Information Processing and Management 57 (2020) 102140

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Opinion spam detection: Using multi-iterative graph-based model

Shirin Noekhah[a,b,*], Naomie binti Salim[a], Nor Hawaniah Zakaria[a]

[a] School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, UTM, 81300, Johor, Malaysia
[b] Department of Computer Science and Engineering, College of Engineering, Komar University of Science and Technology, KUST, Sulaimani, Iraq

ARTICLE INFO

Keywords:
Opinion spam detection
Heterogeneous graph-based structure
Spammer
Group of spammers
Feature fusion

ABSTRACT

The demand to detect opinionated spam, using opinion mining applications to prevent their damaging effects on e-commerce reputations is on the rise in many business sectors globally. The existing spam detection techniques in use nowadays, only consider one or two types of spam entities such as review, reviewer, group of reviewers, and product. Besides, they use a limited number of features related to behaviour, content and the relation of entities which reduces the detection's accuracy. Accordingly, these techniques mostly exploit synthetic datasets to analyse their model and are not able to be applied in the context of the real-world environment. As such, a novel graph-based model called "Multi-iterative Graph-based opinion Spam Detection" (MGSD) in which all various types of entities are considered simultaneously within a unified structure is proposed. Using this approach, the model reveals both implicit (i.e., similar entity's) and explicit (i.e., different entities') relationships. The MGSD model is able to evaluate the 'spamicity' effects of entities more efficiently given it applies a novel multi-iterative algorithm which considers different sets of factors to update the spamicity score of entities. To enhance the accuracy of the MGSD detection model, a higher number of existing weighted features along with the novel proposed features from different categories were selected using a combination of feature fusion techniques and machine learning (ML) algorithms. The MGSD model can also be generalised and applied in various opinionated documents due to employing domain independent features. The output of the MGSD model showed that our feature selection and feature fusion techniques showed a remarkable improvement in detecting spam. The findings of this study showed that MGSD could improve the accuracy of state-of-the-art ML and graph-based techniques by around 5.6% and 4.8%, respectively, also achieving an accuracy of 93% for the detection of spam detection in our synthetic crowdsourced dataset and 95.3% for Ott's crowdsourced dataset.

1. Introduction

Online reviews by consumers in commercial websites continue to have an undeniable effect on the sale of products and services through posting both positive and negative reviews to fame/defame the business. Seen as an opportunity, many companies exploit this situation by offering promotions or hiring people to post fake reviews, called spam reviews either for the company's products or relating to the products of competitors. Given there is no controlling mechanism in opinionated websites for detecting and filtering spam such as product and service reviews (Jindal & Liu, 2007), they can be easily be generated. Unlike other spam types (e.g., search engines, email, recommendation systems, blogs, online tagging, social networks, web forums, SMS, click bots and bot-generated search traffic); opinion spam is sneaky and varies due to the diverse behaviour of reviewers. Even though online reviews offer a

* Corresponding author at: SCRG research group, N28 a, School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia
E-mail addresses: nshirin2@live.utm.my (S. Noekhah), naomie@utm.my (N.b. Salim), hawaniah@utm.my (N.H. Zakaria).

https://doi.org/10.1016/j.ipm.2019.102140
Received 31 July 2018; Received in revised form 4 October 2019; Accepted 7 October 2019
Available online 18 October 2019
0306-4573/ © 2019 Elsevier Ltd. All rights reserved.

valuable, if the not, an important source of information, the credibility and reliability of reviews are often neglected or ignored by both companies and researchers.

Importantly, companies use consumer reviews to analyse their feedback regarding their products and services and to observe market trends. Whereas, researchers examine the nature and type of reviews to create different opinion mining applications such as opinion summarisation, feature-based opinion mining, sentiment analysis and classification, and opinion extraction. Nowadays, spamming is a widespread phenomenon in opinionated websites. It has been reported that more than 33% of reviews on the Internet are spam reviews and is increasing (Ye & Akoglu, 2015). As such, the detection of spam due to cleansing these opinionated documents has become an essential pre-processing step in developing opinion mining applications and for market analysis.

