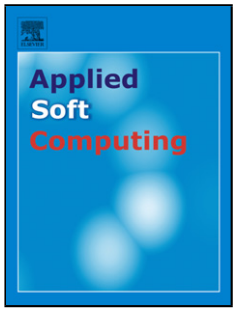



Title: SpamSpotter: An Efficient Spammer Detection Framework based on Intelligent Decision Support System on Facebook

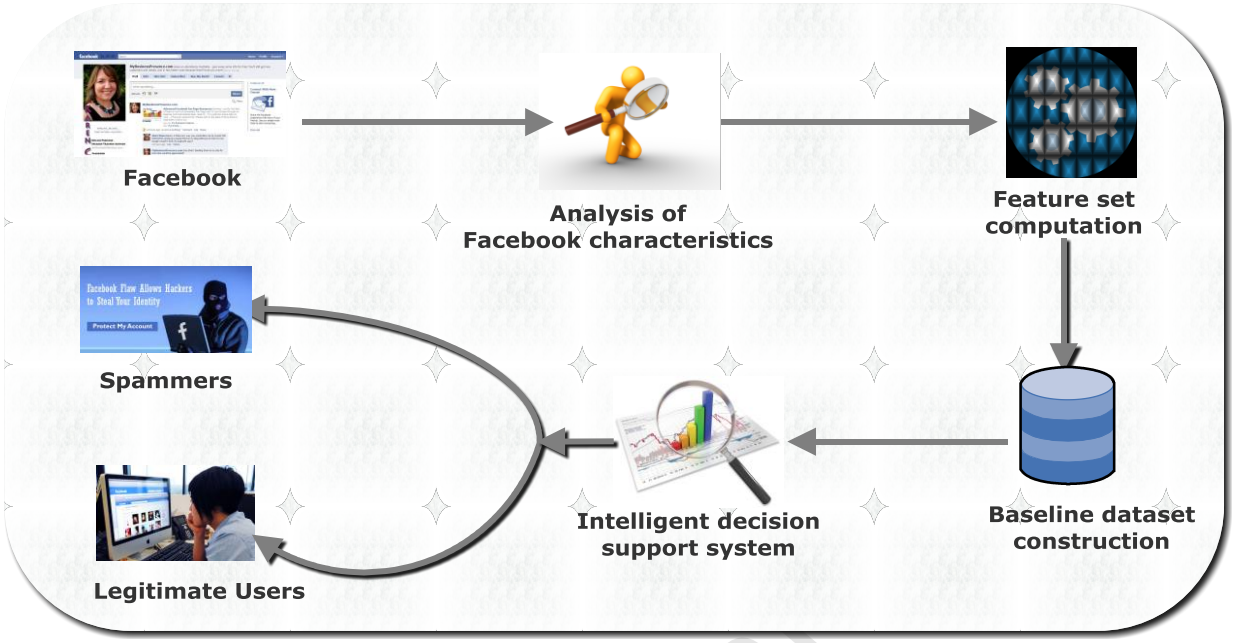
Author: Shailendra Rathore Vincenzo Loia Jong Hyuk Park

PII: S1568-4946(17)30571-9
DOI: http://dx.doi.org/doi:10.1016/j.asoc.2017.09.032
Reference: ASOC 4479

To appear in: *Applied Soft Computing*

Received date: 5-2-2017
Revised date: 31-8-2017
Accepted date: 15-9-2017

Please cite this article as: Shailendra Rathore, Vincenzo Loia, Jong Hyuk Park, SpamSpotter: An Efficient Spammer Detection Framework based on Intelligent Decision Support System on Facebook, *<![CDATA[Applied Soft Computing Journal]]>* (2017), http://dx.doi.org/10.1016/j.asoc.2017.09.032



Highlights

- Spam activities on Facebook are studied.
- A novel feature set is introduced to the task of spammer detection on Facebook.
- A baseline dataset of Facebook user profiles is constructed.
- We propose a SapmSpotter framework based on Intelligent Decision Support System.

# SpamSpotter: An Efficient Spammer Detection Framework based on Intelligent Decision Support System on Facebook

Shailendra Rathore[a], Vincenzo Loia[b], Jong Hyuk Park[a,*]

[a]*Department of Computer Science and Engineering, Seoul National University of Science and Technology, (SeoulTech) Seoul 01811, Korea*
[b]*Department of Management and Innovation Systems, University of Salerno, Italy*

**Abstract**

Facebook is one of the most popular and leading social network services online. With the increasing amount of users on Facebook, the probability of broadcasting spam content on it is also escalating day by day. There are a few existing techniques to combat spam on Facebook. However, due to the public unavailability of critical pieces of Facebook information, like profiles, network information, an unlimited number of posts and more, the existing techniques do not work efficiently for detecting many spammers. In this paper, we propose an efficient spammer detection framework (We called as SpamSpotter) that distinguishes spammers from legitimate users on Facebook. Based on Facebook's recent characteristics, the framework introduces a novel feature set to facilitate spammer detection. We use a baseline dataset from Facebook that included 300 spammers and 700 legitimate user profiles. The baseline dataset contains a set of features for each profile, which are extracted using a novel dataset construction mechanism. In addition, an intelligent decision support system that uses eight different machine learning classifiers on the baseline dataset is designed to distinguish spammers from legitimate users. To evaluate the efficiency and accuracy of our proposed framework, we implemented and compared it with existing frameworks. The evaluation results demonstrate that our proposed framework is accurate and efficient to deliver first-rate performance. It attains a higher accuracy of 0.984 and Mathew correlation coefficient of 0.977.

*Keywords:* Spammer detection, Facebook, Intelligent decision support system, Social network service, Machine learning

## 1. Introduction

Social Network Services (SNSs), such as Facebook[1] , LinkedIn[2] , and Twitter[3] , are some of the most popular applications of the Web. In this recent era, a very large amount of web users use SNSs to stay in touch with friends and family, meet new people, make work-related connections with others, and so on [1, 2, 3]. Among all SNSs, Facebook is a leading online SNS. According to Zephoria's digital marketing report [4], presently over 1.79 billion monthly active Facebook users and more than 1.18 billion users log into Facebook daily. With the escalating number of users on Facebook, the probability of broadcasting spam content on it is also increasing day by day [5]. According to a report from Nexgate [6], YouTube and Facebook deliver the most spam contents in comparison with other SNSs. The ratio of spam on YouTube and Facebook to the other SNSs is 100 to 1. The effect of spam on Facebook is already substantial. A spam post on Facebook is potentially viewed by all users and their friends. Even worse, it might cause malicious or unacceptable behavior in public discussions. This type of behavior can degrade the utilization experience

*Corresponding author
*Email address:* jhpark1@seoultech.ac.kr (Jong Hyuk Park)

[1]https://www.facebook.com/
[2]https://www.linkedin.com/
[3]https://www.twitter.com/

for other Facebook users. For example, sometimes spam is transmitted through a user clicking on bad links or installing malicious software.

The spam post or content on Facebook is transmitted by a malicious user, who is known as a spammer. Their ultimate goal is to seek financial gain. Spammers use various methods to achieve this goal. For instance, spammers can entice a Facebook user to their websites that contain advertisements (ads) or multimedia contents, such as exciting or saucy videos and images. Spammers get money each time a user clicks on their ads or contents. Therefore, spammers can use Facebook as an efficient medium for attracting a large audience and can generate a lot of money. This technique of making money is easy for spammers because a Facebook user exerts little effort when clicking on ads or multimedia contents. Most of the time, many users do not correctly recognize what they are clicking on. Another traditional method used by spammers is a phishing attack, in which spammers obtain the user credentials, such as password or credit card details. Furthermore, Spammers can also use malware, in which software is installed on the user's computer to accumulate their credential data. Nevertheless, these two techniques, phishing and malware, are used less because both techniques require more effort from the spammers to spread their spam behaviors.

Some techniques have been proposed by academia and various industries to provide possible solutions for identifying and detecting spammers on Facebook. However, these techniques have been discussed for solving some challenges and important issues as follows; Hongyu Gao et al. [7] proposed a technique that detects Facebook spam and its campaign using the URLs posting feature. However, it has been observed that multiple advertising and self-promotional campaigns on Facebook do not make use of URLs feature at all to distribute spams. In addition, the techniques proposed by Gianluca Stringhini et al. [8] and Faraz Ahmed et al. [9] do not exploit an absolute and efficient set of features to detect spammers on Facebook. In the other researches [10, 11, 12, 13, 14], it has also been studied that current techniques, which are used to combat spam in other SNSs platforms such as Twitter, cannot be directly applied to Facebook, because of the public unavailability and uncertainty of the critical pieces of Facebook information such as profiles, network information, and the unlimited number of posts. Currently, existing techniques for detecting spammers on Facebook are less effective due to the following three main issues: (1) The use of an inadequate set of features, (2) public unavailability and uncertainty about critical pieces of Facebook information, and (3) lower accuracy and higher response time.

In this paper, we propose a novel SpamSpotter framework based on the Intelligent Decision Support System (IDSS) that handles the above-mentioned three issues and challenges as follows: (1) The framework includes some newly proposed and existing features of Facebook user profiles to detect spammers, which solves the problem of the inadequate set of features in existing spammer detection system; (2) the framework constructs a baseline dataset to resolve the issue of public unavailability and uncertainty about critical pieces of Facebook information; and (3) the issue of lower accuracy and higher response time is resolved by deploying the IDSS system with our framework. It maps user profile data to the classification of a user profile as a spammer or legitimate. It maintains the properties of data representation, including availability, that play a critical role in ensuring the success of the spammer detection system. A decision process in IDSS gathers and analyzes all relevant information about the user from Facebook to maximize the effectiveness and accuracy of spammer detection. The extended use of machine learning classifiers in IDSS provides a fast response that is important for detecting a spammer on Facebook. In addition, the IDSS allows the Facebook service provider to quickly gather and process information in various ways in order to detect spammers. The main contributions of this paper are as listed below:

- This paper provides an analysis of the recent Facebook characteristics and proposes a set of novel profile and content-based features to facilitate the detection of spammers on Facebook.
- To address the issue of the public unavailability of critical pieces of Facebook information, this paper offers a special dataset construction mechanism that constructs a baseline dataset of Facebook user profiles.
- The major contribution of this paper is that we have developed a SpamSpotter framework, which relies on an IDSS to solve the problem of detecting spammers on Facebook. The IDSS implements a *decisionmaker* function that provides a fast response and high accuracy rate for spammer detection on Facebook.

