A SEMI-SUPERVISED LEARNING APPROACH FOR TACKLING TWITTER SPAM DRIFT

NIDDAL IMAM

School of Computing, Media and the Arts,
Teesside University, England, UK[*]
niddal21@gmail.com

BIJU ISSAC

Computing and Information Sciences,
Northumbria University, England, UK
bissac@ieee.org

SEIBU MARY JACOB

School of Science, Engineering and Design,
Teesside University, England, UK
s.jacob@tees.ac.uk

Twitter has changed the way people get information by allowing them to express their opinion and comments on the daily tweets. Unfortunately, due to the high popularity of Twitter, it has become very attractive to spammers. Unlike other types of spam, Twitter spam has become a serious issue in the last few years. The large number of users and the high amount of information being shared on Twitter play an important role in accelerating the spread of spam. In order to protect the users, Twitter and the research community have been developing different spam detection systems by applying different machine-learning techniques. However, a recent study showed that the current machine learning-based detection systems are not able to detect spam accurately because spam tweet characteristics vary over time. This issue is called 'Twitter Spam Drift'. In this paper, a semi-supervised learning approach (SSLA) has been proposed to tackle this. The new approach uses the unlabeled data to learn the structure of the domain. Different experiments were performed on English and Arabic datasets to test and evaluate the proposed approach and the results show that the proposed SSLA can reduce the effect of Twitter Spam Drift and outperform the existing techniques.

Keywords: Semi-supervised learning; twitter spam; machine learning; spam drift.

1. Introduction

The Online Social Networks (OSNs) such as Facebook, WhatsApp, and Twitter have become a very important part of our daily lives nowadays. People use them to make friends, communicate with each other, read the news, and share their stories. Twitter, which was founded in 2006, has become one of the most popular microblogging

[*] School of Computing, Media and the Arts, Teesside University, Middlesbrough, England, TS13BX, UK

NOTE: The preprint of an article to be published in International Journal of Computational Intelligence and Applications (IJCIA), Vol 18, Issue 2, 2019, 18 pages;

platforms since then (Chen et al. 2015c). According to Mateen et al., in one month, Twitter has two million users sharing 8.3 million tweets per hour (2017). A Twitter generic profile consists of three components: the account's tweets, followers, and friends. In addition to the account components, there are several Twitter-specific features, such as mentions, hashtags, and retweets (Grier, et al. 2010). Users can only post messages (tweets) up to 140 characters. These tweets can contain text, hashtags, mentions, and shortened URLs.

Unfortunately, due to the high popularity of Twitter, it has become very attractive to spammers. In Twitter, spammers tweet for several goals, such as to spread advertisement, disseminate pornography, spread viruses, phishing, or simply just to compromise a system's reputation. (Benevenuto, et al. 2010). Also, El-Mawass and Alaboodi (2015) added that a tweet is considered spam if it is not composed purely of text. Instead of this, it may contain a hashtag, a mention, a URL or an image. In 2014, Twitter was flooded with spam tweets that were sent by several compromised accounts (Chen et al. 2015b). Recently, several NatWest Bank customers were victims of phishing attacks. Criminals posted spam tweets that looked similar to the NatWest customer support account and directed users to a phishing site (Al-Zoubi, Alqatawna, and Faris 2017). Spammers use trending hashtags to direct users to unrelated topics. Also, spammers use mentions to spread spam tweets. The most important part of spam tweets is the shortened URLs, which enable spammers to deceive users (Miller et al. 2014). Different studies showed that about 5% to 6% of messages in Twitter are spams (Eshraqi, Jalali, and Moattar 2015; Chen et al. 2015a).

Consequently, the research community and Twitter have proposed several spam detection systems to protect users. Twitter has applied rules against spammers or those who behave abnormally. For instance, users who are frequently sending friend requests, sending duplicate content, mentioning others, or posting tweets containing only a URL are considered spammers (Chen et al. 2015c.) Also, Twitter provides different options to its users to report spammers, such as selecting report @username, clicking on the report icon, or clicking on report conversation (*Twitter Help Center* n.d.). However, spammers are using different ways to evade detection by buying followers or mixing spam tweets with normal tweets. This motivates the research communities to develop new, innovative mechanisms (Mateen et al. 2017).

In 2015a, Chen et al. conducted a study by collecting a dataset and analyzing the characteristics of Twitter spammers. The study showed that the current machine learning-based detection systems are not able to detect Twitter spam accurately because spam tweet characteristics vary over time. This issue is called 'Twitter Spam Drift'. The reason behind the Twitter spam drift problem is that, as researchers are developing new spam detection mechanisms, spammers are also trying to avoid these mechanisms (Chen et al. 2017). The problem was introduced first by Chen et al. (2015a). In their study, they found that the average value of spam tweet features varies as the days go on while it is more stable for non-spam tweets.

In this paper, a semi-supervised learning approach (SSLA) has been proposed to tackle the Twitter spam drift where we used unlabeled data to learn the structure of the domain. Different experiments were performed on English and Arabic datasets to test and evaluate the proposed approach and the results show that the proposed SSLA can reduce the effect of Twitter Spam Drift and outperform the existing techniques. The paper is organized as follows. Section 2 is the related works, section 3 is the proposed approach, section 4 is the methodology used, section 5 is the comparative analysis of the results and section 6 is the conclusion.

