	0.49

Account Category: Non-Spammer

Risk Score/Index: **0.5008**

Risk Level: **Medium**

Response Strategy: **Transference**

Figure 6.5: Non-Spam account with medium risk

3) *Spam Account with Medium Risk:* The result of an online assessment of a spam account classified as medium risk is shown in Figure 6.6. The value of risk index is estimated at 0.5888. This account has an abnormal value of follower to following ratio as compared to the legitimate account with medium risk in Figure 6.5. This spam account is specifically used to distribute pornographic contents. The account also purchased large number of followers to boost its reputation; however, the system is able to detect an imbalance value in the number of followers and friends of the account along with other spammer's behaviors.

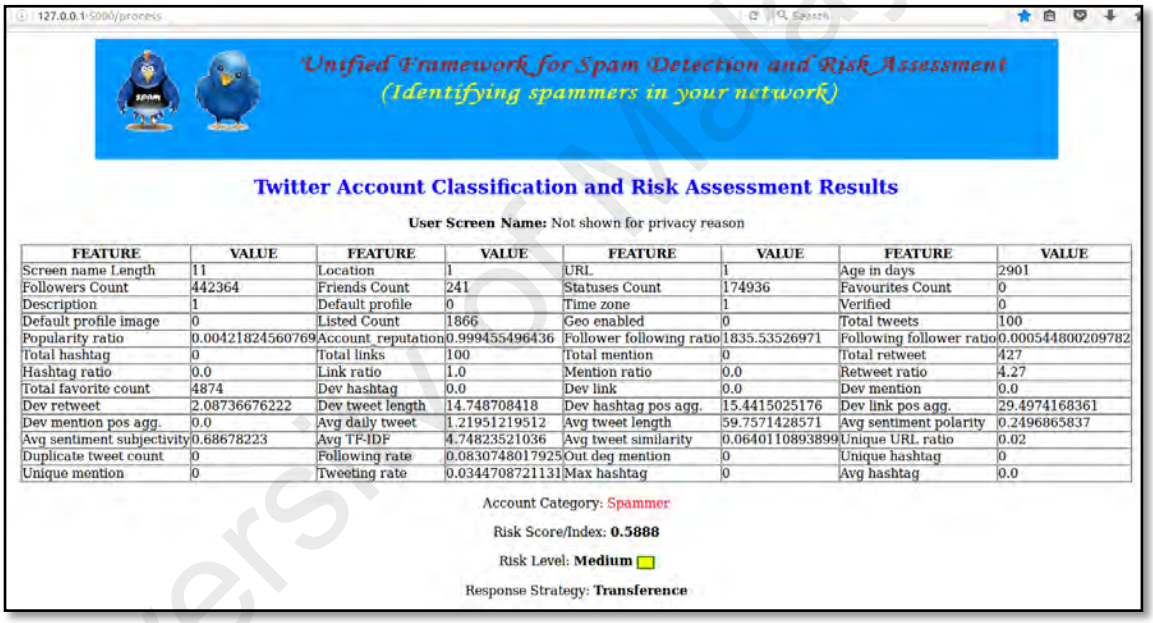


Figure 6.6: Spam account with medium risk

4) *Spam Account with High Risk:* Figure 6.7 presents the result of a spam account classified as high risk. The risk index is estimated at 0.795. Despite the fact that this account has not posted any link, during analysis of the account timeline, friends and followers, it was discovered that a majority of this account friends and followers are spammers.

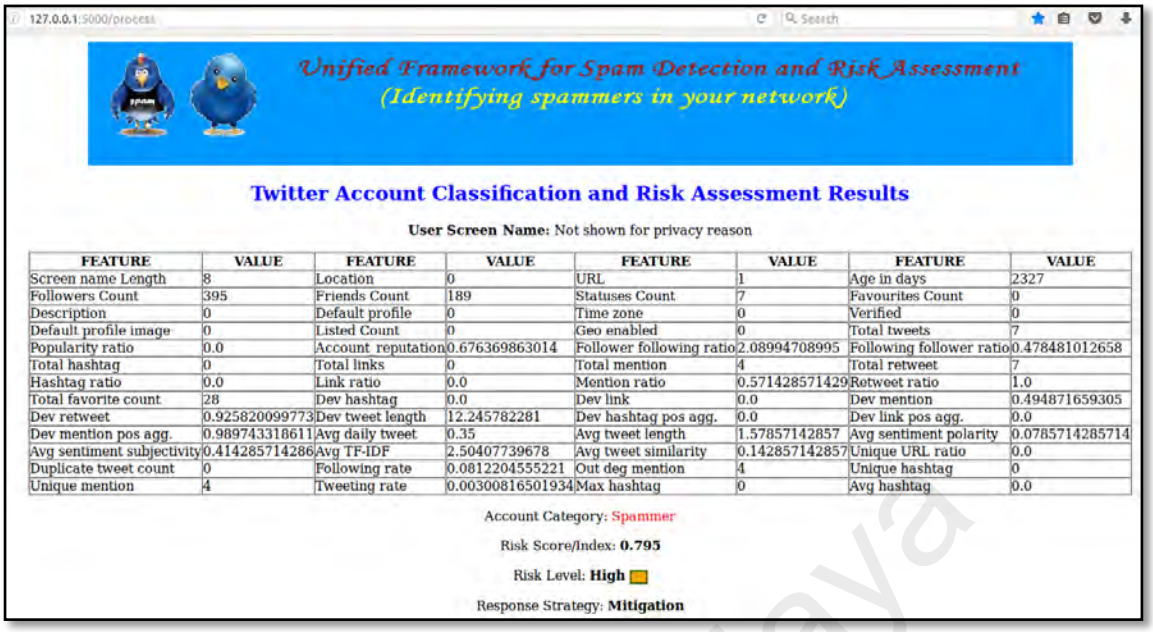


Figure 6.7: Spam account with high risk

5) *Spam Account with Very High Risk:* Figure 6.8 shows a sample spam account classified as very high risk with risk index estimated as 1. This account has a very high ratio of following to follower. This shows that the spam account is used to follow a large number of legitimate users with the hope that a majority of them will follow back.

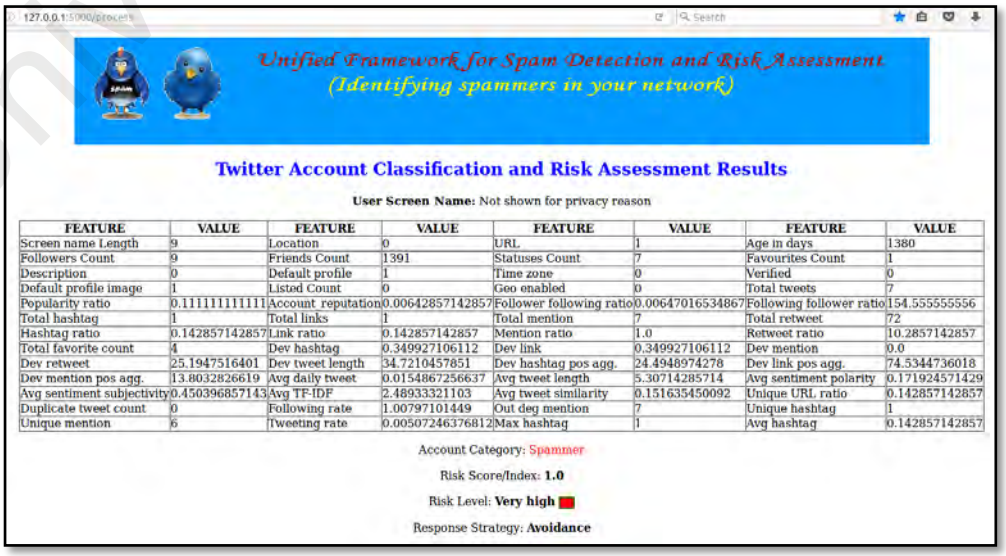


Figure 6.8: Spam account with very high risk

6.3.2 Spam Message Detection

This section presents the running application using sample tweets and mobile SMS data to evaluate the performance of the proposed SMDM for both Twitter and mobile spam message detection. This section is divided into four subsections as follows:

1) *Twitter Spam Message Detection:* This section presents the results of the spam message detection web module embedded in the prototype. In the first case shown in Figure 6.9, the proposed system is able to detect spam message on Twitter despite the fact that no single word was used to compose the tweet except links and question marks. This behavior shows one of the distinguished characteristics of Twitter spammer that has made it difficult for existing bag of words models to effectively detect spam messages on Twitter. Figure 6.10 also presents another sample tweets detected as spam message by the proposed system.

