Journal of Big Data

RESEARCH Open Access

An unsupervised method for social network spammer detection based on user information interests

Darshika Koggalahewa[1], Yue Xu[1*] and Ernest Foo[2]

*Correspondence:
yue.xu@qut.edu.au
[1] School of Computer Science, Queensland University of Technology, Brisbane, Australia
Full list of author information is available at the end of the article

Abstract

Online Social Networks (OSNs) are a popular platform for communication and collaboration. Spammers are highly active in OSNs. Uncovering spammers has become one of the most challenging problems in OSNs. Classification-based supervised approaches are the most commonly used method for detecting spammers. Classification-based systems suffer from limitations of "data labelling", "spam drift", "imbalanced datasets" and "data fabrication". These limitations effect the accuracy of a classifier's detection. An unsupervised approach does not require labelled datasets. We aim to address the limitation of data labelling and spam drifting through an unsupervised approach. We present a pure unsupervised approach for spammer detection based on the peer acceptance of a user in a social network to distinguish spammers from genuine users. The peer acceptance of a user to another user is calculated based on common shared interests over multiple shared topics between the two users. The main contribution of this paper is the introduction of a pure unsupervised spammer detection approach based on users' peer acceptance. Our approach does not require labelled training datasets. While it does not better the accuracy of supervised classification-based approaches, our approach has become a successful alternative for traditional classifiers for spam detection by achieving an accuracy of 96.9%.

Keywords: Spam detection, Peer acceptance, Information interest, Unsupervised learning, Classification

Introduction

Online Social Networks (OSNs) provide a wide range of communication and collaboration opportunities. They also provide a platform to build up new social relationships. OSNs are immensely vulnerable to spam.

OSN spammers are a set of users who manipulate the social media platform through their activities. Twitter defines its spammers as users who manipulate their platform. Platform manipulation activities are any form of behaviours which intend to negatively impact the experience of Twitter users. Some of these behaviours are: "Posting duplicate or very similar content across multiple accounts", "Posting multiple updates in an attempt to manipulate or undermine Twitter trends", "Posting multiple, duplicate updates on one accounts" etc.. [1]. OSNs contain a variety of "Social spams" such as fraudulent reviews, malicious links, click-baiting and likejacking, bulk messages, verbal abuse, and fake followers. Genuine users are users in the social network who do not show any platform manipulation activities. The most popular paradigm to detect spammers is the supervised classification-based approach. Support Vector Machines (SVM), Random forest and Artificial Neural Networks (ANNs) are among the most popular classification-based methods used for spam detection [2]. Classifiers often use OSN features for spam detection. Researchers change or introduce new features and combine multiple features to improve the accuracy of classifiers [3, 4]. Classifiers developed for spam detection suffered with open issues of data labelling and spam drifting [5–7].

The existing dominant spam detection models are supervised classification-based approaches. One key limitation of supervised learning models, which motivates our research, is the requirement of labelled datasets. It is mandatory to use labelled training data sets to train classifiers [6]. The process of data labelling as well as the quality of the labels are the two most challenging issues in data labelling. The training dataset is usually labelled manually. The generation of the labelled dataset is error-prone. It requires more time and extensive human effort. The quality of the labelled data is also a big concern where it brings lots of incertitude and biased decisions. Domain expertise is essential in labelling, but still different experts will generate different labels for the same elements [8]. Data labelling also introduces unnecessary privacy considerations. The collection of OSN data for labelling involves various privacy and ethical considerations.

To remain undetected, spammers shift their behaviors over time. Thus, "Classifiers trained on older datasets find it difficult to detect new forms of attacks" [5]. Therefore, it is essential to relabel the training data to represent the change in behaviour. Existing spammer detection techniques use a variety of features for their detection. Feature engineering is also a challenging task with changing behaviours. Hence, classification-based approaches for spam detection suffer from limitations of data labelling and spam drifting [4, 5]. The suitability and the sustainability of supervised classification-based approaches are in jeopardy with these limitations. The vibrant nature, diversity and complexity of OSNs made us doubt on whether classification-based techniques are practical for combatting social spammers in OSNs. Therefore, an unsupervised approach for spammer detection would be an ideal solution since it does not require labelled training datasets. This research is motivated from aforementioned limitations in supervised spammer detection in OSNs.

"Homophily theory suggests that people with similar interests are likely to connect" [9]. As proposed by Cardoso et al. [10] and Weng et al. [11], if two users reciprocally follow each other in a social network, then they have the same information interest. It is assumed that two users who post on a common topic that their posts' content should be similar. The similarity of the two users' posts should reflect their common interest in the topic and the content of the posts should also indicate the level of similar interest. Because of the similar topic interest and content, we say this indicates an agreed common interest. If this common information interest exists over multiple topics or in the majority of their content for both users, we consider that the users accept each other as valid users that are likely to connect. The likelihood of connection is the basis of Homophily theory. We can extend Homophily theory to all users in a community and based on their information interests, we can estimate how many users are likely to connect with one another. The majority of the users in a community are assumed to share information interest, we can say that they accept the other users in that community as reliable users.

As justified by Sykes et al. [12] and Asher et al. [13] a person in a social community would be a reliable individual when he/she is accepted and recognized by their peers. Thus he/she will not be a suspicious person. If an individual shows an acceptable behavior, then their peers may not hesitate to accept him as a member of the community. Thus, lower peer acceptability indicates that the individual does not belong to the same group and thus is suspicious. Therefore, the peer acceptability of a user in a social network could be a strong measurement to uncover suspicious users in a network. In our research, we define the peer acceptance (PA) as the "readiness of the community members to accept an individual's reliability in a social network". Peer acceptance is "the degree to which a child or adolescent is socially accepted by peers" in the fields of sociology and psychology. It comprises "the level of peer popularity and the ease with which a child or adolescent can initiate and maintain satisfactory peer relationships" [13, 14]. The connections or relationships (such as friendships) could be used to determine peer acceptance. Therefore, we introduce a novel method to determine the peer acceptability of a user in a social network.

Users post various messages on social networks. They may involve different topics in their posts to reflect or highlight their ideas. Therefore, a user's information interest could be derived from the post content. People tend to use a topic relevant to the content of their post. Thus, when multiple users post on the same topic, all their post content should be relevant to the topic and thus similar to each other. The similarity between user's post content with the peers' post content in a topic could be used to determine the peer acceptance of a user in the same topic. We followed the above idea to calculate users' pairwise interest similarity by using their post content in terms of each topic. Finally, user's peer acceptance is calculated using content similarity over multiple topics. For genuine users, the peer acceptance should be consistent across all the topics that they are interested in. Hashtags in tweets can represent topics. The inclusion of a hashtag in a user's posts is an indication of a shown interest on that topic. Nevertheless, compared to the post content of other users on the same topic, usually spammers do not show a similar interest with other users. In our approach, we use peer acceptability at three levels: the peer pairwise single directional level (Peer Acceptance), the peer pairwise bi-directional level (Mutual Peer Acceptance), and the community level (Overall Peer Acceptability). Peer acceptability among two individuals is assessed at the pairwise levels. Mutual peer acceptance is consistent over multiple topics. Two users who have mutual peer acceptance should have posts with multiple common topics and similar content. Globally a user should be accepted by all members of the community. For a user's global acceptance, the user's post content should be similar to the post content of users in each topic in the community. Therefore, user's peer acceptance over all users is assessed at the community level. Researchers in sociology and psychology have proven that an individual's inconsistent behaviors are one of the main reasons for peer rejection in the community [15, 16].

The proposed approach to generate a user's peer acceptance of another user is based on the two users' common shared topics. The approach can be biased when a user does not have many topics used in their posts. This results in a smaller number of common topics between a pair of users. User's information interests can be divided into two types. Some users are interested in diversified areas where they may use many topics in their posts. Some other users may be interested in very few selected areas where their topic usage is very low. Hence it is essential to determine the user's topic interest diversity to reduce the bias in the proposed approach. Our approach first categorizes the users into two groups "Focused" and "Diverse" based on their information interest distribution over all topics. From each user's posts, a topic model can be generated using a topic modelling technique such as Latent Dirichlet Allocation (LDA) [17]. Based on the topic model, a set of features are extracted and used to represent the user's interest distribution over the topics.

This paper proposes a two-stage unsupervised spam detection approach. In stage 1, clustering techniques are used to cluster users into two groups, "Focused" and "Diverse". This grouping is based on their information interest distribution and we considered whether users have a focused or diverse interest over multiple topics. In stage 2, users are assessed using our proposed peer acceptance criterion based on a user's common shared interest over multiple topics to identify potential spammers. The novelty of our proposed approach is that it does not require labelled training data like traditional classification approaches. Stage 2 uses the pairwise peer acceptance calculated for each user to generate the overall peer acceptability of a user in the community. The user is considered as a genuine user if a majority of the users accepts that user as their peer. Otherwise he is considered a spammer. The threshold for determining the majority of users in the focused group is different from that in the diverse group.

Unsupervised learning methods do not use labelled datasets as input. The algorithms presented in this paper are pure unsupervised because these algorithms work together to detect spammers by using an un-labelled dataset, i.e., whether the users in the datasets are spammers or not is unknown. This paper proposed two algorithms. Algorithm 1 generates uesrs' peer acceptance based on users' shared information interest. Algorithm 2 detects spammers based on users' peer acceptance. Clustering (k-means algorithm), which is a typical unsupervised technique, is used to cluster users into focused group and diverse group based on their information interest distribution.

The Social Honeypot [18], HSpam14 [19] and The Fake Project [20] datasets are used to evaluate our approach. All three datasets are publicly available. Two of the best classification-based spam detection approaches were used as baseline systems to compare with our proposed approach. The results are encouraging, and our peer acceptance-based unsupervised spam detection could be a potential alternative for traditional spam detection approaches. Most importantly, it does not require labelled training datasets. Even though our results are not better than the best classifiers, they are very close to the results generated by the classification-based systems. This paper makes the following contributions.

- A new concept of peer acceptance and mutual peer acceptance is proposed to describe user interest sharing.
- A novel method to assess users' peer acceptance based on users' common shared interest is proposed.

- A fully unsupervised two-stage spam detection approach is proposed as a successful alternative for traditional classification-based systems.

The paper is structured as follows. Section "Related work" offers a high-level overview of supervised spam detection approaches. The section of "Our unsupervised spam detection approach" presents our proposed approach. The evaluation and experimental results are presented in Section "Experiments and evaluation". The "Discussion" section discusses the relevant findings and observations. Section "Conclusion" concludes the research work.

Related work

Online social networks (OSNs) are flooded with social spammers. They appear in various forms such as fake accounts, fake reviews, malicious links, bulk messages. Social spammers take advantage of OSN platforms to publish malicious content, spread phishing scams and perform promotional campaigns [2, 3]. Social bots are very popular and highly influential in OSNs. But their honesty is questionable where people deploy social bots for both genuine and malicious purposes. Assenmacher et al. [21] studied the different types of social bots and their influence for the social network users. They highlighted the impact and the importance of detecting social bots in OSNs. They further investigated that, a very little amount of work is published on bot detection and detecting harmful behaviours in OSNs. De Paoli [22] investigated the importance of detecting social bot behaviours in OSNs and how to validate these behaviours and interaction between human and the social bots. Goswami et al. studied the user behaviours and users social interaction towards review fraud detection in social networks. They introduced a set of features to measure user behaviours. Their features were mainly based on user accounts with statistical measurements [23]. Malicious users often post malicious content such as fake news, spam emails, etc. Detection of malicious content will provide an indication of malicious users. Spammers are sophisticated, and they often alter their actions and spamming tactics to appear as genuine users. The popularity, structure, and user-friendliness of these OSNs generate an active platform for spammers [24, 25]. Traditional classification-based approaches for combating spammers first extract a set of features from the data collected from an OSN. Then an existing classification algorithm will be applied to build a classifier by using the extracted feature set to detect spammers. They embed malicious links/URLs in their tweets and encourage the user to click them. Due to the limited number of characters allowed in tweets with the usage of shortening URLs, tweets are highly vulnerable to spamming activities. Compared to other OSNs, Twitter spams are hard to detect [24, 26]. A labelled training dataset is mandatory for classification-based approaches. Hence, people use manual annotation or other labelling techniques to label tweets.