2. The Related Works

The researchers have been drawn to the spam problem in Twitter since 2010 due to the popularity of the platform (Chris Griery et al. 2010). Different machine learning (ML) techniques were applied to detect spam tweets. In 2016, Lin *et al.* compared and evaluated the detection accuracy, stability, and scalability for 9 machine learning algorithms. The results of the experiment showed that Random Forest and C5.0 outperformed the other algorithms due to their superior detection accuracy. Chen *et al.* (2015a) collected a large dataset and labelled approximately 6.5 million spam tweets. They divided the dataset into balanced and imbalanced sets to study the impact of spam to non-spam ratio. They used seven machine learning algorithms: Random Forest, C4.5, Decision Tree, Bayes Network, Naïve Bayes, K Nearest Neighbor, and Support Vector Machine. They found that when using an imbalanced dataset, which simulated the real-world scenario, the classifiers' ability to detect spam tweets was reduced. On the other hand, when features are discretized, the performance of classifiers improved.

Different from the above works that used supervised and unsupervised techniques for detecting spammers in Twitter, there have been various works that used the semi-supervised technique for detecting different types of spam. Igor Santos *et al.* (2011) used an Local and Global Consistency (LLGC) that was provided by the Semi-Supervised Learning and Collective Classification package to detect unknown malware. The aim of the study was to find the minimum number of labelled instances needed to assure a suitable performance by using LLGC. The result showed that the proposed approach can achieve the best accuracy when a training set size is 65%. Driessens *et al.* proposed an approach that uses a simple two-stage idea that can improve its predictive accuracy by using unlabeled data. The algorithm that was used in this study was Yet Another Two Stage Idea (YATSI), which uses a classification or regression in the first step and weighted the nearest neighbour in the second step (2006).

Most recent works have shown that the detection accuracy of the above-mentioned ML algorithms decrease as time goes on due to the change of the spam tweets' characteristics (Chen et al. 2015a; Chen et al. 2017). They referred to this issue as Twitter Spam Drift. Chen *et al.* were the first to study the Twitter Spam Drift problem and, in their work, they proposed a novel Asymmetric Self-Learning Approach (ASL). The proposed approach has three components: Training Stage, Online Detection, and ASL. The approach was able to reduce the impact of Twitter Spam Drift by enabling the classifier to extract

'changed spam' information from the incoming tweets. Experimental results showed that, when applying the ASL approach, both detection rate and F-measure were improved (Chen et al. 2015a). Moreover, Chen et al. 2017 proposed a new scheme called Learning From Unlabeled tweet (Lfun) to tackle Twitter Spam Drift. Lfun has two components: LDT is to learn from detected spam tweets and LHL is to learn from human labelling. Lfun can detect changed spam tweets by learning from unlabeled tweets and updating the classifier's training process. Experimental results showed that Lfun outperformed the traditional classifiers, such as Random Forest and SVM, and reduced the impact of Twitter Spam Drift.

3. The Proposed Approach

In this paper, the Twitter Spam Drift problem is studied, and a new idea called a semi-supervised learning approach (SSLA) is proposed. The aim of SSLA is to tackle the Twitter Spam drift by using a semi-supervised learning technique that combines labelled data and unlabeled data to create better learners. The current machine learning approaches used for detecting Twitter spam cannot overcome the problem of drifted Twitter spam because the statistical features of spam tweets are changing over time. As a result, the accuracy of the traditional machine learning algorithm is decreasing gradually as time goes on. In order to solve this problem, this paper proposes a semi-supervised learning approach (SSLA). The SSLA is a type of Machine-Learning technique that is useful when the number of labelled instances is limited (Santos, Nieves, and Bringas 2011). The aim of SSLA is to combine labelled and unlabeled data to create better learners. The SSLA has been used to evaluate different applications, such as software fault detection, text classification, spam email detection, quantitative structure-activity modelling, and so forth (Sigdel, et al. 2014). The SSLA was chosen to solve the Twitter Spam Drift problem for several reasons. First, the SSLA uses a combination of labelled and unlabeled data at the same time. The unlabeled data is used by the SSLA to learn the structure of the domain. For example, the unlabeled data helps to capture the underlying distribution of the data (Driessens et al. 2006). Thus, the SSLA does not rely solely on labelled data for classification, which can help when dealing with changed spam. Second, SSLA is more applicable than supervised learning approaches when the amount of unlabeled data is huge (Crawford et al. 2015). This attribute is very important when dealing with changed spam tweets. Third, the SSLA lowers the effort of labelling a large dataset, which is very expensive and time-consuming, while maintaining high accuracy rates (Santos et al. 2011).

There are various semi-supervised techniques that have been reported in the literature. However, generic or wrapper and non-generic are the most common types of semi-supervised learning. Generic or wrapper-based techniques can be formulated on top of any supervised classification techniques, such as self-training and YATSI. On the other hand, non-generic-based techniques take advantage of unlabeled data to improve the learning models, such as Transudative Support Vector Machine (TSVM) and Semi-Supervised Support Vectored Machine (S3VM) (Sigdel et al. 2014). In this study, YATSI

is going to be used. YATSI is a semi-supervised classification algorithm that can be built on top of any supervised classification algorithm and the nearest neighbourhood algorithm. YATSI is introduced by Driessens *et al.* (2006) and Figure 1 presents the details of the YATSI algorithm in pseudo code.

YATSI consists of two stages. In the first stage, an initial prediction model, which generated on the training set and prediction for unlabeled instances are determined by using a supervised classifier. In the second stage, the actual predictions for unlabeled instances are determined by using the nearest neighbourhood algorithm (Sigdel et al. 2014; Saputro, Kusumawardani, and Fauziati 2016). In this study, Random Forest is used as the base classifier and Filtered Neighbor Search as the nearest Neighbor Search algorithm. The Random Forest algorithm was chosen because various studies showed that it can detect spam tweets with high accuracy (Chen et al. 2015a; Lin et al 2016; Meda et al. 2016; Chen et al. 2017). The results from Lin et al (2016) is shown below in table 1 and figure 2. C5.0 and random forest achieved more than 90% accuracy when trained with 200k tweets, where random forest was the most accurate. For the other algorithms like GBM, Naive Bayes, Neural Network and Deep Learning, once the size of training data reaches 20k, there is no substantial increase observed in accuracy.