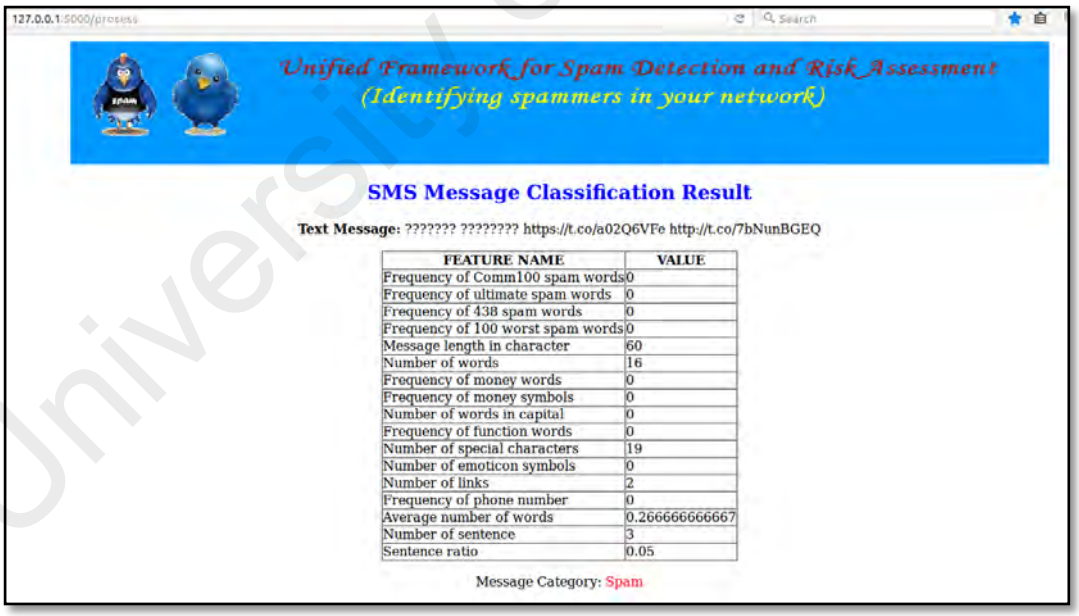


Figure 6.9: Twitter spam message with no word except links

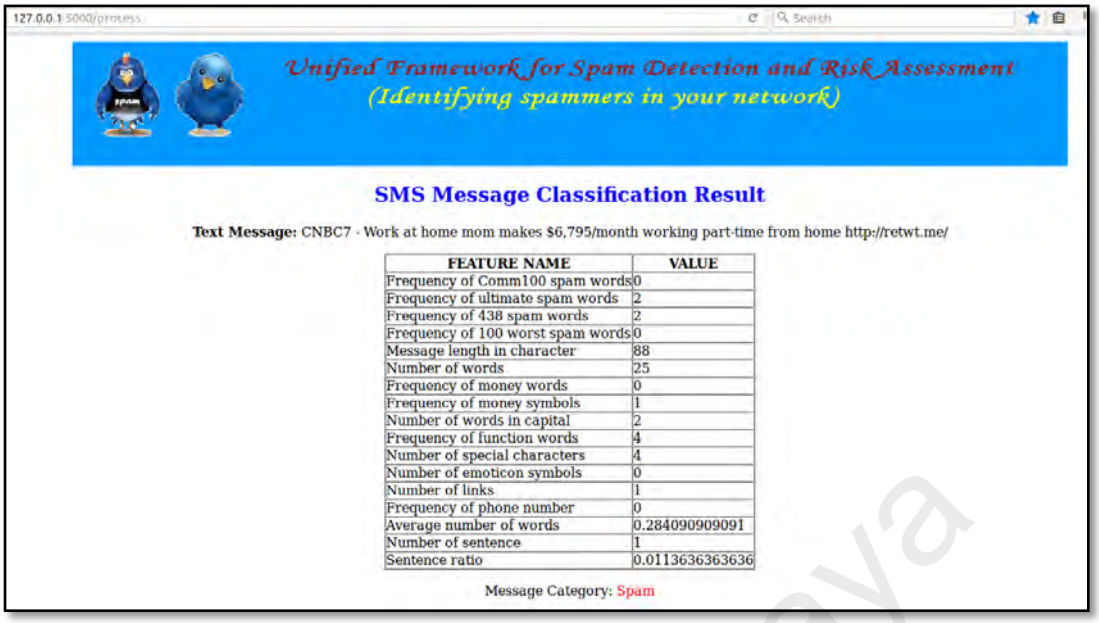


Figure 6.10: Twitter spam message with words

2) *Twitter Non-spam Message Detection:* Figure 6.11 shows how the proposed system was able to classify legitimate message on Twitter. Without using the bag of words approach, this study is able to separate spam from legitimate tweets as demonstrated in Figure 6.11.

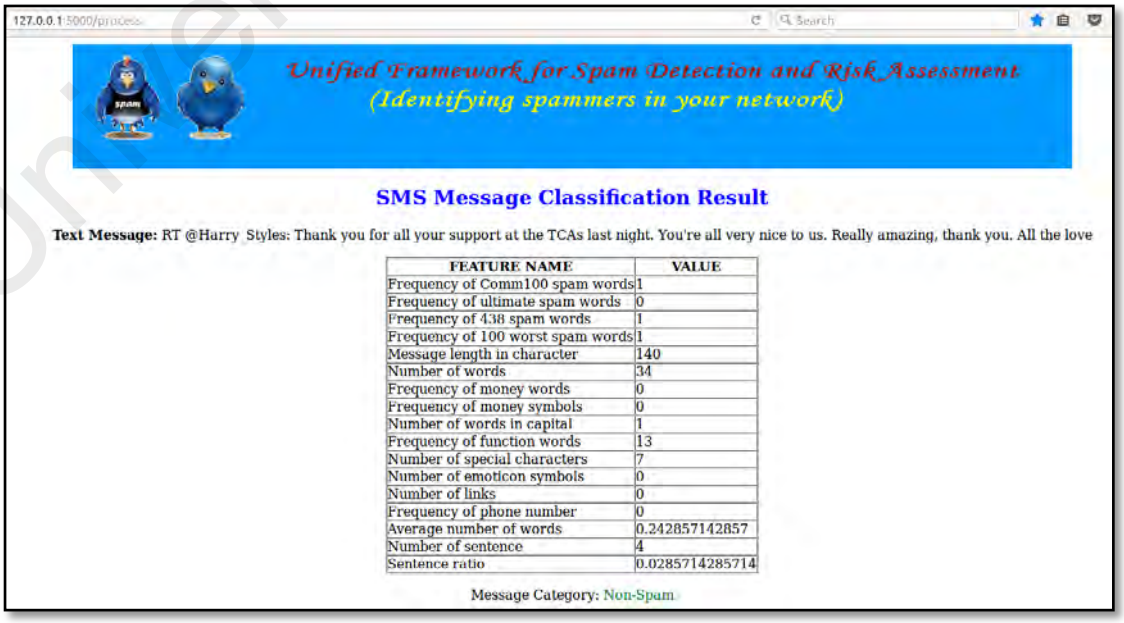


Figure 6.11: Twitter legitimate tweet classification

3) *Mobile Spam Message Detection:* In order to evaluate the capability of the proposed system in identifying SMS spam messages, Figure 6.12 shows a sample mobile spam message classified by the system. This figure confirms that the proposed SMDM web module embedded within the unified framework is promising for both mobile and Twitter spam message detection. Figure 6.13 shows another sample SMS message utilized to test the capability of the proposed system for mobile spam message detection.

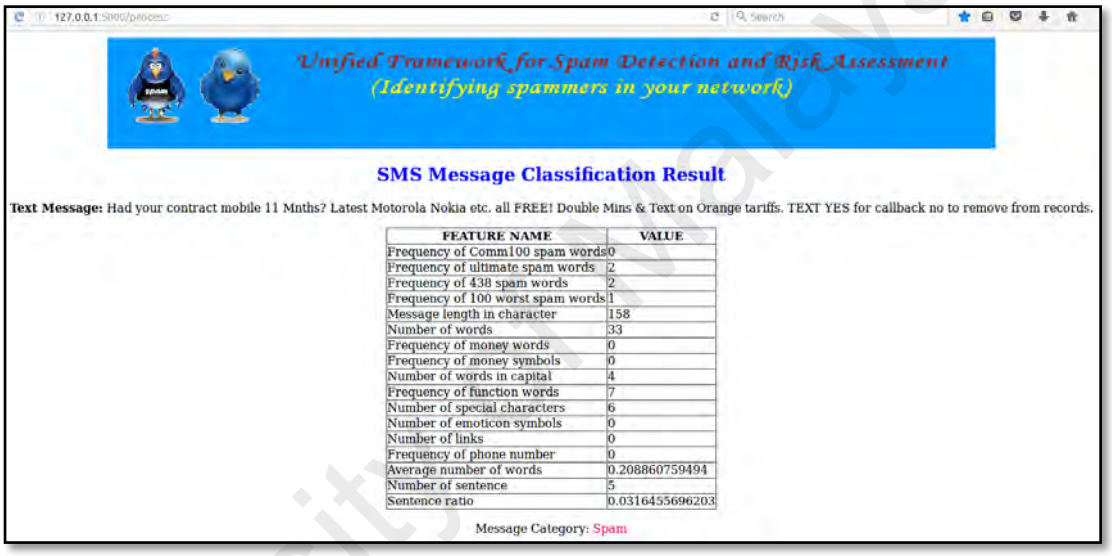


Figure 6.12: Mobile SMS spam message detection

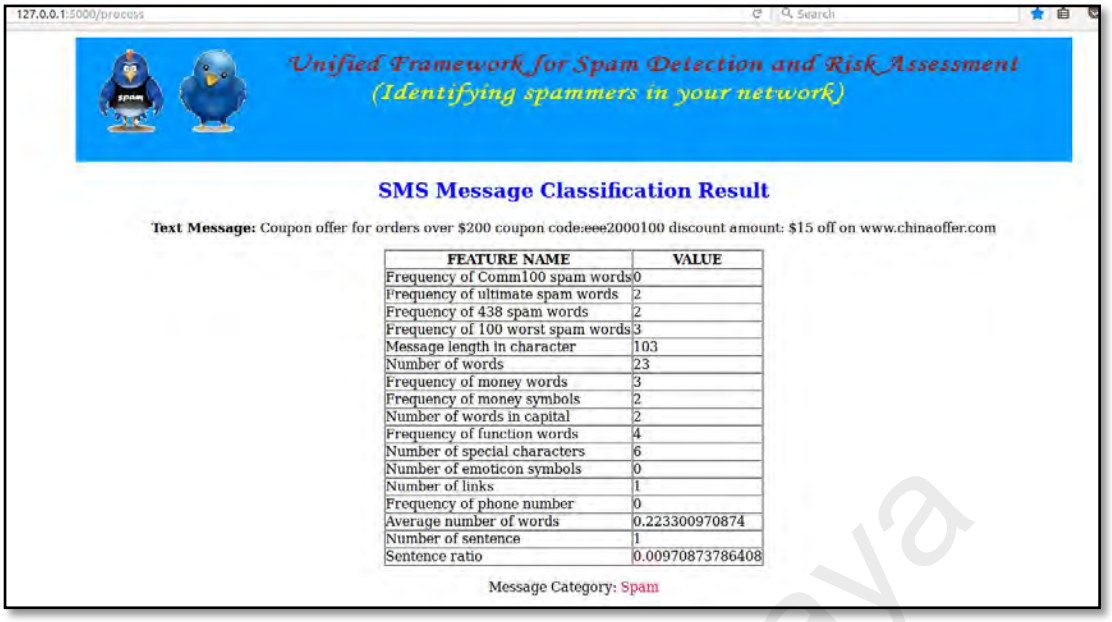


Figure 6.13: Mobile SMS spam message detection

4) *Mobile Non-spam Message Detection:* Figure 6.14 presents the result obtained from the proposed system when used to classify legitimate mobile SMS message. This figure shows that the proposed SMDM is able to separate SMS spam message from legitimate messages.