Furthermore, over the last decade, a considerable number of techniques have been proposed for opinion spam detection (Shehnepoor, Salehi, Farahbakhsh, & Crespi, 2017). Machine learning (ML), (i.e., supervised, unsupervised and semi-supervised) is the most popular approach for opinion spam detection by applying diverse types of features. These features include behaviour (e.g., the review's rating, the number of feedback and date of review), content (e.g., the reviewer's sentiment, POS tags and similarity rate) and relation-based (e.g., the number of reviews by the reviewer, burstiness of the reviewer and average rating of the product or services) features (Ott, Choi, Cardie, & Hancock, 2011).

Even though the results applying ML techniques are quite prominent, they suffer from the lack of acquiring a gold-standard labelled dataset, unbalanced data samples (Al Najada & Zhu, 2014), requiring a large number of labelled datasets (Mukherjee, Liu, & Glance, 2012) and a human evaluator (although not entirely accurate in this regard) (Ott et al., 2011). However, most of the techniques mentioned above only focus on the content features of the review. Whereas, in some opinionated websites, such as the 'Apple App Store', review content is unavailable (Savage, Zhang, Yu, Chou, & Wang, 2015). Importantly, pre-processing is a critical phase which affects the performance of any opinion spam detection technique. The effect of applying various standard Natural Language Processing (NLP) pre-processing steps on the performance of existing ML algorithms (i.e., Support Vector Machine (SVM) and Naïve Bayes (NB)) has been investigated in Etaiwi and Naymat (2017).

In order to detect spam reviews and to extract the topics associated with the reviews and sentiments, as in Dong, Ji et al. (2018), an unsupervised topic-sentiment joint probabilistic model (UTSJ) based on Latent Dirichlet Allocation (LDA) model was proposed. Although the focus was on the semantic structure of reviews, the results demonstrated the efficiency of the model compared with other baseline linguistic-based models. Existing work in this area has mainly focused on employing classical classification techniques. However, in Dong, Yao et al. (2018), an end-to-end supervised joint model was proposed which combined the autoencoder and neural random forest method for opinion spam detection. This work also tends to evaluate the effect of features and adjust different parameters in respective models where the results proved that these methods outperformed other so-called state-of-the-art techniques achieving an accuracy of 96%. A unified unsupervised model has also been proposed in Liu and Pang. (2018) based on the abnormities and deviations that exist in the behaviour of spammers compared to normal users. Here, a set of abnormity signals related to target-dependent sentiment information were proposed to model the latent content deviation in the 'Amazon' review dataset.

Aside from ML approaches, some studies have used a graph-based approach to detect opinion spam entities, simultaneously. For instance, Wang, Xie, Liu, and Yu (2011) proposed a novel heterogeneous graph, for the first time, based on the honesty of the review, trustiness of the reviewer and reliability of the score. However, this method ignored the review content and intra-relationships of the entities. On the other hand, Ye and Akoglu developed a two-step scalable network-based method to identify the group of spammers along with their targeted products based on their network footprints (Ye & Akoglu, 2015). However, this model produced unrealistic assumptions of reviewer centralities in the network. The problem of loose spammer group detection has also been addressed by Wang et al. through exploiting bipartite graph projection (Wang, Hou, Song, Li, & Kong, 2016). This model only focused on group spammer detection. Even though many researchers have investigated opinion spam detection, there remains the need to acquire a comprehensive model to cover existing limitations and weaknesses.

1.1. Research objectives and contributions

Feature engineering (i.e., feature extraction along with a feature selection process) remains one of the significant challenges in the detection process to identify a combination of features. Features, either related to the content or behaviour of entities (including the review, reviewer, product and group of reviewers), do not have the same role and standing in opinion spam detection. MGSD (Multi-iterative Graph-based opinion Spam Detection) model on the other hand, efficiently assigns a weight to the features according to their importance during the detection process. In this research, a lower weight is assigned to low-impact features (e.g., domain-dependent features), and a higher weight is given to high-impact features. Ten novel features (e.g., Review Group agreement, Sentiment-Rate difference, and Similar rating reviews) are proposed to extract precise information from exiting entities. Aside from that, in this model, twenty (20) of the most important sets of existing features detected by supervised classifiers, such as SVM and NB, are applied. The feature fusion technique is also used to determine the most effective and helpful combination of features. To our knowledge, this is the first time that this number of features has been applied to track spamming activities from the different perspectives of entities.

