Detecting spam accounts on Twitter

Zulfikar Alom
DiSTA, University of Insubria
Varese, Italy
mzalom@uninsubria.it

Barbara Carminati
DiSTA, University of Insubria
Varese, Italy
barbara.carminati@uninsubria.it

Elena Ferrari
DiSTA, University of Insubria
Varese, Italy
elena.ferrari@uninsubria.it

***Abstract*—Social networks have become a popular way for internet surfers to interact with friends and family members, reading news, and also discuss events. Users spend more time on well-known social platforms (e.g., Facebook, Twitter, etc.) storing and sharing their personal information. This information together with the opportunity of contacting thousands of users attract the interest of malicious users. They exploit the implicit trust relationships between users in order to achieve their malicious aims, for example, create malicious links within the posts/tweets, spread fake news, send out unsolicited messages to legitimate users, etc. In this paper, we investigate the nature of spam users on Twitter with the goal to improve existing spam detection mechanisms. For detecting Twitter spammers, we make use of several new features, which are more effective and robust than existing used features (e.g., number of followings/followers, etc.). We evaluated the proposed set of features by exploiting very popular machine learning classification algorithms, namely k-Nearest Neighbor (k-NN), Decision Tree (DT), Naive Bayesian (NB), Random Forest (RF), Logistic Regression (LR), Support Vector Machine (SVM), and eXtreme Gradient Boosting (XG-Boost). The performance of these classifiers are evaluated and compared based on different evaluation metrics. We compared the performance of our proposed approach with four latest state of art approaches. The experimental results show that the proposed set of features gives better performance than existing state of art approaches.**

***Index Terms*—Spam detection, Twitter, Machine learning, Social network security.**

I. Introduction

In the last few years, online social networks (e.g., Facebook, Twitter, and LinkedIn) have become one of the major way for internet surfers to communicate with friends, express opinions, and talk about any events. Twitter, a microblogging service launched in 2006, is one of the most popular online social network, where users post messages of around 140 characters, known as *"tweet"*. Twitter has 330 million active users that post about 500 million tweets every single day [1]. This huge popularity attracts the attention of spammers who use Twitter for malicious aims, including spreading malicious URLs within tweets, spreading rumors, sending unsolicited message to other users.

According to Twitter privacy policy [2], a Twitter account having a high number of followings but low number of followers is considered a spam account. Indeed, followers of an account reflect the popularity and reputation of that account. Most of the proposals presented in literature have designed spammer detection mechanisms based only on account-related features [3], [4] like: number of followers/followings, number of tweets, etc. These features have been used for calculating several scores, for instance: the *fifo score*, representing the ratio of the number of an account's followings to its followers; the *reputation score*, computed as the ratio of the number of an account's followers over the sum of its followers and followings. However, Twitter spammers can avert these spam detection mechanisms by buying followers through third-party marketplace [5], making these features not always effective [6]. This motivates us to design new and more robust features to detect Twitter spammers.

To achieve this goal, we have considered both graph-based features (i.e., triangle count of user's network, the ratio of triangle count to the number of followers of a user, and the ratio of bi-directional links) and content-based features (i.e., unique URL ratio, URL to tweet ratio, average tweets per day of a user, and average likes per tweet of a user). To assess our detection method, we selected, from the popular social honeypot dataset [3], 325 Twitter accounts, where 168 are considered legitimate users and 157 spam users. We have used several machine learning classification algorithms for distinguishing between spammers and non-spammers accounts. Through the experiments, we show that the proposed set of feature gives better performance than existing state of art approaches. Moreover, our results show that, among all considered classifiers, Random Forest classifier gives the better performance. By using this classifier, our suggested features can achieve 92% precision and 91% F1-score.

In summary, the paper contains the following main contributions:

- we design a set of novel graph-based and content-based features that have been proved to be powerful for spam account detection on Twitter;
- we use seven machine learning algorithms, namely: k-Nearest Neighbor (k-NN), Decision Tree (DT), Naive Bayesian (NB), Random Forest (RF), Logistic Regression (LR), Support Vector Machine (SVM), and eXtreme Gradient Boosting (XGBoost) to classify spam and legitimate users;
- we compare the proposed set of features with [7], [8], [9], and [10] showing that our features provide better accuracy;
- by using the feature ranking method (i.e., information

IEEE/ACM ASONAM 2018, August 28-31, 2018, Barcelona, Spain
978-1-5386-6051-5/18/$31.00 © 2018 IEEE

gain), we ranked the top 10 most influencing detection features among all the features used by state of the art approaches.

The remainder of this paper is organized as follows. Section II presents related work. Section III explains the novel graph-based and content-based features that are proposed to detect spammers. Section IV explains the spam detection models. Section V illustrates the experimental results, and, finally, Section VI concludes the paper.

II. Related Work

In the literature, many papers addressed the problem of spammers detection on Twitter along with other social networking websites. In the following, we introduce the proposals for detecting spammers' profiles and spam tweets, which are based on user's profile and behaviours.