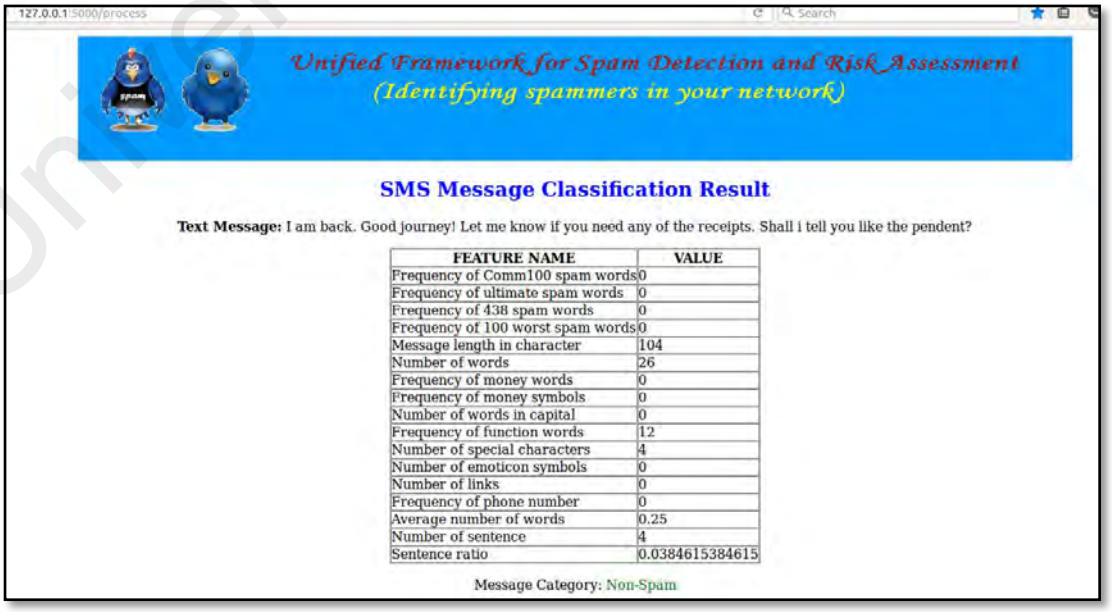


Figure 6.14: Mobile SMS legitimate message detection

6.4 Advantages and Limitation

The previous section has presented the capability of the proposed unified framework to detect spam message and spam account in SMCM as well as categorizing accounts on Twitter based on their risk level through the developed prototype, this section highlights several advantages of the proposed system as follows:

a) *Co-detection:* As opposed to existing frameworks, which focused on a specific problem category such as spam message detection or spam account detection, this study proposed a system that has the capability of co-detecting both spam message and spam accounts within a single framework. The three web modules interact to achieve the overall goal of the proposed system.

b) *Risk assessment:* Apart from detecting both spam message and spam account within a single framework, the proposed system in this study also has the capability to categorize Twitter accounts based on their risk level using live assessment mode. Using the proposed Fuzzy AHP structure, the developed prototype is able to applied four risk indicators, low, medium, high, and very high, to present the risk level associated with a Twitter account in addition to categorizing account as spam or legitimate. This analysis will help user on the social network to make an informed decision on a specific account under investigation.

c) *Analytics results:* The proposed web modules embedded within the system provide more insights on the performance of the unified framework for spam detection and risk assessment. The simplicity of the web modules make it easy for any user to analyze any Twitter account provided the screen name or user id is known. The proposed system is flexible and produces dynamic results based on the specific web module that is being assessed online. In other word, the web modules present a range of custom results and give a broader view of the performance of the proposed unified framework.

d) *Easy-to-use graphical interfaces:* User interacts with the system through easy-to-use graphical interfaces. The results of the prototype performance are presented using graphical interfaces. Additionally, the web modules provide easy-to-read outputs through graphical interfaces, which serve as an advantage to non-technical user of the system.

e) *Live/online assessment mode:* The web modules provide the capabilities of the proposed framework to examine live data from Twitter social network. Twitter accounts can be easily analyzed online without any delay. The results of the analyses can be viewed online, which enable user of the system to make a wise decision immediately based on the online results.

The proposed system also has a number of limitations discussed as follows:

1) *Update of global weights:* The global weights obtained from the Fuzzy AHP analysis used to develop the prototype are fixed values. In the future, a more sophisticated method of updating the global weights of the alternative indicators based on time management can be addressed to improve the performance of the proposed system.

2) *External Resources:* As the web modules for spam account and risk assessment depend on external server (i.e Twitter API server), they rely on the efficiency of the server to aggregate information for the system analysis. If the external server is down, the results of spam account detection and risk assessment web modules that utilize live data from Twitter cannot be obtained. The performances of these web modules rely on the network availability to communicate and exchange information due to the nature of the World Wide Web.

3) *Vulnerabilities:* Just like other web application, the proposed web modules are vulnerable to SQL injection, web server, and browser vulnerabilities as they can be used as a weak point to exploit the system.

6.5 Summary

In this chapter, the implementation phases of the proposed unified framework are discussed in details by providing some examples cases to highlight the performance of the proposed system for spam detection and risk assessment. These details include the composition of the web modules, system architecture, and Use cases in order to show how the various entities within the system interact.

The chapter explains how the proposed unified framework can be implemented for online assessment mode. The chapter also presents some of the advantages and limitations of the proposed system.

CHAPTER 7: CONCLUSION

The chapter summarizes the study by revisiting its aim and objectives as well as presenting the achievements of the research and highlighting its limitations. The chapter also discuses suggestions for future directions to enhance the proposed framework.

7.1 Research questions and research objectives

This study aims to develop a novel unified framework that co-detects spam message and spam account in SMCM as well as assess the risk level of microblogging social network accounts. Section 1.5 had detailed the five research objectives of this study. Therefore, this section aims to answer the following research questions: a) RQ1: What are the features to identify spammers on microblogging social network using hybrid method? b) RQ2: How can a unified framework be developed to detect both spam message and spam account in short message communication media? c) RQ3: How can a spam risk assessment model for microblogging social network be developed? d) RQ4: What is the effect on performance accuracy when the proposed unified framework is compare with existing approaches? e) RQ5: Can a prototype of the proposed framework achieve promising results when deploy in an online environment?

Objective 1: To investigate features for spam account detection in microblogging social network using hybrid method.

The first objective is provided to answer the RQ1 of this study. To accomplish this objective, a thorough literature review was first conducted by considering the most related articles published in online scholarly journals extracted from digital libraries. These libraries include the Association for Computing Machinery (ACM), the Institute of Electrical and Electronics Engineers (IEEE), Elsevier, and Springer. Recent literature extracted from journals, conference papers were considered and analytical issues were

investigated. After critical investigation of the current state-of-the-art approaches, this study proposed taxonomies of the features as well as methods for spam account detection. Consequently, it was revealed that using an approach based on hybrid analysis would offer better results for spam account detection in micrologging network. Based on this reason, this study proposed 69 features, which covered five categories including user profile, content, mention network, timing, and automation. A number of unique features were proposed to compliment the state-of-the-art. The features were evaluated using ten (10) classification algorithms and the results revealed that the proposed features provide a significant improvement in performance.

Objective 2: To design a unified framework for spam message and spam account detection in short message communication media.

This objective addresses the RQ2 earlier highlighted. Having investigated the limitations of the existing framework for spam message and spam account detection in SMCM, this study proposed a unified framework that incorporates the capability of co-detecting both spam message and spam account. Two models were first introduced in the proposed unified framework, which include SADM and SMDM. The SADM addresses spam account detection while the SMDM focuses on spam message detection. Due to the low performance of existing spam message detection framework that is based on the traditional bag-of-words approach, this study introduced 18 unique features to detect spam message in both mobile and Twitter microblog. In addition, the discriminating features for spam account detection were investigated using bio-inspired evolutionary computation.

Objective 3: To design a ranking scheme for spam risk assessment model in microblogging social network.

The purpose of this objective is to address the RQ3 raised in this study. To complete the proposed unified framework, a model for spam risk assessment of microblogging social network accounts that is based on Fuzzy AHP is incorporated at the lower layer of the proposed framework. Ramik Fuzzy AHP was adopted to develop the SRAM model. The model incorporates different components to achieve the goal of assessing the spam risk level of microblogging social network accounts.

Objective 4: To evaluate the performance of the proposed unified framework by validating it using evaluation studies at different stages.