- This paper provides an experimental evaluation of the proposed framework and the experiment results clarify why the framework can attain excellent performance.
- In this paper, we also graphically analyze the dissimilarities between spammers and legitimate users based on each of the proposed features.

The rest of the paper is organized as follows; Section 2 discusses the various existing techniques for spammer detection in SNSs. In Section 3, we propose a SpamSpotter framework architecture and describe its major components: analysis of Facebook characteristics, feature set computation, baseline dataset construction, and IDSS. Section 4 presents experimental evaluation of proposed framework and its comparison with existing frameworks. Finally, we conclude our paper in Section 5.

## 2. Related works

The success of Facebook has attracted the attention of many security researchers. Since this SNS platform is intensely based on the concept of a network of trust, those who abuse this network trust might cause substantial consequences. With the rapid development of SNS, social spam has attracted a significant amount of attention from both various industry and academia. Considerable work has been done for detecting spam in SNS.

In 2010, Hongyu Gao et al. [7] proposed a technique for identifying and characterizing a launched spam campaign by using asynchronous wall posts on Facebook. This technique is mainly based on features, such as wall posts, URL characteristics, post ratios, and other features of malicious accounts. The proposed approach detects spammers and their campaigns by utilizing the URLs posted on Facebook wall posts. Furthermore, Gianluca Stringhini et al. [8] designed a honey profile-based spam detection system, which creates a set of 900 honey profiles on three popular social networking communities: Facebook, Myspace and Twitter. Subsequently, it records the type of messages and contacts that the honey profiles received. The system analyzes the recorded data and can recognize the anomalous activity of users who interacted with the honey profiles. On the basis of activity analysis, the authors suggested six features to categorize spam profiles from legitimate profiles.

In 2011, Xin Jin et al. [15] proposed a data mining-based system to detect spam in SNS. The system extracted three types of features: content, text, and social networking features from SNS to construct a training dataset. A General Activity Detection (GAD) clustering algorithm was used for verifying the spam post from the dataset.

In 2012, Hongyu Gao et al.' s research [16] presented an online spam filtering system that can be applied as a component of SNS to detect spam messages generated by SNS users. In order to do so, the system exploits a set of novel features associated with spam campaigns on Facebook and Twitter. In the same year, Yin Zhu et al. [17] proposed a Supervised Matrix Factorization technique with Social Regularization (SMFSR) for detecting spammers on SNSs. It exploits both the social relations and activities of the user in a greatly scalable and groundbreaking way to detect spammers.

In 2013, Faraz Ahmed et al.'s research [9] presented a generic statistical scheme for identifying a spam profile on multiple SNSs: Facebook and Twitter. This scheme is based on real datasets, including both spam and ordinary profiles, crawled from Twitter and Facebook platforms. A set of 14 generic statistical features was used to identify spam profiles. These features are common to both Twitter and Facebook networks. The effectiveness of used features is evaluated by using three different classification algorithms: Naive Bayes, Jrip, and J48. Xia Hu et al. [18] introduced a unified spam detection framework that collectively uses content and network information to identify spammers in micro blogging.

In 2014, De Wang et al. [19] proposed a cross-site spam detection technique that can be used across all SNSs to detect spam. This technique relies on three major components: assembly and mapping, pre-filtering, and classification. The assembly and mapping component is used to extract the spam features across multiple SNSs. The extracted features are filtered using a pre-filtering component and a real dataset is prepared. The classification component uses associative and cross-domain classification to distinguish spam webpages, messages, and profiles from legitimate ones.

In 2015, Varun Kacholia et al. [10] developed a tool for detecting spam in video-hosting SNSs. It detects spam in the metadata associated with the user-generated video using the concept of clustering.

Recently, Linqing Liu et al.'s research [11] presented a smart spammer detection tool that relies on the topic model of Latent Dirichlet Allocation (LDA). The spammer is identified on the basis of two topic features: a) The Global Outlier Standard Score (GOSS), which exposes users interests in global topics; and b) The Local Outlier Standard (LOSS), which exposes the users interests in local topics. The major advantage of this approach is that it detects a smart spammer who highly pretends to be a legitimate user. Furthermore, Chao Chen et al. [12] implemented a novel Lfun scheme to detect drifted Twitter spam in real time. The scheme solves the drifting issue of spam tweets by thoroughly analyzing their statistical features. Similarly, Rajendra Kumar Roul et al.'s research [13] presented a combined approach to detect spam web pages. Their approach combines link and content-based methods. The link-based method applies personalized page ranking to distinguish spam webpages from legitimate ones, while the content-based method exploits the Part of Speech (POS) ratio and density test to filter spam webpages. Mohit Agrawal et al. [14] proposed an unsupervised spam detection framework that applies a Stochastic Approach for Link-Structure Analysis (SALSA) algorithm over the dataset collected from a popular Dutch SNS called Hyves.

In summary, the fundamental concept of the existing approaches for spammer detection in SNS is to extract a feature set that separates spammer profiles from legitimate ones and supplies that feature set into different machine learning classifiers for identifying inappropriate activity. The different performance might be attained by different classifiers as a result of dissimilarities existing in the used data sources and features. In general, this paper follows a similar idea, but with the different aspects listed below.

- Our proposed framework takes into account highly effective and selected features for identifying spammers on Facebook and attained excellent performance results, with accuracy rate as high as 0.984 and Mathew Correlation Coefficient (MCC) of 0.977. This is the greatest result ever obtained in comparison with the existing researches in the area of spam detection on Facebook. However, a small variation in performance results might be due to the different accumulated dataset with different contents, but this significant enhancement in the results from using our framework is still substantial and comparable.
- We also included multimedia content, such as videos and photos, to design the feature set. The significance of each selected feature was measured and validated using the feature selection algorithm.
- Having implemented our proposed framework over the two existing feature sets of [8] and [9], shows why the proposed framework is effective and why it can attain higher performance than other existing frameworks.

## 3. SpamSpotter framework

In this section, we discuss our proposed SpamSpotter framework for identifying spammers on Facebook.

### *3.1. SpamSpotter architecture*

Our framework identifies the spammers on the basis of two types of Facebook features: a) Profile-based features b) Content-based features. The technique used by the framework relies on the machine learning classifier to classify Facebook users into two categories: spammers and legitimate-users.

The architecture of SpamSpotter framework is shown in Fig. 1. The framework is composed of four major components: Analysis of Facebook characteristics, Feature set computation, Baseline dataset construction, and IDSS. The analysis of Facebook characteristics analyzes the various recent characteristics of Facebook and identifies which characteristics of Facebook are responsible for spreading spam behaviors on Facebook. Based on the identified characteristics, the feature set is computed by using feature set computation. The baseline dataset construction component constructs a baseline dataset by extracting the computed feature set from various user profiles on Facebook. The baseline dataset is supplied to IDSS that distinguish spammers from legitimate users on Facebook.

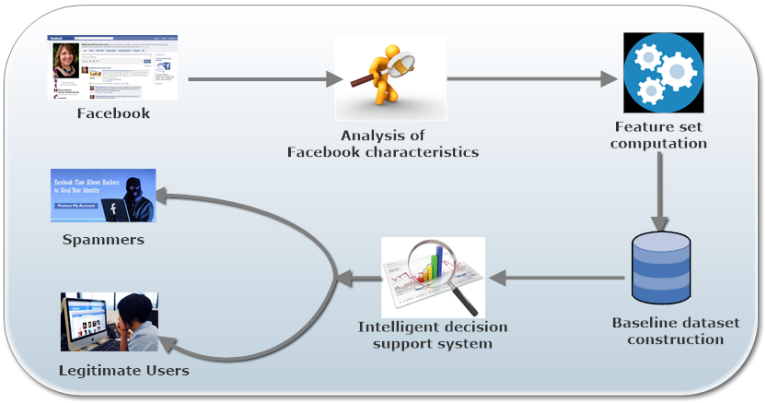



Figure 1: Architecture of SpamSpotter framework

### *3.2. Analysis of Facebook characteristics*

In order to design the SpamSpotter framework for distinguishing spammers from legitimate users on Facebook, we analyzed its recent characteristics that are described below.

*Community.* The community characteristic of Facebook involves events, groups, and liked pages. The event community provides a way for the user to share his or her thoughts on upcoming events or occasions. Since, shared thoughts can influence a large number of users in a short period of time; a spammer can share spam information about the event and infect a huge amount of people who participated in that event. The probability of a user being a spammer also depends on the number of events in which the user is participating. Moreover, the group community on Facebook connects a user with a specific set of people, such as coworkers, teammates, or family. It is a place where a user can share documents, photos, videos, and updates with a specific set of users. Like the event community, since this feature also influences a large number of users in a short period of time, a spammer can spread spam information in a group and infect huge amount of its members.

*Posting.* Posting is a major way for users to communicate on Facebook. A user can share messages/posts on community pages and other users profiles. A shared user post can be viewed by the user's friends or everyone on Facebook. However, a spammer can take advantage of this posting feature and share spam posts on community pages. This type of spam affects a large number of the users who have liked these pages. In addition, a spammer can share spam posts on an individual user profile for financial or other gain.

*URLs.* Website URLs are a way to share interesting information on SNSs. Facebook users can share website URLs with their colleagues for transmitting some of their favorite information. On Facebook, URLs can be shared in the form of posts and comments. However, spammers can use this URL feature for sharing the link to a malicious website. They can short the URLs by using URLs shortening facilities, such as bit.ly. Short URLs can conceal the original URLs and vague the suspicious websites behind them. As an outcome, this offers an opportunity for spammers to spam and phish.

*Photos and videos.* This feature allows users to upload albums, photos, and videos onto Facebook. However, according to the case study by Xin Jin et al. [15], spammers can post spam photos/videos on Facebook. For instance, spammers may trick users by posting spam videos/photos instead of sharing malicious URLs on Facebook. The spam videos/photos shared by spammers may appear as a Like or other buttons on a user's timeline [20]. When users click on the spam videos/photos, they may be redirected to the spammers websites. Later, spammers may steal the user's credentials using those websites. This type of spam spreading technique is called Like-Jacking [21].

*Tagging.* Facebook provides a tagging feature in which a user can tag other users. For instance, a user can tag a photo to show who is in that photo or post a status update and say who he or she is with. However, a spammer can tag a large number of users in a single post for spreading malicious content to a large audience with little effort.