Algorithm: High level pseudo code for the two-stage YATSI algorithm

Input: a set of labeled data D_l and a set of unlabeled data D_u, an of-the-shelf classifier C and a nearest neighbor number K; let $N = |D_l|$ and $M = |D_u|$

Step 1:
Train the classifier C using D_l to produce the model M_l
Using the model M_l to "pre-label" all the examples from D_u
Assign weights of 1.0 to every example in D_l
and of $F \times (N/M)$ to all examples in D_u
Merge the two sets D_l and D_u into D

Step 2:
For every example that needs a prediction:
Fined the K-nearest neighbors to the example from D to produce set NN
For each class:
Sum the weights of the examples from NN that belong to that class
Predict the class with the largest sum of weights.

Fig. 1. YATSI Algorithm (Driessens et al. 2006)

Table 1. The training and testing datasets with different spam to non-spam ratios (Lin et al., 2016)

	Training Data		Testing Data	
Dataset	No of spam tweets	No of non-spam tweets	No of spam tweets	No of non-spam tweets
1	1000	1000	100000	100000
2	10000	10000	100000	100000
3	100000	100000	100000	100000

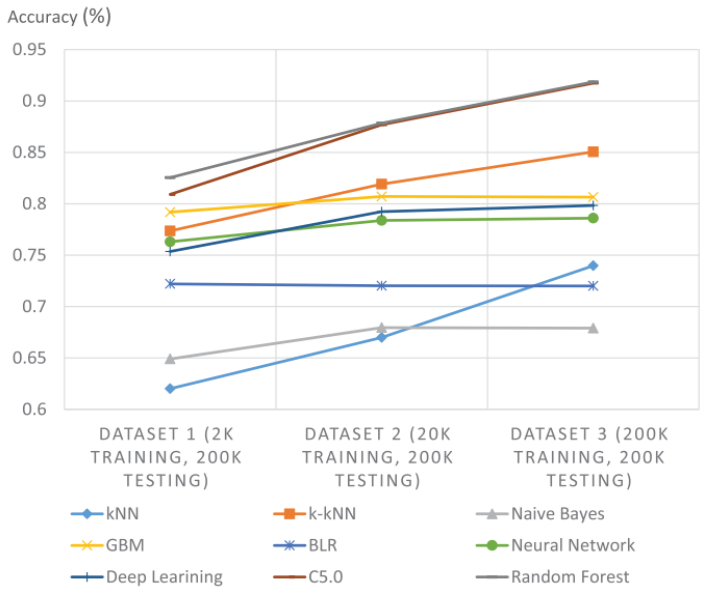

Fig. 2. Detection Accuracy (%) of 9 Algorithms using Dataset 1, 2 and 3 with the ratio of spams and non-spams being 1:1 (Lin et al., 2016)

The nearest Neighbor Search algorithm was chosen because when used with Random Forest, YATSI's accuracy improved as in Table 2. The Day 6 of dataset 1 is used, where details of datasets are explained in the next few sections.

Table 2. The Best Nearest Neighbor Search Algorithm

YATSI	TPR	FPR	Precision	F-Measure
Random Forest + KDTree	76.5	23.5	76.5	76.5
Random Forest + KDTree	76.5	23.5	76.5	76.5
Random Forest + LinearN	76.5	23.5	76.5	76.5
Random Forest + FilteredNeighbourSearch	83.9	16	84	83.9
Random Forest + CoverTree	83.9	16	84	83.9
Random Forest + BallTree	77.9	22	78.1	77.9

4. The Methodology Used

4.1. *Data Collection*

Two datasets were used in this study. The first dataset was built by Chen et al. (2015a), and it was collected in a period of 10 consecutive days. The dataset contains 100K spam tweets and 100K non-spam tweets for each day totaling to 2 million tweets, and it is available online at NSCLab. The second dataset was built by the author of this current study. The process of dataset collection was adopted from different previous studies (Benevenuto et al. 2010; Al Twairesh et al. 2016; Chen et al. 2015a El-Mawass and Alaboodi 2016). The dataset of around 500 tweets was collected by using Twitter Streaming Application Programing Interference (API) in a period of 5 consecutive days. The public streaming API enables accessing 1% of all the public tweets, but those sent by protected accounts or direct messages cannot be accessed (Choeikiwong and Vateekul 2016). The tweets were collected from Arabic trending hashtags in different domains like social, political and entertainment. Table 3 provides the description of the collected dataset.

Table 3. Description of The Second Dataset

Days	Hashtags	Number of Tweets	Spam	Non-Spam
1	#الى_متى_قضايانا_لا_تحل_الا_بترند #ودي_صراحة	108	44	64
2	#يستوطن_قلبي	108	50	58
3	#كيف_ترتيب_عايلتك	103	33	70
4	#ماذا_تطلب_من_وزارة_الصحة	81	25	56
5	#معك_حكي	80	35	45

4.2. *Building and Labelling Dataset*

The authors of the first dataset Chen *et al.* focused on tweets that contained URLs because, after inspecting hundreds of tweets, they found that most of the spam tweets have embedded URLs (2015a). Spammers take advantage of URLs and use them to direct victims to malicious sites. The authors used Trend Micro's Web Reputation Technology (WRT) to check the tweets' URLs. Trend Micro's WRT was chosen by the authors because it maintains a large dataset of URL reputation records that were acquired from their customers. All the URLs were checked, and tweets that contained a malicious URL were defined as spam.