This objective provides answer to RQ4 established in this study. Several stages of evaluation were conducted with each stage producing promising results that demonstrate the capability of the proposed framework using different evaluation metrics. In addition, the performances of SADM and SMDM were compared with related studies and the outcome revealed the superiority of the proposed framework for detecting spam message and spam account in SMCM. Consequently, the overall performance of the proposed SRAM model was verified by considering a significant number of accounts for manual verification. The result also shows the applicability of the SRAM model to categorize account based on risk level, such as low, medium, high, and very high risk.

Objective 5: To implement a novel prototype of the proposed framework for practical evaluation in an online environment.

To answer RQ5, a novel prototype of the proposed framework is implemented using Flask Python web framework and MySQL database management system. In

addition, the composition of the prototype was discussed using Use Case diagram of UML modeling language. The prototype provides three web modules to address the overall goal of this study. Each module is evaluated by considering real life cases. The evaluation result shows the applicability of the prototype to detect spam message and spam account as well as assess account risk level.

7.2 Contributions of the study

This study proposed a novel framework for detecting both spam message and spam account in short message communication media like mobile and Twitter microblog. The framework is also capable of assessing the risk level of Twitter microblogging social network account by proposing methodology to rate, rank, and categorize Twitter account based on their risk level. The study identified the decision factors as well as alternative indicators to aid the estimation of the risk level. The proposed unified framework comprises of three models, which were studied in details, and their performances evaluated.

At each evaluation stage of the proposed framework, different sub-objectives were established in order to achieve the main research objectives. In general, the overall aim of this study is to establish a novel approach to develop a system, which has the capability for detecting spam and assessing risk. To further show the performance of the proposed unified framework, this study presents a prototype implementation of the framework using three main web modules. The details of the achievement are as follows:

1) *Taxonomies:* In Chapter 2, this study discussed two taxonomies based on features and methods to detect malicious accounts as well as their behaviors in social networks. The taxonomies provide review of existing related studies. These

taxonomies will assist future researchers to gain more insights into the domain of malicious behavior detection in social networks.

2) *Spam detection and risk assessment issues:* In Chapter 3, several issues and challenges that hindered effective performance of the existing models were identified. The aim was to establish a unified framework for spam detection and risk assessment that can produce improved results. By presenting issues with existing risk assessment studies as well as their strengths, this study proposed a different approach in the risk assessment stage to address human vagueness and uncertainty when judging criteria and alternative indicators. Bio-inspired evolutionary approach was employed to identify important indicators to model the risk assessment.
3) *Establishment of new set of features:* To achieve the goal of this study, eighteen (18) unique set of features were proposed to develop spam message detection model. Additional thirty three (33) features were introduced in this study to complement existing features for detecting spam account on Twitter microblog. These features enabled the proposed unified framework to achieve better performance during the evaluation stages. The study ranked the five categories of features utilized for spam account detection during evaluations. It was established through experimental results that the first feature category with best performance is user profile. This is followed by automation, content, timing, and network features respectively.
4) *Twitter and mobile spam message detection:* This study has established a model called SMDM, which categorized both Twitter, and mobile messages as spam or legitimate by proposing a novel set of features for spam message detection that are different from traditional bag of words approach. To ascertain the feasibility of the proposed SMDM for spam message detection, several experiments were conducted and their results show positive outcomes (see Chapter 5). Based on the findings from the experiments, it was established that Random Forest classification algorithm

was the best for this classification task. The performance of the Random Forest algorithm was promising based upon the various evaluation metrics utilized in this study. This study also established the superiority of the proposed model by comparing its performance with related models. As opposed to the traditional bag of words approach, which rely on word analysis, the proposed SMDM embedded within the unified framework identified spam pattern that contains only links and special characters even though the message has no single word (see Chapter 6). Additionally, the proposed model was tailored towards addressing the distinguished characteristics of messages posted on SMCM, some of which contain abbreviations, special characters, emoticons, and idioms.

5) *Twitter spam account detection:* This study proposed SADM model, which categorized Twitter account as spam or legitimate. By establishing a novel set of features for spam account detection, this model enables the proposed framework to have the capability of identifying spam account on Twitter microblog (see Chapters 4 and 5). To establish the feasibility of the proposed SADM, several experiments were conducted and their results show positive outcomes (see Chapter 5). Based on the findings from the experiments, it was established that ensemble-based classification algorithms, such as Random Forest and Decorate, are suitable for the proposed SADM. The performances of these algorithms were promising based upon the various evaluation metrics utilized in this study. This study established the superiority of the proposed SADM by comparing its performance with related models using baseline method (see Chapter 5).

6) *Spam risk assessment model:* In addition to detecting spam message and spam account, this study established a model for assessing the risk level of Twitter accounts. Four risk linguistic variables, low, medium, high, and very high, were used to quantify account risk level. The model enables the proposed unified

framework to rate, rank, and categorizes Twitter account based on their risk level. To establish the feasibility of the proposed spam risk assessment model (SRAM), several experiments were conducted and their results show positive outcomes (see Chapter 5). The performance of the model was further established through a web module embedded within the proposed system for live demonstration (see Chapter 6).

7) *Novel unified framework for spam detection and risk assessment:* By integrating the three models established in this thesis, this study proposed a novel unified framework to address the problem of spam detection and risk assessment in SMCM. The performance of ten machine learning algorithms have been studied in order to select the best algorithm for the proposed spam detection models. With the aid of the Fuzzy AHP and the risk assessment structure established in this study, a new risk estimation model was proposed and integrated within the unified framework.
8) *Evaluation stages:* The three models embedded within the unified framework were critically evaluated by establishing stages to examine the performance of the proposed framework. The results of these evaluations were used to justify the applicability of the proposed framework in real live environment.
9) *Implementation of the proposed unified framework:* To widen the investigation based on the feasibility of the proposed unified framework and show its practical application within the context of online assessment mode, a prototype was developed (see Chapter 6). As an extension to the evaluation study, the prototype implementation phase involved the development of a web-based system, which focuses on the web modules to depict the functionalities of the proposed framework. In order to illustrate the implementation stage, UML modeling language was used to show the various entities within the proposed system. In addition, the results of the three web modules embedded within the system were presented using some

snapshots of different cases. The system outputs demonstrated that this study has achieved its aim and objectives as stated in Chapter 1.

7.3 Limitations of the study

The previous section has discussed the achievement of this study, which is aimed at developing a system for spam detection and risk assessment, specifically for SMCM. However, during the cause of this study, a number of limitations and challenges were encountered, which are discussed as follows for future reference:

1) *Performance evaluation:* Although the performance of the proposed unified framework has been demonstrated through the three models integrated within the framework, the performance of these models can still be further improved to achieve reduced false alarm rates.
2) *Feature extraction time:* The features used for developing the SADM require a considerable amount of time to be able to extract for all the accounts in the Twitter dataset due to the millions of tweets involved. In the future experiment, a more sophisticated feature extraction module can be developed to improve the complexity of the feature extraction stage.
3) *Label samples:* Although previous researchers have provided the labeled samples in the public datasets used for mobile spam message detection, this study adopted Twitter suspension algorithm method for identifying labeled samples for the private dataset collected from Twitter in order to have a consistent and efficient approach to label the spam accounts. However, this approach requires that the dataset be left for a longer period before more labeled samples can be identified. This issue can be addressed in the future study.
4) *Message length:* Despite the large number of labeled samples used to evaluate the proposed spam message detection model on Twitter spam message corpus, the

domain specific words commonly used on Twitter still have little effect on the classification performance as compared with mobile SMS spam corpora. The approach used in this study, which combined various spam words utilized by spammers on Twitter to extract useful feature for the proposed model improves the performance of the classification algorithms. In the future experiment, a sophisticated method can be developed to expand the domain specific words using natural language processing technique before the actual features proposed in this study are extracted.

5) *Noise data:* During the feature extraction stage from the Twitter corpus, it was discovered that a lot of messages on Twitter are noisy due to the nature of the social network. In this study, the length of the messages used in the evaluation results was set to 100 characters and more. This is to ensure that some of the noisy data are filtered out from the experimental analysis. This challenge can be addressed in the future experiment using a more robust method.

6) *Expert assumption:* In this study, assumptions were made regarding the comparison of one criterion or alternative indicator to another based on the subjective opinion of the expert. These assumptions involve the strategy used to provide the weighting scales for each judgment matrix developed during the risk assessment modeling. While AHP method provides consistency measure to accept or reject the assumptions used to construct the comparison matrices, the CR ratios obtained can be reduced further to provide judgments that are more consistent.

7) *Method comparison:* As discussed in Chapter 4, this study applied Ramik fuzzy AHP approach; therefore, the results presented for the risk assessment evaluation study are based on this methodology. However, it is important to consider a situation where the results of different fuzzy AHP methods such as extent analysis and Ramik approach are compared in order to provide more findings on the

performance of the proposed SRAM based on variation in the fuzzy AHP methods applied to compute the global weights.

8) *Rating threshold:* The rating threshold provides opportunity to establish the risk level membership function used in this study. This rating threshold was based on the distribution of the risk scores computed for both spam and legitimate account categories. This means that the approach used to obtain the rating threshold is based on local computation method in relation to the distribution of spam and legitimate accounts in the ground truth dataset.

7.4 Research implications

The overall goal of this study is to provide a robust spam detection and risk assessment framework that incorporates three unique models namely SADM, SMDM and SRAM. This aim was achieved by conducting experiments to show the applicability of each model integrated within the proposed framework. At each stage of the evaluation study, the performance of the individual model was assessed using different standard evaluation metrics. The findings of this research demonstrate that hybrid feature learning is effective to build a robust spam account detection model that prevents evasion of spam filter. The study also proposed a novel method for feature representation that benefits both mobile and microblogging spam message detection. In addition, a model for risk assessment of microblogging social network accounts was proposed.