The majority of previous studies [3], [7], [8], [11] are based on user-behavior and tweet content-based spam detection. Lee et al. [3] used 10 machine learning classifiers and two different data sets for detecting spammers on Twitter. Ameen et al. [7] used four machine learning classification algorithms and 13 content-based attributes. Similarly, Ala'M et al. [8] used four machine learning classifiers and some of the most common user-based and content-based features for detecting spammers on Twitter. Benevento et al. [11] compared two approaches for detecting spam profiles and spam tweets. Initially, they built their model to identify spam profiles based on user-based features. They considered 23 user-based attributes, such as number of followers/followings, number of tweets, age of an account, etc. By using the SVM classifier their work achieves 84.5% accuracy. Then, they used both user-based and content-based features to classify the tweets into spam and non-spam categories, by achieving 87.6% accuracy. Hai Wang et al. [12] used a social graph model, by leveraging on the followings and followers relationships. They extracted from Twitter around $25K$ users, and 20 recent tweets for each user, along with $49M$ friends/followers relationship. To assess the detection method, they used four different classifiers (i.e., DT, NN, SVM, NB) to classify users into spammers and legitimate users. In the experiments, Naive Bayesian classifier gives the best performance: 91.7% precision and 91.7% F1-score. Wang et al. [13] focused on spam tweets detection rather than detecting spammers accounts. They used two hand-labeled data sets (i.e., Social honeypot and 1KS-10KN) and four feature sets, i.e., user-based, content-based, n-gram, and sentiment features.

Some hybrid approach, like Herzallah et al. [10], used user behaviours, graph-based and tweet content-based features to detect spammers on Twitter. They used popular machine learning algorithms for classifying users into spammers and non-spammers categories. Sing et al. [9] used three feature sets, namely trust score, content-based and user-based features and four machine learning classification algorithms for classifying the users into spammers and non-spammers.

As summary, Table I lists the used features, classifiers, data sets and the results of the approaches described so far.

One key issue is that spammers normally evade user-based detection features by obtaining more followers. Since, spammers repeat posting with malicious URLs for attracting legitimate users to visit particular sites, so their tweets show strong homogeneous characteristics. For this reason, many researchers used tweet content-based features (i.e., tweet similarity, duplicate tweet count, etc.) to detect spam accounts. Although, using simple techniques, such as posting heterogeneous tweets and act as normal users, spammers can avoid these detection techniques. On the other hand, we propose graph-based features, which are based on social relationships in the real world scenario, for instance, legitimate users usually follow accounts whose owners are their close friends, relatives or family members, as such these accounts are likely to have a relationship together and build a triangle in their networks. Likewise, they follow each other and increase number of bi-directional links between them. However, spammers can change their tweets by changing words but they cannot change the URLs since they want users to visit a particular website. Moreover, spammers intend to post more tweets with URLs than legitimate users, so we have considered URLs to tweet ratio and average tweet per day features. We also proposed average likes per tweet, spammers usually posted more tweets but they could not get response i.e., likes from users, so this feature may be useful for the detection process. More information about our proposed features are given in the following section III.

III. Features

In this section, we present the proposed set of features, consisting of three graph-based and four content-based features.

Graph- based features. Twitter allows users to build their own social graph. A social graph represents the following and follower relationships among users. From the social graph of a target user u, we extract three graph-based features: (1) triangle count of user u's network, (2) the ratio of triangle count to number of followers of u, and (3) the ratio of bi-directional links from the users' social graph.

Triangle count of user u's network: We compute the total number of triangles of u's network, $(Triangle\_Count(u))$. In the social network, a triangle exists if a user/node has two adjacent nodes which, in turn, are also adjacent to each other. To find out spammer and non-spammer users, we make use of this feature because legitimate users usually follow accounts whose owners are their close friends, colleagues, or family [6], [14], as such these accounts are likely to have a relationship together. Therefore, an high number of triangles implies that the user is legitimate. On the other hand, spammers usually blindly follow other accounts, these accounts do not know each other and have a lower relationship among them. Thus,

TABLE I: Summary of the reviewed Twitter spam detection papers

Ref. ID	Features	Classifiers	Dataset	Results
[11]	User-based Tweet content-based	SVM	1065 users' profiles 355 spammers 710 legitimate users	Accuracy with only user attributes : 84.5% both user and content attributes: 87.6%
[3]	User-based Tweet content-based	Decorate, LogitBoost HyperPipes, Bagging RandomSubSpace, BFTree, FT SimpleLogistic, LibSVM Classification Via Regression	500 users' profiles 168 spammers 332 legitimate users	Decorate classifier gives the best accuracy: 88.98%
[12]	Graph-based Tweet content-based	DT, Neural Network SVM, NB	500 users' profiles 3% spam users' account	NB gives the best performance Precision: 91.7% Recall: 91.7% F1-score: 91.7%
[13]	User-based Tweet content-based n-gram Sentiment	NB, k-NN, SVM DT, RF	2 hand labeled data sets Social Honeypot Dataset 1KS-10KN Dataset	RF gives the best performance with F1-score: 94%
[8]	Tweet content-based User-based	DT Multilayer Perception k-NN, NB	82 users' profiles	NB gives the highest accuracy: 95.7%
[7]	User-based Tweet content-based	NB, J48 RF, IBK	1183 users' profiles 355 spammers 828 legitimate users	RF gives the best performance with accuracy: 92.95%
[10]	Graph-based User-based Tweet content-based	NB, SVM MLP, k-NN, AD Tree J48, RF	210 users' profiles 100 spammers 110 legitimate users	k-NN gives the best performance with accuracy: 99.05% Precision: 98.33% Recall: 100% F1-score: 99.13%
[9]	Trust-based Tweet content-based User-based	Bayes Net, Logistic, J48 RF, AdaBoostM1	19581 users' profiles 11059 spammers 8522 legitimate users	RF gives the best performance with accuracy: 92.1%

compared to legitimate users, spam users will have a smaller number of triangles.