The second dataset, which was built by the author of this study, focused mainly on advertisers in Twitter as spammers. Advertisers or promoters are types of spammers who use Twitter to publicize themselves. These advertisers could be a company, an organization, or an individual (Sinha et al. 2016). Also, some of the individual advertisers use Twitter for selling and buying followers. According to Twitter Rules, any

accounts that sell or buy Twitter usernames may either be temporarily blocked or subject to permanent suspension. Additionally, the Twitter rules state that promoting third-party services or apps to get more followers is considered as a spam activity (*The Twitter Rules* 2017). Another type of spammers who were taken into consideration was Trending Topics spammers who flood hashtags with repeated tweets. The dataset was labelled manually, and accounts that have one of the previously mentioned activities are classified as spam. Although labelling a dataset is time-consuming, it helps in understanding Twitter account characteristics (cited in Al Twairesh et al. 2016). One of the main characteristics of spam tweets in Arabic trending hashtags is that most of the spam tweets are for advertisement. However, no spam tweet that contained a malicious URL was found. That may be because most of the tweets that contain malicious content can be detected more easily by today's machine learning systems.

4.3. *Feature Selection*

Feature selection is a very important step in machine learning-based classification tasks (cited in Chen et al. 2017). The authors of the first dataset Chen *et al.* extracted 12 lightweight features to detect spam tweet that included: account_age, no_follower, no_following, no_user favourites, no_list, no_tweets, no_retweets, no_hashtag, no_user mention, no_URLs, no_char, and no_digts (2015a).

In the second dataset, *Tweepy python* wrappers, which have been used in different papers (Twairesh et al. 2016; Sinha et al. 2016), were used to extract data from Twitter API. A python script was written by using *Tweepy* to collect data and compute features, and the script is available for public use online. The collected tweets were structured by the JavaScript Object Notation (JSON) format, which is a lightweight data-interchange format (*Introducing JSON* n.d). Two types of features were used: account-based features and tweet content-based features, and the total number of features was 13 lightweight statistical features as described in Table 4. Various studies showed that extracting lightweight features is more efficient for timely detection because the longer a spam exists, the more victims it can compromise (Chen et al. 2017; Sinha et al. 2016; Lin et al 2016). Also, Al Twairesh *et al.* used a new feature, which is a phone number (2016). Al Twairesh *et al.*'s study revealed that most of the advertisement tweets contained a phone number. Thus, adding phone numbers as a feature can improve the spam detection. After extracting the features, the file was saved as a Comma Separated Values (CSV) file to be able to construct the dataset properly. CSV format saves data in a table structured format. Then, the dataset was saved as an Attribute Relation File Format (ARFF) to be able to use it in WEKA, which will be described in the following section.

4.4. *Preprocessing*

The first dataset in this study is very large as it contains one million spam tweets and one million non-spam tweets, with 100K spam tweets and 100K non-spam tweets for 10 days. In order to use it in WEKA, the dataset was reduced to 10K spam tweets and 10K non-spam tweets for each day. First, the duplication was removed from the dataset, and each

day's dataset was randomized. Next, each dataset was split into a training set and a testing set for supervised ML approach, and a training set, a testing set, and an unlabeled set for the semi-supervised ML approach.

Similarly, the second dataset in this study was randomized, and the duplication was removed. Each dataset was then split into a training set and a testing set for supervised ML approach and a training set, a testing set, and an unlabeled set for the semi-supervised ML approach. The splitting percentages are discussed in the following section

Table 4. Extracted Features and Feature Description (Lin et al 2016; Sinha et al 2016)

Feature Category	Feature Name	Description
Account-based features	account_age	The number of days since the creation of an account
	no_followers	The number of followers of an account
	no_friends	The number of friends an account has
	no_favorites	The number of favourites an account received
	no_lists	The number of an account is a member of
	no_reputation	The ratio of the number of followers and the sum of followers and friends of an account
	no_statuses	The number of tweets an account has
Tweet content-based features	no_words	The number of words in a tweet
	no_chars	The number of characters in a tweet
	no_hashs	The number of hashtags in a tweet
	no_urls	The number of URLs in a tweet
	no_phone	The number of phone numbers in a tweet
	no_mentions	The number of mentions in a tweet

4.5. *Experiments and Evaluations*

In this section, the experimental procedure that was followed to verify the efficiency of the proposed approach in tackling the Twitter spam drift problem is provided. Different experiments were performed. First, the performance of 4 supervised ML algorithms was evaluated and the most accurate one was used for the rest of the experiments. Second, each dataset was split into different percentages and the split that gave the best result was used for the rest of the experiment in both approaches. Finally, to confirm the result, the proposed approach was compared with the supervised ML approach.

So, in summary, two different benchmarks have been used in this study: a supervised ML approach and a semi-supervised ML approach. Two different datasets were used. The first dataset was made up of 200K tweets for 10 consecutive days – 10K spam tweets and 10K non-spam tweets for each day. It is described by 12 lightweight features. The second dataset contained 500 tweets for 5 consecutive days with 100 tweets each day and is described by 13 lightweight features.

The chosen metrics for evaluating the performance of the different experiments are as follows: the true positive rate (TPR), the false positive rate (FPR), the precision, and the F-measure. The TPR is the ratio of spams that were correctly identified by the total number of actual spams. The FPR is the proportion of non-spams that were incorrectly identified as spams in the total number of actual non-spams. The precision is the ratio of

correctly classified spams to the total number of tweets that were identified as spams. The F-measure is a measurement of prediction accuracy that combines both the precision and recall (Lin et al 2016).