The domain dependency of features is a further issue in opinion spam detection techniques. Though some techniques cannot be generalised since they exploit the features which are either domain dependent (e.g., AVP Choo, Yu, & Chi, 2017, helpful ratings Goswami, Park, & Song, 2017) or not publicly available (e.g., IP address Li, Chen, Liu, Wei, & Shao, 2014 and spatial patterns Li, Chen, Mukherjee, Liu, & Shao, 2015). One of the major significances of the MGSD model is that its features are domain-independent, which contributes to the flexibility of the model in applying in different domains.

Furthermore, given each entity is not isolated from other entities, either the same or different type(s), relation-based features can reveal the inter- and intra-relationships that exist among the entities which are usually ignored by existing opinion spam detection

techniques. Since there is no comprehensive model which can evaluate all entities in a unified structure, the primary objective of this study is to propose a heuristic graph-based opinion spam detection model. Also, given MGSD is an unsupervised model, so there is no need for a gold-standard labelled dataset, which in itself is a challenge in existing supervised techniques. The proposed model is expected to reveal more subtle spamming activities with a higher accuracy rate and with less complexity in its structure compared to existing ML and graph-based techniques.

The main contributions of this study are as follows:

- A novel heterogeneous graph (MGSD) model is proposed to illustrate the intra- and inter-relationships among entities (e.g., to capture both singleton and multiple spam entities).
- Various sets of existing content, behaviour and relation-based features are exploited, with a new set of features defined to improve spamming detection accuracy. Importance of the features is analysed (the features' weight) by applying the feature fusion technique to determine the most effective combination of weighted features.
- A multi-iterative algorithm is designed to update the spamicity of entities during a finite number of iterations based on the spamicity score of their adjacent neighbours. The relationships among neighbours' nodes though cannot be captured if only the spamicity score of each entity, without considering its neighbours, is calculated.

Accordingly, the structure of this paper is organised into seven sections. Section 1, this chapter, provides an overview and background to the study along with anticipated contributions. Section 2 discusses previous studies relating to the context of this study. Sections 3 and 4 describe the definitions and proposed model, which is followed by Section 5, presenting the implementation and analysis of the proposed model. Section 6 presents the results of the evaluation and existing techniques, which is followed by Section 7 that summarises the proposed model and presents overall conclusions. Finally, Section 8 provides opportunities for future research.

2. Related studies

Opinion spam detection and its challenges is a relatively new research field needing further investigation and attention. While traditional techniques have adopted a human annotation approach in order to detect spam reviews, their accuracy was near to chance more than their accuracy (around 50%) (Ott et al., 2011). As such, greater focus on automated opinion spam detection is required. In the following sub-sections, major methods from different two perspectives, including ML and graph-based, are discussed regarding their uniqueness, results and limitations.

2.1. Supervised learning approach

The ML approach is often employed for opinion spam detection to reduce and optimise the feature vectors acquired from the feature selection procedure. Supervised ML approaches learn from experience (labelled dataset) as training samples used to predict or classify testing samples. Jindal and Liu were the first researchers who performed review spam detection using the Logistic Regression (LR) classifier based on duplication and the near duplication concept in www.Amazon.com (Jindal & Liu, 2007). Similarly, in Narayan, Rout, and Jena (2018), various sets of features, along with the sentiment score and different classifiers were applied. Due to the lack of a gold-standard labelled dataset, Ott, Cardie, and Hancock (2013) collected a labelled spam dataset written by Amazon Mechanical Turk (AMT). They proposed a supervised detection model by exploiting both linguistic and psychological features. Although they generated a gold-standard dataset, their models depended on human effort (unreal spammers) to write the spam reviews, and their flexibility was limited to different domains.

However, expert spammers do not write fake reviews in such a way that can be detected easily using detection methods. Sun, Morales, & Yan (2013) proposed a novel technique to produce synthetic fake reviews and detected them by extracting semantic coherence and flow smoothness of the reviews. Also, pre-processing that effects on the detection accuracy have been investigated in Etaiwi and Naymat (2017) by applying various ML algorithms, such as SVM and NB. In their analysis, they used a limited number of linguistic features which reduced the precision of detection. In another study, Dong, Yao et al. (2018) presented an end-to-end unified model to influence the requested properties employing the Autoencoder and random forest method. Here, a random decision tree model was employed to control the global parameter of the learning process. Even though the model was accurate, it employed a set of complicated formulas which reduced its performance. Whereas, in Li, Qin, Ren, and Liu (2017), a novel convolutional neural network model was presented to learn from both sentence and document representations of reviews. Here, the researchers calculated the importance of weights of each sentence in conjunction with the document representation.