Neudert et al. analyzed the effect of posting spam content in real world scenarios [27]. They investigated the impact of junk news sources in three recent European election campaigns. Their study reveals the importance of detecting such behaviours and sources in real time and impact of such sources to a real campaign. Hence it is essential to analyse the post content representation to detect malicious behaviours in OSNs. Post content representation is extended for multiple dimensions with the aid of the latest advancements in text mining technologies. Wang et al. [28] introduced a generic content-based trust model for spam detection by combining information quality attributes and text features. Yang et al. [29] developed a system that compares the posts in two different Twitter accounts. The system was further improved by Chu et al. [30] with the aid of content similarity measures. However, the content similarity was not prioritized in their approach, and they used URL based features as prominent features. Wu [31] developed a hybrid method by combining a system and neural networks. He defined spam behaviors as rules for spam detection.It is essential to identify contextual features and user behaviour based features in addition to the existing low level features [32].

The state-of-the-art approaches employed deep learning methods and similarity-based methods as optimized classifiers for spam detection [33, 34]. Kudugunta and Ferrara [35] has introduced a deep learning-based method for spam bot detection. Their method tried to minimize the number of features used for classification. They also tried to reduce the size of the training dataset used for the classification. El-Mawass et al. [36] developed a hybrid social spam detection approach using Markov Random Fields, where they used a combination of users' features and content-based similarities. There were some semi-supervised approaches that use Matrix Factorization, where they combined the user's content and behaviors to detect social spammers [37]. There is some work to calculate the semantic similarity between contents. Various methods were used to calculate the content similarity. Ontology-driven approaches, Hybrid approaches, feature-driven approaches, and Content-based approaches can be considered as popular methods with combined techniques to generate similarity [38]. Li et al. [33] proposed a "hyponymy" based method to calculate the semantic similarity by using the WordNet lexical database. Homophily theory states that "contact between similar people occurs at a higher rate than among dissimilar people" [9]. "The pervasive fact of homophily means that cultural, behavioral, genetic or material information that flows through networks will tend to be localized" [12]. Pirro [38] investigated about the semantic similarity approaches and concluded that statistical approaches based on the word co-occurrence had high computational complexity and that approach is not suitable for large text collections.

There are different classification algorithms such as Decision Tree [39], Artificial Neural Networks and Deep Learning models [8, 40], Support Vector Machines [41], Naïve Bays etc. Supervised learning algorithms are applied in variety of fields such as object recognition [42, 43], object detection [40, 44, 45], image and colour analysis [46–48] and natural language processing which includes a variety of tasks such as language detection, question answering, language understanding and translations [49–51]. For spam detection many of the current OSNs use supervised learning algorithms [6, 7, 52].

Classification is the most common method for spam detection where people try to learn the classifiers using the features extracted from OSN data sets. These classification-based spam detection approaches suffered from limitations; data collection, data labelling, spam drift, imbalance datasets and data fabrication [24, 26, 38, 53]. Out of these problems, collecting and labelling data as ground truth is vital. A set of recent research works also summarizes the same set of open issues in Twitter spam detection [6, 7, 54–56]. They also highlighted another set of issues: (1) Feature selection and multi-dimensional features, (2) Adversarial machine learning attacks, (3) Biasness of the models and datasets. They remain as open issues for other classifier based approaches [57–59]. Many

OSN platforms do not allow to collect large amounts of data for research purposes. For example, Twitter allows only 1% of its network to be collected as sample data. Hence the collected datasets are usually biased and do not represent the entire social network. The most accurate method for data labelling is manual annotation. However, it is highly time-consuming and requires more human effort. Another method is blacklisting, which needs human effort and suffers from inaccuracy as well. The unavailability of labelled data or poorly labelled data will negatively effect the performance of classifiers.

Spam drifting is the problem of spammer features' changing through the life cycle. Spammers will change their features overtime. After the datasets are collected and features are extracted, the spam features can fluctuate. Hence classifiers trained on previously collected data may not be suitable for spam detection in new datasets since these features fluctuate over time. This will badly effect dynamic spam detection over time. Moreover, to avoid detection, spammers act as benign users by imitating genuine behaviors. Spammers apply some sophisticated social engineering techniques for data fabrication. These issues of classification for spam detection encourage the need for some alternative approaches. In addition to that, the viability of classification-based technologies for dynamic spam detection over time becomes more challenging with the changing OSN trends.

Our Unsupervised spam detection approach

We present a two-stage, unsupervised approach for spam detection by using users' information interest. People use topics (hashtags, titles, article headings) to emphasize the content and the idea of their posts in OSNs.

We assume that ordinary people often post and share similar content for same set of topics while spammers are not sharing similar content with their peers over multiple topics. For genuine users, the content of their posts is generally conformant with the topics. Spammers often insert trending topics in their posts to expand their social media reach. Posting irrelevant content for a topic is suspicious. Posting content that does not match the topic is a suspicious behaviour. We can assume that a user is a spammer if the user posts irrelevant content related to multiple topics. Further, if the content of a user's posts is unrelated to most users regarding the same topic, we consider that the user is not interested in the topic and could be a potential spammer. Users are free to insert their hash tags based on their own understanding and they have the freedom to label themselves. Some tags are only marked by few people or even one person which makes no significance.

If an ordinary person uses a tag in his posts, the use of the tag is intentional, and the tag should be relevant to the content he posted. It could be possible that a user may have used a tag in some posts with a different understanding of the tag or a different intention to many other users. But it is hard for a genuine user to have different intentions when using tags with other users across the majority of shared topics. Our method checks all topics to ensure that a user is identified as a spammer only if the majority of the user's posts do not contain shared content with other users. Therefore, even if a user had a different intention for a topic and is an outlier for that topic, the user may not be identified as a spammers if the user has similar intentions with many other users for most shared topics.

There are several studies that support the above assumption. Yousukkee and Wisitpongphan [60] investigated the relevance of social media messages to the topic and the content. They calculated a relevance score using bag-of-words and found that spammers tend to post irrelevant content to the topic. Several studies have shown that the spammers are posting similar content, or they have near duplicate tweets for frequent topics while ordinary people do not show such behaviours [57, 61, 62]. It is an indication that spammers post content that is irrelevant to the topic. Several studies on movie reviews investigated that, "A normal user usually has a relatively consistent attitude toward a specific movie, which means a user could be a spammer if he or she gives a completely opposite review to the same movie" [63, 64]. These studies showed that ordinary people are posting relevant content to a given topic while spammers are providing irrelevant content to a given topic.

Our aim is to identify spammers in OSNs by using users' information interests. It is possible to use the shared information interest to assess the acceptance of an individual in a social network as perceived by others. This acceptance can be used to identify spammers in social networks. A selection of representative words extracted from all of the posts belonging to a topic can be used to represent the content of that topic. A user's content interest can be measured by the frequency of representative words in their posts. If a user exhibits a similar information interest very similar to another user's interest across multiple shared topics, they have a high peer acceptance. In Section "Users' peer acceptance" , we propose a new concept, peer acceptance, to describe the interest sharing between users.

The measure of a user's acceptance against the community is determined by the overall peer acceptability. We assume that spammers do not join an online community to share information instead they aim to post malicious messages. Therefore, our hypothesis is that spammers do not share similar content interest or have a high overall peer acceptability and that they would not be accepted by peers in the community. The common shared interest metrics used to determine overall peer acceptability can be utilized for spam detection.

Users have different interests in various topics. Some people may have very focused interests in a few topics, while others have a diverse interest over many topics. For users with diverse content interest, their content similarity could be lower. Lower content similarity leads to lower peer acceptability for such users.

If the same criterion is used to assess all users, because of the lower peer acceptability, users with diverse interest will be detected as spammers (i.e., false positive). To deal with this problem, we used clustering to separate users into two groups,"focused" and "diverse". We applied clustering algorithms based on features derived from user's information interest to generate the two groups of users. In this paper, a topic modelling technique is applied to generate a topic model from users' tweets. The topic model provides users' information interest distribution over a set of latent topics. A set of features were learnt based on the topic model. The clustering is performed based on the learnt features to generate two groups,"focused" and"diverse".

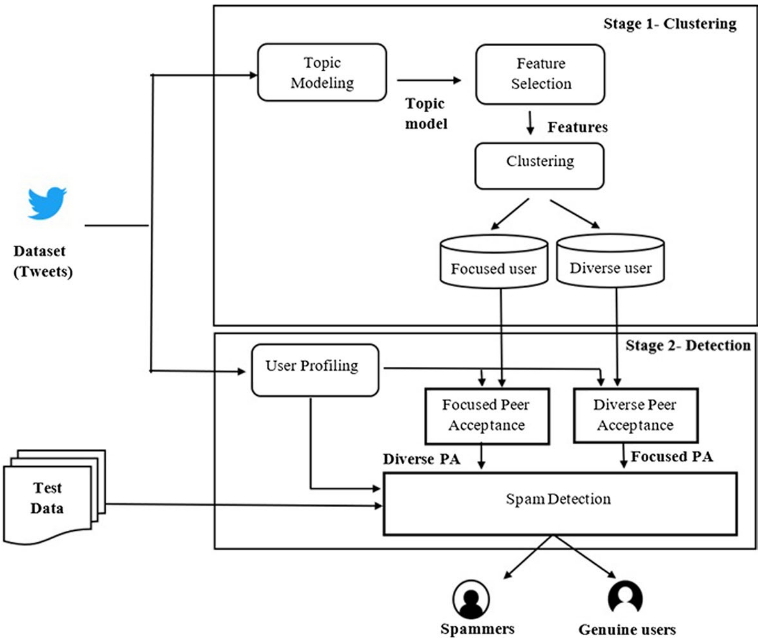

Fig. 1 The architecture of the two-stage spam detection system

In this paper, we propose a two-stage approach to detect spammers based on users' peer acceptance, consisting of a clustering stage and a detection stage. Fig. 1 depicts the architecture of the two-stage spam detection approach.

In Stage1, we perform the clustering based on user interest distribution and, in Stage 2, we do the spam detection based on peer acceptance. In Stage 1, a topic model is constructed from users' tweets first, then the features discussed in Section "Representation of user's information interest using LDA topic models" are extracted from the topic model to represent each user. Based on the extracted features from the topic model, users are clustered into two groups. In Stage 2, firstly, user profiles discussed in Section "Peer acceptance" are generated. Then the proposed peer-acceptance algorithm PeerAcceptance is used to generate the pairwise peer acceptance matrix for each group separately. The last step in Stage 2 is to categorize a test user as spam or genuine by assessing the user's peer acceptability against a threshold. If the user's peer acceptability is below the threshold of the cluster, we detect that the user as a spammer and otherwise a genuine user. (Section "Spam detection using peer acceptance and mutual peer acceptance" and "Thresholds used in the experiment" provide detailed descriptions about the thresholds used in the approach and the derivation of these thresholds based on the experimented datasets). Section "User clustering based on interest distribution", describes the approach for clustering, Section "Users' peer acceptance" describes the peer acceptance model with mutual peer acceptance distance, and section "Spam detection using peer acceptance and mutual peer acceptance" describes the peer acceptance-based spam detection method.

User clustering based on interest distribution

If a person focuses on a particular topic, his attention and concentration is mainly on that topic. He often thinks, discusses, and deals with that topic rather than dealing with other topics. In social networks, they post on a few selected topics. These users are focused users. If a user deals with group or range of various topics belong to a wide variety of domains, he has a diverse interest [65]. These users often include many topics in their posts. They are diverse users. Peers may not accept such users who focus on few topics due to fewer shared topics. Thus, these focused users could be wrongly identified as spammers. Hence, use of the same peer acceptance threshold for both user groups (focused and diverse) would be biased. In our two-stage spammer detection approach, users are clustered into two groups. This section describes the steps for clustering the users into focused and diverse groups.

Representation of user's information interest using LDA topic models

Topic modelling is a technique used to discover hidden topics of a given document collection. There are different approaches for topic modelling. LDA is a probabilistic method used to generate a probability distribution over k topics for a given document where each topic is represented by a probability distribution over words.