*Hashtag.* A user can make a topic/phrase/event as a clickable link by using a hashtag. It helps the user to find the post related to the topics/phrases/events that they are interested in. A hashtag can be added by writing # (the number sign) along with the topic/phrase/event. However, spammers can use a large number of hashtags in their post for spreading malicious activity to a large number of users. According to Rebecca Hiscott [22], a hashtag is considered to spam if it is utilized too frequently in a single post on Facebook. A legitimate post on Facebook should contain a maximum of three hashtags.

*Comments and Likes.* A user can appreciate or promote a shared post on Facebook by clicking the Like button below the post or by writing comment about the post.

### *3.3. Feature set computation*

In order to compute a feature set, we generalize Facebook SNS as a combination of various characteristics such as community, posting, URLs, photos and videos, Tagging, hashtag, comments, and likes. Based on the work of [8, 9, 15, 20, 22], we found that an analysis of these characteristics can be used to distinguish the behaviors and activities of spammer and legitimate user profiles. For instance, legitimate users have many legitimate friends, while spammers are fraudulent profiles that never reply to comments. The spammer profiles may share a greater number of attractive photos and videos designed to spread their spam behaviors more quickly. They generally use attractive photos as their profile picture in order to attract others users. Therefore, we analyzed each of the characteristics (as provided in the earlier subsection) and computed a novel set of features that can be used to detect spammers on Facebook. We categorized the selection of features for computation of the feature set into two categories of profile-based features and content-based features. The selection of both types of features is based on the existing studies [8, 9, 15, 20, 22] and our analysis of Facebook characteristics. The selection and computation of each feature for both categories are explained below.

*Profile-based features ($f^{(p)}$).* The use of features related to a user's SNS profile is a legacy of previous work on the detection of spammers in SNS. The selection of these features is based on the behaviors and activities of user profiles with regard to a given profile's interaction rate and content posting rate. Based on our analysis of Facebook characteristics, we observed that spammer and legitimate user profiles show a significant difference in the social interaction rate in terms of number of friends, number of a community, and number of followings. Moreover, the two user profiles show remarkable dissimilarities in the content posting rate in terms of total number of posts shared, average number of posts shared per day, total number of URLs shared, average number of URLs shared per day, total number of photos/videos shared, and average number of photos/videos shared per day. These significant differences in the social interaction rate and the content posting rate can be used to distinguish spammer and legitimate user profiles, and thereby assist the detection of spammers on Facebook. Therefore, in our framework, we computed ten profile-based features that depict the behavior of the user on his or her Facebook account. Concise details of each feature are shown in Table 1.

*Content-based features ($f^{(c)}$).* These types of features are based on the recent activities of the user's Facebook profile. The features are computed from the contents associated with $n$ recent posts of the user's profile, such as photos/videos, tags, comments, likes, and URLs associated with $n$ recent posts. Based on our analysis of Facebook characteristics, we found that both spammer and legitimate user profiles show different behaviors in terms of the sharing of content associated with the post. For instance, spammers may associate content with the post, such as spam words or malicious URLs, in order to spread malicious behaviors, as well as scandalous photo/videos to attract a large audience or number of users to their profile. We measured this difference in behavior in terms of the fraction of posts containing spam words, URLs, photos/videos, and

rate of comments, likes, tags, and hashtags per post. We also observed that spammers share repeated URLs and multiple links. Therefore, the average rate of URLs repetition and the maximum number of links per post were also measured. Based on this measurement, we computed nine content-based features from $n$ recent posts of the user profile. Each feature is described in Table 2.

Table 1: List of profile-based features

| No. | Profile-based features | Reference |
|---|---|---|
| 1. | Number of friends: $f_1^{(p)}$ | [8, 9] |
| 2. | Number of followings: $f_2^{(p)}$ | New |
| 3. | Number of Community: $f_3^{(p)}$ | [9] |
| 4. | The age of the user account (in days): $f_4^{(p)}$ | New |
| 5. | Total number of posts shared: $f_5^{(p)}$ | [8, 9] |
| 6. | Average number of posts shared per day: $f_6^{(p)} = f_5^{(p)}/f_4^{(p)}$ | New |
| 7. | Total number of URLs shared: $f_7^{(p)}$ | [8, 9] |
| 8. | Average number of URLs shared per day: $f_8^{(p)} = f_7^{(p)}/f_4^{(p)}$ | New |
| 9. | Total number of photos/videos shared: $f_9^{(p)}$ | [15, 20] |
| 10. | Average number of photos/videos shared per day: $f_{10}^{(p)} = f_9^{(p)}/f_4^{(p)}$ | New |

### *3.4. Baseline dataset construction*

To build the SpamSpotter framework, we needed a labeled collection of Facebook user profiles, which are pre-classified into spammers and legitimate user profiles. The labeled collection is a baseline dataset containing the most significant features of several Facebook user profiles. However, this type of dataset is not publicly available, and, therefore, a special dataset construction mechanism was developed in which a web crawler using Facebook API was used to collect a real dataset from information that is publicly available in a user's Facebook profile. This mechanism solves the problem of collecting critical pieces of Facebook information, like profiles, an unlimited number of posts, and so on. A functional block diagram of the mechanism that we developed is shown in Fig. 2. The overall working procedure of the mechanism is summarized in the steps laid out below.

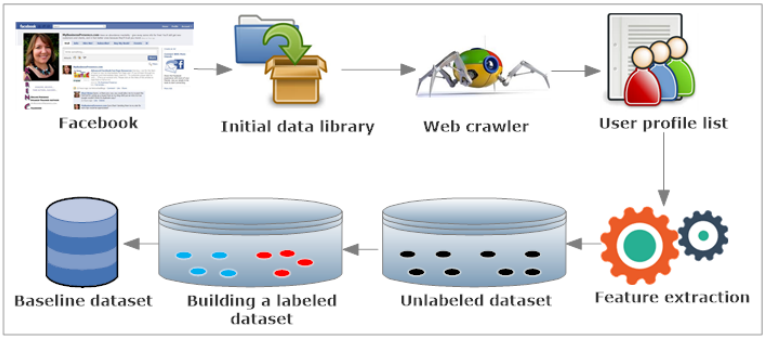



Figure 2: Baseline dataset construction

***Step 1***. An initial data library was created by selecting spammer and legitimate user profiles from Facebook. This library contained 30 legitimate user profiles and 30 spammer profiles. The legitimate user profiles were selected from the set of public profiles that were more active on Facebook, e.g. profiles that generated comments and posts very frequently, or which had a higher reaction and interaction rate on Facebook. A

Table 2: List of content-based features

| No. | Content-based features | References |
|---|---|---|
| 1. | Fraction of the posts containing spam words: $f_1^{(c)} = P_{sw}/n$, where $P_{sw}$ is total number of posts containing spam words. | New |
| 2. | Fraction of the posts containing URLs: $f_2^{(c)} = P_{url}/n$, where $P_{url}$ is total number of posts containing URLs. | New |
| 3. | Fraction of the posts containing photos/videos: $f_3^{(c)} = P_{pv}/n$, where $P_{pv}$ is total number of posts containing photos/videos. | New |
| 4. | Average number of comments per post: $f_4^{(c)} = P_{cmnt}/n$, where $P_{cmnt}$ is total number of comments in $n$ recent posts. | New |
| 5. | Average number of likes per post: $f_5^{(c)} = P_{liks}/n$, where $P_{liks}$ is total number of likes in $n$ recent posts. | New |
| 6. | Average number of tags in a post (Rate of tagging): $f_6^{(c)} = P_{tags}/n$, where $P_{tags}$ is total number of tags present in $n$ recent posts. | [9] |
| 7. | Average rate of URLs repetition: $f_7^{(c)} = U_{rptd}/U_{totl}$, where $U_{rptd}$ is total number of repeated URLs in $n$ recent posts and $U_{totl}$ is total number of URLs present in $n$ recent posts. | [9] |
| 8. | Maximum number of links present in a post: $f_8^{(c)} = max\{P_1, P_2...P_i...P_n\}$, where $P_i$ is total number of links present in $i^{th}$ post. | New |
| 9. | Average number of hashtags present in a post: $f_9^{(c)} = P_{hstg}/n$, where $P_{hstg}$ is total number of hashtags present in $n$ recent posts. | [22] |

social media analytic tool called NapoleonCat [4] was used to determine the activeness of each user profile. This tool enables an in-depth analysis of any public profile on Facebook, and measures the activity index of the user profile in terms of number of its comments, posts, interactions, and reactions rate. The spammer profiles were selected from the set of public profiles that were involved in malicious activity too often, e.g. profiles that shared malicious URLs or links, scandalous videos, much pornographic or enticing information, and many links directed to dirty websites, and that broadcast texts that make sympathetic appeals or threats to other users in order to circulate malicious messages.

***Step 2***. User profile list: The user profile list $ul$ was created by crawling the list of followings and friends for each profile in the initial data library. For this process, a web crawler was used. Thus, we successfully crawled 1,000 Facebook profiles and created user profile list $ul$. The created $ul$ was an unlabeled mixture of spammer and legitimate user profiles.

***Step 3***. Feature Extraction: The features of each profile $u_i$ in user profile list $ul$ were extracted using a novel features extraction algorithm, which is shown in Algorithm 1. The overall working procedure of the algorithm for a profile $u_i$ is as follows:

(a) Profile-based features ($f_{u_i}^{(p)}$): These features were extracted using Facebook-Graph API [23] and were saved into the unlabeled dataset $D$.