The ratio of u's triangles to number of u's followers: To evade the *triangle_ count* feature, spammers could create many fake accounts so as to form triangles, by building links among these fake accounts. Moreover, spammers can purchase followers, thus sometimes the number of followers of spammers is greater than the one of legitimate users. Thus, this feature $(RateTNF(u) = \frac{Triangle\_Count(u)}{N_{fer}(u)})$, where, $N_{fer}(u)$ refers to the number of followers of user u, can help us to detect spam users, even if spam users generate fake triangles in their social networks.

The ratio of bi-directional links of u: When any two users' accounts follow each other, we refer to this as a bi-directional link. The number of bi-directional links of an account reflects the reciprocity between an account holder and its followings. In general, spammers follow huge amount of legitimate users, but they cannot force them to follow back, thus their number of bi-directional links will probably be low. On the other hand, legitimate users usually follow their family members, friends, and co-workers who will follow them back. It means that the number of bi-directional links of legitimate users will be high. So, we can use this feature for distinguishing between spammers and legitimate users. We define ratio of bi-directional links as follows: $Rate_{bilink}(u) = \frac{N_{bilink}(u)}{N_{fer}(u)+N_{fing}(u)}$, where $N_{bilink}(u)$ refers to the number of followings of a user u which follow him back, whereas $N_{fer}(u)$ is the number of followers of user u, and $N_{fing}(u)$ is the number of followings of user u.

Content-based features. These features are properties related to text of tweets. In previous work, many researchers used content-based features i.e., duplicate tweets, suspicious words, repeated words, tweet time patterns, for detecting spammers. Thus, we consider four new content-based features that can isolate spammers from non-spammers, namely: 1) Unique URL ratio, 2) URL to tweet ratio, 3) Average tweets per day of user u, and 4) Average likes per tweet of user u.

Unique URL ratio: A way to gain money from spammer activities is to force legitimate users to visit a particular site. As such, spammers post the same URL several times. $URate_{url}(u)$ is the ratio of the number of unique URLs posted by user u to the number of total URLs posted by him/her. A higher $URate_{url}(u)$ means that user u is a legitimate user. Similarly, the lower $URate_{url}(u)$ is, the higher is the chance of being a spammer account. We define unique URL ratio as follows : $URate_{url}(u) = \frac{N_{unique\_URLs}(u)}{N_{all\_URLs}(u)}$.

URL to tweets ratio: Spammers post a huge number of URLs compared to legitimate users. This feature $Rate_{url\_tweet}(u)$ defines the ratio of number of URLs posted by a user u to the number of tweets posted by him/her. A high value of this feature means that user u is a spam user. Likewise, a lower value means a higher chance of being a legitimate user.

URL to tweets ratio is defined as follows : $Rate_{url\_tweet}(u) = \frac{N_{all\_URLs}(u)}{N_{tweets}(u)}$.

Average tweets per day of u: This feature refers to the ratio of the number of tweets posted by a user u to the age of an account (days). More precisely, $Avg_{tweet}(u) = \frac{N_{tweets}(u)}{Age(u)}$. For making money or spreading fake news, spammers tend to post more tweets than legitimate users. Thus, a higher value of $Avg_{tweet}(u)$ means that user u is likely to be a spam user, whereas a lower value of $Avg_{tweet}(u)$ means an higher chance of being a legitimate user.

Average likes per tweet of u: This feature defines the ratio of the number of likes of user u's tweets over the number of tweets posted by u. It is expressed by $Avg_{likes}(u) = \frac{N_{likes}(u)}{N_{tweets}(u)}$. Since, spam users do not get more likes for their tweets, so a high value of $Avg_{likes}(u)$ means user u is a legitimate user, whereas a lower value means user u is a spam user.

IV. Spam Detection Models

There are many classification algorithms that have been already used to detect spam users in online social networks. In order to evaluate the effectiveness of our proposed features in identifying spam users, we used seven classification models, namely, Naive Bayes, k-NN, Decision Tree, Random Forest, Logistic Regression, SVM, and XGBoost, briefly described in what follows [15].