For LDA, each document d_i is represented as a multinomial distribution θ_i over a set of latent topics, $Z = \{z_1, z_2, ...z_k\}$, i.e., $\theta_i =< p(z_1|d_i), p(z_2|d_i), ..., p(z_k|d_i) >$, and $p(z_j|d_i)$ indicates the proportion of topic z_j in document d_i. θ_i is called the topic distribution for document d_i. Each topic z_j is also a multinomial distribution ϕ_j over a set of words $W = \{w_1, w_2, ..., w_M\}$, $\phi_j =< p(w_1|z_j), ..., p(w_M|z_j) >$, $\sum_{i=1}^{M} p(w_i|z_j) = 1$. ϕ_j is called the topic representation for the topic z_j.

A Set of users in OSN is represented as $U = \{u_1, u_2, ..., u_m\}$, where each user $u_i \in U$,d_i is considered as one document concatenating all tweets posted by u_i. A topic model can be generated from the document collection $\left\{d_1, d_2, ..., d_m\right\}$. For a user u, the entropy of the user's topic distribution θ_i can be used to measure the certainty of user's interest over the topics, as defined in Eq. 1 below, for simplicity, u is used instead of d.

$$E(u) = -\sum_{i=1}^{k} p(z_i|u) log_2 p(z_i|u) \tag{1}$$

If the entropy of a user's topic distribution is high, it indicates that the user has an evenly distributed interest over topics, meaning that the user has a diverse interest. On the other hand, a low entropy value indicates that the user is interested in a small range of topics. In addition to using the entropy of a user's topic distribution, we adopted two other types of features proposed by [66] as the other features for the clustering. The two types of features are Global Outlier Standard Score (GOSS) and Local Outlier Standard Score (LOSS), as defined in Eqs. (2) and (3), respectively. Liu et al. originally used these features for classification, we applied them to derive our clusters.

Global Outlier Standard Score (GOSS) measures how a user's tweet content is related to other users over a certain topic z_k, $x_{ik} = p(z_k|u_i)$.

$$GOSS(x_{ik}) = \frac{x_{ik} - \mu(x_k)}{\sqrt{\sum_i (x_{ik} - \mu(x_k))^2}} \tag{2}$$

where $\mu(x_k) = \frac{\sum_{i=1}^{m} x_{ik}}{m}$ is the average topic interest to the topic z_k over all users.

The value of $GOSS(x_{ik})$ indicates user u_i's degree of interest to topic z_k. Because GOSS is normalized across all users, $GOSS(x_{ik}) \gg GOSS(x_{jk})$ indicates that the user u_i is more interested in topic z_k compared to user u_j who is interested in the same topic. The extreme higher or lower values of $GOSS(x_{ik})$ indicates the user has a high interest or a very low interest for that topic. If we extract K topics for each user, we will end up with a vector of K features for each user, $GOSS(u_i) =< GOSS(x_{i1}), ..., GOSS(x_{iK}) >$. Local Outlier Standard Score (LOSS) measures the content interest of a user over topics based on the user's own contents.

$$LOSS(x_{ik}) = \frac{x_{ik} - \mu(x_i)}{\sqrt{\sum_k (x_{ik} - \mu(x_i))^2}} \tag{3}$$

where $\mu(x_i) = \frac{\sum_{k=1}^{K} x_{ik}}{K}$ is the average topic interest over all topics for user u_i. Similar to GOSS, we will have K LOSS features for each user, $LOSS(u_i) =< LOSS(x_{i1}), ..., LOSS(x_{iK} >)$.

Clustering users into two groups: focused vs diverse

Clustering is an unsupervised machine learning technique which find clusters of objects that share similar characteristics. Clustering techniques are designed to find hidden strictures in data. In clustering unlabelled objects are grouped in to clusters where objects in same cluster are similar to each other compared to objects in other clusters. In other words, data points in same cluster should have similar features while data points in different clusters have different features.

Once the features, i.e., topic entropy, GOSS features and LOSS features, were generated, a clustering method such as K-means can be used to cluster users into two groups using the $2K + 1$ topic features as a vector, denoted as $< GOSS(u), LOSS(u), E(u) >$, to represent each user. In the experiments reported in Section "Experiments and evaluation", the topic model used 25 LDA topics. The number of topics for the LDA model is an experimental value. We used cluster evaluation metrics (Silhouette score) to analyze the quality of clustering using the different number of topics from 5 to 35. Using 25 topics achieves the best performance. We assumed that the best clustering results would be provided with the best number of topics for LDA. In the experiments reported in Section "Experiments and evaluation", the number of topics in the LDA topic model is set to 25. The group whose users have a higher average topic entropy is considered the diverse user group, while the group whose users have a lower topic entropy is considered the focused user group. For the three datasets, the entropy is higher for most of the users in cluster 1, and the users in cluster 2 have a comparably lower entropy for the topic distribution. From the distribution of the entropy, we can assume that cluster 1 consists of diverse users and cluster 2 contains more focused users. Figure 2 depicts the entropy of

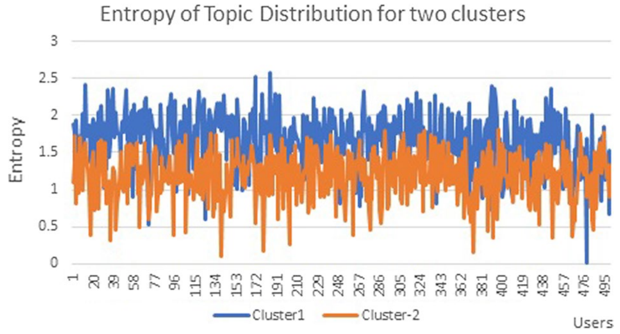

Fig. 2 Entropy of the topic distribution in Social Honeypot dataset

the topic distribution for the first 500 users in each cluster for the social honeypot dataset, which shows that the users in cluster 1 have higher entropy values.

It should be mentioned that both clusters contain spammers as well as genuine users. For the focused cluster, the number of spam users is comparably low in all the three datasets. As described earlier, it is essential to address the nature of user's information interest. A higher entropy of a user's topic distribution indicates that the user has an evenly distributed interest over the topics meaning that the user has a diverse interest. Low entropy is an indication of the user's interest in only some of the topics. The value of $LOSS(x_{ij})$ is an indication of the degree of user u_i's interest in a certain topic z_j.

The extreme higher or lower values of LOSS indicate that the user is highly interested in some topic or a very low interest for that topic. Figure 3 depicts the user distribution in each cluster. We can see that both clusters consist of spam and genuine users. The number of focused users is relatively lower than that of the diverse users.

The behaviour of LOSS features is quite similar to topic entropy. The LOSS values overall topics are uneven for each focused user while they should be quite similar for each diverse user.

Figure 4 depicts the average standard deviation of LOSS value of each topic over all users in the focused cluster (i.e., cluster 2) and the diverse cluster (i.e., cluster 1). The average standard deviation of LOSS values overall topics in the diverse cluster is 0.026. It is much lower than the average standard deviation of LOSS values over all topics in the focused cluster, which is 0.104. It is evident from the figure that the focused user's interest is focused on some topics with much higher LOSS values, which makes the standard deviation higher. On the other side, a diverse user's LOSS feature values are similar over all topics, which makes the standard deviation of the LOSS values lower. In summary, the focused users have a lower entropy value, while the average LOSS and GOSS values are higher. For diverse users, their entropy is higher, while the average GOSS and LOSS values are lower.

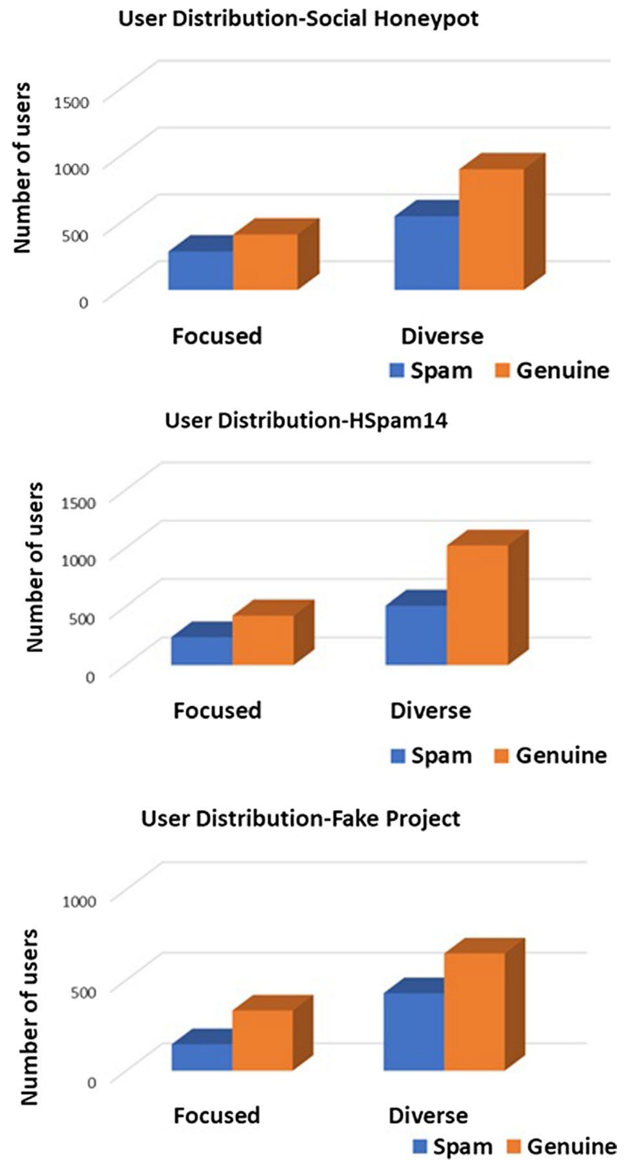

Fig. 3 Distribution of users in each cluster for different datasets

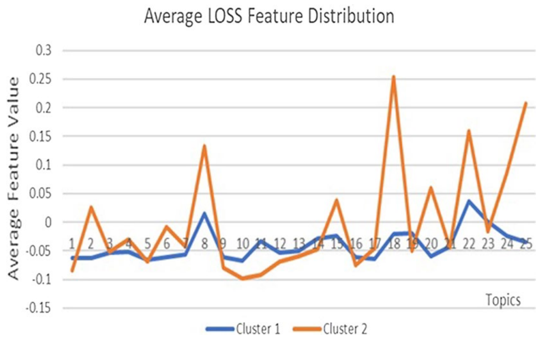

Fig. 4 Average of LOSS feature across all topics in Social Honeypot dataset

Users' peer acceptance

The content in social media posts (i.e., Tweets) covers various topics. Tweets often include hashtags to indicate the topics. We can derive a user's information interest from the user's tweets. To generate peer acceptance among users, we propose the following approach.

Peer acceptance

A set of users in an OSN and a set of hashtags used by the users are represented as $U = \{u_1, u_2, ..., u_m\}$ and $T = \{t_1, t_2, ..., t_n\}$, respectively. The hashtags in T are frequent topics based on the percentage of tweets that contain each topic. These frequent topics are extracted from all the tweets posted by all users in U . Let P_i be a document constructed by concatenating all the tweets posted by user $u_i \in U$. Based on word tf-idf values on the collection $\{P_1, ...P_m\}$, the top frequent words in P_i, denoted as W_i, are selected for user u_i. Overall, $W = \bigcup_{u_i \in U} W_i$ contains all the frequent words for all users.

(1) *Representation of user's information interest: Users' information interest* can be reflected from their posts. Therefore, users' posts can be used to derive each user's information interest. From users' posts, we can construct a tensor $\mathcal{CI} \in \mathbb{N}^{|U|x|T|x|W|}$ to represent each user's profile, where $\mathcal{CI}(u_i, t_j, w_k)$ is the term frequency of w_k in user u_i's tweets in topic t_j. $\overrightarrow{\mathcal{CI}}(u_i, t_j) \in \mathbb{N}^{|W|}$is the vector to represent user u_i's interest in the topic t_j.