Similarly, Ren and Ji (2017) explored document-level representation to train the gated recurrent neural network model for opinion spam detection. The experimental results, by integrating discrete and neural features for in-domain and cross-domain (three domains datasets) demonstrated the efficiency of this model. The researchers claimed that the neural model could be generalised easier compared to discrete models. In both of the above models, the semantic representation of reviews was considered, which decreases the flexibility of the model for the domains in which there is insufficient linguistic information.

Character n-gram as the main representation for opinion spam detection has been employed in intra- and cross-domain classifications by Cagnina and Rosso (2017). Here, the researchers proved that utilising character n-grams in tokens resulted in significant output with a low dimensionality representation. Character n-gram has also been used to detect deceptive controversial opinions. For example, Sánchez-Junquera, Villaseñor-Pineda, Montes-y-Gómez, and Rosso (2018) considered the opinions expressed for three

domains, including abortion, death penalty and personal feelings. NB and SVM classifiers with a binary weighting scheme were used to implement the model. The researchers discovered that using the character n-grams representation approach was more effective compared to using psycholinguistic features in order to detect deceptive opinions in these domains by achieving simplicity and high performance.

Likewise, Zhang, Du, Yoshida, and Wang (2018) proposed an opinion spam detection model called DRI-RCNN (Deceptive Review Identification by Recurrent Convolutional Neural Network) by exploiting word contexts and deep learning. Max-pooling and ReLU (Rectified Linear Unit) filters were applied to transfer words' recurrent convolutional vectors to a review vector. Aside from using a complicated model structure, the model only focused on the structure of reviews and neglected the behaviour-based features.

2.2. Unsupervised learning approach

Unlike the supervised approach, the unsupervised approach automatically generates the class label without requiring any prior knowledge. Here, Sandulescu and Ester (2015) investigated singleton spammers applying knowledge-based semantic similarity evaluation and Latent Dirichlet Allocation (LDA). Whereas, burstiness in temporal patterns of reviewers was investigated by Xie, Wang, Lin, and Yu (2012) based on the arrival pattern of normal and suspicious reviewers through a multi-dimensional time series model. Rating distortion is a further concept of suspicious behaviour investigated by Wu, Greene, Smyth, and Cunningham. (2010) examining the nature of spam reviews distorting the trend and popularity ranking of hotels. In Dong, Ji et al. (2018), to detect spam reviews and extract the topics and polarities from the reviews, an Unsupervised Topic-Sentiment Joint probabilistic model (UTSJ) based on Latent Dirichlet Allocation (LDA) model was proposed. Here, the UTSJ model was found to outperform its benchmark models for opinion spam detection in both balanced and unbalanced datasets in different domains. Further, Liu and Pang (2018) detected spammers based on their abnormal and deviated behaviours compared with normal users by defining several signals of abnormalities and deviation dimensions to detect review spammers. Lastly, target-dependent sentiment information was also used to model content-based review deviation. Although given it implements deep content analysis, the main limitation of this model was regarding its performance time.

2.3. Semi-supervised learning approach

The main limitation of the supervised approach is its dependency on a vast number of labelled data samples. As such, to overcome this issue, researchers in the field of opinion spam detection proposed a semi-supervised approach to benefit from the accuracy of the supervised approach while not requiring many labelled data samples. Chengzhang and Kang proposed a three-view model called tri-training, which used a scoring method to identify a spammed store (Chengzhang & Kang, 2015). The results proved that the tri-training model outperformed both the two-view co-training and single-view models. Likewise, Hernández Fusilier, Montes-y-Gómez, Rosso, and Guzmán Cabrera (2015) focused on the detection of positive and negative opinion spam by employing PU-learning, where they analysed the role of the review's sentiment in spam detection. The results indicated that negative spam reviews were more difficult to detect compared to positive reviews. Moreover, it was proved that using one classifier for detecting both types of spam reviews was more efficient compared to employing two different classifiers.