Naive Bayesian (NB). The Naive Bayesian classifier is based on the probability theory (i.e., Bayes theorem) [15]. This model is widely used because it gives good performance and requires less computational time for training the model. The main assumption of this algorithm is that the features of a dataset are independent, it means that the probability of one attribute does not affect the probability of the other. However, let us consider C represents the class (i.e., spammer or non-spammer) and D defines a Twitter user's profile, which may belong to class spammer or non-spammer. The Naive Bayesian classifier, which is based on Bayes theorem, can be described as follows [16] : $P(C|D) = \frac{P(D|C)*P(C)}{P(D)}$, where, $P(C)$ and $P(D)$ are the probability of C and D respectively. These are called the prior probability and their values can be computed from the training data. $P(D|C)$ is called the conditional probability, which means the probability of D given that C happens. $P(C|D)$ means the probability of C given that D happens, it is called the posterior probability. Due to the page limitation, we do not provide more details about conditional probability and the statistical model behind NB, by referring the interested readers to [15], [16].

k-Nearest Neighbor (k-NN). k-Nearest Neighbor classifier is a very simple supervised learning algorithm, which stores all available instances and classifies new instances based on the similarity measure (e.g., distance functions). The instances are classified by majority vote of their neighbors, the instances being assigned to the class which is the most common amongst its k nearest neighbors. If k = 1, then the instance is simply assigned to the class of its nearest neighbor [17].

Decision Tree (DT). The Decision tree is an extension of ID3 algorithm which uses Entropy and Information Gain to construct a decision tree [18]. It is a very powerful and widely used classifier because it is very simple and gives good results by using less memory space than other algorithms. The major disadvantage of this algorithm is that it takes long time for training the classification model. Generally, Decision tree classification method is divided into two phases: tree building and tree pruning [16]. In the tree building phase, it recursively partitions a dataset using depth-first greedy approach or breadth-first approach until all the data items belong to the same class label. In the tree pruning phase, it works for improving the classification accuracy by minimizing overfitting problem. The Decision tree consists of root, internal and leaf nodes, where the root node is the topmost node of decision tree, internal node corresponds to a test condition on an attribute, branch corresponds to results of the test conditions, and a leaf node corresponds to a class label.

Random Forest (RF). Random Forest is a very flexible and easy to use machine learning classifier that consists of a collection of tree structured classifiers [19]. It randomly selects the features to construct a collection of decision trees. It is one of the most widely used algorithms, because of its simplicity and it can be used for both classification and regression tasks.

Logistic Regression (LR). Logistic regression is an extension of simple regression method. In logistic regression, the output of linear regression is passed through the activation function. Softmax function can be used as an activation function, it is a very popular function to calculate the probabilities of the events. It can be defined as follows: $\sigma(z)_j = \frac{\exp^{z_j}}{\sum_{k=1}^{K}\exp^{z_k}}\ for\ j = 1, 2, .., K$, where z is a vector of the inputs to the output layer (since we have 2 output class i.e., spam and non-spam, so there are 2 elements in z) and j is the index of output units. The output values of this function is always in the range [0,1], and the sum of the output values is equal to 1. Generally, LR is intended for binary classification, so that we classify input to class 1 when the output of this function is closed to 1 (i.e., output > 0.5) and classify to class 2 when the output is closed to 0 (i.e., output ≤ 0.5) [20].

Support Vector Machine (SVM). Support Vector Machine is probably one of the most popular machine learning algorithm. It performs classification by finding the hyperplane that maximizes the margin between the two classes. The vectors which define the hyperplane are called the support vectors. An ideal SVM produces a hyperplane which completely separates the vectors into two non-overlapping classes. However, perfect separation may not be possible, so in this case, SVM finds the hyperplane which maximizes the margin and minimizes the mis-classifications [21].

TABLE II: Characteristics of the dataset

Property	Value
$Number\ of\ Twitter\ accounts$	325
$Number\ of\ followings$	21676
$Number\ of\ followers$	5039
$Number\ of\ tweets$	6500
$Number\ of\ extracted\ URLs$	2506
$Number\ of\ unique\ URLs\ extracted$	1346
$Number\ of\ triangles$	5037

eXtreme Gradient Boosting (XGBoost). XGBoost has become a widely used and popular machine learning algorithm. It is an implementation of gradient boosted decision trees. More particularly, it is an ensemble method which sequentially adds predictors and corrects the previous models. However, instead of assigning weights to the classifiers after every iteration, this method fits the new model to new residuals of the previous prediction and then minimizes the loss [22].

V. Experiments and Results

In this section, we present the results of experimental carried out to show the effectiveness of the proposed set of features.

A. Data Collection

In order to build our model, we need a dataset of Twitter users classified as spammers and legitimate users. For this reason, we used the Twitter Social Honeypot dataset [3] in which users have been already classified as spammers and legitimate users based on tweet content, user behavioral and topological features. The authors created and manipulated 60 social honeypot accounts on Twitter to attract spammers. Thereafter, they used Expectation-Maximization (EM) clustering algorithm and then manually grouped their harvested users into spammers and legitimate users. The dataset consists of $41,499$ user accounts, with pre-classified accounts of $22,223$ spammers and $19,276$ legitimate users that were captured during an eight month period in 2010. In this dataset, most of the users do not have their list of followings/followers; hence values of their interaction and novel graph-based features (i.e., triangle count, bi-directional links) will be zero, which forces classifiers to be biased for spamming detection. Therefore, we consider only those users who have a complete list of followers and followings. Moreover, we also excluded those users who posted tweets in non-English languages. We randomly selected 325 seed Twitter accounts including 168 legitimate users and 157 spammer users' profile from this dataset. Since the dataset is quite old, we have manually checked 20% of collected dataset and found that all checked data are still labeled correctly. To extract the followers and followings relationship among these seed users, a web crawler was developed based on Twitter API [23]. In addition, we collected 20 most recent tweets with the URLs. The basic characteristics of the dataset are shown in Table II.