(2) *Representation of topics:* Eq. (4) below calculates the average interest of all users in a topic t (i.e., the centroid vector for topic t), which is derived from vectors $\overrightarrow{\mathcal{CI}}(u_i, t), i = 1, 2, ..., m$. The content of the topic t can be represented by $\overrightarrow{T}_t$.

$$\overrightarrow{T}_t = \frac{1}{m} \sum_{u_i \in U} \mathcal{CI}(u_i, t) \tag{4}$$

Users often include multiple topics in their tweets. But their degree of interest to each topic might be varied. Therefore, we determine the set of most interested topics for a certain user. The topics which are similar to the topic representation $\overrightarrow{T}_t$ are deemed as the set of topics that the user is most interested in. In contrast, the topics that are dissimilar to the topic representation $\overrightarrow{T}_t$ are the topics that the user is not interested in.

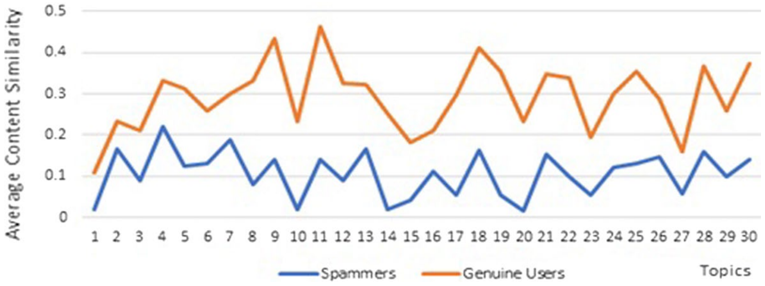

Fig. 5 Variation of average content similarity distribution for spammers and genuine users

To determine the set of interested topics, we calculate the similarity between a user's content vector of each topic and the topic centroid vector. Eq. (5) defines the user's topic set, where ω is a threshold of the minimum similarity.

$$\mathcal{UT}(u) = \left\{ t | t \in T, sim(\overrightarrow{\mathcal{CI}}(u,t), \overrightarrow{T}_t) \geq \omega \right\} \tag{5}$$

In order to investigate the behavior of content similarity across multiple topics for spammers and genuine users, we have conducted an experiment to compare spammers and genuine users in terms of the similarity between users' post content and the topic that the posts belong to. Figure 5 compared the average similarity of users' content $\overrightarrow{\mathcal{CI}}(u_i, t_i)$ to topic representation T_i for each topic between spammers and genuine users. The results are from the social honeypot dataset. It is evident from the figure that content similarity for the identified topics is different between the two user groups, spam and genuine. Genuine users have a higher content similarity compared to spammers over all the topics. It confirms our assumption that there is a clear difference between spammer's and genuine user's post contents for any given topic.

(3) *Peer Acceptance based on information interest:* People perceive the world with the same or different perceptions. Thus, they may have the same or different opinions/ideas about the incidents of the world. Users in a social network can partially share their information interest with other users. They may also do not share much interest with their peers. Therefore, peer acceptance aims to assess the amount of information shared from one user with another user across their common topics. We consider that the more a user shares his/her interest with another user, the more the user accepts the other user in terms of information interest. We consider that the content interest of two users for their common topics can be used to determine the peer acceptance of the two users. We define the peer acceptance of u_i to u_j as the ratio between the common interest of the acceptor u_j and the acceptee u_i across their shared topics and the acceptor u_j's interest for all the topics of u_j. Let the peer acceptance of user u_i to u_j be $PA(u_i, u_j)$, where u_i is the acceptee of the relationship and u_j represents the acceptor, $PA(u_i, u_j)$ is defined as follows:

$$PA(u_i, u_j) = \frac{\sum_{t_k \in \mathcal{UT}(u_i) \bigcap \mathcal{UT}(u_j)} (sim(T_k, \mathcal{CI}(u_j, t_k)) * sim(\mathcal{CI}(u_i, t_k), \mathcal{CI}(u_j, t_k))}{\sum_{t_k \in \mathcal{UT}(u_i)} sim(T_k, \mathcal{CI}(u_j, t_k))} \tag{6}$$

In Eq. (6), the numerator defines user u_j' s content interest in the topics commonly shared by both users weighted by the similarity between the two users' interests, while the denominator defines user u_j's content interest for all the user's topics.

The steps of generating the peer acceptance among two users are explained in the *Peer acceptance* algorithm.

```
Algorithm 1:Peer acceptance
Input : User set $U$
        User content profile $\mathcal{CI}$ ,
        Topic content representation $T_j, j = 1, ...n$ ,
        Representative topics for each user $\mathcal{UT}(u_i) i = 1, ..., m$
Output : Peer Acceptance matrix $PA \in \mathbb{R}^{|U|*|U|}$
1:  for each user $u_a \in U$ do
2:    for each user $u_b \in U$ do
3:      $PA(u_a, u_b) = 0$//initializing the Peer Acceptance
4:      $\mathbb{C} := \mathcal{UT}(u_a) \bigcap \mathcal{UT}(u_b)$//common topics among two users.
5:      $S_n := 0, S_d := 0$ //initial value for the numerator and denominator in
        Equation 6
6:      for each $t_k \in \mathbb{C}$ do
7:        $S_n := S_n + sim(T_k, \mathcal{CI}(u_b, t_k)) * sim(\mathcal{CI}(u_a, t_k), \mathcal{CI}(u_b, t_k))$
8:      for each $t_j \in \mathcal{UT}(u_b)$ do
9:        $S_d := S_d + sim(T_j, \mathcal{CI}(u_b, t_j))$
10:     $PA(u_a, u_b) := \frac{S_n}{S_d}$
11: Return $PA$
```

A Peer Acceptance matrix, denoted as PA, is the output of the Algorithm 1: *Peer acceptance* algorithm, which contains peer acceptance for each pair of users calculated using Eq. (6). For each user pair in the dataset, as described in lines 3 and 4, we generate the common topics used by both users. As described in lines 6 and 7, we sum up the content similarity between each topic and the acceptor u_b as well as the content similarity between two users over all common topics. As described in lines 9 and 10, we sum up the similarity between each topic and the acceptor u_b over u_b's topics. Finally, the ratio between these two is considered as the peer acceptance of u_a to u_b. The Peer acceptance matrix for all the users will be generated from algorithm 1 as the final outcome. It is asymmetric i.e., for a pair of users, $PA(u_a, u_b)$ is not necessarily the same as to $PA(u_b, u_a)$. Since peer acceptance is bi-directional, we considered the peer acceptance in one direction as well as bidirectional (mutual peer acceptance) as two separate components.

Mutual peer acceptance

Spammers who promote the products or services usually post near-duplicate tweets which often make these spammers seem to be genuine users because the content of their tweets is similar. Spam campaigns are the other reason for such tweets. The original HSpam14 dataset contains 71.5% of near-duplicate tweets out of 3.3 million spam tweets. Posting duplicate tweets occasionally are expected for genuine users. In contrast, for the HSpam14 dataset, only 7% of the near-duplicate tweets are posted by genuine users [61]. Genuine users may not post such near duplicate tweets. Further, the majority of the near duplicate tweets contain the frequent hashtags that we used in the experiment. As a result of this higher content similarity, the pairwise peer acceptance will be high for users who post many near duplicate tweets. This leads us to identify users with higher content similarity with others as genuine users in the final detection. Algorithm 1 generates the peer acceptance matrix for all user pairs in the dataset. In this matrix, there are three possible pairs of user combinations. They are genuine–genuine, spam–spam, spam–genuine pairs. Peer acceptance is asymmetric, i.e., for a pair of users, $PA(u_a, u_b)$ is not necessarily the same as $PA(u_b, u_a)$.

To measure the difference, we define the mutual peer acceptance distance of two users as $MPAD(u_a, u_b) = |PA(u_a, u_b) - PA(u_b, u_a)|$. An experiment is conducted to observe the behaviour of mutual pairwise peer acceptance.

In the experiment, it is noted that the mutual peer acceptance distance $MPAD(u_a, u_b)$ is much smaller for spammer pairs compared to the genuine user pairs or genuine-spam pairs. In general, a pair of spammers is mutually closed to each other when compared to a genuine user pair. This is because two spammers in the same topics often post tweets with high similarity or duplicate content, which makes their mutual peer acceptance distance small. In contrast, genuine users did not exhibit such behaviour in their tweets. Figure 6 depicts the mutual pairwise peer acceptance distance in two user groups, spam and genuine, for all three user combinations (spam-spam, spam-genuine, genuine-genuine). Figure 6 shows the average mutual pairwise peer acceptance distance between user pairs of 100 randomly selected users from fake project dataset. Due to space limitation, we only randomly selected 100 users.

In fact, we conducted the experiment for all users and ensured the same behavior among all user pairs. The average mutual pairwise distances for the three groups are 0.182968, 0.186097 and 0.086125 for genuine-genuine, genuine-spam, and spam-spam, respectively. The average distances clearly show that the spam user pairs have much lower mutual peer acceptance distance than that of genuine-genuine and genuine-spam pairs.

It is clearly evident from Fig. 6 that the mutual peer acceptance distances of spammer pairs are much lower, while genuine-genuine or spam- genuine user pairs have diversified mutual pairwise distances.

A user's mutual peer acceptance distance to the users in a cluster is calculated as below, where *c* can be *d* or *f* representing the diverse or focused cluster, respectively.

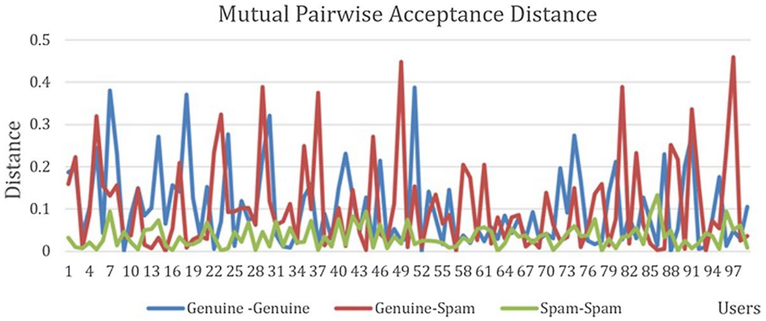

Fig. 6 Mutual Peer Acceptance for different user groups

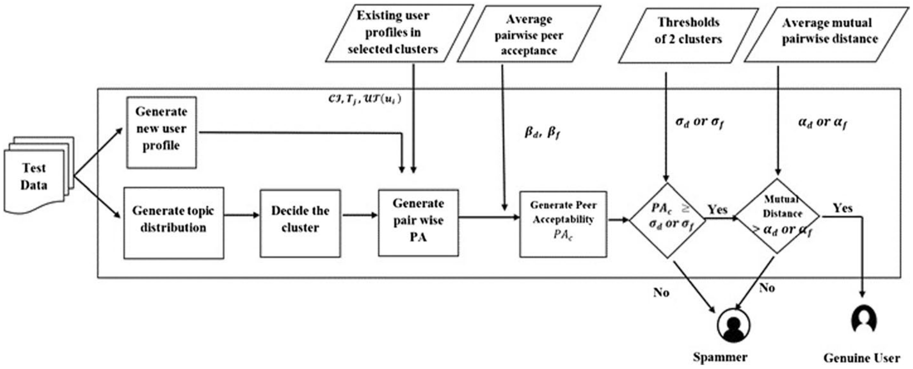
Existing user profiles in selected clusters
Average pairwise peer acceptance
Thresholds of 2 clusters
Average mutual pairwise distance
$CJ, T_j, UT(u_i)$
β_d, β_f
σ_d or σ_f
α_d or α_f
Test Data
Generate new user profile
Generate topic distribution
Decide the cluster
Generate pair wise PA
Generate Peer Acceptability PA_c
$PA_c \geq \sigma_d$ or σ_f
Yes
Mutual Distance $> \alpha_d$ or α_f
Yes
No
No
Spammer
Genuine User

Fig. 7 Spam detection process for new users

$$MPAD_c(u) = \frac{\sum_{v \in U_c} MPAD(u, v)}{|U_c|} \tag{7}$$

Both peer acceptance and mutual peer acceptance is used for the spam detection process.

Spam detection using peer acceptance and mutual peer acceptance

As depicted in Fig. 1 and explained in Section "User clustering based on interest distribution", in the first stage of the two-stage spam detection approach, users are clustered into two groups. In Stage 2, firstly the proposed Algorithm 1 is used to generate the pairwise peer acceptance matrix for each group, separately. Secondly, for each group, three thresholds are to be generated.