2.4. Graph-based approach

Given the existing problems in the ML approach, (i.e., lack of a gold-standard dataset, insufficient applied features and ignoring the entities' relationships), many researchers attempted to use a graph-based structure to represent opinion spamming activities. For instance, Wang et al. (2011) proposed a heterogeneous review graph, for the first time, which illustrated that the entities' relationships were based on the honesty of the reviews, trustiness of the reviewer and reliability of the store. However, their model could not be generalised given their research only focused on stores where on the majority of websites (e.g., Amazon.com), limited information was available. Lu, Zhang, Xiao, and Li (2013) attempted to detect spamming through using a united framework called the 'Review Factor Graph' (RFG) by considering the relationship among reviews and reviewers. This model applied a complicated mathematical formula which reduced its performance. Moreover, they considered helpfulness as a highly accurate feature to detect spammers, while this feature could also be spammed.

Unlike previous studies, Fayazbakhsh and Sinha (2012) considered review, reviewer, and product and their relations through a unified network-based structure. Although, their method only focused on content-dependent features which reduced the accuracy and could only detect a limited range of spamming activities. More recently, Shehnepoor et al. (2017) proposed a heterogeneous graph framework, called 'NetSpam', which applied features as network information to map spam detection into classifications. However, their method exploited a limited number of features and could not be used for the detection of group spammers.

By analysing the techniques mentioned above, it can be understood that a graph-based approach can reveal more subtle spamming activities compared with ML techniques with an acceptable accuracy rate. However, there is still no comprehensive model that can evaluate all entities in a unified structure and apply the most effective combination of features. Moreover, the graph-based approach does not need annotated data and has less complexity in its structure compared with supervised ML. In this study, we propose a heterogeneous graph-based model to detect spamming activities by considering inter- and intra-relationships which exist among all entities. In this case, the model does not only detect group activities, but it can also identify singleton spammed entities. In addition, we define a set of new features along with applying existing features and benefits from using a feature fusion technique to determine the most effective combination of features. Therefore, the main objective of this study is to capture the spamming influence

of each entity on its neighbours within the graph structure. As such, a novel multi-iterative algorithm is developed to update the spamicity score of the entity during a finite number of iterations.

3. Definitions

MGSD is a heterogeneous graph-based model that is applied for spam detection in opinionated documents. This structure represents various sets of entities, including review, reviewer, group of reviewers, products, and their relationships.

a) ***Heterogeneous graph:*** Graph-theory (proposed by Leonhard Euler 1735) has a broad range of applications, in such fields as biology, engineering, medicine and computer science. Given graph theory is an excellent modelling tool to effectively model entities and their relations, the theory has been applied in the opinion spam detection model of this study. Indeed, based on the review of literature, there are a vast number of spam reviews distributed throughout the web by spammers who post those spam reviews for targeted products. Therefore, all these relations are required to be considered through a united structure since they influence each other. In the graph structure, $G = (V, E)$, each entity sample is represented as a node V, and their relation, if one exists, is shown as edge E. Since the nature of entities is different, the proposed model is a heterogeneous graph.
b) ***Relations:*** Previous studies have suggested that the detection of spamming activities by considering only reviews, reviewers, or their targets cannot be effective and efficient. Therefore, MGSD, as a flexible and linearly scalable model, has been presented. The main focus of this study is to determine and analyse both inter- and intra-relationships that exist among entities and their influence on each other based on the joint and disjoint features. In graph $G = (V, E)$, v is an entity type ($v \in V$; e.g., review) which contains n samples $v_i = \{v_{i1}, v_{i2}, \ldots, v_{in}\}$:

$$Intra - relationship(R) = \begin{cases} 1, & \text{if } R(v_{in}, v_{im}) \neq \varnothing \\ 0, & \text{if } R(v_{in}, v_{im}) = \varnothing \end{cases} \quad (1)$$

and,

$$Intra - relationship(R) = \begin{cases} 1, & \text{if } R(v, w) \neq \varnothing \\ 0, & \text{if } R(v, w) = \varnothing \end{cases} \quad (2)$$

where $R(i, j)$ depicts whether two nodes have a relation with each other or not. For example, if $reviewer_1$ and $reviewer_2$ (i.e., nodes from the same entity) work together to put a higher spamming effect on one product, then they have an intra-relationship. Alternatively, if $reviewer_1$ writes $review_1$ (i.e., nodes from different entities), then consequently, these two nodes have an inter-relationship.