As an implication for research practice, the results gathered in this study further raised new research questions. For instance, what are the additional features based on hybrid analysis that could improve the performance of the proposed SADM? What is the implication of such findings on the overall performance of the proposed framework? Similarly, extensive analysis may be conducted to show the cross-platform applicability

of the proposed SMDM when considering different SMCM. Since social network user receives friendship request from unknown accounts almost on a daily basis, the impact of this risk taking behavior may be huge on the victim. It is possible to reduce this risk impact with the aid of a robust risk management system that uncovers hidden behavioral patterns of the account under investigation. The findings of this research suggest different response strategies, which play a significant role in reducing the risk impact.

7.5 Suggestions for future work

This section highlights a number of suggestions for future research outside the scope of this study. These are discussed as follows:

1) *Cross-platform verification:* In order to identify spam account on microblogging social network, this study analyzed only the data collected from Twitter microblog due to its openness and robust API for data collection. Future research can investigate the possibility of developing a spam detection and risk assessment framework that would leverage different microblogging networks to provide more insights on spammer's behaviors across many networks.
2) *Identification of more salient features:* As stated in the previous section, the performance accuracy of the spam account detection model still need to be improved to provide a more efficient framework. This can be achieved in the future study by investigating additional salient features to counter evasion strategies used by spammers. This in turn will improve the performance of the spam risk assessment model proposed in this study.
3) *Improving tweet quality:* One of the challenges faced in this study is the need to deal with the quality of tweets posted on Twitter due to the characteristics of Twitter microblog. The future work should investigate how to develop a robust approach based on natural language processing techniques to address this problem. This

would have a positive impact on the performance of the proposed spam message detection model embedded within the unified framework.

4) *Data streaming:* One of the challenges hindering the development of a dynamic risk assessment system for social networks is the need to deal with the speed at which social network data is evolving. To address the drifting nature of messages shared on social networks, there is a need to investigate an approach that is different from the batch learning method used in this study. The future work should explore the possibility of adapting streaming algorithms to develop potential spam detection and risk assessment system.
5) *Distributed framework:* The continuous increase in the number of messages posted on social networks demands for a more robust and scalable method of data analysis. For instance, technology such as Big data that supports distributed processing of large amount of data can be applied in the future research to uncover large hidden patterns and relationships among social spammers.
6) *Semi-supervised learning:* As discussed in the limitation section, one of the problems faced in this study is identification of more labeled samples to train and validate the selected classification algorithms. Future study should investigate the applicability of semi-supervised learning method, which enables small number of labeled samples to be used with large unlabeled samples to develop robust and more efficient machine leaning models.

REFERENCES

Ab Razak, M. F., Anuar, N. B., Salleh, R., & Firdaus, A. (2016). The rise of "malware": Bibliometric analysis of malware study. *Journal of Network and Computer Applications, 75*, 58-76.

Abeer, A., Maha, H., Nada, A., & Hemalatha, M. (2016). Security issues in social networking sites. *Int. J. Appl. Eng. Res., 11*(12), 7672-7675.

Abu-Nimeh, S., Chen, T. M., & Alzubi, O. (2011). Malicious and Spam Posts in Online Social Networks. *Computer, 44*(9), 23-28. doi: 10.1109/mc.2011.222

Adamic, L., & Adar, E. (2005). How to search a social network. *Social Networks, 27*(3), 187-203.

Adewole, K. S., Anuar, N. B., Kamsin, A., Varathan, K. D., & Razak, S. A. (2016). Malicious accounts: Dark of the social networks. *Journal of Network and Computer Applications*. doi: http://dx.doi.org/10.1016/j.jnca.2016.11.030

Aggarwal, A., Rajadesingan, A., & Kumaraguru, P. (2012). *PhishAri: Automatic realtime phishing detection on twitter.* In: eCrime Researchers Summit (eCrime), 2012.

Ahmad, F., & Sarkar, A. (2016). Analysis of Dynamic Web Services: Towards Efficient Discovery in Cloud. *Malaysian Journal of Computer Science, 29*(3).

Ahmed, F., & Abulaish, M. (2012). *An MCL-based approach for spam profile detection in online social networks.* In: 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom).

Ahmed, F., & Abulaish, M. (2013). A generic statistical approach for spam detection in Online Social Networks. *Computer Communications, 36*, 1120-1129. doi: 10.1016/j.comcom.2013.04.004

Al-garadi, M. A., Varathan, K. D., & Ravana, S. D. (2016). Cybercrime detection in online communications: The experimental case of cyberbullying detection in the Twitter network. *Computers in Human Behavior, 63*, 433-443.

Al Hasan, M., Chaoji, V., Salem, S., & Zaki, M. (2006). *Link prediction using supervised learning.* In: SDM'06: Workshop on Link Analysis, Counter-terrorism and Security.

Almaatouq, A., Shmueli, E., Nouh, M., Alabdulkareem, A., Singh, V. K., Alsaleh, M., . . . Alfaris, A. (2016). If it looks like a spammer and behaves like a spammer, it must be a spammer: analysis and detection of microblogging spam accounts. *International Journal of Information Security*, 1-17.

Almeida, T., Hidalgo, J. M. G., & Silva, T. P. (2013). Towards sms spam filtering: Results under a new dataset. *International Journal of Information Security Science, 2*(1), 1-18.

Alsaleh, M., Alarifi, A., Al-Salman, A. M., Alfayez, M., & Almuhaysin, A. (2014). *TSD: Detecting Sybil Accounts in Twitter.* In: 2014 13th IEEE International Conference on Machine Learning and Applications (ICMLA).

Anuar, N. B. (2012). *Incident prioritisation for intrusion response systems.* (Doctor of Philosophy PhD), Plymouth University.

Atluri, A. C., & Tran, V. (2017). Botnets Threat Analysis and Detection *Information Security Practices* (pp. 7-28): Springer.

Baker, S., Ponniah, D., & Smith, S. (1999). Risk response techniques employed currently for major projects. *Construction Management & Economics, 17*(2), 205-213.

Balakrishnan, V., Humaidi, N., & Lloyd-Yemoh, E. (2016). Improving document relevancy using integrated language modeling techniques. *Malaysian Journal of Computer Science, 29*(1).

Başaran, B. (2012). A critique on the consistency ratios of some selected articles regarding fuzzy AHP and sustainability.

Bates, C. L., & Illg, J. J. (2011). Spam risk assessment: Google Patents.

Ben-Gal, I., Ruggeri, F., Faltin, F., & Kenett, R. (2007). Bayesian networks, encyclopedia of statistics in quality and reliability: John Wiley and Sons.

Benevenuto, F., Rodrigues, T., Almeida, V., Almeida, J., & Gonçalves, M. (2009). *Detecting spammers and content promoters in online video social networks.* In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval.

Bhat, S. Y., & Abulaish, M. (2013). *Community-based features for identifying spammers in online social networks.* In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining.

Bhat, S. Y., Abulaish, M., & Mirza, A. A. (2014). Spammer classification using ensemble methods over structural social network features. *2014 Ieee/Wic/Acm International Joint Conferences on Web Intelligence (Wi) and Intelligent Agent Technologies (Iat), Vol 2*, 454-458. doi: 10.1109/wi-iat.2014.133

Bhattacharya, M., Islam, R., & Abawajy, J. (2016). Evolutionary optimization: a big data perspective. *Journal of Network and Computer Applications, 59*, 416-426.

Bilge, L., Strufe, T., Balzarotti, D., & Kirda, E. (2009). *All your contacts are belong to us: automated identity theft attacks on social networks.* In: Proceedings of the 18th international conference on World wide web. ACM.

Boyd, D. M., & Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication, 13*(1), 210-230.

Bozan, Y. S., Çoban, Ö., Özyer, G. T., & Özyer, B. (2015). *SMS spam filtering based on text classification and expert system.* In: 2015 23nd Signal Processing and Communications Applications Conference (SIU).

Byun, D.-H. (2001). The AHP approach for selecting an automobile purchase model. *Information & Management, 38*(5), 289-297.

Cao, C., & Caverlee, J. (2015). Detecting Spam URLs in Social Media via Behavioral Analysis. *Advances in Information Retrieval 37th European Conference on IR Research, ECIR 2015, Vienna, Austria, March 29 - April 2, 2015. Proceedings*, 703.

Cao, J., Li, Q., Ji, Y., He, Y., & Guo, D. (2016). Detection of Forwarding-Based Malicious URLs in Online Social Networks. *International Journal of Parallel Programming, 44*(1), 163-180.

Cao, Q., Sirivianos, M., Yang, X., & Pregueiro, T. (2012). *Aiding the detection of fake accounts in large scale social online services.* In: Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation.

Carpinter, J., & Hunt, R. (2006). Tightening the net: A review of current and next generation spam filtering tools. *Computers & Security, 25*(8), 566-578.

Chan, P. P. K., Yang, C., Yeung, D. S., & Ng, W. W. Y. (2014). Spam filtering for short messages in adversarial environment. *Neurocomputing, 155*, 167-176. doi: 10.1016/j.neucom.2014.12.034

Chang, D.-Y. (1996). Applications of the extent analysis method on fuzzy AHP. *European Journal of Operational Research, 95*(3), 649-655.

Chen, C.-M., Guan, D., & Su, Q.-K. (2014). Feature set identification for detecting suspicious URLs using Bayesian classification in social networks. *Information Sciences, 289*, 133-147.

Chen, L., Yan, Z., Zhang, W., & Kantola, R. (2015). TruSMS: a trustworthy SMS spam control system based on trust management. *Future Generation Computer Systems, 49*, 77-93.

Chen, S.-J., & Hwang, C.-L. (2012). *Fuzzy multiple attribute decision making: methods and applications* (Vol. 375): Springer Science & Business Media.

Chu, Z., Gianvecchio, S., Wang, H., & Jajodia, S. (2012a). Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *IEEE Transactions on Dependable and Secure Computing, 9*(6), 811-824. doi:10.1109/TDSC.2012.75

Chu, Z., Wang, H., & Widjaja, I. (2012b). *Detecting social spam campaigns on Twitter* (Vol. 7341 LNCS).