- One threshold, denoted as β in Algorithm 2 below, is used for determining the acceptance of one user by another user. For a pair of users u_i and u_j, if $PA(u_i, u_j)$ is larger than β, we consider that user u_j accepts user u_i.
- The second threshold, denoted as σ in Algorithm 2, is used for identifying spammers. A person can be considered as a reliable individual, when majority of the members in the group/community accept the individual. The percentage of users in a group who accept a certain user is defined as the user's acceptability. If a user's acceptability is larger than σ, the user is considered as a spammer.
- The third threshold, denoted as α in Algorithm 2, is used to further detect spammers from the users who were considered as genuine based on σ.

The average peer acceptance for all the users in an OSN to is used to calculate the threshold β. The threshold σ is the average entropy of interest distribution of all users in a group. The threshold α is the average pairwise peer acceptance distance of all user pairs in a group. "Thresholds used in the experiment" section provides the details about the definitions of these thresholds in this research. The last step in stage 2 is to categorize a test user as spam or genuine by assessing the user's peer acceptability and the mutual pairwise distance against the thresholds. The user is considered as a genuine user when the peer acceptability of that user is above the threshold β of the cluster and otherwise, a spammer. Finally, the set of genuine users identified using the peer acceptability will go through a further detection process based on the mutual pairwise distance. Figure 7 illustrates the detail of the spam detection step in Fig. 1. Algorithm 2 below describes the steps in Stage 2. The test dataset contains new users with unseen posts. To determine which cluster, i.e., focused, or diverse, a test user belongs to, first we need to generate the user's LDA topic distribution based on which to calculate the user's topic entropy, GOSS and LOSS features. These features form a vector which is used to determine the cluster to which the user belongs based on the similarity between the user's feature vector and the cluster centroid.

To generate a test user's LDA topic distribution, all the posts of the user are concatenated to form a document. The method proposed by (Hoffman et al. [67]) was used to generate the topic distribution for an unseen document based on an existing topic model. In this paper, the Genism package (Radim and SOJKA 2010) which implemented the method in (Hoffman et al. 67) and the topic model generated in stage 1 from the training users are used to generate the topic distribution for each test user, which is calculated in line 7 in Algorithm 2. Once the LDA topic distribution is generated for a test user, the topic Entropy, GOSS and LOSS features are calculated (line 8 in Algorithm 2). The similarity between the centroid vector of each cluster and the test user's feature vector is calculated and the user is assigned to the cluster, which is closer, as described in lines 9 and 10 in Algorithm 2.

The decision about whether the test user is a spammer or a genuine user is made based on the percentage of users in the assigned cluster who would accept the test user. Before we determine whether the test user is a genuine user or a spammer, we first determine the percentage of users in the assigned cluster who would accept the test user, then make the decision based on the percentage. This percentage is called the test user's peer acceptability, which is calculated in lines 11 to 15 in Algorithm 2. Next, the test user's peer acceptability is compared against the cluster spam threshold σ. It is to determine the user as a genuine user or a spammer. As shown in line 16 in Algorithm 2, if the peer acceptability of the test user is lower than the spam threshold σ of the cluster, we consider that the user is a spammer and otherwise a genuine user.

Algorithm 2: Spam Detection based on Peer Acceptance

```
Input : Focused training user cluster $U_f$ and diverse training user cluster $U_d$
        Topic feature vector of each user $u$ in $U_f \cup U_d$,
        $< GOSS(u), LOSS(u), E(u) >$
        Training user content profile $\mathcal{CI}$, topic content representation $T_j$,
        $j = 1, \dots n$, user topic set $\mathcal{UT}$
        Test dataset $U_T$
Output : User detection: Spam or Genuine
1:  Generate peer acceptance matrix $PA_d$ , $PA_f$ for each cluster by calling Algorithm 1.
2:  Generate the average topic entropy of each cluster, $\sigma_d$ and $\sigma_f$
3:  Generate the average pairwise peer acceptance of each cluster, $\beta_d$, $\beta_f$
4:  Generate the average pairwise peer acceptance distance for each cluster, $\alpha_d, \alpha_f$
5:  Calculate centroid vector $\vec{L}_f$ and $\vec{L}_d$ based on the topic feature vectors of users
    in the focused cluster and the diverse cluster, respectively
6:  for each test user $u \in U_T$ do
7:      Generate $u$'s topic distribution using the method proposed in Hoffman et al., [66]
8:      Generate $u$'s topic feature vector $\vec{L}_u$=$< GOSS(u), LOSS(u), E(u) >$
9:      $S_f := sim(\vec{L}_u, \vec{L}_f)$ AND $S_d := sim(\vec{L}_u, \vec{L}_d)$
10:     if $(S_f > S_d)$ then
11:         Assign $u$ to Focused Cluster, $c :=$ "$f$"
12:     else
13:         Assign $u$ to Diverse Cluster,$c :=$ "$d$"
14:     $Num\_accept := 0$
15:     for each user $v$ in the assigned cluster $U_c$ do
16:         if $PA_c(u, v) >= \beta_c$ then
17:             $Num\_accept := Num\_accept + 1$
18:     $Acceptability := Num\_Accept/|U_c|$
19:     if $Acceptability < \sigma_c$ then
20:         $u$ is a spammer
21:     else
22:         Generate $MPAD_c(u)$
23:         if $MPAD_c(u) > \alpha_c$ then
24:             $u$ is a Genuine User
25:         else
26:             $u$ is a spammer
```

There is a possibility that, the identified genuine users may contain spammers. This is because some spammers post near duplicate tweets. As a result of the near duplicate tweets, their peer acceptability will be high. To address this limitation, for each of the genuine users identified at line 16 of the algorithm, we generate the mutual peer acceptance distance, $MPAD_c(u)$. An experimental threshold α is used to distinguish the spammers from genuine users based on the mutual peer acceptance distance. As illustrated in Fig. 7, in the spam detection process for a test user, a user is considered as a genuine user only if both the user's peer acceptability and the mutual peer acceptance distance are higher than the respective threshold.

Experiments and evaluation

Our unsupervised spam detection approach detects spammers based on users' peer acceptability and mutual peer acceptance distance, calculated using our proposed algorithms. The overall performance of the proposed approach achieved an accuracy of 0.969 for the Fake project dataset, which is very close to the accuracy obtained by using traditional classification-based methods. This section will report the experiments and evaluation results.

Datasets and criteria for evaluation

We use three public datasets for our evaluation.

Each dataset is pre-processed by cleaning missing values and NULL values, removing links and non-English content. Stop words and special characters were also removed except for # and @ characters. Then we selected a set of frequent hashtags as topics. The users who used frequent hashtags in their tweets and have posted at least 25 tweets were chosen to form the user set U.

- Social Honey Pot [18]: The dataset was collected in 2010. It contains labelled users and their tweets for both genuine and spam users. The experiment uses, 1328 genuine users and 841 spammers.
- HSpam14 million Tweets-HSpam14 [19]: The dataset was published in 2015. A combined approach of human annotation with popular classification algorithms was used to label the tweets. In our experiment, if a user had over 30% of spam tweets, we labelled them as a spammer and otherwise a genuine user. We used 1450 genuine users and 750 spammers in our experiment.
- The Fake Project dataset [20]: Institute of Informatics and Telematics of the Italian National Research Council (CNR) released this dataset in 2017 with both twitter spambots and genuine accounts. We used 989 genuine users and 574 spammers in our experiment. The dataset is prepared by removing non-English tweets and filtering users with a fewer number of posts.

Thresholds used in the experiment

There are three different experimental thresholds used in our approach. Each threshold is experimentally tested based on the best detection accuracy. For the two clusters, focused and diverse, the threshold values can be different. The first set of thresholds, β_f and β_d, is designed to determine the peer acceptance among two users. The test user is considered to be accepted by a user in the assigned cluster if the test user's peer acceptance with the user is above the average pairwise peer acceptance of all users in the cluster. Therefore, we set β_f and β_d to be the average pairwise peer acceptance of the users in the two clusters, respectively. Let U_f and U_d denote the focused and diverse clusters, PA_f and PA_d be the pairwise peer acceptance matrix of the two clusters, respectively, β_f and β_d can be calculated as follows:

$$\beta_f = \frac{\sum_{u \in U_f} \sum_{v \in U_f} PA_f(u,v)}{|U_f|^2}, \beta_d = \frac{\sum_{u \in U_d} \sum_{v \in U_d} PA_d(u,v)}{|U_d|^2} \tag{8}$$

The second set of thresholds, i.e., the spam threshold σ_d and σ_f, used for determining spammers based on the overall acceptability of the user. To determine the type of the user (spam or genuine), the test user's peer acceptability is compared against the cluster spam threshold. As shown in line 16 in Algorithm 2, a user is considered as genuine, if the peer acceptability of the test user is above the spam threshold of the cluster, and otherwise the user is a spammer.

The entropy of a user's topic distribution can be used to measure the certainty of user's interest over the group of topics. Usually, focused users may be interested in a few topics and may share less topics with other users in the focused cluster, hence the acceptability value is relatively lower. On the other hand, diverse users are interested in many topics, and thus, they can share many topics with other users in the diverse cluster. Therefore, the users in the diverse cluster may have relatively higher acceptability values. So, we cannot use a unified threshold for users in the two groups. Fortunately, the content interest can be reflected by the entropy of topic distributions. For focused users, their content interest is uneven over topics, which makes their entropy of topic distribution lower. On the other hand, for diverse users, their topic entropy is higher because their topic distribution is even. This behavior is consistent with the acceptability of the two user groups. In this research, we use the average topic entropy of the users in a cluster as the spam threshold of the cluster to determine whether a user in the cluster is a spammer or not. σ_d and σ_f can be calculated as follows:

$$\sigma_d = \frac{\sum_{u \in U_d}(-\sum_{i=1}^{k} p(Z_i|u) log_2 p(Z_i|u))}{|U_d|^2}, \sigma_f = \frac{\sum_{u \in U_f}(-\sum_{i=1}^{k} p(Z_i|u) log_2 p(Z_i|u))}{|U_f|^2} \tag{9}$$

The third set of thresholds, α_d and α_f, is designed to detect spammers by using the mutual peer acceptance distance. The spam detection in stage 2 suffered from the problem of higher false-negative due to the high content similarity among spam campaigns. To address this problem, we introduced the mutual pairwise acceptance distance as the last part of Algorithm 2. It identifies spammers from the users who have been wrongly categorized as 'genuine' by the previous step (Line 16) of Algorithm 2. It is the main cause for the high false negatives. To this end, the average pairwise peer acceptance distance of users in each cluster is used as a threshold to detect spammers in each cluster. Let α_d and α_f denote the average pairwise peer acceptance distance for the diverse and focused clusters, respectively, α_d and α_f are calculated as follows:

$$\alpha_d = \frac{\sum_{u \in U_d} \sum_{v \in U_d} MPAD(u,v)}{|U_d|^2}, \alpha_f = \frac{\sum_{u \in U_f} \sum_{v \in U_f} MPAD(u,v)}{|U_f|^2} \tag{10}$$

Methodology

In this paper, we propose methods to extract features from users' tweets that represent users' information interests. Based on that users are divided into 'focused' and 'diverse' groups using clustering techniques. We then proposed methods to assess users' peer-acceptance based on the users' common information interests across different topics. Finally, we proposed an approach to identify spammers based on users' peer-acceptability. The evaluation of our approach is conducted in two parts as described below.