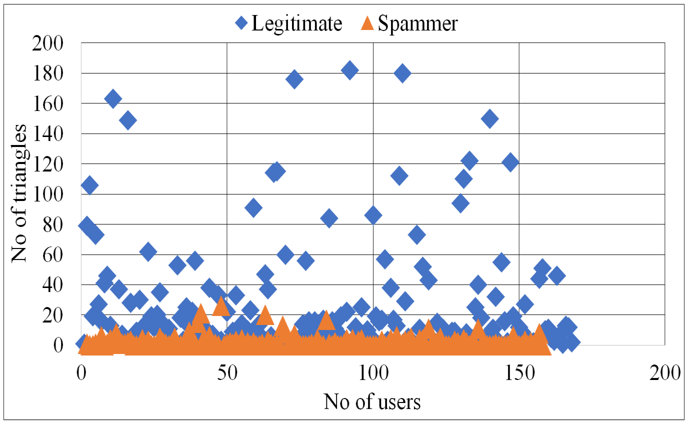

Fig. 1: Number of triangles

		Predicted class	
		Spammer	Non-spammer
True class	Spammer	a	b
	Non-spammer	c	d

TABLE III: Confusion matrix

B. Evaluation Metrics

In the evaluation, we consider the confusion matrix illustrated in Table III, where a means the number of spammers that have been correctly classified, b represents the number of spammers which are misclassified as non-spammers, c expresses the number of non spammers which are misclassified as spammers, and d refers to the number of non-spammers that have been correctly classified. We used four widely adopted machine learning metrics, that is: accuracy, precision, recall, and F1-score.

Accuracy (A) is ratio of the total number of correctly classified instances of both classes over the total number of all instances in the dataset and is expressed by: $A = \frac{(a+d)}{(a+b+c+d)}$. Precision (P) refers to the ratio of the number of correctly classified instances to the total number of instances and is expressed by: $P = \frac{a}{(a+c)}$. Recall (R) defines the ratio of the number of instances correctly classified to the total number of predicted instances and is expressed by: $R = \frac{a}{(a+b)}$. Finally, F1-score (F1) is measured as a weighted average of the precision and recall, and is defined as: $F1 = \frac{2P*R}{(P+R)}$.

C. Data Analysis

In this section, we analyze the collected dataset. As we can see from Fig. 1, showing the characteristics of graph-based features, the number of triangles for legitimate users is higher than those for spammers. Likewise, from Fig. 2, we see that the ratio of the number of triangles to number of followers for legitimate users is higher than for spammer users. Fig. 3 shows that the number of bi-directional links of each account which reflects reciprocity between user accounts is higher for legitimate users than for spammer users.

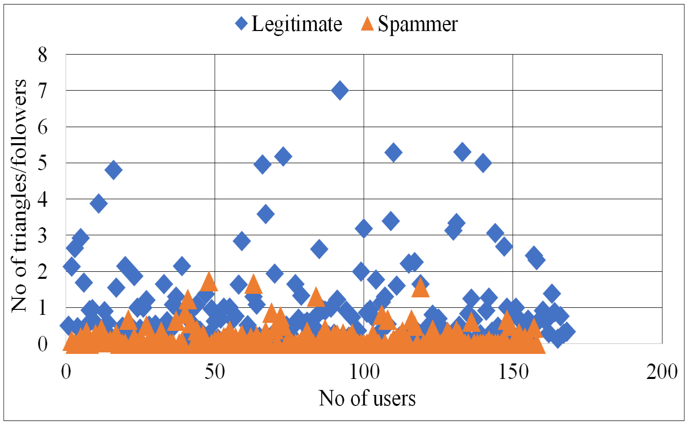

Fig. 2: Number of triangles/followers

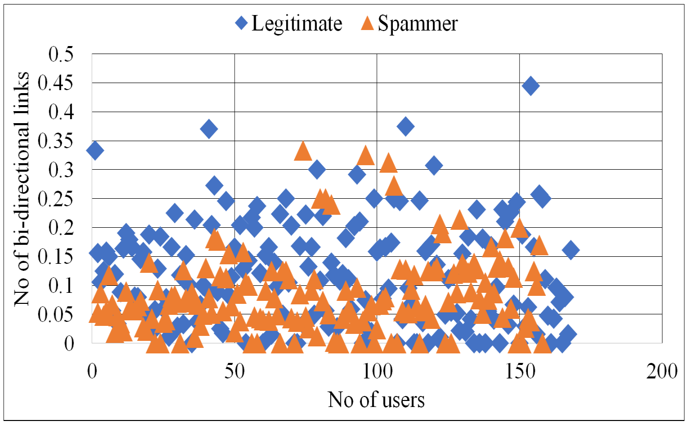

Fig. 3: Number of bi-directional links

Fig. 4 - 5 show the differences between the considered content-based features of spammers and legitimate users. From Fig. 4, we see that legitimate users tend to post a unique link in their tweets. As we expected, spammer users tend to post more links in their tweets than legitimate users, but from Fig. 5, we see that the number of unique links does is almost the same for non-spammers and spammer users.