Clauset, A., Shalizi, C. R., & Newman, M. E. (2009). Power-law distributions in empirical data. *SIAM review, 51*(4), 661-703.

Cohen, W. W. (1995). *Fast effective rule induction.* In: Proceedings of the twelfth international conference on machine learning.

Conti, M., Poovendran, R., & Secchiero, M. (2012). *FakeBook: Detecting Fake Profiles in On-Line Social Networks.* In: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM).

Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., & Tesconi, M. (2015). Fame for sale: Efficient detection of fake Twitter followers. *Decision Support Systems, 80*, 56-71. doi: 10.1016/j.dss.2015.09.003

Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., & Tesconi, M. (2017). The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. *arXiv preprint arXiv:1701.03017*.

Cui, X. (2016). Identifying Suspended Accounts In Twitter. http://scholar.uwindsor.ca/etd/5725/

Danezis, G., & Mittal, P. (2009). *SybilInfer: Detecting Sybil Nodes using Social Networks.* In: NDSS.

Davis, C. A., Varol, O., Ferrara, E., Flammini, A., & Menczer, F. (2016). *BotOrNot: A system to evaluate social bots.* In: Proceedings of the 25th International Conference Companion on World Wide Web.

Delany, S. J., Buckley, M., & Greene, D. (2012). SMS spam filtering: methods and data. *Expert Systems with Applications, 39*(10), 9899-9908.

Devineni, P., Koutra, D., Faloutsos, M., & Faloutsos, C. (2015, 25-28 Aug. 2015). *If walls could talk: Patterns and anomalies in Facebook wallposts.* In: 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM).

DMR. (2015). By The Numbers: 150+ Amazing Twitter Statistics. from http://expandedramblings.com/index.php/march-2013-by-the-numbers-a-few-amazing-twitter-stats/10/

Echeverría, J., & Zhou, S. (2017). TheStar Wars' botnet with> 350k Twitter bots. *arXiv preprint arXiv:1701.02405*.

Egele, M., Stringhini, G., Kruegel, C., & Vigna, G. (2015). Towards Detecting Compromised Accounts on Social Networks. *IEEE Transaction on Dependable and Secure Computing*. doi: 10.1109/TDSC.2015.2479616

El-Alfy, E.-S. M., & AlHasan, A. A. (2016). Spam filtering framework for multimodal mobile communication based on dendritic cell algorithm. *Future Generation Computer Systems*.

Elyashar, A., Fire, M., Kagan, D., & Elovici, Y. (2013). *Homing socialbots: intrusion on a specific organization's employee using Socialbots.* In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining.

ENISA. (2014). *An evaluation framework for national cyber security strategies*. Heraklion: European Union Agency for Network and Information Security.

Ezpeleta, E., Zurutuza, U., & Gomez Hidalgo, J. M. (2015). An Analysis of the Effectiveness of Personalized Spam Using Online Social Network Public Information. In A. Herrero, B. Baruque, J. Sedano, H. Quintian & E. Corchado (Eds.), *International Joint Conference: Cisis'15 and Iceute'15* (Vol. 369, pp. 497-506).

Ezpeleta, E., Zurutuza, U., & Hidalgo, J. M. G. (2016). *Short Messages Spam Filtering Using Sentiment Analysis.* In: International Conference on Text, Speech, and Dialogue.

Ferrara, E., & Fiumara, G. (2012). *Mining and Analysis of Online Social Networks*. (PhD doctoral thesis), University of Messina, Italy.

Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (2014). The rise of social bots. *arXiv preprint arXiv:1407.5225*.

Fire, M., Kagan, D., Elyashar, A., & Elovici, Y. (2014). Friend or foe? Fake profile identification in online social networks. *Social Network Analysis and Mining, 4*(1), 1-23.

Forman, E. H., & Gass, S. I. (2001). The analytic hierarchy process—an exposition. *Operations research, 49*(4), 469-486.

Galán-García, P., de la Puerta, J. G., Gómez, C. L., Santos, I., & Bringas, P. G. (2014). *Supervised Machine Learning for the Detection of Troll Profiles in Twitter Social Network: Application to a Real Case of Cyberbullying.* In: International Joint Conference SOCO'13-CISIS'13-ICEUTE'13.

Gani, K., Hacid, H., & Skraba, R. (2012). *Towards multiple identity detection in social networks.* In: Proceedings of the 21st international conference companion on World Wide Web. ACM.

Gao, H., Hu, J., Wilson, C., Li, Z., Chen, Y., & Zhao, B. Y. (2010). *Detecting and characterizing social spam campaigns.* In: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement.

Gao, S., Ma, X., Wang, L., & Yu, Y. (2016). *Spammer detection based on comprehensive features in Sina Microblog.* In: 2016 13th International Conference on Service Systems and Service Management (ICSSSM).

Ghosh, S., Viswanath, B., Kooti, F., Sharma, N. K., Korlam, G., Benevenuto, F., . . . Gummadi, K. P. (2012). Understanding and combating link farming in the twitter social network. *Proceedings of the 21st International Conference World Wide Web*, 61.

Gong, N. Z., Frank, M., & Mittal, P. (2014). SybilBelief: A semi-supervised learning approach for structure-based Sybil detection. *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, 9*(6), 976-987.

Google. (2015). Google safe browsing API. Retrieved 25th November, 2015, from http://code.google.com/apis/safebrowsing/

Grier, C., Thomas, K., Paxson, V., & Zhang, M. (2010). @spam: the underground on 140 characters or less. *Proceedings of the 17th ACM conference on Computer and communications security*, 27-37.

Grzymala-Busse, J. W. (2010). Rule induction. *Data mining and knowledge discovery handbook*, 249-265.

Gupta, A., & Kaushal, R. (2015). Improving Spam Detection in Online Social Networks.

Guzella, T. S., & Caminhas, W. M. (2009). A review of machine learning approaches to spam filtering. *Expert Systems with Applications, 36*(7), 10206-10222.

Haimes, Y. Y. (2015). *Risk modeling, assessment, and management*: John Wiley & Sons.

Harsule, S. R., & Nighot, M. K. (2016). N-Gram Classifier System to Filter Spam Messages from OSN User Wall *Innovations in Computer Science and Engineering* (pp. 21-28): Springer.

Heidemann, J., Klier, M., & Probst, F. (2012). Online social networks: A survey of a global phenomenon. *Computer Networks, 56*(18), 3866-3878. doi: http://dx.doi.org/10.1016/j.comnet.2012.08.009

Hillson, D. (1999). *Developing effective risk responses.* In: Proceedings of the 30th Annual Project Management Institute Seminars & Symposium.

Hu, X., Tang, J., Zhang, Y., & Liu, H. (2013). *Social spammer detection in microblogging.* In: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence.

Igawa, R. A., Barbon Jr, S., Paulo, K. C. S., Kido, G. S., Guido, R. C., Júnior, M. L. P., & da Silva, I. N. (2016). Account classification in online social networks with LBCA and wavelets. *Information Sciences, 332*, 72-83.

ISO. (2009). 73: 2009. *Risk management—Vocabulary*.

Jang, J.-S. R., Sun, C.-T., & Mizutani, E. (1997). Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence.

Javelin Strategy & Research. (2016). 2016 Identity fraud: Fraud hits an inlection point. Retrieved 3rd March 2016, from https://www.javelinstrategy.com/coverage-area/2016-identity-fraud-fraud-hits-inflection-point

Jiang, J., Shan, Z., Sha, W., Wang, X., & Dai, Y. (2012). Detecting and Validating Sybil Groups in the Wild2012 32nd International Conference on Distributed Computing Systems Workshops (pp. 127): IEEE. doi: 10.1109/ICDCSW.2012.9

Jin, L., Takabi, H., & Joshi, J. B. (2011). *Towards active detection of identity clone attacks on online social networks.* In: Proceedings of the first ACM conference on Data and application security and privacy.

Karami, A., & Zhou, L. (2014). Improving static SMS spam detection by using new content-based features.

Karchefsky, S., & Rao, H. R. (2017). Toward a Safer Tomorrow: Cybersecurity and Critical Infrastructure *The Palgrave Handbook of Managing Continuous Business Transformation* (pp. 335-352): Springer.

Karlsson, J., Wohlin, C., & Regnell, B. (1998). An evaluation of methods for prioritizing software requirements. *Information and software technology, 39*(14-15), 939-947.

Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis* (Vol. 344): John Wiley & Sons.

Kaur, G., & Chhabra, A. (2014). Improved J48 classification algorithm for the prediction of diabetes. *International Journal of Computer Applications, 98*(22).

Kégl, B. (2013). The return of AdaBoost. MH: multi-class Hamming trees. *arXiv preprint arXiv:1312.6086*.

Khan, U. U., Ali, M., Abbas, A., Khan, S., & Zomaya, A. (2016). Segregating Spammers and Unsolicited Bloggers from Genuine Experts on Twitter. *IEEE Transactions on Dependable and Secure Computing*. doi: 10.1109/TDSC.2016.2616879

Kharratzadeh, M., Renard, B., & Coates, M. J. (2015). Bayesian topic model approaches to online and time-dependent clustering. *Digital Signal Processing*. doi: http://dx.doi.org/10.1016/j.dsp.2015.03.010

Kiruthiga, S., Kola Sujatha, P., & Kannan, A. (2014). *Detecting cloning attack in Social Networks using classification and clustering techniques.* In: 2014 International Conference on Recent Trends in Information Technology (ICRTIT).

Klir, G. J., & Yuan, B. (1996). *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A. Zadeh*: World Scientific Publishing Co., Inc.