(b) Content-based features ($f_{u_i}^{(c)}$): For extracting these features, a recent post list $pl_{u_i}$ of profile $u_i$ was crawled using Facebook-Graph API. For each recent post $p_j$ in $pl_{u_i}$, the raw features $f_{p_j}^{(raw)}$ were extracted and saved as an XML file $\chi$. Furthermore, $f_{u_i}^{(c)}$ was computed through parsing XML $\chi$. The computed value of $f_{u_i}^{(c)}$ was saved in the unlabeled dataset $D$. Thus, we obtained an unlabeled

[4] https://napoleoncat.com/

**Algorithm 1** Feature extraction algorithm

**Input:** A user profile list $ul$ crawled from Facebook
**Output:** Unlabeled dataset $D$

**while** $ul \neq \phi$ **do**
  $D \leftarrow u_i$, where $u_i \in ul$
  $f_{u_i}^{(p)} \leftarrow \phi$, where$f_{u_i}^{(p)}$ is $u_i$'s profile-based features
  $f_{u_i}^{(c)} \leftarrow \phi$, where$f_{u_i}^{(c)}$ is $u_i$'s content-based features
  **Extract** $f_{u_i}^{(p)}$
  **Save** $D \leftarrow f_{u_i}^{(p)}$
  **Crawl** $u_i$'s recent post list $pl_{u_i}$
  **while** $pl_{u_i} \neq \phi$ **do** {for each post $p_j$ in the $pl_{u_i}$ }
    $f_{p_j}^{(raw)} \leftarrow \phi$, where $f_{p_j}^{(raw)}$ is raw feature of $p_j$
    **Extract** $f_{p_j}^{(raw)}$
    **Save** $\chi \leftarrow f_{p_j}^{(raw)}$, where $\chi$ is XML file
  **end while**
  **Pars**e $\chi$ and **Compute** $f_{u_i}^{(c)}$
  **Save** $D \leftarrow f_{u_i}^{(c)}$
  $\chi \leftarrow \phi$
  $u_i \leftarrow u_i + 1$
**end while**

dataset $D$ that contained the extracted features of each profile $u_i$ in the user profile list $ul$. Finally, we successfully assembled the unlabeled dataset for 6 months from January 7-July 5, 2016. The unlabeled dataset contained 1,000 profiles, around 200K posts, and the extracted features of each profile.

***Step 4***. Building a labeled dataset: To build a labeled dataset, each profile in the unlabeled dataset $D$ was manually labeled as a spammer or as a legitimate profile on the basis of its characteristics. In the manual labeling process, a user profile was labeled as a spammer if it displayed any one of the following characteristics: a) the sharing of phishing or malicious links or URLs; b) the posting of scandalous videos, or the broadcasting of malicious messages containing chain letters and phishing text that usually ask for private information or money; c) the sharing of a large amount of advertisements or URLs that promote online shopping websites; and d) the dissemination of much pornographic or enticing information and many links directed to dirty websites. To determine the first characteristic, Google Safe Browsing technology [5] was applied, and the rest of the characteristics were measured by scanning the posted contents of each user profile. Finally, based on these characteristics, we labeled 300 profiles as spammers and the rest of the 700 profiles as legitimate profiles, which we then used as the baseline dataset in our framework.

Finally, the constructed baseline dataset contains a labeled collection of 1,000 Facebook user profiles in which 300 profiles are labeled as spammer and 700 profiles are labeled as being legitimate. Each profile in our baseline dataset is represented with their features value and label as follows:

$\langle$ Profile number , $f_1^{(p)}$, $f_2^{(p)}$, $f_3^{(p)}$, $f_4^{(p)}$, $f_5^{(p)}$, $f_6^{(p)}$, $f_7^{(p)}$, $f_8^{(p)}$, $f_9^{(p)}$,$f_{10}^{(p)}$, $f_1^{(c)}$, $f_2^{(c)}$, $f_3^{(c)}$, $f_4^{(c)}$, $f_5^{(c)}$, $f_6^{(c)}$,$f_7^{(c)}$, $f_8^{(c)}$, $f_9^{(c)}$, Label $\rangle$.

We calculated the time complexity for constructing a baseline dataset by computing the run time of the Algorithm 1. Let, the time complexity of extracting and computing features for a profile is $d$ and $|ul|$ is the total number of profiles in the baseline dataset, then:

[5] https://www.google.com/transparencyreport/safebrowsing/diagnostic/

Time complexity for constructing a baseline dataset B = (Total number of profile in baseline dataset) × (Time complexity of extracting and computing features for a profile)

$$B = |ul| * d. \tag{3.1}$$

*3.5. Intelligent decision support system*

In this subsection, an IDSS is introduced to detect spammer profiles in Facebook. Let $G = (V, E)$ be a SNS. Here, $V$ denotes the set of all user profiles and $E$ refers the set of relations between the user profiles. We denote the two different views or features of each profile (namely, profile-based features as $f^{(p)}$ and content-based features as $f^{(c)}$), as discussed in subsection 3.3. For a given unknown user profile $x_i \in V$, all features is denoted by the feature vector $f_{x_i} = (f_{x_i}^{(p)}, f_{x_i}^{(c)})$. The baseline dataset is denoted as $BD$, which we prepared in earlier subsection. The detection of a spammer profile in $G$ using IDSS can be formulated as follows: Given a new user profile $x_i \in V$, its corresponding feature vector is $f_{x_i} = (f_{x_i}^{(p)}, f_{x_i}^{(c)})$ and baseline dataset $BD$, then,

$$[\alpha_{x_i}^l, \alpha_{x_i}^s] = decisionmaker(BD, f_{x_i}) \Rightarrow y = \{Legitimate, Spammer\} \tag{3.2}$$

where, *decisionmaker* is a function that determines the class $y$ to which the user profile $x_i$ belongs to. This function takes two inputs, baseline dataset $BD$ and feature vector $f_{x_i}$, and returns the probability set $[\alpha_{x_i}^l, \alpha_{x_i}^s]$, where, $\alpha_{x_i}^l$ and $\alpha_{x_i}^s$ denote the calibrated probabilities of legitimate and spammer classes, respectively, for the profile $x_i$. The *decisionmaker* function is provided using the machine learning classification algorithm. As per other existing studies [8, 9], on spammer detection in Facebook, several machine learning classification algorithms, such as Random Forest, Decision Tree, and Bayesian Network, can be used to provide the *decisionmaker* function for spammer detection on Facebook. However, the Bayesian Network (BN) classifier has excellent performance over all other classification algorithms due to the reasons given below.

(1) The BN classifier is noise robust. Generally, a Facebook user profile cannot be predicted as being a spammer with confidence even through some features of the user profile match the training examples. This demonstrates that the relationship between the features and the prediction of spammers is non-deterministic. The BN classifier treats the non-deterministic relationship between features and the prediction of spammers as random variables and defines their relationship using calibrated probability. While other classifiers, such as the decision tree, cannot handle these kind of confounding factors and noisy data.

(2) The class label (spammer or legitimate) for a user profile on Facebook is predicted based on the specific behavior or pattern of the user. The probability of a user being a spammer is computed based on their behavior, instead of providing a common rule. Moreover, the BN classifier is an efficient and simple classification algorithm.

The BN classifier works on the concept of the well-known Bayes theorem, which is defined using following equation [24].

$$P(X|Y) = (P(Y|X)P(X))/P(Y) \tag{3.3}$$

where, $P(X|Y)$ is a conditional probability of $X$ with the consideration that the given $Y$, $P(Y|X)$ is the probability of $Y$ with the given $X$, $P(X)$ and $P(Y)$ denotes the probability of $X$ and $Y$, respectively.

The BN classifier depends on the assumption of conditional independent features, which is required for detecting spammers on Facebook. With the assumption of feature independency, the calibrated probability for both classes of spammer and legitimate can be calculated by using the Bayes theorem. Let $x_i = (f_1, f_2, ....., f_d)$ be a vector that denotes each Facebook user profile with the value of its features $(f_1, f_2, ....., f_d)$. A baseline dataset $BD$ that contains a labeled collection of Facebook user profiles with their features value is given. Then, the calibrated probabilities of both classes for user profile $x_i$ is provided using the following equations:

$$\alpha_{x_i}^l = P(C_l|(f_1, f_2, ...., f_d)) = P(C_l)\prod_{i=1}^{d} P(f_i|C_l) \tag{3.4}$$

$$\alpha^{s}_{x_i} = P(C_s|(f_1, f_2, ...., f_d)) = P(C_s)\prod_{i=1}^{d} P(f_i|C_s) \tag{3.5}$$

where, $P(C_l|(f_1, f_2, ...., f_d))$ stands for the conditional probability of the legitimate class for the user profile $x_i$ with the given value of its features, $P(C_l)$ denotes the prior probability of legitimate class $C_l$ in baseline dataset $BD$. $P(f_i|C_l)$ is the probability of feature $f_i$ with the given legitimate class $C_l$ in the baseline dataset $BD$. Similarly, $P(C_s|(f_1, f_2, ...., f_d))$ stands for the conditional probability of the spammer class for the user profile $x_i$ with the given value of its features. $P(C_s)$ denotes the prior probability of spammer class $C_s$ in the baseline dataset $BD$, and $P(f_i|C_s)$ is the probability of the feature $f_i$ with the given spammer class $C_s$ in baseline dataset $BD$.

After calculating the calibrated probabilities $\alpha^{l}_{x_i}$ and $\alpha^{s}_{x_i}$, the IDSS makes a decision based on the two assumptions provided below.

(1) If $(\alpha^{l}_{x_i} < \alpha^{s}_{x_i})$, then, the IDSS classifies the user profile $x_i$ as being in the spammer class.

(2) If $(\alpha^{l}_{x_i} > \alpha^{s}_{x_i})$, then, the IDSS classifies the user profile $x_i$ as being in the legitimate class.

The BN classifier calculates the calibrated probabilities of both classes with the assumption of feature independency. However, many dependent feature vectors existed in the baseline dataset $BD$. Equations 3.4 and 3.5 show that each probability in the set $\{P(f_1|C_l), P(f_2|C_l), ...., P(f_d|C_l)\}$ and the set $\{P(f_1|C_s), P(f_2|C_s), ...., P(f_d|C_s)\}$ need to be independent of each other. If the interdependency of these probabilities is not achieved, then the BN classifier will return incorrect results and its accuracy will be substantially diminished. To address this issue, a method is required to eliminate the dependency between the features as far as possible. However, the Support Vector Machine (SVM) is a method that is capable of classifying training instances with dependent features into different classes [25]. Hence, we propose a new method of elimination for the BN classifier, in which an elimination process based on SVM is used to remove training instances that are wrongly classified into incorrect classes by the BN classifier due to the dependent features. Thus, the feature dependencies are decreased while the accuracy of the BN classifier is increased. The overall procedure of the elimination method is described as follows:

The baseline dataset $BD$ is first trained by applying the BN method. For each user profile in the baseline dataset $BD$, the BN classifier generates a corresponding class. Therefore, the following set of results is obtained:

$$(u_1, v_1), (u_2, v_2), ...., (u_n, v_n) \tag{3.6}$$

where $u_i \in R, v_i \in \{C_l, C_s\}, i = 1, 2, ....n$ denotes the user profiles in the baseline dataset $BD$ and their corresponding classes; and n stands for the number of user profiles in the baseline dataset $BD$.