- First, we evaluate the features for deriving the focused and diverse user groups. Our objective was to prove the suitability of the selected features for user clustering. Three sets of features can be used for the clustering : GOSS, LOSS, and entropy features. There are multiple feature combinations possible for user clustering and the clustering quality can be different using different feature combinations. In the Section "Feature selection and Impact of clustering on spam detection", we use two evaluation metrics, the Silhouette Coefficient and the Davis Bouldin Score, to evaluate the quality of clusters for different feature combinations to determine the best feature combination.
- Then, we evaluate the performance of the proposed spam detection approach. We use the evaluation metrics, Accuracy, Precision, Recall and F-measure. Let TP denote true positive which is the number of correctly identified spammers, TN denote true negative, which is the number of correctly identified genuine users, FP denote false positive which is the number of genuine users who are wrongly identified as spammers, and FN denote false negative which is the number of spammers who are wrongly identified as genuine users. The four evaluation metrics are defined below:
  $\text{Accuracy} = (\text{TP} + \text{TN})/\ (\text{TP} + \text{TN} + \text{FP} + \text{FN})$
  $\text{Precision} = \text{TP}/(\text{TP} + \text{FP})$
  $\text{Recall} = \text{TP}/(\text{TP} + \text{FN})$
  $\text{F1} = 2 * \text{Precision} * \text{Recall}/(\text{Precision} + \text{Recall})$

Feature selection and impact of clustering on spam detection

In the proposed approach, users are clustered into two groups, diverse and focused groups. It is essential to select the best set of features for clustering by evaluating the quality of the clusters generated using different sets of features. We used three main criteria to measure the impact of clustering (with different feature combinations) on spam detection.

- Cluster quality evaluation metrics
- Accuracy of spam detection
- Average number of common topics

(1) Cluster quality evaluation metrics

In our experiment, the quality of clustering is evaluated by using two cluster quality evaluation metrics, Silhouette Coefficient [68] and the Davis Bouldin index [69]. The Davis-Bouldin Index evaluates intra-cluster similarity and inter-cluster differences where a lower score would provide a better clustering. The Silhouette Index measures how similar an object to its own cluster compared to other clusters. The higher Silhouette coefficient value would define a better clustering.

In our approach, we have three features to choose, which are the entropy of user topic distribution, LOSS and GOSS, as discussed in Section "Representation of user's information interest using LDA topic models". We tested different feature combinations for all the three datasets and used the two metrics to evaluate the quality of clustering. Table 1 provides the evaluation result for each feature combinations.The best result for each dataset is highlighted in bold.

Based on the clustering quality, from Table 1 we can see the LOSS + Entropy would be the best feature combination for clustering the users for Fake Project and HSpam14 datasets because these two features depend on the user's own content while GOSS depends on other users' content. Nevertheless, for the social honeypot dataset, the best feature combination for clustering is all the three features, i.e., LOSS + GOSS + entropy.

(2) Accuracy of spam detection The main purpose of clustering is to improve the detection results. Hence it is important to analyse how each feature combination would effect detection. Table 2 provides the detection performance results generated by our proposed unsupervised spam detection approach for each dataset under different feature combinations. The best detection accuracy results are achieved by using all three features, i.e., GOSS + LOSS + Entropy, as highlighted in Table 2.

(3) Average number of common topics We analysed the average number of common topics among the users in each cluster created based on different feature combinations. Table 3 contains the average number of common topics for each cluster with different feature combinations. The number of common topics effects the detection

Table 1 Results of cluster quality evaluation metrics

Dataset	Features	Silhouette coefficient	Davis Bouldin Score
Social Honeypot	GOSS + LOSS	0.133783289	3.619251614
	GOSS + LOSS + Entropy	**0.185440031**	3.878652971
	GOSS + Entropy	0.084125872	3.624939240
	LOSS + Entropy	0.148410937	**3.547088595**
The Fake Project	GOSS + LOSS	0.171869530	3.244455815
	GOSS + LOSS + Entropy	0.170521355	3.167608666
	GOSS + Entropy	0.171822136	3.125462915
	LOSS + Entropy	**0.180362480**	**2.941402933**
HSpam14	GOSS + LOSS	0.165421365	2.69321541
	GOSS + LOSS + Entropy	0.158961042	2.73569256
	GOSS + Entropy	0.129411538	2.51342332
	LOSS + Entropy	**0.172345361**	**2.49823176**

Table 2 Detection results for cluster feature combinations

Dataset	Features	Accuracy	Precision	Recall
Social Honeypot	GOSS + LOSS	0.934	0.92	0.91
	GOSS + LOSS + Entropy	**0.967**	**0.96**	**0.95**
	GOSS + Entropy	0.927	0.91	0.90
	LOSS + Entropy	0.950	0.94	0.93
The Fake Project	GOSS + LOSS	0.954	0.94	0.93
	GOSS + LOSS + Entropy	**0.969**	**0.95**	**0.96**
	GOSS + Entropy	0.947	0.94	0.91
	LOSS + Entropy	0.955	0.95	0.93
HSpam14	GOSS + LOSS	0.927	0.91	0.88
	GOSS + LOSS + Entropy	**0.949**	**0.92**	**0.93**
	GOSS + Entropy	0.923	0.91	0.86
	LOSS + Entropy	0.937	0.92	0.89

Table 3 Number of average common topics for different feature combination

Dataset	Features	Average number of common topics	Focused
		Diverse	
Social Honeypot	GOSS + LOSS	15	14
	GOSS + LOSS + Entropy	**18**	**17**
	GOSS + Entropy	17	15
	LOSS + Entropy	18	16
The Fake Project	GOSS + LOSS	18	16
	GOSS + LOSS + Entropy	**22**	**19**
	GOSS + Entropy	16	14
	LOSS + Entropy	20	17
HSpam14	GOSS + LOSS	21	16
	GOSS + LOSS + Entropy	**24**	**18**
	GOSS + Entropy	20	15
	LOSS + Entropy	23	18

accuracy. The clustering with higher number of common topics would give better detection results. Table 3 shows that the highest numbers of average common topics are from the combination of three features.

Out of the three criteria used, the combination of three features is the best for two of them. (Average number of common topics and accuracy of spam detection). Since two of the evaluation criteria was in favour of three feature combination, we choose the combination of GOSS + LOSS + Entropy, for clustering in our approach.

Evaluation of spam detection

The users' peer acceptance, calculated using Algorithm 1, is a basic measurement which can be used to detect spammers. To reduce the bias in user peer acceptance caused by users' different interest diversity, users are clustered into focused and diverse groups.

Table 4 Results of spam detection for Social Honeypot dataset

Model	Accuracy	Precision	Recall	F-measure
SDPA	0.864	0.81	0.84	0.83
SDPAC	0.915	0.92	0.86	0.89
Improvement% (SDPA → SDPAC)	5.9%	13.6%	2.4%	7.2%
SDPACM	0.967	0.96	0.95	0.96
Improvement% (SDPAC → SDPACM)	5.7%	4.3%	10.5%	7.9%

Table 5 Results of spam detection for "The Fake Project" dataset

Model	Accuracy	Precision	Recall	F-measure
SDPA	0.881	0.83	0.86	0.84
SDPAC	0.928	0.92	0.88	0.90
Improvement% (SDPA → SDPAC)	5.3%	10.8%	2.3%	7.1%
SDPACM	0.969	0.95	0.96	0.96
Improvement%(SDPA → SDPACM)	4.4%	3.3%	9.1%	6.7%

Integrated with clustering, the spam detection based on peer acceptance can increase the detection accuracy by 6 percent compared to the detection accuracy achieved by using the peer acceptance only, according to our experimental results to be reported below. As discussed in Section "Mutual peer acceptance", in order to address the high false negative problem caused by high content similarity in spam posts, the mutual peer acceptance distance is used to further differentiate spammers from genuine users. In this section, the effect to spam detection by using the peer acceptance, clustering, and the mutual peer acceptance distance will be evaluated separately. Accordingly, we have three proposed models to be evaluated in this section:

- SDPA: Spam Detection based on Peer Acceptance
- SDPAC: Spam Detection based on Peer Acceptance with Clustering
- SDPACM: Spam Detection based on Peer Acceptance with Clustering and Mutual peer acceptance distance

Two classification-based systems and K-means clustering algorithm as an unsupervised algorithm were chosen as the baseline models for the evaluation.

- DNNBD: Deep Neural Networks for Bot Detection [35].
- SMD: Seven Months with the Devils [18].
- K-mean: K-means clustering algorithm.

Both systems used different supervised classification-based approaches with various feature combinations for their detection. We tested DNNBD for their best performing model (AdaBoost Classifier With SMOTENN). Compared to the results mentioned in the paper, the results we obtained for our own implementation is little lower. We think that this due to the selection of a subset of users from the original dataset with a smaller

Table 6 Results of spam detection for Hspam14 dataset

Model	Accuracy	Precision	Recall	F-measure
SDPA	0.840	0.75	0.80	0.77
SDPAC	0.900	0.86	0.84	0.85
Improvement% (SDPA → SDPAC)	7.1%	14.7%	5.0%	10.4%
SDPACM	0.949	0.92	0.93	0.93
Improvement% (SDPAC → SDPACM)	5.4%	7.0%	10.7%	9.4%

Table 7 Results comparison with the baseline models

Dataset	Model	Accuracy	Precision	Recall	F-measure
Social Honeypot	SDPACM	0.967	0.96	0.95	0.96
	DNNBD	0.975	0.96	0.95	0.96
	SMD	0.982	0.97	0.98	0.98
	K-Means	0.704	0.53	0.48	0.50
The Fake Project	SDPACM	0.969	0.95	0.96	0.96
	DNNBD	0.979	0.98	0.97	0.98
	SMD	0.984	0.99	0.98	0.98
	K-Means	0.796	0.57	0.89	0.70
Hspam14	SDPACM	0.949	0.92	0.93	0.93
	DNNBD	0.977	0.96	0.97	0.96
	SMD	0.978	0.98	0.95	0.97
	K-Means	0.662	0.63	0.49	0.65

number of tweets. Original dataset has more spammers than genuine users and contains many non-English tweets. 10-fold cross-validation is used in the experiments. When selecting clustering algorithms as a baseline, we used both DBScan and K-Means where K-means gave us a better result compared to DBScan. Tables 4, 5, 6, 7 provide the experiment results for the three datasets. The results are generated based on 90%:10% training to testing separation. From the results, we can see, peer acceptance alone (i.e., model SDPA) did not provide a higher accuracy where the accuracy is between 84–88% in all three datasets. The analysis of results indicated that false positives are higher. This is due to the diversified content interest in users. Both spam and genuine users have a diversified content interest. But the genuine users have a higher diversified content interest compare to spammers. To address this limitation, we extended our model with clustering. Using peer acceptance plus clustering (i.e., model SDPAC), we were able to increase the accuracy of the detection by around 5–7% by eliminating a set of false positives. The model is further extended by integrating the mutual pairwise distance, which is the SDPACM model. The accuracy of the SDPACM model is 0.967, 0.969, and 0.949 for the three datasets, respectively, which are quite close to the best performance achieved by the classification-based model SMD. Using peer acceptance and clustering plus mutual peer acceptance distance (model SDPACM), we were able to increase the accuracy around 4–5% by eliminating a set of false negatives. Table 7 provides the comparison of results with the two classification-based baseline systems DNNBD and SMD.

Even though the performances are not better than the baseline systems, they are much closer to the classifiers' results. The gap is less than 3% for all three datasets.

The Precision improvement is 10–15 % while Recall improvement is 9–10% for all three datasets. It is evident from the improvements that false positives are reduced by SDPAC, and False negatives are reduced by SDPACM.

Figure 8 depicts the comparison of results obtained for three datasets for three different models introduced in our approach.

Discussion

The approach introduced in this paper is a pure unsupervised model for spam detection which does not require a labelled dataset. It can be considered as a successful alternative for traditional supervised spam detection techniques. One novelty of the paper is the introduction of the new concept of Peer Acceptance into spam detection models. Peer acceptance is a widely recognized concept in areas of sociology and psychology [5, 12, 13]. It is the first time that the peer acceptance concept is used in spam detection models. People often believe reliable users and their post contents [70, 71]. Our peer acceptance methods can be extended to find reliable users and their posts. Moreover, the proposed approach can be modified to determine the change of user's content interest overtime so that it can be utilized to address the spam drifting problem. The discussion includes details and observations on peer acceptance, clustering and the features used to cluster users.