Kondratovich, E., Baskin, I. I., & Varnek, A. (2013). Transductive Support Vector Machines: Promising Approach to Model Small and Unbalanced Datasets. *Molecular Informatics, 32*(3), 261-266.

Kontaxis, G., Polakis, I., Ioannidis, S., & Markatos, E. P. (2011). *Detecting social network profile cloning.* In: 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops).

Kuhlman, D. (2009). *A python book: Beginning python, advanced python, and python exercises*: Dave Kuhlman.

Lee, K., Caverlee, J., & Webb, S. (2010a). *Uncovering Social Spammers: Social Honeypots plus Machine Learning.* In: SIGIR 2010: Proceedings of the 33rd Annual International ACM SIGIR Conference on Research Development in Information Retrieval.

Lee, K., Caverlee, J., & Webb, S. (2010b). *Uncovering social spammers: social honeypots+ machine learning.* In: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval.

Lee, S., & Kim, J. (2014). Early filtering of ephemeral malicious accounts on Twitter. *Computer Communications, 54*, 48-57.

Leskovec, J. (2015). Stanford large network dataset collection. from https://snap.stanford.edu/data/

Li, Q., & Li, X.-F. (2007). Research on short message spam filtration based on support vector machine.

Li, Y., Xiao, R., Feng, J., & Zhao, L. (2013). A semi-supervised learning approach for detection of phishing webpages. *Optik-International Journal for Light and Electron Optics, 124*(23), 6027-6033. doi:http://dx.doi.org/10.1016/j.ijleo.2013.04.078

Lin, M.-S., Chiu, C.-Y., Lee, Y.-J., & Pao, H.-K. (2013). *Malicious URL filtering—A big data application.* In: 2013 IEEE International Conference on Big Data.

Lin, P.-C., & Huang, P.-M. (2013). A study of effective features for detecting long-surviving Twitter spam accounts. *2013 15th International Conference on Advanced Communications Technology (ICACT)*, 841.

Liu, D., Mei, B., Chen, J., Lu, Z., & Du, X. (2015). Community Based Spammer Detection in Social Networks. In X. L. Dong, X. Yu, J. Li & Y. Sun (Eds.), *Web-Age Information Management* (Vol. 9098, pp. 554-558).

Main, W., & Shekokhar, N. (2015). Twitterati Identification System. In H. Vasudevan, A. R. Joshi & N. M. Shekokar (Eds.), *International Conference on Advanced Computing Technologies and Applications* (Vol. 45, pp. 32-41).

Manurung, H. M. (2004). *An evolutionary algorithm approach to poetry generation.* (Doctor of Philosophy PhD), University of Edinburgh. Retrieved from https://www.era.lib.ed.ac.uk/handle/1842/314

Markines, B., Cattuto, C., & Menczer, F. (2009). *Social spam detection.* In: Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web.

Martinez-Romo, J., & Araujo, L. (2013). Detecting malicious tweets in trending topics using a statistical analysis of language. *Expert Systems with Applications, 40*, 2992-3000. doi: 10.1016/j.eswa.2012.12.015

Mccord, M., & Chuah, M. (2011). Spam detection on twitter using traditional classifiers *Autonomic and trusted computing* (pp. 175-186): Springer.

Melville, P., & Mooney, R. J. (2003). *Constructing diverse classifier ensembles using artificial training examples.* In: IJCAI.

Miller, Z., Dickinson, B., Deitrick, W., Hu, W., & Wang, A. H. (2014). Twitter spammer detection using data stream clustering. *Information Sciences, 260*, 64-73. doi: 10.1016/j.ins.2013.11.016

Mislove, A., Viswanath, B., Gummadi, K. P., & Druschel, P. (2010). *You are who you know: inferring user profiles in online social networks.* In: Proceedings of the third ACM international conference on Web search and data mining.

Mohaisen, A., Yun, A., & Kim, Y. (2010). *Measuring the mixing time of social graphs.* In: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. ACM.

Morrison, K. (2013). How Social Media Spam Erodes Brand Trust. Retrieved 20/05/2016, from http://www.adweek.com/digital/how-social-media-spam-erodes-brand-trust/

Mulamba, D., Ray, I., & Ray, I. (2016). *SybilRadar: A Graph-Structure Based Framework for Sybil Detection in On-line Social Networks.* In: IFIP International Information Security and Privacy Conference.

Mustafa, M. A., & Al-Bahar, J. F. (1991). Project risk assessment using the analytic hierarchy process. *IEEE transactions on engineering management, 38*(1), 46-52.

Narudin, F. A., Feizollah, A., Anuar, N. B., & Gani, A. (2014). Evaluation of machine learning classifiers for mobile malware detection. *Soft Computing*, 1-15.

Nettleton, D. F. (2013). Survey: Data mining of social networks represented as graphs. *Computer Science Review, 7*, 1-34. doi: 10.1016/j.cosrev.2012.12.001

Nguyen, H. (2013). Research report 2013 state of social media spam.

NIST. (2013). *Security and privacy controls for federal information systems and organizations* (4th ed.). Gaithersburg: National Institute of Standards and Technology.

Noriega, L. (2005). Multilayer perceptron tutorial. *School of Computing. Staffordshire University*.

Onnela, J.-P., Saramäki, J., Hyvönen, J., Szabó, G., Lazer, D., Kaski, K., . . . Barabási, A.-L. (2007). Structure and tie strengths in mobile communication networks. *Proceedings of the National Academy of Sciences, 104*(18), 7332-7336.

Özdağoğlu, A., & Özdağoğlu, G. (2007). Comparison of AHP and fuzzy AHP for the multi-criteria decision making processes with linguistic evaluations.

Pappa, G. L., Ochoa, G., Hyde, M. R., Freitas, A. A., Woodward, J., & Swan, J. (2014). Contrasting meta-learning and hyper-heuristic research: the role of evolutionary algorithms. *Genetic Programming and Evolvable Machines, 15*(1), 3-35.

Pérez-Rosés, H., Sebé, F., & Ribó, J. M. (2016). Endorsement deduction and ranking in social networks. *Computer Communications, 73*, 200-210.

PhishTank. (2015). Phishtank API. Retrieved 25th November 2015, from http://www.phishtank.com/

Precup, R.-E., & Hellendoorn, H. (2011). A survey on industrial applications of fuzzy control. *Computers in Industry, 62*(3), 213-226.

Quinlan, J. R. (2014). *C4. 5: programs for machine learning*: Elsevier.

Ramík, J., & Korviny, P. (2010). Inconsistency of pair-wise comparison matrix with fuzzy elements based on geometric mean. *Fuzzy Sets and Systems, 161*(11), 1604-1613.

Ratkiewicz, J., Conover, M., Meiss, M., Gonçalves, B., Flammini, A., & Menczer, F. (2011). *Detecting and Tracking Political Abuse in Social Media.* In: ICWSM.

Rijal, S. (2016). Risk Management using RISK MATRIX. Retrieved 20th January, 2017, from http://www.suvashrijal.com/

Roli, F., Kittler, J., & Windeatt, T. (2004). *Multiple Classifier Systems: 5th International Workshop, MCS 2004, Cagliari, Italy, June 9-11, 2004, Proceedings* (Vol. 5): Springer Science & Business Media.

Ruan, X., Wu, Z., Wang, H., & Jajodia, S. (2016). Profiling Online Social Behaviors for Compromised Account Detection. *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, 11*(1), 176-187. doi: 10.1109/tifs.2015.2482465

Saaty, T. L. (2005). *Theory and applications of the analytic network process: decision making with benefits, opportunities, costs, and risks*: RWS publications.

Saaty, T. L. (2008). Decision making with the analytic hierarchy process. *International journal of services sciences, 1*(1), 83-98.

Sadan, Z., & Schwartz, D. G. (2011). Social network analysis of web links to eliminate false positives in collaborative anti-spam systems. *Journal of Network and Computer Applications, 34*(5), 1717-1723.

Sallaberry, A., Zaidi, F., & Melançon, G. (2013). Model for generating artificial social networks having community structures with small-world and scale-free properties. *Social Network Analysis and Mining, 3*(3), 597-609.

Savage, D., Zhang, X., Yu, X., Chou, P., & Wang, Q. (2014). Anomaly detection in online social networks. *Social Networks, 39*, 62-70. doi: 10.1016/j.socnet.2014.05.002

Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review, 1*(1), 27-64.

Schneider, F., Feldmann, A., Krishnamurthy, B., & Willinger, W. (2009). *Understanding online social network usage from a network perspective.* In: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference.

Schoenherr, T., Tummala, V. R., & Harrison, T. P. (2008). Assessing supply chain risks with the analytic hierarchy process: Providing decision support for the offshoring decision by a US manufacturing company. *Journal of Purchasing and Supply Management, 14*(2), 100-111.

Schütze, H. (2008). *Introduction to Information Retrieval.* In: Proceedings of the international communication of association for computing machinery conference.

Shyni, C. E., Sundar, A. D., & Ebby, G. S. E. (2016). Spam profile detection in online social network using statistical approach. *Asian Journal of Information Technology, 15*(7), 1253-1262.

Singh, M., Bansal, D., & Sofat, S. (2014). Detecting Malicious Users in Twitter using Classifiers. *ACM International Conference Proceeding Series*, 247.

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing, 14*(3), 199-222.

Stanford University. (2008). Chi-Squared feature selection. from http://nlp.stanford.edu/IR-book/html/htmledition/feature-selectionchi2-feature-selection-1.html

Statista. (2016). Leading social networks worldwide as of April 2016, ranked by number of active users (in millions). from http://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

Stein, T., Chen, E., & Mangla, K. (2011). Facebook immune systemProceedings of the 4th Workshop on Social Network Systems. ACM (pp. 8): ACM. doi: 10.1145/1989656.1989664

Stringhini, G., Kruegel, C., & Vigna, G. (2010). *Detecting spammers on social networks.* In: Proceedings of the 26th Annual Computer Security Applications Conference. ACM.