a) ***Spamicity:*** Spamicity is a concept (score) that determines to what extent and with what probability an entity's sample is spam among a group of samples (Mukherjee et al., 2012). In order to calculate the spamicity, we consider a diverse set of features of that sample along with its inter- and intra-relationships with other entity's/entities' sample(s). Moreover, spamicity is not only used to detect spammed entities but can also be considered to rank them based on that score.
b) ***Multi-iterative algorithm:*** The MGSD model monitors the behaviour of entities and iteratively updates their spamicity using a novel multi-level iterative algorithm to achieve high accuracy and performance. However, the spamicity calculation of entities is not a discrete process. In a graph structure, this score depends on both sets of extracted entities' features and their related nodes. Therefore, after calculating the spamicity for each node, the score needs to be updated based on its relation(s), which result in a new spamming score. The multi-iterative algorithm continues until the stopping condition is satisfied (i.e., after ten iterations or if the difference between previous and current scores of all samples is < 0.05).

4. MGSD: multi-iterative graph-based spam detection model

The MGSD model structure, along with its phases, is illustrated in Fig. 1.

In this study, data collection was performed using the Amazon dataset and human crowdsourced (i.e., synthetic reviews) reviews. Since the reviews were human-generated content, they contain different types of noises, such as emoji, special characters, acronyms, URLs and emotional words (e.g., 'hahaha' and 'wow'). In this case, the noise removal process can enhance the efficiency of the spam detection model. Although the emoji and emotional words can be used in opinion mining applications, they cannot be discriminative for spam and non-spam reviews. These words can be used by both genuine reviewers and spammers to write their reviews. In spamming activities, the spammers attempt to write the spam reviews which follow the genuine review's style and use emoji and emotional words to achieve this goal. Acronyms can also be found in both spam and non-spam reviews, and therefore cannot reflect any spamming signal. Though, unlike email spam, using URLs is not common practice in opinion spamming activities. Similar to acronyms, single URLs can exist in spam and non-spam reviews, although readers can easily detect the reviews with multiple URLs as uninformative reviews, and can consequently be discarded. The analysis of the Amazon dataset indicated that there was a small portion of reviews that contained multiple URLs. Therefore, in this study, URLs will be removed to improve processing efficiency.

Pre-processing steps are required for transferring text from human language to a machine-readable format for further processing.

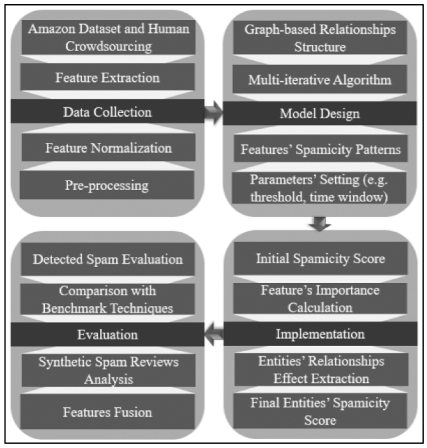

Fig. 1. MGSD model structure.

After a text is obtained, text normalisation (pre-processing) tasks are applied to prepare the data to extract the desired features. These tasks include noise removal, removing extra white spaces and punctuations, text segmentation and stop words removal. In designing the model, the graph-based structure was established to illustrate the entities and reveal their inter- and intra-relationships. In addition, the novel multi-iterative algorithm with its parameters (such as spamming threshold and time window size) and spamming patterns of entities was also developed. The feature's importance was analysed, applying the feature fusion technique to determine the best combination of features. According to this evaluation, the appropriate weight was assigned to the features to improve the accuracy of the proposed model. The spamicity of entities in the initial (i.e., considered features' values of each entity) and final (i.e., consider features related to their relationships) phases, after performing a finite number of iterations, was calculated. Finally, the accuracy of the MGSD model was assessed by employing human evaluators to compare with existing prominent techniques. The details of the phases, as mentioned above, are explained in the following section.

4.1. Feature extraction

Feature extraction is the main phase of developing any data mining application. In opinion spam detection, there are various sets of features, including content, behaviour and relation-based, which can help to detect spam entities (Noekhah, Salim, & Zakaria, 2017). While a large number of behaviour-based features are applied in opinion spam detection, a limited number of content-based features were applied due to high processing costs. In this study, the most significant set of features, which give remarkable results, was selected. In addition, we proposed ten new features that have never been used before to improve accuracy.

***Content-based features*:** The text of reviews often contains valuable information, either as linguistic or semantic, which is extracted through a diverse set of features. In this study, an effective set of content-based features was used that mainly focused on sentiment, similarity and syntactic structure of the review's content.