Next, the nearest neighbor for each user profile is determined. For each user profile, if it and its nearest neighbor belong to the same class, then the user profile is retained. Otherwise, the user profile (and the corresponding class information) is removed from the baseline dataset $BD$. The user profile is removed because it is categorized into an incorrect class as a result of the feature dependency between its nearest neighbor user profile and itself. In a nutshell, the user profiles with dependent features are removed from the baseline dataset $BD$. It further improves the classification accuracy of the BN. The nearest neighbourliness of two user profiles can be determined by computing their distance from each other. Let $A = (a_1, a_2, ...., a_d)$ and $B = (b_1, b_2, ...., b_d)$ be two vectors that denote two user profiles with the feature values $(a_1, a_2, ...., a_d)$ and $(b_1, b_2, ...., b_d)$, then their distance can be computed by using the following vector operation:

$$dis(A, B) = \sqrt{\sum_{k=1}^{d}(a_k - b_k)} \tag{3.7}$$

Based on the concept of SVM, the two nearest neighbor user profiles (closed in distance) very likely belong to the same class. Otherwise, the results of the classification need to be re-evaluated [25]. Algorithm 2 describes the pseudo-code of the elimination method for the BN classifier, where $U = \{u_1, u_2, ...., u_n\}$ and $V = \{v_1, v_2, ...., v_n\}$ denotes the sets of the user profiles and their corresponding classes, respectively.

Let $Y = \{y_1, y_2, ...., y_n\}$ be the set of classification results produced by the BN classifier. This results is re-evaluated by the elimination method, where the elimination of user profiles with dependent features from set $Y$ is performed and the output set $Y^1$ of user profiles is used for further classification by BN classifier.

**Algorithm 2** Elimination method for BN classifier

**Input:** $U = \{u_1, u_2, ...., u_n\}$, $V = \{v_1, v_2, ...., v_n\}$, and $Y = \{y_1, y_2, ...., y_n\}$
**Output:** $Y^1$

**Initialization** $i = 1, j = 1$
**while** $(i \leq n)$ **do**
  **while** $(j \leq n)$ **do**
    **if** $i \neq j$ **then**
      $dis_{ij} \leftarrow dis(u_i, u_j)$
    **end if**
    $j = j + 1$
  **end while**
  $i = i + 1$
**end while**
**while** $(i \leq n)$ **do**
  $temporary \leftarrow \infty$
  $nearestneighbor \leftarrow i$
  **while** $(j \leq n)$ **do**
    **if** $i \neq j$ **then**
      **if** $(temperary > dis(u_i, u_j))$ **then**
        $nearestneighbor \leftarrow j$
        $temporary \leftarrow dis(u_i, u_j)$
      **end if**
    **end if**
    $j = j + 1$
  **end while**
  **if** $(v_i \neq v_{nearestneighbor})$ **then**
    Eliminate $y_i$ from $Y$
  **end if**
  $i = i + 1$
**end while**
**RETURN** $Y$

The proposed elimination method with the BN classifier was used to provide a *decisionmaker* function in the IDSS. In order to provide a better understanding, the overall working procedure of the IDSS is illustrated in Algorithm 3 and explained below.

***Step 1***. For a given SNS $G$, first select an input test set $x$. This set contains a list of profiles that need to be tested for spam behavior. The most important part of the IDSS is constructing the baseline dataset $BD$, as discussed in the earlier subsection.

***Step 2***. For each profile $x_i$ in the set $x$, the appropriate features are extracted by using the feature extraction mechanism described in Subsection 3.4 and the feature vector is computed. The *decisionmaker* function calculates the calibrated probabilities $\alpha^l_{x_i}$ and $\alpha^s_{x_i}$ for the prediction of the legitimate and spammer classes by using the baseline dataset $BD$ and computed feature vector $f_{x_i}$. After calculating the probabilities, if $(\alpha^l_{x_i} < \alpha^s_{x_i})$, then the user profile $x_i$ is classified as being a spammer and is added to the list of spammer profiles $(y)$.

**Algorithm 3** Overall working procedure of the IDSS

**Input:** A input test set $x$ of user profiles
**Output:** $y$ list of all spammer profiles in $x$

1: **Initialization** $y \leftarrow \phi$
2: **Construct** Baseline dataset $BD$
3: **while** $x \neq \phi$ **do**
4: $temp \leftarrow x_i$, where $x_i \in x$
5: **Compute** $f_{x_i} = (f_{x_i}^{(p)}, f_{x_i}^{(c)})$
6: $[\alpha_{x_i}^{l}, \alpha_{x_i}^{s}] = decisionmaker(BD, f_{x_i})$
7: **if** $(\alpha_{x_i}^{l} < \alpha_{x_i}^{s})$ **then**
8: $y = y \cup temp$
9: **end if**
10: $x_i \leftarrow x_{i+1}$
11: **end while**
12: **RETURN** $y$

***Step 3***. Step 2 is repeated for each profile until set $x$ is empty. Finally, the IDSS returns ($y$) a list of all spammer profiles.

***Time complexity***. The time complexity of the overall working procedure of the IDSS can be calculated by computing the running time of each statement shown in Algorithm 3. In the second statement, the baseline dataset is constructed. Let $B$ is the time complexity of constructing the baseline dataset $BD$. Statements 4 to 10 are located inside the while loop (statement 3). Let $|x|$ is the number of profiles in the input test set $x$, $d$ refers to the time complexity of computing the feature vector for each profile in the input test set $x$, and $C$ is the time complexity of the *decisionmaker* function, then, the time complexity for the while loop is $|x| * d * C$. Hence, the worst-case time complexity of IDSS is $T = O(B + |x| * d * C)$.

***Illustrative example***. To better understand how the proposed framework works, we will now present an example. Suppose A is a Facebook user profile. We can determine whether this profile is a spammer or legitimate by using our framework, which involves the steps laid out below.

***Step 1***. In this step, extract the profile-based features $f^{(p)}$ and content-based features $f^{(c)}$ of the user profile A by using the feature extraction mechanism described in Subsection 3.4. The values of the extracted features are shown as: $\langle$ Profile number , $f_1^{(p)}$, $f_2^{(p)}$, $f_3^{(p)}$, $f_4^{(p)}$, $f_5^{(p)}$, $f_6^{(p)}$, $f_7^{(p)}$, $f_8^{(p)}$, $f_9^{(p)}$,$f_{10}^{(p)}$, $f_1^{(c)}$, $f_2^{(c)}$, $f_3^{(c)}$, $f_4^{(c)}$, $f_5^{(c)}$, $f_6^{(c)}$,$f_7^{(c)}$, $f_8^{(c)}$, $f_9^{(c)}$ $\rangle$= $\langle$A, 39, 300, 907, 200, 1000, 5, 850, 4.25, 922, 4.61, 0.15, 0.49, 0.55, 0.56, 0.47, 40, 0.20, 4, 14$\rangle$.

***Step 2***. The framework prepares a baseline dataset $BD$ using the special dataset construction mechanism described in Subsection 3.4. The baseline dataset contains a collection of user profiles with their features. Each user profile in the baseline dataset is manually labeled as being a spammer or legitimate profile based on their characteristics. Our framework constructs a baseline dataset that contain 300 spammers and 700 legitimate profiles with their feature values.

***Step 3***. The framework supplies the feature values for user profile A and the baseline dataset $BD$ to the IDSS. Here, we used Weka software [24] in the IDSS to provide the *decisionmaker* function using the elimination method based BN classifier. Weka returns the probability set [0.5, 0.7], where, 0.5 and 0.7 are the calibrated probabilities of the legitimate and spammer class, respectively, for the profile A. The IDSS compares these two probabilities that returns the profile A is a spammer as being ($0.5 < 0.7$).

## 4. Experiment and evaluation

In order to evaluate the performance of the proposed framework, we deployed it in a real time environment and conducted different experiments. The details and results of our experiments are described below.

### *4.1. Evaluation methodology*

The experimental evaluation was performed in Weka [26], which is a data mining tool that evaluates a classification model using numerous machines learning classifiers. Here, we evaluated the performance of the SpamSpotter framework using different classifiers in Weka. To carry out our evaluation, we applied a 10-fold cross-validation technique [27] over the constructed baseline dataset, and we took into account the two classes of Facebook user profiles - spammer and legitimate. The aim of this technique is to forecast and approximate how accurate the proposed framework will be worked in practice. The 10-fold cross validation technique splits the dataset into 10 random subsets of equal size, out of which nine subsets are used for training and the other for testing. The same procedure is repeated until all 10 subsets have been utilized as the testing set exactly once. The results reported are an average of the results of all ten runs and are described using a confusion matrix, as shown in Table 3, where, *tp*: true positive shows the quantity of

Table 3: Confusion matrix

| Actual | Classified | |
|---|---|---|
| | Spammer | Legitimate |
| Spammer | tp | fn |
| Legitimate | fp | tn |

spammer profiles correctly classified as spammer profiles, $fp$: false positive shows the quantity of legitimate profiles incorrectly classified as spammer ones, $fn$: false negative indicates the quantity of spammer profiles incorrectly classified as legitimate ones, and $tn$: true negative stands for the quantity of legitimate profiles correctly classified as legitimate ones.

We used the confusion matrix as a metric to evaluate the performance of the SpamSpotter framework. We used the standard evaluation measures given below.

(a) Accuracy: the proportion of correctly classified profiles in the total classified profiles.

$$Accuracy = (tp + tn)/(tp + fn + fp + tn) \tag{4.1}$$

(b) Positive Predictive Value (PPV): the fraction of the correctly classified spammer profiles in the profiles classified as spammer.

$$PPV = tp/(tp + fp) \tag{4.2}$$

(c) Sensitivity: the fraction of the correctly classified spammer profiles in the total amount of spammer profiles.

$$Sensitivity = tp/(tp + fn) \tag{4.3}$$

(d) F-score: the harmonic mean of PPV and Sensitivity.

$$F - score = 2tp/(2tp + fp + fn) \tag{4.4}$$

(e) Mathew Correlation Coefficient (MCC): the estimator of the correlation between the classified results and the actual results.

$$MCC = (tp * tn - fp * fn)/\sqrt{(tp + fn)(tp + fp)(tn + fp)(tn + fn)} \tag{4.5}$$

(f) The Area-Under-the-Curve (AUC): stands for the area under the Receiver Operating Characteristic (ROC) curve [25]. The ROC shows the ratio of true positive samples with the false positive samples. This ratio is more approach to 1, and then the classifier is more efficient.