4.6. *Supervised Learning Approach*

To evaluate the performance of the classifier when dealing with Drifted Twitter Spam, WEKA which is an open software for data analysis and data mining was used (cited in Goyal, Chauhan, and Parveen 2016). WEKA provides a collection of machine learning algorithms and data preprocessing tools. It has been funded by the New Zealand government since 1993 (Hall et al. 2009). WEKA has been used often in the literature for analyzing Twitter spam (Goyal, Chauhan, and Parveen 2016; El-Mawass, and Alaboodi 2016; Al Twairesh et al. 2016).

In the study, several experiments were done with Chen *et al*. (2015) first dataset to choose the best supervised ML algorithm that was going to be used in the final comparison. First, the dataset was evaluated by using four popular supervised ML algorithms: LibSVM, Bayes Network, J48, and Random Forest. Day 1 dataset with 10K spam tweets and 10K non-spam tweets was chosen at random, and the dataset was split into 80% training data and 20% test data. Figure 3 shows that Random Forest outperforms the other classifiers in all metrics. Thus, Random Forest was chosen to be used for the supervised ML approach.

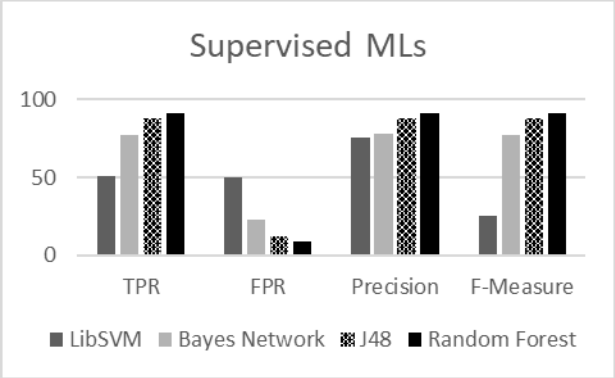

Fig. 3. Supervised ML Algorithms

Table 5. Dataset Split Percentages (Supervised ML Approach)

Dataset Split	Classifier	TPR	FPR	Precision	F-Measure
60:40	Random Forest	90.4	9.6	90.5	90.4
70:30	Random Forest	90.5	9.5	90.7	90.5
80:20	Random Forest	91.3	8.7	91.5	91.3

Second, an experiment was carried out to find the best split percentage for the dataset. The Day 1 of dataset 1 was chosen again for this experiment, and the dataset was split into three groups: 60:40, 70:30, and 80:20. Table 5 shows that when splitting the dataset into 80% training dataset and 20% test dataset, the classifier (Random Forest) performance improved. Finally, the Random Forest, and 80% training dataset and 20% test dataset were chosen to be used for the Twitter Spam Drift final experiment.

4.7. *Semi-Supervised Learning Approach*

Like the supervised ML approach, several experiments were done with Chen *et al.* (2015a) first dataset to choose the best semi-supervised ML algorithm that was going to be used in the final comparison. First, the YATSI algorithm was chosen to be used for the semi-supervised ML approach. However, as YATSI does not specify any specific algorithm to be used, the experiment was run by using four supervised ML algorithms LibSVM, Bayes Network, J48, and Random Forest. Day 1 dataset was chosen at random, and the dataset was split into 70% unlabeled dataset, 15% training dataset and 15% test dataset. Figure 4 shows that Random Forest obtained the best results compared to the other classifiers. Thus, Random Forest was chosen to be used for the semi-supervised ML approach because it has been proven in the literature that it can detect spam more accurately than other algorithms (Chen et al. 2015a; Lin et al 2016; Meda et al. 2016) as in figure 2.

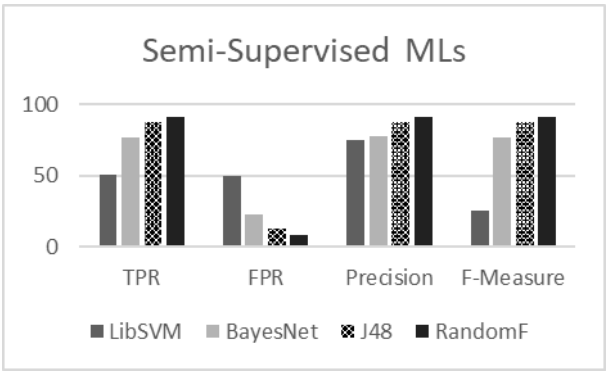

Fig. 4. Semi-Supervised YATSI algorithm and Four Supervised ML Algorithms as Base Classifiers

Table 6. Dataset Split Percentages (Semi-Supervised ML Approach)

Dataset Split	YATSI	TPR	FPR	Precision	F-Measure
30:70	Random Forest	84.8	15.2	86.8	84.8
40:60	Random Forest	83.7	16.3	83.7	83.7
20:80	Random Forest	82.2	17.9	82.2	82.1

Second, the experiment was run to find the best split percentage for the dataset. The Day 1 of dataset 1 was chosen for this experiment, and the dataset was randomly split into three groups: 30:70, 60:40, and 80:20. Table 6 shows that, when splitting the dataset into 30% training and test dataset and 70% unlabeled dataset, YATSI performance improved. Finally, the base classifier, Random Forest, and 30% training and test dataset and 70% unlabeled dataset were chosen to be used for the Twitter Spam drift final experiment.