Peer acceptance based on shared information interest

The research is based on the assumption that "genuine users' post content is consistent with the topic of the posts". Therefore, a set of users who used the same topic should have a consistent posting behaviour, and their post content should be relevant to the topic. The same consistent behaviour should exist over multiple topics. Spammers' behaviour is inconsistent across multiple topics. Spammers often insert frequent topics of the social network to increase their reachability. Thus, spammers post relevant content to some topics while a majority of their post content is irrelevant to the topic

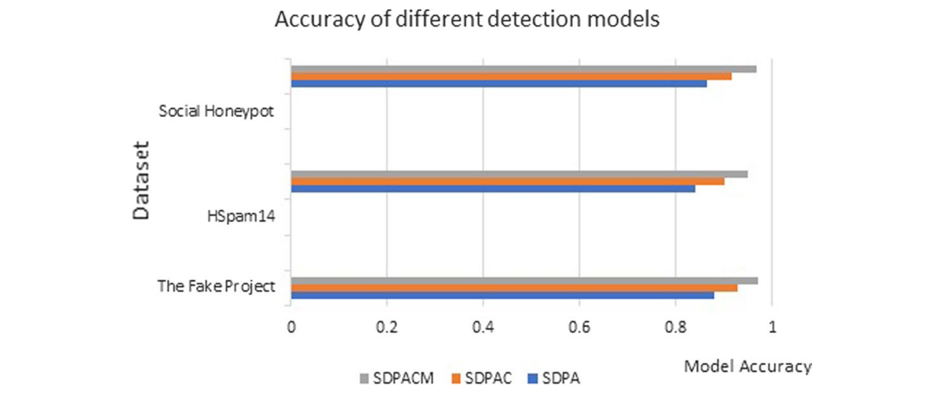

Fig. 8 Accuracy of different detection models

of the post. We generate our peer acceptance model based on this assumption and our observation of users' behavior reflected in the datasets. Peers will reject a user who displays contradictory common shared interest across multiple topics. Thus, that user is considered as a suspicious user in the network. We used the overall peer acceptability of a user to identify spammers in a social network. There are multiple thresholds used in our approach. For the pairwise peer acceptance threshold β_f and β_d, we used the average peer acceptance of the users in each cluster because it is an indication of the average content interest change over multiple topics of the users in the same cluster. For the overall peer acceptability threshold σ_f and σ_d, we used the average entropy of the users' topic distribution in a cluster. The entropy is higher for the diverse users and it is low for the focused users. Our unsupervised spam detection approach would be a competitive alternative to existing supervised classification-based spam detection approaches. Traditional supervised spam detection approaches require large training datasets. Our approach is unique in that it does not include any labelled datasets. This is vital for spam detection approaches where spammers change their behaviors over time. Further analysis of the results indicates that a set of genuine users have detected as spammers which made a set of higher false positives.

We strongly believe that the presence of false positives might be due to the following reasons.

- Being a member of the social network, some users post a smaller number of posts and they use fewer topics in their post content. They may have focused on only a few topics. Since they do not have much content to be shared with other users, the pairwise peer acceptance will be very low. As a result, they have a low level of peer acceptance in the social network. Thus, they will become unreliable to many of their peers and hence considered as spammers.
- We believe that these users are isolated users with less social interactions in the social network. But often they are not spammers. It is difficult to calculate the shared information interest for such users. We tried to address this problem to a certain extent by introducing clustering before applying peer acceptance.

The clustering will separate the users into focused and diverse clusters where we employ different thresholds to calculate the peer acceptability of users in each cluster. Frequent words and their synonyms are used to represent the user's content. An enhanced content representation can be used to improve the accuracy of the detection. As a result of diversified content and topic usage in social media interactions, there were topics with a few frequent words. Empirical results show that the representation of such topics is not strong compared to topics with more frequent words. For the users who used those topics, their content similarity would be high. As a result, they ended up with a higher peer acceptability with other users in the same cluster. The analysis of such users shows that they have used similar or closely related contents under the same topic. Spammers tend to send the same spam message or content with different frequent topics used in the social network. To address this problem, we have used the mutual peer acceptance distance. The analysis shows that spammer pairs have a lower mutual peer acceptance distance compared to genuine user pairs or genuine-spam user pairs.

Two-stage approach for spam detection

The common shared interest-based peer acceptance is used for spam detection in the early stage of our approach. The detection accuracies were around 84-88 %. To handle the higher false negatives and false positives we had in stage one, the approach was extended with another stage. Stage 2 addresses the problem of higher false positives of genuine users identified as spammers. The analysis indicates that the approach can be biased when a user does not have many topics used in their posts. This results in a smaller number of common topics between pairs of users. Some other users may have interested in very few selected areas where their topic usage is very low. Hence, we have extended our approach to determine the user's topic interest to avoid bias in the proposed approach. An LDA based topic model is used for clustering. The clustering was essential to determine whether a user's information interest is diverse or focused. The feature selection for clustering was challenging since we need to design content-based features which can be used to detect the user's focused or diverse interest. Among the three features used for clustering, entropy is directly related to the user's content distribution. Higher entropy of topic distribution indicates that the user has an evenly distributed interest over the topics, while users who have a diverse interest show lower entropy of their topic distribution, indicating that the user is interested in some of the topics, but not in the others. Such users are focused users who are only interested in very few topics. The results of the clustering evaluation were in favor of features LOSS and Entropy. However, for Social the Honeypot dataset, GOSS + LOSS + Entropy were the best features for clustering. Further, we analyzed the same feature combinations for the detection results. The detection accuracies indicate that the best results are obtained by using a combination of the three features, LOSS, GOSS, and Entropy. Even though the cluster quality is in favor of LOSS + Entropy, we achieved higher results for a combination of three features. Since the detection is most important, we choose all three features for the clustering. In Stage 2, the problem of higher false negatives due to higher content similarity by spam users was also addressed. We used the mutual pairwise peer acceptance to filter some spammers who were identified as genuine users by using peer acceptance only.

Datasets for experiments

The datasets used for the experiments are publicly available datasets. We had a filtering criterion for our experiments. By analyzing the post distribution in each dataset, we set a filter of the minimum number of posts. Some users in the data set have very few posts, and they were removed from the dataset. At the same time, spammers' posts are often higher than the genuine users. The average post count is high for spammers in all three datasets. The frequent topics in all three datasets were not evenly distributed among two user groups. Some topics were mainly used by the spammers. It is essential to have a considerable amount of post content for content profiles. In the topic selection, we considered a minimum of five common topics for each user. The datasets were labelled using a set of non-content features such as follower/followee count, number of URLs, account features etc. This has negatively impacted on our experiments. Because our features are totally content-based, and the detection is also done on content-based features. The usage of hashtags in the dataset was also a concern for the experiments. Spammers have a higher usage of hashtags compared to genuine users in their posts.

Impact of mutual peer acceptance distance (MPAD)

Analysis indicates that the mutual peer acceptance for genuine-genuine or spam-genuine users are highly diversified, and their average MPAD is much higher compared to the spam-spam user group. Such a diversified average distance is expected since their content is different. As mentioned earlier, this behavior does not exist in spammer pairs due to their similar content usage. Compared to the peer acceptability of a user, MPAD is not such a strong filter to identify spammers. This is due to the variation of the spam content usage. Spam groups or campaigns often use similar or near duplicate tweets. At the same time, there are some other spammers who did not exhibit such behaviors. Further, compared to genuine-genuine and spam-genuine pairs, the spam-spam group shows a much smaller average distance. Hence, we used that as a measure to detect spammers in our final stage. The strength of this measure could be reduced with a more diversified set of spammers present in the community. According to our investigation, this measure is useful since the first two stages of the approach also try to distinguish the spammers from genuine users. The results indicate that MPA + PA could be still useful as a feature for spam campaign detection.

Algorithm complexity

Algorithm 1 generates the peer acceptance matrix which provides the peer acceptance values between every pair of users. For a dataset with m users and c chosen hashtags, the algorithm complexity is O(cm^2). Not losing generality, we can treat c as a constant number once the frequent hashtags are determined. In this case, the complexity of Algorithm 1 is O(m^2).

The LDA topic modelling method was used to generate user interest representations based on which users are grouped into the focused and diverse clusters. The complexity of the LDA method is O(km) [72] where k is the number of topics and m is the number of users.

Algorithm 2 is for spammer detection. At the beginning of Algorithm 2, Algorithm 1 is called to generate the user peer acceptance matrix with complexity of O(m^2). For each testing user, the users in either the focused or the diverse cluster will be compared with the testing user to determine whether the testing user is a spammer or not. At the worst case, the complexity of this part is O(m2). Overall, the complexity of Algorithm 2 would be O(m^2).

Conclusion

A novel two-stage unsupervised spam detection approach is introduced in the research. User's information interest is derived through users' post content. Peer acceptance of a user in an OSN is derived from the user's information interest, and it is used to detect spammers. Notably, no labelled training datasets are needed for our method. Three publicly available datasets were used for experiments, and results were compared with two of the best classification-based systems. The results are close, and the highest accuracy achieved is 96.9%. It is much closer to the best performance of the classification-based system with an accuracy of 98.4%. We received a promising set of results. As per our investigation, the inaccuracy is due to misrepresentation of users with fewer topics. In our approach, we used frequent words with synonyms to represent the user's information interest. Set of users are often considered dissimilar to other users since they use very few topics and have not shared much of the content with users. Such users will be considered unacceptable to their peers in the final detection. We assume, genuine users have a higher shared information interest compared to spammers. The overall peer acceptability of a spammer should be lower. In an extreme situation where an ordinary person has a different mindset to the common opinion in all or the majority of his topics, the system may incorrectly identify that user as a spammer. But when the number of topics used is large, probability of such a situation is very low. Though we tested our approach only with Twitter, the same approach can be used in any other online social networks. The advantages of the system can be described as follows. The approach does not require labelled dataset. Since it is unsupervised, a new training model is not required when the data is changed. Our approach is a pure content-based approach with a limited set of features. Finally it delivers a pure unsupervised approach for spammer detection.

Acknowledgements
Not applicable.

Authors' contributions
DK: investigation, methodology, data curation, software, validation, writing-original draft. YX: conceptualization, methodology, formal analysis, validation, writing-review & editing, supervision. EF: methodology, validation, writing-review & editing, supervision. All authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
The data that support the findings of this study are available at following repositories, Social Honeypot: http://infolab.tamu.edu/data/. The Fake Project: http://mib.projects.iit.cnr.it/dataset.html. HSpam14: https://personal.ntu.edu.sg/axsun/datasets.html. But restrictions apply to the availability of these data.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]School of Computer Science, Queensland University of Technology, Brisbane, Australia. [2]School of Information and Communication Technology, Griffith University, Brisbane, Australia.

Received: 11 August 2021 Accepted: 19 December 2021
Published online: 10 January 2022

References

1. Hinesley K. A reminder about spammy behaviour and platform manipulation on twitter. Twitter: Technical report; 2020.
2. Hua W, Zhang Y. Threshold and associative based classification for social spam profile detection on twitter. In: 2013 Ninth International Conference on Semantics, Knowledge and Grids; 2013. p. 856–864.