D. Evaluation

We evaluate our proposed approach through performance comparison and feature ranking, by using different machine learning tools.

Performance comparison: In this experiment, we compare the performance of our approach (E) with four existing state of art approaches, namely: (A) [7], which used 12 features; (B) [8], which used 10 features; (C) [9], which used 10 features; and (D) [10], which used 17 features.

We selected these four approaches for comparison because these are the latest published state of the art approaches for spam detection on Twitter, and it was possible to extract all of the features they considered from our dataset. We conducted our evaluation by using seven different machine learning classifiers [24], namely: k-NN, DT, NB, RF, LR, SVM, and XGBoost. For each machine learning classifier, we compute four performance metrics: accuracy, precision, recall, and F1-score.

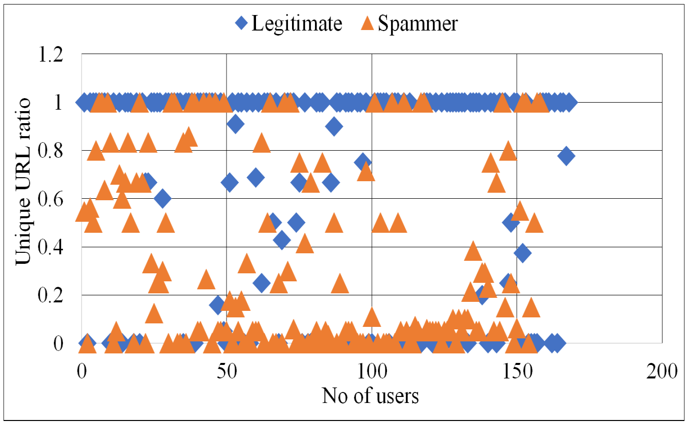

Fig. 4: Unique URL ratio

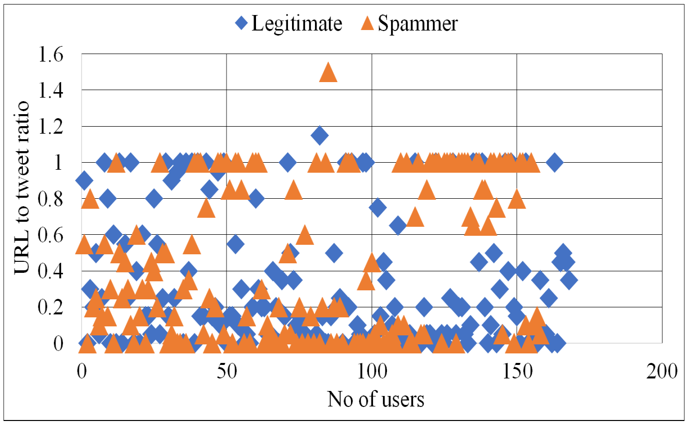

Fig. 5: URL to tweets ratio

As shown in Fig. 6 - 9, our proposed approach outperforms every considered approaches. More particularly, from Fig. 6, we can see that the accuracy of our approach (i.e., RF of E) is greater than the others. It reaches highest accuracy of 91% for RF classifier and lowest accuracy of 74% for NB. Likewise, from Fig. 7, we can see that the precision of our approach is greater than the other approaches. It achieves highest precision value of 92% for both RF and XGBoost classifiers. Especially, under the NB the precision value of our approach is 0.04% lower than approach D and 0.01% lower than approach A. Similarly, for SVM the precision value of our approach is 0.08% lower than approach C. On the other hand, we can see that the precision of the other five machine learning classifiers are the highest. In the same way, from Fig. 8, we can see that the recall value of E is lower than approach C and D under SVM and Naive Bayes respectively, whereas the recall value of the other five machine learning classifiers are the highest.