5. Comparative Analysis of Results and Discussion

In this section, the prediction accuracy of the proposed approach, SSLA, was compared with the prediction accuracy of the chosen supervised learning approach to tackle the Twitter Spam Drift problem. Several experiments were performed by using both datasets. First, the Day 1 dataset performance was evaluated by using Day 1 training data and testing data from Day 1 to Day 10 for both approaches. Figure 5 presents the experimental results in terms of the F-measure. We can see that the F-measure of SSLA is more stable, with prediction accuracy above 80%, whereas the F-measure of the supervised learning approach is significantly decreasing. Especially on Day 9, the F-measure drops from 90% to below 60%. Second, the Day 6 dataset performance was evaluated by using Day 6 for training data and testing data from Day 1 to Day 10 for both approaches. Figure 6 demonstrates the second experimental results in terms of F-measure. We can see that the prediction accuracy of SSLA remained above 80% in all 10 days, unlike the prediction accuracy of supervised learning approach, which fluctuated between about 85% and 60%.

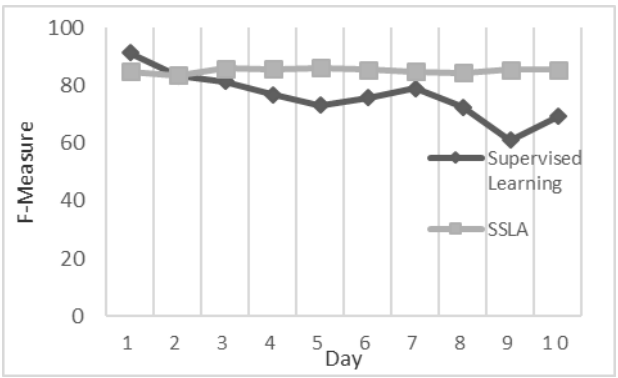

Fig. 5. Comparison between Supervised and Semi-Supervised approaches (training on day 1 and testing on day 1-10)

In addition, the second dataset was used to confirm the efficiency of the proposed approach (SSLA). The Day 1 training dataset and Day 1 to Day 5 testing data were used for this experiment. Figure 7 presents the experimental results using the second dataset in

terms of F-measure. We can see that SSLA performed much better than the supervised learning (Random Forest). The prediction accuracy of SSLA increases slightly from 75% to 81%. However, there was a major drop in the prediction accuracy of the supervised learning (Random Forest) from 95% to 75%

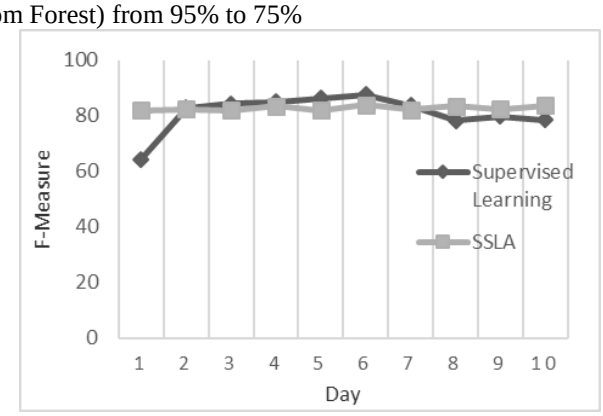

Fig. 6. Comparison between Supervised and Semi-Supervised approaches (training on day 6 and testing on day 1-10)

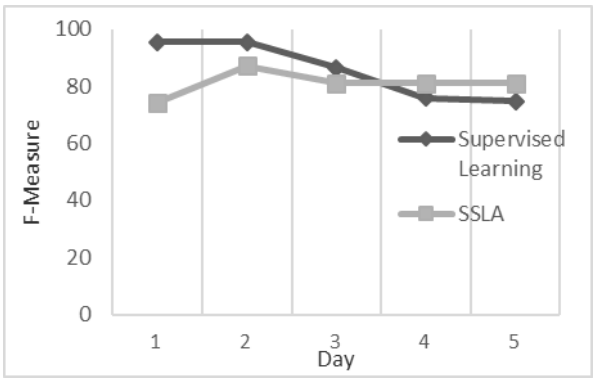

Fig. 7. Comparison between Supervised and Semi-Supervised approaches (training on day 1 and testing on day 1-5)

The last experiment was carried out to improve the accuracy of the proposed approach. It has been proven that changing the value of the weighting parameter F, which gives as much weight to the training set as to testing set, improved the YATSI performance (Driessens et al. 2006; Choeikiwong and Vateekul 2016).

The weight is used to reduce the influence of the test set, and it can be calculated by the following equation: weight = $p \times$ (*training instances/ test instances*). p is a parameter that can be defined by a user to raise or lower the importance of the test-set (Pfahringer, Driessens, and Reutemann 2015). Additionally, Choeikiwong and Vateekul (2016) define the weighting strategy as the amount of trust on the unlabeled data, and it applies to a distance during the process of finding a neighbour. By default, the weight of the labelled data is set to 1, but the weight of unlabeled data is equal to $F \times (N/M)$. N refers to the

number of labelled data and *M* refers to the number of unlabeled data. *F* is a parameter that can be defined by a user to show the amount of trust on the unlabeled data. In Figure 8 when changing the weighting parameter F to 0.1, the prediction accuracy of the SSLA improved slightly. Especially on Day 6, the improvement percentage almost reached 5%.