*4.2. Performance evaluation of the SpamSpotter framework*

In this subsection, we present a thorough evaluation of our framework using the evaluation methodology presented above. For conducting our evaluation, we employed the following eight different machine learning classifiers: Bayesian Network (BN) [24], Random Forest (RF) [28], Decorate (DE) [29], Decision Tree (J48) [30], Jrip (JR) [31], k-Nearest Neighbors (kNN) [32], Support Vector Machine (SVM) [25], and Multinomial Ridge Logistic Regression (LR) [33]. We used elimination method (proposed in Subsection 3.5) with BN classifier and the libSVM algorithm [25, 34, 35] to train the SVM classifier. The gamma ($\gamma$) and cost ($C$) parameters of libSVM were boosted using a grid search algorithm along with a 10-fold cross validation technique. In the grid search, 10-fold cross validation over the baseline dataset for the different pair of $(\gamma, C)$ was performed and the best pair with the optimal cross validation accuracy was selected. In our experiment, for selecting different pairs of $(\gamma, C)$, we kept the range of cost $C$ between 20 to 50 and range of gamma $\gamma$ between 0.1 to 0.9. Similarly, the ridge penalizing parameter of the LR model and the k parameter of the kNN classifier were adjusted using a cross-validation parameter selection algorithm. We built and trained all eight classifiers using our baseline dataset constructed in Subsection 3.4. Then, we applied 10-fold cross validation in order to evaluate the performances of each trained classifier. Table 4 summarizes the performance results of all classifiers in terms of standard evaluation measures as previously described. The highest value of each measure is in bold.

As clearly seen in Table 4, it can be observed that all of the classifiers have an excellent classification capability. In particular, BN, RF, and J48 achieved an AUC as high as 0.989. The BN classifier achieved the highest performance in terms of accuracy, PPV, F-score, MCC, and AUC. There is a slight difference of around 0.01 in accuracy for the classifiers BN, RF, and DE. Furthermore, the RF classifier obtained the highest sensitivity and is slightly similar to the sensitivity achieved by the DE classifier. Similarly, the JR classifier obtained the lowest performance in terms of accuracy, F-score, and MCC.

Table 4: Performance of SpamSpotter framework using eight different machine learning classifiers

| Algorithm | | Evaluation measure | | | | | |
|---|---|---|---|---|---|---|---|
| | | Accuracy | PPV | Sensitivity | F-score | MCC | AUC |
| BN | Bayesian Network | **0.984** | **0.987** | 0.980 | **0.984** | **0.977** | **0.989** |
| RF | Random Forest | 0.983 | 0.983 | **0.983** | 0.983 | **0.977** | **0.989** |
| DE | Decorate | 0.982 | 0.981 | 0.982 | 0.982 | 0.973 | 0.983 |
| J48 | Decision Tree | 0.977 | 0.978 | 0.977 | 0.977 | 0.965 | **0.989** |
| JR | Jrip | 0.950 | 0.955 | 0.944 | 0.950 | 0.911 | 0.981 |
| kNN | k-Nearest Neighbors | 0.961 | 0.953 | 0.969 | 0.961 | 0.931 | 0.980 |
| LR | Logistic Regression | 0.976 | 0.975 | 0.978 | 0.976 | 0.962 | 0.986 |
| SVM | Support Vector Machine | 0.977 | 0.973 | 0.981 | 0.977 | 0.964 | 0.977 |

*4.3. Comparison with existing work*

In order to validate the effectiveness of our framework on detecting spammer profiles, we employed all eight classifiers over two different datasets build using the feature set in [8] and [9]. Since, we did not have the actual accumulated datasets from the authors of [8] and [9]. Therefore, we reproduced their effort (datasets) by using the diverse set of features proposed in their corresponding papers. Table 5 summarizes the results. The highest values for each measure are in bold.

The following observation can be made using Table 4 and Table 5.

(1) All the classifiers built using our feature set have significantly higher performance, when compared with the other ones built with the feature sets of [8] and [9]. In particular, BN, RF and J48 attained an AUC as high as 0.989. However, the performance using our features set slightly surpasses or somewhat matches with that of using feature set of [9]. This is because some features in our set is similar to the features in the set of [9]. Thus, from these comparisons of our work with the works of [8] and [9], we have analyzed that our proposed features are more effective for spammer detection in Facebook.

Table 5: Performance of classifiers based on the feature set by existing work

| Algorithm | | Evaluation measure | | | | | |
|---|---|---|---|---|---|---|---|
| | | Accuracy | PPV | Sensitivity | F-score | MCC | AUC |
| *Performance of classifiers over the dataset that we build using the feature set in Gianluca Stringhini et al. [8]* | | | | | | | |
| BN | Bayesian Network | **0.891** | 0.891 | 0.891 | **0.891** | **0.883** | 0.898 |
| RF | Random Forest | **0.891** | 0.888 | **0.893** | **0.891** | **0.883** | 0.898 |
| DE | Decorate | 0.890 | 0.891 | 0.889 | 0.890 | 0.880 | 0.897 |
| J48 | Decision Tree | 0.888 | 0.889 | 0.887 | 0.888 | 0.874 | **0.899** |
| JR | Jrip | 0.876 | **0.894** | 0.858 | 0.876 | 0.836 | 0.897 |
| kNN | k-Nearest Neighbors | 0.866 | 0.866 | 0.866 | 0.866 | 0.832 | 0.883 |
| LR | Logistic Regression | 0.869 | 0.866 | 0.873 | 0.869 | 0.839 | 0.896 |
| SVM | Support Vector Machine | 0.889 | 0.885 | **0.893** | 0.889 | 0.876 | 0.889 |
| *Performance of classifiers over the dataset that we build using the feature set in Faraz Ahmed et al. [9]* | | | | | | | |
| BN | Bayesian Network | 0.971 | 0.973 | **0.969** | **0.971** | **0.951** | **0.985** |
| RF | Random Forest | **0.972** | **0.974** | **0.969** | **0.971** | **0.951** | 0.983 |
| DE | Decorate | 0.969 | **0.974** | 0.964 | 0.969 | 0.943 | 0.975 |
| J48 | Decision Tree | 0.958 | 0.955 | 0.960 | 0.958 | 0.928 | **0.985** |
| JR | Jrip | 0.943 | 0.943 | 0.943 | 0.943 | 0.897 | 0.975 |
| kNN | k-Nearest Neighbors | 0.944 | 0.951 | 0.936 | 0.943 | 0.897 | 0.964 |
| LR | Logistic Regression | 0.917 | 0.911 | 0.925 | 0.918 | 0.845 | 0.964 |
| SVM | Support Vector Machine | 0.949 | 0.957 | 0.940 | 0.948 | 0.907 | 0.948 |

(2) In addition, with the all three feature sets, the LR, kNN and JR classifiers had the worse performance. However, the SVM classifier attained remarkable performance, especially with the feature set from [8]. SVM achieved only a slightly worse performance than BN, RF, and DE, and better than J48, which scored extremely high when evaluated with the AUC measure.

(3) Finally, BN and RF were the most efficient and consistent classifiers in terms of overall performance.

### *4.4. Global Significance of our features set*

Motivated by the excellent performance achieved by our feature set, we went further and evaluated the global significance of it for the detection of spammers on Facebook. In order to assess the significance of the single feature among all the features, we used the $\chi^2$ test available on Weka. The final result of the test is shown in Table 6, which summarizes all of the features with their rank of significance. It can be easily observed form Table 6 that the two high-ranked features are a) the average number of tags in a post, and b) the total number of community. The higher significance of these two features can be explained from a general point of view. Usually, on Facebook, spammers frequently exploit the tagging feature because they can easily spread malicious contents to a large audience by tagging multiple people and pages in a single malicious post. Meanwhile, the intention of a legitimate user is to utilize the tagging feature for spreading the legitimate post and information with known users. Therefore, they only tag selected users in a single post. Like the tagging feature, spammers can join large number of communities (liked pages, events, groups). After joining communities, spammers can post spam messages on these pages and infect huge amounts of people with less effort.

### *4.5. Discussion and analysis of features*

After providing the evaluation of our framework and feature set, in this subsection, we then give our analyzation of the dissimilarity between legitimate users and spammers on the basis of top 10 highly ranked features selected from Table 6. For analysis, we randomly selected 500 user profiles comprised of 150 spammer profiles and 350 legitimate profiles from our baseline dataset. We labeled each selected profile with

Table 6: Significance rank of all features in our feature set

| | Feature | Rank |
|---|---|---|
| $f_6^{(c)}$ | Average number of tags in a post | 1 |
| $f_3^{(p)}$ | Total number of community | 2 |
| $f_6^{(p)}$ | Average number of posts shared per day | 3 |
| $f_8^{(p)}$ | Average number of URLs shared per day | 4 |
| $f_{10}^{(p)}$ | Average number of photos/videos shared per day | 5 |
| $f_2^{(c)}$ | Fraction of the posts containing URLs | 6 |
| $f_2^{(p)}$ | Number of followings | 7 |
| $f_3^{(c)}$ | Fraction of the posts containing photos/videos | 8 |
| $f_1^{(p)}$ | Number of friends | 9 |
| $f_9^{(c)}$ | Average number of hashtags present in a post | 10 |
| $f_4^{(c)}$ | Average number of comments per post | 11 |
| $f_5^{(c)}$ | Average number of likes per post | 12 |
| $f_5^{(p)}$ | Total number of posts shared | 13 |
| $f_9^{(p)}$ | Total number of photos/videos shared | 14 |
| $f_7^{(p)}$ | Total number of URLs shared | 15 |
| $f_4^{(p)}$ | The age of the user account | 16 |
| $f_8^{(c)}$ | Maximum number of links present in a post | 17 |
| $f_1^{(c)}$ | Fraction of the posts containing spam words | 18 |
| $f_7^{(c)}$ | Average rate of URLs repetition | 19 |

an arbitrary number (identity value) ranging from 1 to 500. Fig. 3 and Fig. 4 depicts the variation of each selected feature with respect to both spammer and legitimate user profiles.