From Fig. 9, we can see that the F1-score of our work

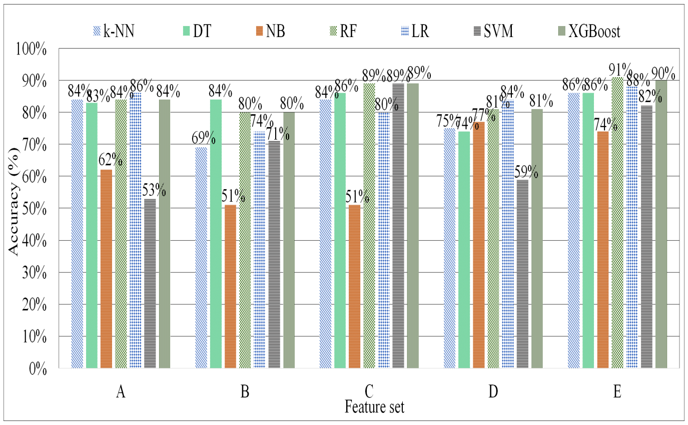

Fig. 6: Accuracy

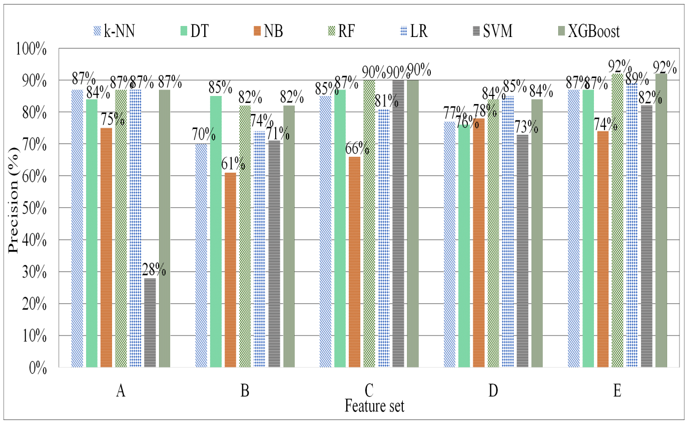

Fig. 7: Precision

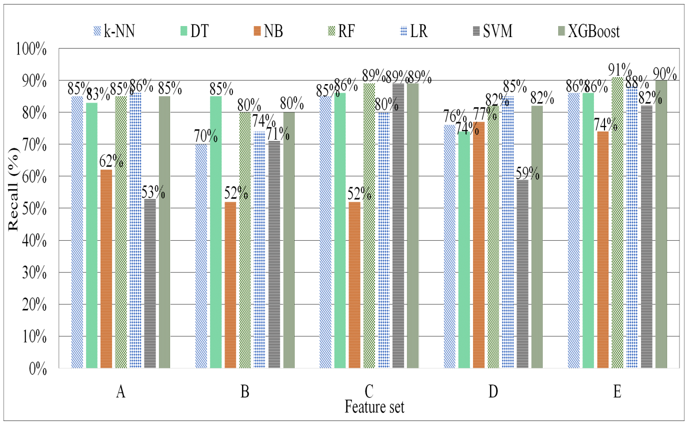

Fig. 8: Recall

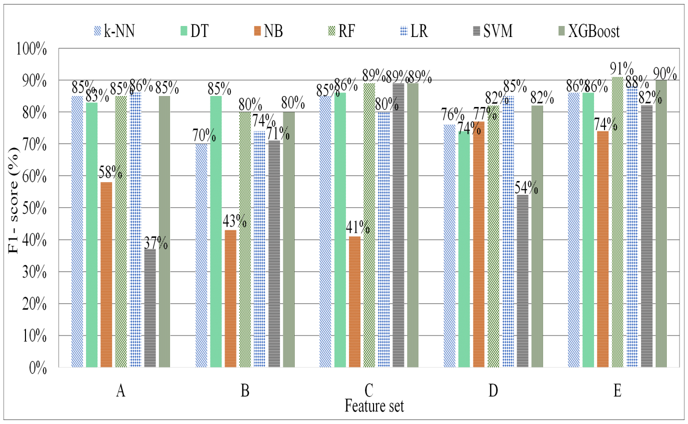

Fig. 9: F1-score

under all machine learning classifiers is also the highest. More particularly, the highest F1-score of our approach is 91% (RF in E), and the lowest F1-score of our approach is 74% (NB in E). From the above discussion, we see that our new feature set is more effective to detect Twitter spammers than other existing approaches.

Feature ranking: In order to verify the importance of the considered features, we used feature selection method. It is also known as variable selection or attribute selection, which is the process of selecting relevant features in terms of the target learning problem. The purpose of feature selection is to remove redundant and irrelevant features because these features can reduce the learning accuracy and the quality of the model. However, we used information gain feature selection method, that are available on Weka [25]. Weka supports feature selection via information gain using the $Info\_Gain\_Attribute\_Eval$ attribute evaluator. It calculates the information gain (i.e., entropy) for each attribute. This value vary from 0 (i.e., no information) to 1 (i.e., maximum information). The attributes that contribute more information will have a higher information gain value and can be selected, whereas those that do not add much information will have a lower score and can be removed. Information gain can be calculated as follows: $IG(C, P_i) = H(C) - H(C|P_i)$, where C is the output class, P_i and H is the entropy.

The result listed in Table IV indicates the top most 10 important attributes among 55 features. Interestingly, we see that 5 of our features are included among the top 10 important features. The first and third most important attributes in the list are the number of triangles and the number of triangles to number of followers.

Furthermore, we verify the importance of the top 10 features, by measuring the F1-score. We calculated F1-score consider-

TABLE IV: Top 10 features

Rank	Information gain
1	Number of triangles
2	Age of an account (days)
3	Number of triangles to number of followers
4	Number of followers
5	Number of tweets
6	Average tweets per day
7	Unique URL ratio
8	Average likes per tweet
9	Reputation
10	Fifo ratio

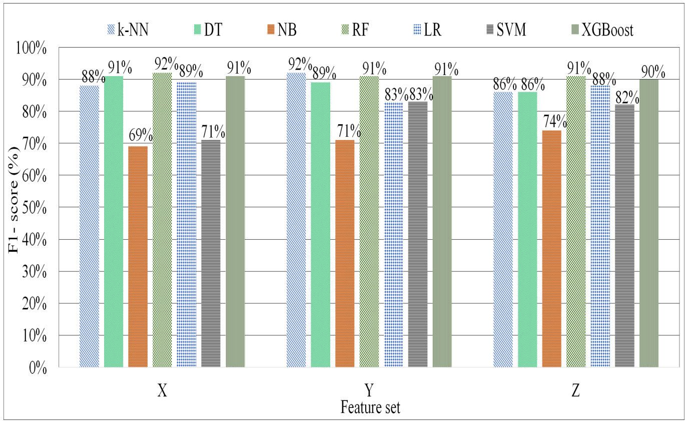

Fig. 10: F1-score

ing: (1) all top 10 attributes, which we labeled as X, (2) 5 of our features that are included among the top 10 ones, which we labeled as Y, and (3) all of our 7 features, which labeled as Z. From Fig. 10, we can see that the highest F1-score is 92% for RF, when we consider all top 10 features (RF in X), whereas we get the highest F1-score of 92% for k-NN classifiers on Y, and 91% for both RF and XGBoost. For Z, we get the highest F1-score of 91% for RF, and the lowest F1-score is around 74% (NB in Z).