Tan, E., Guo, L., Chen, S., Zhang, X., & Zhao, Y. (2013). *UNIK: unsupervised social network spam detection.* In: Proceedings of the 22nd ACM international conference on Conference on information & knowledge management.

Thomas, K., Grier, C., Song, D., & Paxson, V. (2011). *Suspended accounts in retrospect: an analysis of twitter spam.* In: Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference.

Thomas, K., & Nicol, D. M. (2010). *The Koobface botnet and the rise of social malware.* In: 2010 5th International Conference on Malicious and Unwanted Software (MALWARE).

Tran, N., Li, J., Subramanian, L., & Chow, S. S. (2011). *Optimal sybil-resilient node admission control.* In: INFOCOM, 2011 Proceedings IEEE.

Twitter. (2016). The twitter rules. Retrieved 28th January, 2016, from https://support.twitter.com/articles/18311

Twitter rate limit. (2015). Twitter rate limit for search/tweets REST API calls. from https://dev.twitter.com/rest/public/rate-limits

URIBL. (2015). URIBL API. Retrieved 25th November, 2015, from http://uribl.com/

Vance, A. (2009). Data analysts captivated by R's power. *New York Times, 6*(5.4).

Varol, O., Ferrara, E., Davis, C. A., Menczer, F., & Flammini, A. (2017). Online human-bot interactions: Detection, estimation, and characterization. *arXiv preprint arXiv:1703.03107*.

Veeralakshmi, V., & Ramyachitra, D. (2015). Ripple Down Rule learner (RIDOR) Classifier for IRIS Dataset. *Issues, 1*(1), 79-85.

Viswanath, B., Bashir, M. A., Crovella, M., Guha, S., Gummadi, K. P., Krishnamurthy, B., & Mislove, A. (2014). *Towards detecting anomalous user behavior in online social networks.* In: Proceedings of the 23rd USENIX Security Symposium (USENIX Security)}.

Viswanath, B., Post, A., Gummadi, K. P., & Mislove, A. (2011). An analysis of social network-based sybil defenses. *ACM SIGCOMM Computer Communication Review, 41*(4), 363-374.

Vlasselaer, V. V., Meskens, J., Van Dromme, D., & Baesens, B. (2013). *Using social network knowledge for detecting spider constructions in social security fraud.* In: 2013 Ieee/Acm International Conference on Advances in Social Networks Analysis and Mining (Asonam).

Wallace, M. (2016). Mitigating Cyber Risk in IT Supply Chains. *Global Bus. L. Rev., 6*, 4.

Wallen, C. (2015). Cyber risk and threat management: A discussion of tools and techniques.

Wang, A. H. (2010a). Detecting Spam Bots in Online Social Networking Sites: A Machine Learning Approach. In S. Foresti & S. Jajodia (Eds.), *Data and Applications Security and Privacy Xxiv, Proceedings* (Vol. 6166, pp. 335-342).

Wang, A. H. (2010b). Detecting spam bots in online social networking sites: a machine learning approach *Data and Applications Security and Privacy XXIV* (pp. 335-342): Springer.

Wang, A. H. (2010c). *Don't follow me: Spam detection in twitter.* In: Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on.

Wang, C., Zhang, Y., Chen, X., Liu, Z., Shi, L., Chen, G., . . . Lu, W. (2010). A behavior-based SMS antispam system. *Ibm Journal of Research and Development, 54*(6). doi: 10.1147/jrd.2010.2066050

Wang, G., Mohanlal, M., Wilson, C., Wang, X., Metzger, M., Zheng, H., & Zhao, B. Y. (2012a). Social turing tests: Crowdsourcing sybil detection. *arXiv preprint arXiv:1205.3856*.

Wang, G., Wilson, C., Zhao, X., Zhu, Y., Mohanlal, M., Zheng, H., & Zhao, B. Y. (2012b). *Serf and turf: crowdturfing for fun and profit.* In: Proceedings of the 21st international conference on World Wide Web.

WEKA. (2016). The University of Waikato. Retrieved 2nd February, 2016, from http://www.cs.waikato.ac.nz/ml/weka/

Widenius, M., & Axmark, D. (2002). *MySQL reference manual: documentation from the source*: " O'Reilly Media, Inc.".

Wu, F., Shu, J., Huang, Y., & Yuan, Z. (2016). Co-Detecting Social Spammers and Spam Messages in Microblogging via Exploiting Social Contexts. *Neurocomputing*.

Wüest, C. (2010). The risks of social networking. *Symantec Corporation*.

Xin-fang, S. (2013). *Survey of model and techniques for online social networks.* In: 2013 8th International Conference on Computer Science & Education (ICCSE). IEEE.

Yang, C., Harkreader, R., & Gu, G. (2013). Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers. *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, 8*(8), 1280-1293. doi: 10.1109/tifs.2013.2267732

Yang, C., Harkreader, R. C., & Gu, G. (2011). *Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers.* In: Recent Advances in Intrusion Detection.

Yang, W., Shen, G.-W., Wang, W., Gong, L.-Y., Yu, M., & Dong, G.-Z. (2015). Anomaly Detection in Microblogging via Co-Clustering. *Journal of Computer Science and Technology, 30*(5), 1097-1108. doi: 10.1007/s11390-015-1585-3

Yang, Z., Wilson, C., Wang, X., Gao, T., Zhao, B. Y., & Dai, Y. (2014). Uncovering social network sybils in the wild. *ACM Transactions on Knowledge Discovery from Data (TKDD), 8*(1), 2.

Yoon, J. W., Kim, H., & Huh, J. H. (2010). Hybrid spam filtering for mobile communication. *Computers & Security, 29*(4), 446-459. doi: http://dx.doi.org/10.1016/j.cose.2009.11.003

Yoon, Y. S., & Lee, H.-W. (2016). *Click it or not? the effect of providing information of the unknown's trustworthiness and discovering hidden relationship on risk-taking behavior.* In: Information and Communication Technology Convergence (ICTC), 2016 International Conference on.

Yu, H., Gibbons, P. B., Kaminsky, M., & Xiao, F. (2008). *Sybillimit: A near-optimal social network defense against sybil attacks.* In: IEEE Symposium on Security and Privacy, 2008. SP 2008. .

Yu, H., Kaminsky, M., Gibbons, P. B., & Flaxman, A. (2006). Sybilguard: defending against sybil attacks via social networks. *ACM SIGCOMM Computer Communication Review, 36*(4), 267-278.

Zainal, K., & Jali, M. Z. (2015). A Perception Model of Spam Risk Assessment Inspired by Danger Theory of Artificial Immune Systems. *Procedia Computer Science, 59*, 152-161.

Zhang, H.-y., & Wang, W. (2009). *Application of Bayesian Method to Spam SMS Filtering.* In: 2009 International Conference on Information Engineering and Computer Science.

Zhang, L. (2010). Comparison of classical analytic hierarchy process (AHP) approach and fuzzy AHP approach in multiple-criteria decision making for commercial vehicle information systems and networks (CVISN) project.

Zhang, X., Zhu, S., & Liang, W. (2012). *Detecting spam and promoting campaigns in the Twitter social network.* In: 2012 IEEE 12th International Conference on Data Mining.

Zhang, Y., & Lu, J. (2016). Discover millions of fake followers in Weibo. *Social Network Analysis and Mining, 6*(1), 1-15.

Zheng, X., Zeng, Z., Chen, Z., Yu, Y., & Rong, C. (2015). Detecting spammers on social networks. *Neurocomputing, 159*, 27-34. doi: 10.1016/j.neucom.2015.02.047

Zhi Yang, Jilong Xue, Xiaoyong Yang, Xiao Wang, & Dai, Y. (2015). VoteTrust: Leveraging friend invitation graph to defend against social network Sybils. *IEEE Transaction on Dependable and Secure Computing*. doi: 10.1109/TDSC.2015.2410792

Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning, 3*(1), 1-130.

Zuo, X., Blackburn, J., Kourtellis, N., Skvoretz, J., & Iamnitchi, A. (2016). The power of indirect ties. *Computer Communications, 73*, 188-199.

LIST OF PUBLICATIONS AND PAPERS PRESENTED

1. K. S. Adewole, N. B. Anuar, A. Kamsin, K. D. Varathan and S. A. Razak (2017). Malicious accounts: Dark of the social networks. *Journal of Network and Computer Applications*, 79(2017), pp. 41-67.

2. K. S. Adewole, N. B. Anuar, A. Kamsin, A. K. Sangaiah (2017). SMSAD: a framework for spam message and spam account detection. *Multimedia Tools and Applications*, pp. 1-36. DOI: https://doi.org/10.1007/s11042-017-5018-x.

3. K. S. Adewole, N. B. Anuar and A. Kamsin (2016). Ensemble based streaming framework for spam detection and risk assessment in microblogging social networks. *In proceedings of the 5th International Conference on Computer Science and Computational Mathematics (ICCSCM 2016)*, pp. 161-171. Published by Science and Knowledge Research Society. Available at https://www.iccscm.com/cms/

4. K. S. Adewole, N. B. Anuar and A. Kamsin (2016). Spammers Detection in Microblogging Social Networks Based on Content and Structural Analysis. *Paper presented at Postgraduate Research Excellence Symposium (PgRES 2016)*, organized by Faculty of Computer Science & Information Technology, University of Malaya.

5. Ibrahim Abaker Targio Hashem, Victor Chang, Nor Badrul Anuar, Kayode Adewole, Ibrar Yaqoob, Abdullah Gani, Ejaz Ahmed, and Haruna Chiroma (2016). The role of big data in smart city. *International Journal of Information Management*, 36(5), pp. 748-758.