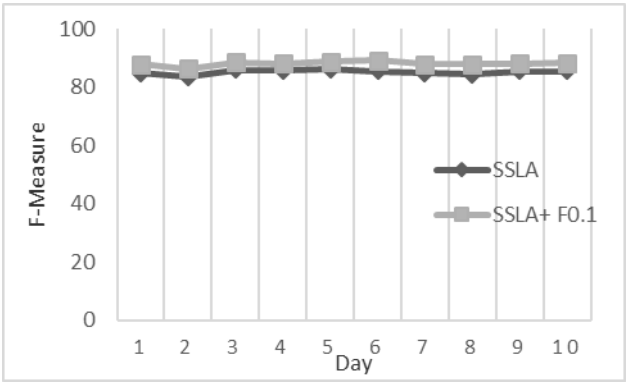

Fig. 8. Comparison between SSLA and SSLA with weighting parameter F value 0.1 (training on day 1 and testing on day 1-10)

Finally, like mentioned before there are some machine learning approaches that are being proposed to deal with the Twitter Spam Drift problem like the ASL and Lfun approaches (Chen et al. 2015a and Chen et al. 2017). These two approaches were able to outperform the traditional ML algorithms (e.g., Random Forest, C4.5, and Decision Tree). The F-measure of the traditional algorithms, such as Random Forest remains stable more than 80% when using ASL as in figure 9. Similarly, the F-measure of Lfun was stable and reached slightly above 80% as in figure 10. However, this study shows that the proposed approach SSLA outperforms both the ASL and Lfun approaches with an F-measure greater than 86% as shown in Figure 11.

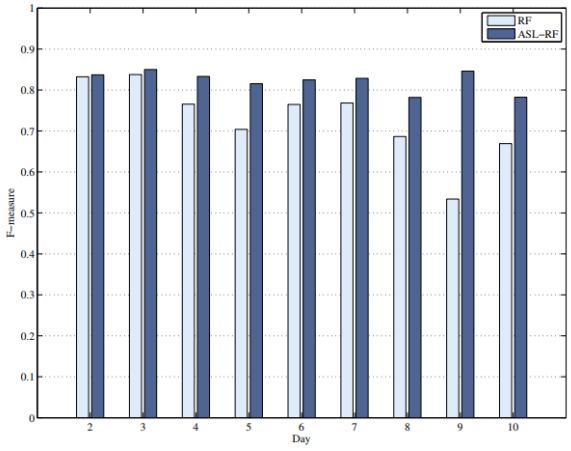

Fig. 9. Spam F-measure Comparisons before and after ASL for Random Forest (Chen et al. 2015a)

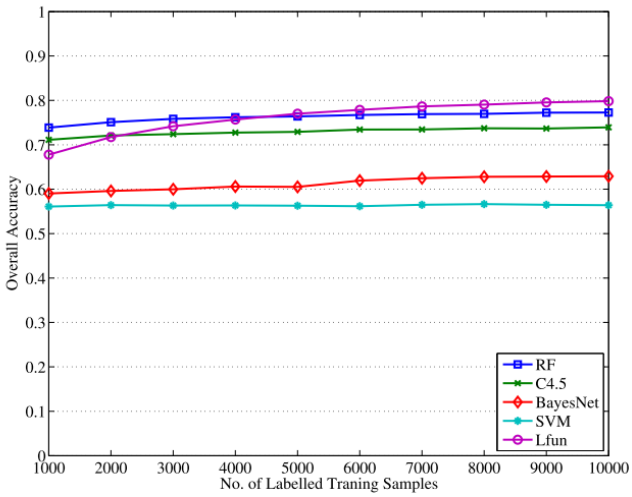

Fig. 10. Comparisons with other algorithms (training on day 4 and testing on day 8) and overall accuracy of Lfun (Chen et al. 2017)

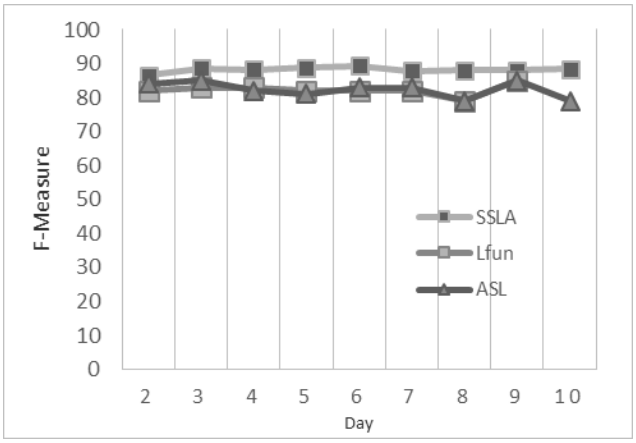

Figure 11. Comparison between ASL, Lfun and SSLA (training on day 1 and testing on day 2-10)

6. Conclusion

Twitter popularity has not only attracted more users, but it also makes it a very attractive platform for spammers. As the number of spammers is growing rapidly, the research community and Twitter have been developing different spam detection systems to protect users. Various machine learning approaches have been proposed to detect spam tweets. However, a recent study pointed out a new problem in Twitter detection systems called Drifted Twitter Spam (Chen et al. 2015). The study shows that spam tweet characteristics are changing over time, which affect the performance of the traditional ML algorithms.

Consequently, this paper introduced a new approach that can reduce the effect of Twitter Spam Drift while detecting spam tweets. The proposed approach, SSLA, is a semi-supervised ML technique that uses the unlabeled data to learn the structure of the domain. Thus, it can detect spam tweets with high accuracy even when the Twitter Spam Drift problem occurs. SSLA uses the YATSI algorithm, which can be built on top of any supervised machine learning algorithms. One of the advantages of using YATSI is that it usually improves the predictive performance of the base classifier (Driessens et al. 2006). Random Forest was used as the base classifier and Filtered Neighbor Search as the nearest Neighbor Search algorithm. The performance of SSLA was evaluated with F-measure values. Various experiments were carried out, and the results showed that SSLA can reduce the effect of Twitter Spam Drift. The F-measure of SSLA was compared to the Random Forest algorithm and some of the currently proposed approaches (e.g., ASL and Lfun), and it was found that SSLA outperforms them.