3. Dang Q, Zhou Y, Gao F, Sun Q. Detecting cooperative and organized spammer groups in micro-blogging community. Data Mining Knowl Discov. 2016;31(3):573–605. https://doi.org/10.1007/s10618-016-0479-5.
4. Gao H, Hu J, Wilson C, Li Z, Chen Y, Zhao BY. Detecting and characterizing social spam campaigns. In: Proceedings of the 10th Annual Conference on Internet Measurement - IMC '10; 2010.
5. Cao C, Caverlee J. Behavioral detection of spam URL sharing: posting patterns versus click patterns. In: 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014); 2014.
6. Rao S, Verma AK, Bhatia T. A review on social spam detection: challenges, open issues, and future directions. Expert Syst With Appl. 2021;186:11572. https://doi.org/10.1016/j.eswa.2021.115742.
7. Neisari A, Rueda L, Saad S. Spam review detection using self-organizing maps and convolutional neural networks. Comput Security. 2021;106:102274. https://doi.org/10.1016/j.cose.2021.102274.
8. Sarr J-MA, Brochier T, Brehmer P, Perrot Y, Bah A, Sarré A, Jeyid MA, Sidibeh M, Ayoubi SE. Complex data labeling with deep learning methods: lessons from fisheries acoustics. ISA Trans. 2021;109:113–25. https://doi.org/10.1016/j.isatra.2020.09.018.
9. McPherson M, Smith-Lovin L, Cook JM. Birds of a feather: homophily in social networks. Ann Rev Sociol. 2001;27(1):415–44. https://doi.org/10.1146/annurev.soc.27.1.415.
10. Cardoso FM, Meloni S, Santanche A, Moreno Y. Topical alignment in online social systems. Front Phys. 2019. https://doi.org/10.3389/fphy.2019.00058.
11. Weng J Lim E-P, Jiang J, He Q. TwitterRank. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining - WSDM '10; 2010.
12. Sykes TA, Venkatesh V, Gosain S. Model of acceptance with peer support: a social network perspective to understand employees' system use. MIS Q. 2009;33:371–93. https://doi.org/10.2307/20650296.
13. Asher SR, Parkhurst JT, Hymel S. Peer rejection and loneliness in childhood. In: Asher SR, Coie JD, editors. Peer rejection in childhood. Cambridge: Cambridge University Press; 1990.
14. Gurucharri C, Selman RL. The development of interpersonal understanding during childhood, preadolescence, and adolescence: a longitudinal follow-up study. Child Dev. 1982;53(4):924. https://doi.org/10.2307/1129129.
15. Sherchan W, Nepal S, Paris C. A survey of trust in social networks. ACM Comput Surveys. 2013;45(4):1–33. https://doi.org/10.1145/2501654.2501661.
16. Lewis JD, Weigert A. Trust as a social reality. Social Forces. 1985;63(4):967. https://doi.org/10.2307/2578601.
17. Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. J Mach Learn Res. 2003;3:993–1022.
18. Lee K, Eoff BD, Caverlee J. Seven months with the devils: a long-term study of content polluters on twitter. In: Adamic LA, Baeza-Yates R, Counts S, editors. ICWSM; 2011.
19. Sedhai S, Sun A. HSpam14. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2015.
20. Cresci S, Pietro RD, Petrocchi M, Spognardi A, Tesconi M. The paradigm-shift of social spambots. In: Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17 Companion; 2017.
21. Al-garadi MA, Varathan KD, Ravana SD. Cybercrime detection in online communications: the experimental case of cyberbullying detection in the twitter network. Comput Hum Behav. 2016;63:433–43. https://doi.org/10.1016/j.chb.2016.05.051.
22. Paoli SD. Not all the bots are created equal: the ordering turing test for the labeling of bots in MMORPGs. Social Media Soc. 2017;3(4):205630511774185. https://doi.org/10.1177/2056305117741851.
23. Goswami K, Park Y, Song C. Impact of reviewer social interaction on online consumer review fraud detection. J Big Data. 2017. https://doi.org/10.1186/s40537-017-0075-6.
24. Thomas K, Grier C, Song D, Paxson V. Suspended accounts in retrospect. In: Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference - IMC '11; 2011.
25. Zhu Y, Wang X, Zhong E, Liu NN, Li H, Yang Q. Discovering spammers in social networks. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence; 2012. p. 171–177.
26. Grier C, Thomas K, Paxson V, Zhang M. @spam. In: Proceedings of the 17th ACM Conference on Computer and Communications Security - CCS '10; 2010.
27. Neudert L-M, Howard P, Kollanyi B. Sourcing and automation of political news and information during three European elections. Social Media Soc. 2019;5(3):205630511986314. https://doi.org/10.1177/2056305119863147.
28. Wang W, Zeng G, Tang D. Using evidence based content trust model for spam detection. Expert Syst With Appl. 2010;37(8):5599–606. https://doi.org/10.1016/j.eswa.2010.02.053.
29. Yang C, Harkreader R, Zhang J, Shin, S Gu G. Analyzing spammers' social networks for fun and profit. In: Proceedings of the 21st International Conference on World Wide Web - WWW '12; 2012.
30. Chu Z, Widjaja I, Wang H. Detecting social spam campaigns on twitter. In: Applied cryptography and network security; 2012. p. 455–472. https://doi.org/10.1007/978-3-642-31284-7_27.
31. Wu C-H. Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks. Expert Syst With Appl. 2009;36(3):4321–30. https://doi.org/10.1016/j.eswa.2008.03.002.
32. Sarker IH, Kayes ASM, Badsha S, Alqahtani H, Watters P, Ng A. Cybersecurity data science: an overview from machine learning perspective. J Big Data. 2020. https://doi.org/10.1186/s40537-020-00318-5.
33. Li CH, Yang JC, Park SC. Text categorization algorithms using semantic approaches, corpus-based thesaurus and WordNet. Expert Syst With Appl. 2012;39(1):765–72. https://doi.org/10.1016/j.eswa.2011.07.070.
34. Mabotuwana T, Lee MC, Cohen-Solal EV. An ontology-based similarity measure for biomedical data–application to radiology reports. J Biomed Inform. 2013;46(5):857–68. https://doi.org/10.1016/j.jbi.2013.06.013.
35. Kudugunta S, Ferrara E. Deep neural networks for bot detection. Inform Sci. 2018;467:312–22. https://doi.org/10.1016/j.ins.2018.08.019.
36. El-Mawass N, Honeine P, Vercouter L. SimilCatch: enhanced social spammers detection on twitter using markov random fields. Inform Process Manage. 2020;57(6):102317. https://doi.org/10.1016/j.ipm.2020.102317.

37. Yu D, Chen N, Jiang F, Fu B, Qin A. Constrained NMF-based semi-supervised learning for social media spammer detection. Knowl Based Syst. 2017;125:64–73. https://doi.org/10.1016/j.knosys.2017.03.025.
38. Pirró G. A semantic similarity metric combining features and intrinsic information content. Data Knowl Eng. 2009;68(11):1289–308. https://doi.org/10.1016/j.datak.2009.06.008.
39. Breiman L. Classification regression trees. New York: Chapman & Hall; 1993.
40. Xu N, Huo C, Zhang X, Cao Y, Meng G, Pan C. Dynamic camera configuration learning for high-confidence active object detection. Neurocomputing. 2021;466:113–27. https://doi.org/10.1016/j.neucom.2021.09.037.
41. An R, Xu Y, Liu X. A rough margin-based multi-task v-twin support vector machine for pattern classification. Appl Soft Comput. 2021;112:107769. https://doi.org/10.1016/j.asoc.2021.107769.
42. Gao W, Wan F, Yue J, Xu S, Ye Q. Discrepant multiple instance learning for weakly supervised object detection. Pattern Recognit. 2022;122:108233. https://doi.org/10.1016/j.patcog.2021.108233.
43. Zhang J, Su H, Zou W, Gong X, Zhang Z, Shen F. CADN: a weakly supervised learning-based category-aware object detection network for surface defect detection. Pattern Recognit. 2021. https://doi.org/10.1016/j.patcog.2020.107571.
44. Yadav SP. Vision-based detection, tracking, and classification of vehicles. IEIE Trans Smart Process Comput. 2020;9(6):427–34. https://doi.org/10.5573/ieiespc.2020.9.6.427.
45. Arulprakash E, Aruldoss M. A study on generic object detection with emphasis on future research directions. J King Saud Univ Comput Inform Sci. 2021. https://doi.org/10.1016/j.jksuci.2021.08.001.
46. Cheplygina V, de Bruijne M, Pluim JPW. Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. Med Image Anal. 2019;54:280–96. https://doi.org/10.1016/j.media.2019.03.009.
47. Muruganantham P, Balakrishnan SM. A survey on deep learning models for wireless capsule endoscopy image analysis. Int J Cogn Comput Eng. 2021;2:83–92. https://doi.org/10.1016/j.ijcce.2021.04.002.
48. Yadav SP, Mahato DP, Linh NTD. Distributed artificial intelligence. 2020. https://doi.org/10.1201/9781003038467.
49. Lauriola I, Lavelli A, Aiolli F. An introduction to deep learning in natural language processing: models, techniques, and tools. 2021. https://doi.org/10.1016/j.neucom.2021.05.103.
50. Torfi A, Shirvani RA, Keneshloo Y, Tavaf N, Fox EA. Natural language processing advancements by deep learning: a survey. 2020. http://arxiv.org/abs/2003.01200.
51. Babić K, Martinčić-Ipšić S, Meštrović A. Survey of neural text representation models. Information. 2020;11(11):511. https://doi.org/10.3390/info11110511.
52. Ligthart A, Catal C, Tekinerdogan B. Analyzing the effectiveness of semi-supervised learning approaches for opinion spam classification. Appl Soft Comput. 2021;101:107023. https://doi.org/10.1016/j.asoc.2020.107023.
53. Crawford M, Khoshgoftaar TM, Prusa JD, Richter AN, Najada HA. Survey of review spam detection using machine learning techniques. J Big Data. 2015. https://doi.org/10.1186/s40537-015-0029-9.
54. Kaur R, Singh S, Kumar H. Rise of spam and compromised accounts in online social networks: a state-of-the-art review of different combating approaches. J Netw Comput Appl. 2018;112:53–88. https://doi.org/10.1016/j.jnca.2018.03.015.
55. Latah M. Detection of malicious social bots: a survey and a refined taxonomy. Expert Syst Appl. 2020;151:113383. https://doi.org/10.1016/j.eswa.2020.113383.
56. Abkenar SB, Kashani MH, Akbari M, Mahdipour E. Twitter spam detection: a systematic review. 2020. http://arxiv.org/abs/2011.14754.
57. Hussain N, Mirza HT, Hussain I, Iqbal F, Memon I. Spam review detection using the linguistic and spammer behavioral methods. IEEE Access. 2020;8:53801–16. https://doi.org/10.1109/access.2020.2979226.
58. Corbett-Davies S, Goel S. The measure and mismeasure of fairness: a critical review of fair machine learning. 2018. http://arxiv.org/abs/1808.00023.
59. Malik MM. A hierarchy of limitations in machine learning. 2020. http://arxiv.org/abs/2002.05193.
60. Yousukkee S, Wisitpongphan N. Analysis of spammers' behavior on a live streaming chat. IAES Int J Artif Intell. 2021;10(1):139. https://doi.org/10.11591/ijai.v10.i1.pp139-150.
61. Sedhai S, Sun A. An analysis of 14 million tweets on hashtag-oriented spamming*. J Assoc Inform Sci Technol. 2017;68(7):1638–51. https://doi.org/10.1002/asi.23836.
62. Tang X, Qian T, You Z. Generating behavior features for cold-start spam review detection with adversarial learning. Inform Sci. 2020;526:274–88. https://doi.org/10.1016/j.ins.2020.03.063.
63. Zhuang L, Jing F, Zhu X-Y. Movie review mining and summarization; 2006.
64. Diao Q, Qiu M, Wu C-Y, Smola AJ, Jiang J, Wang C. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS); 2014. https://doi.org/10.1145/2623330.2623758.
65. Weng L, Menczer F. Topicality and impact in social media: diverse messages, focused messengers. PLOS ONE. 2015;10(2):0118410. https://doi.org/10.1371/journal.pone.0118410.
66. Liu L, Lu Y, Luo Y, Zhang R, Itti L, Lu J. Proceedings of the NAACL student research workshop. In: Detecting "Smart" spammers on social network: a topic model approach. Association for Computational Linguistics; 2016.
67. Hoffman MD, Blei DM, Bach F. Online learning for latent dirichlet allocation. In: Proceedings of the 23rd International Conference on Neural Information Processing Systems, vol 1. NIPS'10. Curran Associates Inc., Red Hook, NY; 2010. p. 856–864.
68. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math. 1987;20:53–65. https://doi.org/10.1016/0377-0427(87)90125-7.
69. Davies DL, Bouldin DW. A cluster separation measure. IEEE Trans Pattern Anal Mach Intell. 1979;1(2):224–7. https://doi.org/10.1109/tpami.1979.4766909.
70. Yao X, Liang G, Gu C, Huang H. Rumors clarification with minimum credibility in social networks. Comput Netw. 2021;193:108123. https://doi.org/10.1016/j.comnet.2021.108123.

71. Westerman D, Spence PR, Heide BVD. A social network as information: the effect of system generated reports of connectedness on credibility on twitter. Comput Hum Behav. 2012;28(1):199–206. https://doi.org/10.1016/j.chb.2011.09.001.
72. Wei X, Croft WB. LDA-based document models for ad-hoc retrieval. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval—SIGIR '06; 2006.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.