Review sentiment reflects the positive, negative or neutral opinion of people, and can be a good indicator for opinion spam detection (Kim, Chang, Lee, Yu, & Kang, 2015). In this study, we also exploited VADER (Hutto & Gilbert, 2014), a rule-based sentiment analysis model to identify the sentiment of review and related features. Besides sentiment manipulation, spammers typically write similar reviews, given they need to write a vast number of reviews within a very short period to significantly change in the product's sales rating. For similarity detection and its derived set of features, cosine similarity algorithm (due to its performance efficiency) was employed. Unlike other existing methods, we considered both linguistic and conceptual similarity using the threshold value of 0.7 (Choo et al., 2017). Syntactic-related features describe different structural aspects of the review's content (e.g., POS tags distribution extracted through applying NLP tasks). The set of applied features used in this study, along with their spamming signal and references are presented in Table 1.

Table 1
Content-based opinion spam detection features.

Feature	Spamming signal	Reference
Review polarity strength (RPS)	Spammers try to increase their influence by promoting or demoting their targeted product(s) or service(s) through extreme positive/negative opinionated content.	Choo et al. (2017)
Number of opinionated terms (OT)	If a review has a few or no adjectives and adverbs, it can be considered as a potential spam review.	Karami, Zhou, and Baltimore (2015)
Personal pronoun (PP)	Frequency of the existence of first-person pronouns in spam reviews is higher than non-spam ones due to emphasis on having self-experience and giving more credit to spam reviews.	Li, Ott, Cardie, and Hovy (2014)

Table 2
Behaviour-based opinion spam detection features.

Feature	Spamming signal	Reference
Rating deviation (RD)	A high rate of deviation from the average product's rating, based on the threshold value of 2 (Mukherjee & Venkataraman, 2014), changes the rating trend of a product.	Savage et al. (2015)
Extremity of rating (EXR)	An extreme positive (5) /negative (1) rating significantly influences the product's rating, which is used to detect spammers.	Choo et al. (2017)
Reviewer rating average (RRA)	Spammers usually give extremely positive or negative ratings to products, so their average rating can be a good spamming indicator.	Lu et al. (2013)
Helpful feedback of review (HF)	Spammers try to collect many positive (helpful) votes for their reviews in order to give credit to themselves. Therefore, a high helpful rating cannot be reliable to judge truthfulness.	Choo et al. (2017)
Helpful rate of reviewer (HRR)	Average helpful ratings of reviewers can be useful to detect spammers as the helpful rating can be manipulated.	Choo et al. (2017)
Review in burstiness TW (RBTW)	Reviews written in burstiness TW can be considered as potential spam reviews.	Xu, Shi, Tian, and Lam (2015)
Max. number of reviews per day (MRD)	Spammers attempt to write many spam reviews within a short period (e.g., one day) to gain more financial benefits.	Mukherjee et al. (2013)
Time gap of first and last reviews (FLG)	Spammers will usually post a vast number of reviews within a short period (number of days between first and last reviews).	Mukherjee and Venkataraman (2014)
Anonymous review (AR)	In some websites, spammers can post reviews without using a username.	Ren, Ji, Yin, and Zhang (2015)

***Behaviour-based features*:** These features imply behaviour information and metadata of entities. Many studies, such as Mukherjee, Kumar, Liu, Wang, and Hsu (2013), have applied them for opinion spam detection since their extraction requires less processing effort compared with content-based features. The rate and time-related features are two significant groups of behaviour-based features.

The rate, either in numeric or star form, is a key part of the product's metadata which spammers use to manipulate the popularity trend of a product. The review's feedback can be used in opinion spam detection as a complementary factor with the review's rating. Aside from the rating concept, the time of the review also provides critical information to detect spamming activities. Many studies have shown that tracking entities in a burstiness time window (i.e., time intervals in which a large number of reviews have been posted) could reveal the abnormal behaviour of entities. The rating and time-related features along with other behaviour-based features, their spamming signals and references are presented in Table 2.