According to the experimental results, it can be observed that the top features identified by the feature selection method (i.e., information gain) gives slightly improved performance. We conclude that the features identified by information gain, which are number of triangles, age of an account (days), and number of triangles to number of followers, are very important and higher influencing features in the process of identifying spam users on Twitter.

VI. Conclusion

In this paper, we designed a new and more robust set of features to detect spammers on Twitter. We considered both graph-based and tweet content-based features, and applied them into seven different machine learning algorithms. In the experiment, Random Forest (RF) gives the better result compared to other algorithms, with an accuracy of 91%, precision 92%, and F1-score 91%. Through the performance comparison analysis, we showed that our proposed solution is feasible and is capable to give better results than other existing state of art approaches.

In the future, we plan to build a more effective model which can easily classify various types of spammers within different types of social networks. In addition, we will work on modifying the machine learning algorithms and apply our method to different social networks.

References

[1] "Twitter Usage Statistics - Internet Live Stats (2018)," accessed on 20 April 2018. [Online]. Available: http://www.internetlivestats.com/twitter-statistics/

[2] "Twitter privacy rules," accessed on 20 April 2018. [Online]. Available: https://help.twitter.com/en/rules-and-policies/twitter-rules

[3] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: social honeypots+ machine learning," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2010, pp. 435–442.

[4] A. T. Kabakus and R. Kara, "A survey of spam detection methods on twitter," *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, vol. 8, no. 3, pp. 29–38, 2017.

[5] "Purchase Twitter followers," accessed on 20 April 2018. [Online]. Available: https://www.buyrealmarketing.com/buy-twitter-followers

[6] C. Yang, R. Harkreader, and G. Gu, "Empirical evaluation and new design for fighting evolving twitter spammers," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1280–1293, 2013.

[7] A. K. Ameen and B. Kaya, "Detecting spammers in twitter network," *International Journal of Applied Mathematics, Electronics and Computers*, vol. 5, no. 4, pp. 71–75, 2017.

[8] A.-Z. Ala'M, H. Faris *et al.*, "Spam profile detection in social networks based on public features," in *Information and Communication Systems (ICICS), 2017 8th International Conference on*. IEEE, 2017, pp. 130–135.

[9] M. Singh, D. Bansal, and S. Sofat, "Who is who on twitter–spammer, fake or compromised account? a tool to reveal true identity in real-time," *Cybernetics and Systems*, pp. 1–25, 2018.

[10] W. Herzallah, H. Faris, and O. Adwan, "Feature engineering for detecting spammers on twitter: Modelling and analysis," *Journal of Information Science*, p. 0165551516684296, 2017.

[11] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on twitter," in *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, vol. 6, no. 2010, 2010, p. 12.

[12] A. H. Wang, "Don't follow me: Spam detection in twitter," in *Security and cryptography (SECRYPT), proceedings of the 2010 international conference on*. IEEE, 2010, pp. 1–10.

[13] B. Wang, A. Zubiaga, M. Liakata, and R. Procter, "Making the most of tweet-inherent features for social spam detection on twitter," *arXiv preprint arXiv:1503.07405*, 2015.

[14] C. Yang, R. C. Harkreader, and G. Gu, "Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers," in *International Workshop on Recent Advances in Intrusion Detection*. Springer, 2011, pp. 318–337.

[15] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," 2007.

[16] S. D. Jadhav and H. Channe, "Comparative study of k-nn, naive bayes and decision tree classification techniques," *International Journal of Science and Research*, vol. 5, no. 1, 2016.

[17] "K Nearest Neighbors," accessed on 22 April 2018. [Online]. Available: http://www.saedsayad.com/k_nearest_neighbors.htm

[18] "Decision Tree," accessed on 20 April 2018. [Online]. Available: http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/

[19] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[20] "Logistic Regression," accessed on 20 April 2018. [Online]. Available: http://dataaspirant.com/2017/03/02/how-logistic-regression-model-works/

[21] "Support vector machine," accessed on 20 April 2018. [Online]. Available: http://www.saedsayad.com/support_vector_machine.htm

[22] "eXtreme Gradient Boosting (XGBoost)," accessed on 20 April 2018. [Online]. Available: https://www.kdnuggets.com/2017/10/xgboost-top-machine-learning-method-kaggle-explained.html

[23] "Twitter Developers. Documentation,," accessed on 22 January 2018. [Online]. Available: https://developer.twitter.com/en/docs

[24] "ML algorithms," accessed on 20 April 2018. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms

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