Fig. 3a depicts the average number of tags in a post. According to Facebook policy, a Facebook user can tag up to 50 people or pages in a photo. Therefore, we utilized 50 as a maximum value. As we observed, the spammers depict the higher rate of tagging because spammers abuse the tagging feature of Facebook and attempt to tag high quantity of user in a single post.

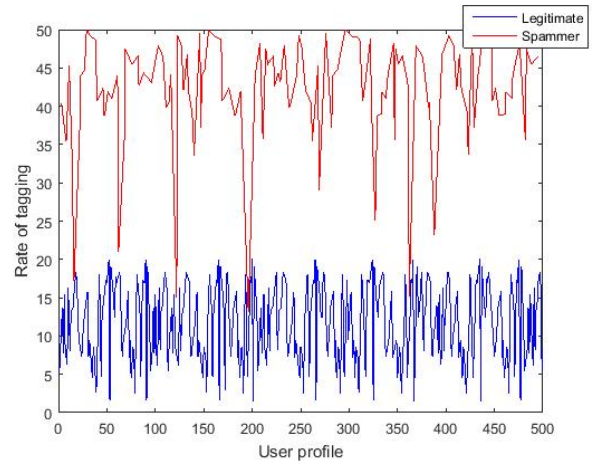
Fig. 3b shows that the community joining rate of spammers is higher as compared to legitimate users. Usually, joining a community is an efficient way for a spammer to transmit spam data more effectively and attract a large audience. A spammer can share a single spam post on community pages and can easily infect all of the members of that particular community.

Moreover, as shown in Fig. 3c that the post sharing rate for spammers is nearly three to five times greater than legitimate users. Generally, spammers utilize the posting feature for spreading malicious or fake information on Facebook and infect a large number of Facebook users in a short period of time. While, legitimate users are mostly less active on Facebook and utilize the posting feature less. Typically, a spammer behaves like an automatic machine that automatically and frequently shares posts on Facebook.

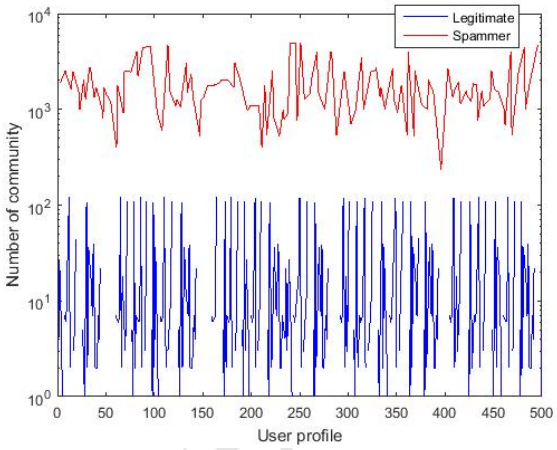
We analyzed that average URL sharing rate of spammers is high as compared with legitimate users, as shown in Fig. 3d. Spammers usually share a large number of spam links that contain the address of malicious websites for spreading their malicious behaviors in a short amount of time, whereas legitimate users share a small number of URLs that link to the latest news, funny videos, famous articles, and many more.

The photos/videos sharing rate is approximately three to five times higher than the legitimate users as shown in Fig. 3e. Spammers frequently share phony videos/photos on Facebook to trick other users and a large audience. While, legitimate users spend less amount of time for sharing photos/videos.

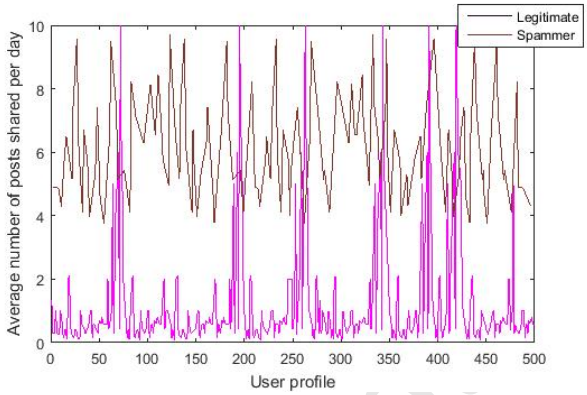
We evaluated the Cumulative Distribution Function (CDF) of the fraction of posts containing URLs ($f_2^{(c)}$) with respect to spammers and legitimate users which is shown in Fig. 3f. We found that spammers share a much larger amount of posts with URLs than legitimate users. As per our observations from Fig.



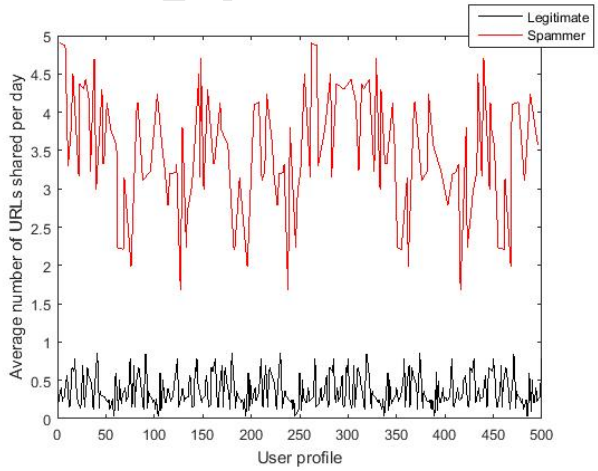

(a) $f_6^{(c)}$



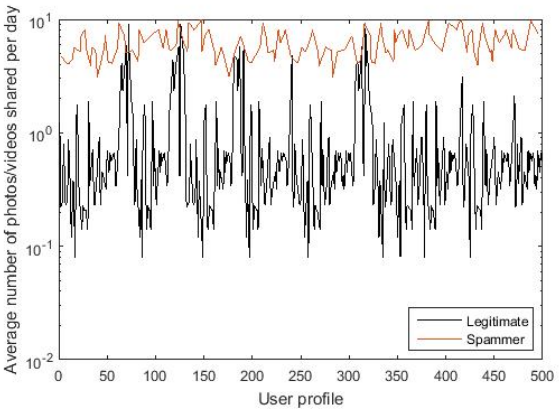

(b) $f_3^{(p)}$



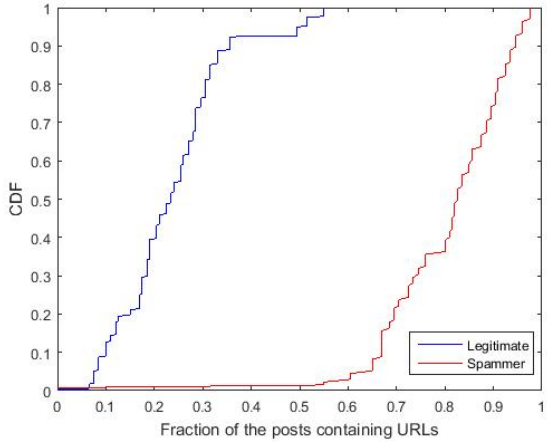

(c) $f_6^{(p)}$




(d) $f_8^{(p)}$

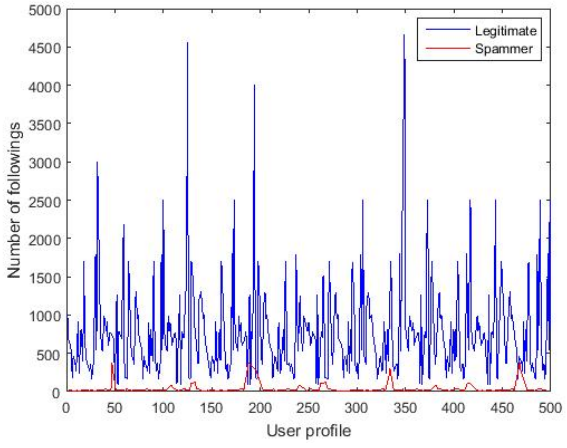



(e) $f_{10}^{(p)}$

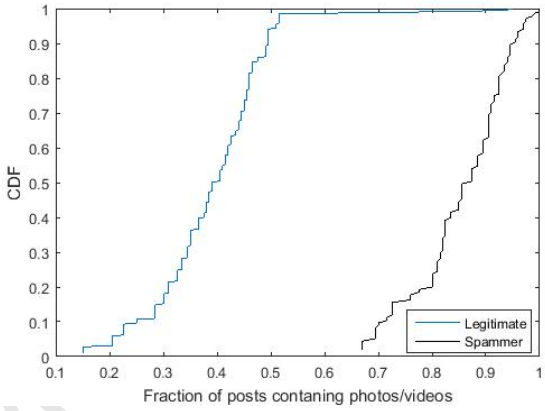



(f) $f_2^{(c)}$

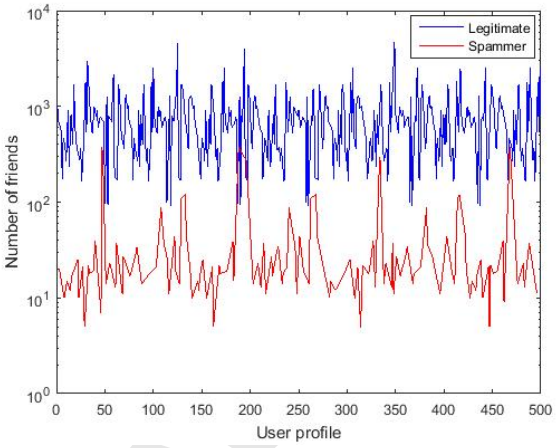
Figure 3: Analysis of dissimilarity between legitimate users and spammers



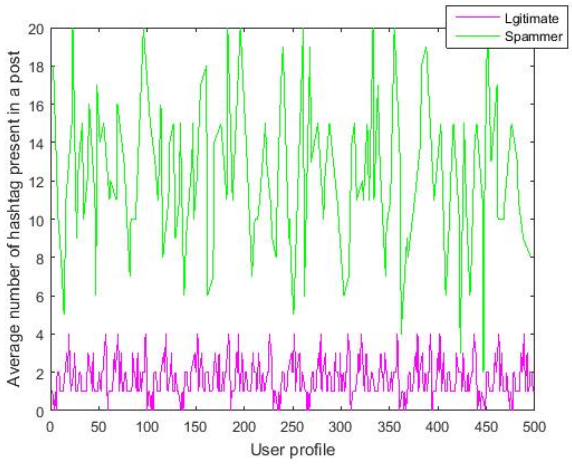

(a) $f_2^{(p)}$




(b) $f_3^{(c)}$




(c) $f_1^{(p)}$




(d) $f_9^{(c)}$

Figure 4: Analysis of dissimilarity between legitimate users and spammers

3f, more than three-fourths posts by each spammer account contain URLs.