References

1. N. Al Twairesh, M. Al Tuwaijri, A. Al Moammar, S. Al Humoud, "Arabic Spam Detection in Twitter", *The 2nd Workshop on Arabic Corpora and Processing Tools 2016 Theme: Social Media*, (2016).
2. A. Al-Zoubi, J. Alqatawna & H. Faris, "Spam profile detection in social networks based on public features", *2017 8th International Conference on Information and Communication Systems (ICICS)*, (2017), pp. 130.
3. F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)* (2010) (Vol. 6, No. 2010, p. 12).
4. 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, (2017)*, vol. 12, no. 4, pp. 914-925.
5. C. Chen, J. Zhang, X. Chen, Y. Xiang & W. Zhou, "6 million spam tweets: A large ground truth for timely Twitter spam detection", *2015 IEEE International Conference on Communications (ICC)*, 2015b, pp. 7065.
6. C. Chen, J. Zhang, Y. Xiang and W. Zhou, Asymmetric self-learning for tackling twitter spam drift. In *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on* (pp. 208-213). (2015a), IEEE.
7. C. Chen, J. Zhang, Y. Xie, Y. Xiang, W. Zhou, M. M. Hassan, A. AlElaiwi, and M. Alrubaian, A performance evaluation of machine learning-based streaming spam tweets detection. *IEEE Transactions on Computational Social Systems*, *2*(3), (2015c), pp.65-76.
8. T. Choeikiwong, and P. Vateekul, Improve Accuracy of Defect Severity Categorization Using Semi-Supervised Approach on Imbalanced Data Sets. In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (Vol. 1) (2016).
9. M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, & H. Al Najada, "Survey of review spam detection using machine learning techniques", *Journal of Big Data*, (2015), vol. 2, no. 1, pp. 23.
10. K. Driessens, P. Reutemann, B. Pfahringer, and C. Leschi, "Using weighted nearest neighbor to benefit from unlabeled data", *PAKDD* Springer, (2006), pp. 60.

11. N. El-Mawass and S. Alaboodi, "Detecting Arabic spammers and content polluters on Twitter", *2016 Sixth International Conference on Digital Information Processing and Communications (ICDIPC)*, (2016), pp. 53.
12. N. Eshraqi, M. Jalali and M. Moattar, "Detecting spam tweets in Twitter using a data stream clustering algorithm", *2015 International Congress on Technology, Communication and Knowledge (ICTCK)*, (2015), pp. 347.
13. S. Goyal, R. K. Chauhan, and S. Parveen, Spam detection using KNN and decision tree mechanism in social network. In *Parallel, Distributed and Grid Computing (PDGC), 2016 Fourth International Conference on* (pp. 522-526). (2016), IEEE.
14. C. Grier, K. Thomas, V. Paxson, and M. Zhang, @ spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security* (pp. 27-37). (2010), ACM.
15. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update", *ACM SIGKDD explorations newsletter, (2009)*, vol. 11, no. 1, pp. 10-18.
16. Json.org. (n.d.). *JSON*. [online] Available at: http://www.json.org/ [Accessed 22 Aug. 2018].
17. G. Lin, N. Sun, S. Nepal, J. Zhang, Y. Xiang & H. Hassan, *Statistical Twitter Spam Detection Demystified: Performance, Stability and Scalability* (2017).
18. M. Mateen, M. Iqbal, M. Aleem and M. Islam, "A hybrid approach for spam detection for Twitter", *2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, 2017, pp. 466.
19. C. Meda, E. Ragusa, C. Gianoglio, R. Zunino, A. Ottaviano, E. Scillia, and R. Surlinelli, Spam detection of Twitter traffic: A framework based on random forests and non-uniform feature sampling. In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on* (pp. 811-817). 2016, IEEE.
20. Z. Miller, B. Dickinson, W. Deitrick, W. Hu, and A. H. Wang, *Twitter spammer detection using data stream clustering* (2014).
21. B. Pfahringer, "A semi-supervised spam mail detector", *Discovery Challenge Workshop* (2006).
22. B. Pfahringer, K. Driessens, and P. Reutemann, (2015). *fracpete/collective-classification-weka-package*. [online] GitHub. Available at: https://github.com/fracpete/collective-classification-weka-package [Accessed 22 Aug. 2018].
23. I. Santos, C. Laorden, and P. G. Bringas, "Collective classification for unknown malware detection", *Security and Cryptography (SECRYPT), 2011 Proceedings of the International Conference on*IEEE, (2011), pp. 251.
24. I. Santos, J. Nieves, and P. G. Bringas, "Semi-supervised Learning for Unknown Malware Detection.", *DCAI* Springer, (2011), pp. 415.
25. K. Saputro, S. Kusumawardani, and S. Fauziati, "Development of semi-supervised named entity recognition to discover new tourism places", *2016 2nd International Conference on Science and Technology-Computer (ICST)*, (2016), pp. 124.
26. M. Sigdel, I. Dinç, S. Dinç, M. Sigdel, M. Pusey, and R. Aygün, "Evaluation of semi-supervised learning for classification of protein crystallization imagery", *IEEE SOUTHEASTCON 2014*, pp. 1.
27. P. Sinha, O. Maini, G. Malik, and R. Kaushal, September. Ecosystem of spamming on Twitter: Analysis of spam reporters and spam reportees. In *Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on* (pp. 1705-1710). IEEE, 2016.

28. Twitter Help Center. (n.d). *The Twitter Rules*. [online] Available at: https://support.twitter.com/articles/18311 [Accessed 15 Nov. 2018]
29. Twitter Help Center. (n.d.). *Reporting spam on Twitter*. [online] Available at https://support.twitter.com/articles /64986 [Accessed 22 Aug. 2018].