Fig. 4a depicts the number of followings of spammer profiles is significantly high, while legitimate profiles have very low number of followings. As we observed that spammers try to follow a bulk amount of legitimate users on Facebook in order to their malicious behavior to a large audience. However, not all spammers follow a large amount of users, instead only 60% of them do that. The reason is that Facebook allows its users to tag or mention any other user in their posts and comments. In other words, spammers do not need to follow legitimate users to get their attention. They can simply tag someone in their posts or they also mention someone in their comments by using the @username format. These posts or comments will appear on that user's Timeline whose username is tagged or mentioned. In this manner, spammers can send out spam posts to legitimate users instead of following them.

Furthermore, we examined that spammer share high amount of posts that contain photos/videos. Usually, spammers use attractive photos/videos to lure normal users/audiences towards to their spam posts. As seen in Fig. 4b, nearly 65% of the posts of each spammer contain photos/videos.

Table 7: Existing research works, their research gaps and comparison with our framework

| **Existing research work** | **Research gap** | **Our framework** |
|---|---|---|
| Jin et al. [15] | This system uses a large set of features to identify spam posts in SNSs such as Facebook. To store this feature set, we require a large database. Therefore, implementation of this system includes some issues related to computer vision, data mining and database. | Our framework identifies spammers rather than spam posts and our feature set are not large. We use only selected and highly effective features. Therefore, it does not have the issue of database to store the huge dataset. |
| Stringhini et al. [8] | This approach uses a special method namely social honeypot for accumulating dataset. However, it needs an inactive observation of long duration to build a satisfactory dataset. Another disadvantage of this approach is that the resulting dataset often biased (it contains only spammers that are actively following other users). | While our proposed framework uses a crawler-based feature extraction and dataset construction mechanism to construct baseline dataset. This mechanism does not require a long duration of inactive observation and resulting dataset is unbiased (contain considerable amount of both spam as well as legitimate profiles). |
| Goa et al. [7], Stringhini et al. [8], Ahmed et al. [9] | The feature sets used by these approaches are inadequate and not updated to detect complete spammers on Facebook. | In our framework, we analyze recent characteristics of Facebook and propose a novel set of feature that includes several new features as well as features from existing work. Therefore, our framework is efficient to detect possible recent spam campaigns including spam campaign due to advertising and self-promotional. |

As shown in Fig. 4c, the variation in the number of friends of spammer profiles is lower as compared to legitimate profiles. Usually, on Facebook, legitimate users interact with a large portion of their friends and friends of friends Therefore, they have a higher probability of accept friend requests and have many legitimate friends. However, spammers usually send a large number of friend requests to other users for gaining the chance to spam, but not all users accept these friend requests due to numerous reasons, such as

they have a low number of mutual friends or they are unknown to the other users.

We also observed that spammers utilize the hashtags too frequently in their posts. As illustrated in Fig. 4d, the average number of hashtags for each spammer profile is approximately above 4. On the other hand, this value for each legitimate user profile is below 4. Usually legitimate users do not frequently use hashtags in their posts.

Finally, we compared our framework with the existing research in the area of spam detection on Facebook. Table 7 summarizes these existing studies and their research gaps, and provides a comparison with our framework.

## 5. Conclusion

In this paper, we studied spam activities in the one of the most popular SNSs called Facebook. This study has made three new contributions to the field of spam detection on Facebook. First, we introduced a set of novel profile and content-based features to the task of spammer detection on Facebook. Second, we developed a special dataset construction mechanism that constructs a baseline dataset of Facebook user profiles. Finally, we proposed a SpamSpotter framework that relies on the IDSS approach implemented by using the machine learning classifier on the baseline dataset to distinguish spammers from legitimate users. Our evaluation results show that the framework achieved excellent performances with 98.4% accuracy, 97.7% MCC, and a 98.4% F-Score. Our findings suggest that our proposed profile and content-based features enhance spammer detection on Facebook. The framework can be integrated with Facebook as a detection module that identifies spammers by gathering and analyzing information from Facebook.

We foresee two directions in which our proposed framework can evolve. First, we intend to discover other enhancements in the IDSS that we used in our framework, such as the use of a deep learning algorithm to implement an IDSS. Second, we intend to construct our baseline dataset for multiple SNSs and to develop a common framework across multiple SNSs.

## Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2017-0-00213, Development of Cyber Self Mutation Technologies for Proactive Cyber Defense)

## Reference

[1] D. H. Lee, Personalizing information using users' online social networks: A case study of citeulike., JIPS 11 (1) (2015) 1–21.

[2] S. S. Ninawe, P. Venkataram, A method of designing a generic actor model for a professional social network, Human-centric computing and information sciences 5 (1) (2015) 25.

[3] J. Wu, F. Chiclana, E. Herrera-Viedma, Trust based consensus model for social network in an incomplete linguistic information context, Applied Soft Computing 35 (2015) 827–839.

[4] Zephoria digital marketing, the top 20 valuable facebook statistics updated january 2017, `https://zephoria.com/top-15-valuable-facebook-statistics/`, (accessed 01.01.2017).

[5] H. Xu, W. Sun, A. Javaid, Efficient spam detection across online social networks, in: Big Data Analysis (ICBDA), 2016 IEEE International Conference on, IEEE, 2016, pp. 1–6.

[6] Nexgate, research report 2013 state of social media spam, `http://nexgate.com/wp-content/uploads/2013/09/Nexgate-2013-State-of-Social-Media-Spam-Research-Report.pdf`, (accessed 02.01.2016).

[7] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, B. Y. Zhao, Detecting and characterizing social spam campaigns, in: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, ACM, 2010, pp. 35–47.

[8] G. Stringhini, C. Kruegel, G. Vigna, Detecting spammers on social networks, in: Proceedings of the 26th annual computer security applications conference, ACM, 2010, pp. 1–9.

[9] F. Ahmed, M. Abulaish, A generic statistical approach for spam detection in online social networks, Computer Communications 36 (10) (2013) 1120–1129.

[10] V. Kacholia, A. Garg, D. Stoutamire, Spam detection for user-generated multimedia items based on concept clustering, uS Patent 9,208,157 (Dec. 8 2015).

[11] L. Liu, Y. Lu, Y. Luo, R. Zhang, L. Itti, J. Lu, Detecting" smart" spammers on social network: A topic model approach, arXiv preprint arXiv:1604.08504.

[12] C. Chen, Y. Wang, J. Zhang, Y. Xiang, W. Zhou, G. Min, Statistical features based real-time detection of drifted twitter spam, IEEE Transactions on Information Forensics and Security.

[13] R. K. Roul, S. R. Asthana, M. Shah, D. Parikh, Detecting spam web pages using content and link-based techniques, Sadhana 41 (2) (2016) 193–202.

[14] M. Agrawal, R. L. Velusamy, Unsupervised spam detection in hyves using salsa, in: Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA) 2015, Springer, 2016, pp. 517–526.

[15] X. Jin, C. Lin, J. Luo, J. Han, A data mining-based spam detection system for social media networks, Proceedings of the VLDB Endowment 4 (12) (2011) 1458–1461.

[16] H. Gao, Y. Chen, K. Lee, D. Palsetia, A. N. Choudhary, Towards online spam filtering in social networks., in: NDSS, Vol. 12, 2012, pp. 1–16.

[17] Y. Zhu, X. Wang, E. Zhong, N. N. Liu, H. Li, Q. Yang, Discovering spammers in social networks, in: Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012, pp. 1–7.

[18] X. Hu, J. Tang, Y. Zhang, H. Liu, Social spammer detection in microblogging., in: IJCAI, Vol. 13, Citeseer, 2013, pp. 2633–2639.

[19] D. Wang, D. Irani, C. Pu, Spade: a social-spam analytics and detection framework, Social Network Analysis and Mining 4 (1) (2014) 1–18.

[20] Govtnaukries, you wont ever use head and shoulder shampoo after watching this video facebook spam, `http://www.govtnaukries.com/you-wont-ever-use-head-and-shoulder-shampoo-after-watching-this-video-facebook-spam/`, (accessed: 05.01.2017).

[21] C. Wisniewski, What is "likejacking"?, `https://www.sophos.com/en-us/security-news-trends/security-trends/what-is-likejacking.aspx`, (accessed: 05.01.2017).

[22] R. Hiscott, The beginner's guide to the hashtag, `http://mashable.com/2013/10/08/what-is-hashtag/#i0IsnwOnYPqB`, (accessed: 06.01.2017).

[23] Facebook for developers, graph api explorer, `https://developers.facebook.com/tools/explorer`, (accessed: 06.01.2017).

[24] I. Rish, An empirical study of the naive bayes classifier, in: IJCAI 2001 workshop on empirical methods in artificial intelligence, Vol. 3, IBM New York, 2001, pp. 41–46.

[25] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology (TIST) 2 (3) (2011) 27.

[26] Weka 3: Data mining software in java, `http://www.cs.waikato.ac.nz/ml/weka/`, (accessed: 06.01.2017).

[27] I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, Data Mining: Practical machine learning tools and techniques, Morgan Kaufmann, 2016.

[28] A. Liaw, M. Wiener, et al., Classification and regression by randomforest, R news 2 (3) (2002) 18–22.

[29] P. Melville, R. J. Mooney, Creating diversity in ensembles using artificial data, Information Fusion 6 (1) (2005) 99–111.

[30] J. R. Quinlan, C4. 5: Programming for machine learning, Morgan Kauffmann 38.

[31] Class jrip, http://weka.sourceforge.net/doc.stable/weka/classifiers/rules/JRip.html.

[32] L. E. Peterson, K-nearest neighbor, Scholarpedia 4 (2) (2009) 1883.

[33] S. Le Cessie, J. C. Van Houwelingen, Ridge estimators in logistic regression, Applied statistics (1992) 191–201.

[34] C. Chantrapornchai, P. Nusawat, Two machine learning models for mobile phone battery discharge rate prediction based on usage patterns, Journal of information processing systems 12 (3) (2016) 436–454.

[35] R. Malhotra, A systematic review of machine learning techniques for software fault prediction, Applied Soft Computing 27 (2015) 504–518.