***Relation-based features*:** In reality, since the entity is not isolated from other entities, (i.e., either the same or of a different type), there is a further group of features called relation-based features which imply that inter- and intra-relationships exist among entities. The graph structure is the most appropriate structure in presenting all entities and revealing their relationships (Wang et al., 2011). Many varied sets of features can be extracted by considering this correlation and applied to identify spammed entities. Relations can be evaluated through reviewer-review (e.g., number of singleton reviewers), product-reviews (e.g., number of product reviews), reviewer-reviewer (e.g., content similarity of a group of reviewers) and review-review (e.g., neighbouring review agreement) features. However, extracting relation-based features is more complicated compared to content and behaviour-based features. These types of features reveal more subtle spamming activities which spammers cannot hide due to three main reasons. These include 1) insufficient knowledge about the entire graph structure; 2) following group rules and collaboration with other group members, and 3)

Table 3
Relation-based opinion spam detection features.

Feature	Spamming signal	Reference
Singleton reviewer (SR)	Spammers commonly use different IDs (singleton spammer) to post reviews to avoid being caught.	Xu et al. (2015)
Group rating deviation (GRD)	The reviews' rating given by spammers who operate within a group is deviated from the average rating of a product in changing the rating trend.	Wang et al. (2016)
Burstiness time window (BTW)	Spammers attempt to post a large number of spam reviews in a limited period which causes burstiness in a time window.	Xu et al. (2015)
Number of Positive/Negative reviews in TW (PNTW)	Spammers attempt to change the rating trend of products by posting many positive/negative reviews in burstiness time intervals.	Xu et al. (2015)
Group burstiness time window (GBTW)	Spammers, who operate in a group, post a large number of reviews within a short time interval for the targeted product(s).	Mukherjee et al. (2012)
Group content similarity (GCS)	Due to the increasing spamming effect within a limited time, spammers operate together, usually writing similar reviews.	Choo et al. (2017)
Rating abused product (RAP)	Spammers post multiple reviews with similar ratings for the same product.	Choo et al. (2017)
Multiple reviews for product (MRP)	Spammers give multiple reviews into the same/group of the product(s) to increase their influence.	Lim, Nguyen, Jindal, Liu, and Lauw (2010)

Table 4
New proposed features for opinion spam detection.

Feature	Description	Spamming signal	Spamming condition
Sentiment-Rate difference (SRD)	Deviation of the sentence's sentiment and its rating	If the positive/negative degree of the review's ratings and content are not matched, it can be considered as a signal for spamming activities.	$SRD = \begin{cases} spam, & if\ \lvert Rate - Sentiment\rvert > 2 \\ non-spam, & if\ \lvert Rate - Sentiment\rvert \leq 2 \end{cases}$
Review group agreemnet (RGA)	The similar polarity of the review with other surrounding reviews due to posting time	The spam review has different polarity compared to the surrounding reviews as it attempts to alter the popularity trend of the product.	$RGA = \begin{cases} spam, & if\ \lvert R_{rate} - AvgRS_{rate}\rvert > 0.8 \\ non-spam, & if\ \lvert R_{rate} - AvgRS_{rate}\rvert \leq 0.8 \end{cases}$
Length deviation (LD)	Difference between the review's length and the average length of the product's reviews	Difference between the review's length and the average length of the product's reviews can be considered as a spamming signal as the review does not match with other reviews.	$LD = \begin{cases} spam, & if\ \lvert R_{len} - Avg(PRs'_{length})\rvert > 0.8 \\ non-spam, & if\ \lvert R_{len} - Avg(PRs'_{length})\rvert \leq 0.8 \end{cases}$
Rate for trend change (RTC)	Opposite review's rating compared to the previous review's rating to change the trend	Spammers attempt to write the review with an opposite rating compared with the previous review's rating to change the rating trend of the product.	$RTC = \begin{cases} spam, & if\ \lvert R_{rate} - Avg(PRs'_{rate})\rvert > 3 \\ non-spam, & if\ \lvert R_{rate} - Avg(PRs'_{rate})\rvert \leq 3 \end{cases}$
Reviewer's similar products' rates (SRP)	The ratio of similar ratings of the reviewer given to different products	The spammer hired by the company posts many reviews with a similar rating for different products of that company to enhance their popularity.	$SRP = \begin{cases} spam, & if\ Avg(RR_{Ps'rate}) > 0.9 \\ non-spam, & if\ Avg(RR_{Ps'rate}) \leq 0.9 \end{cases}$
Product's reviews allcoation (PRA)	The proportion of reviewer's reviews among the entire product's reviews	Companies attempt to hire spammers to write many positive reviews for their unpopular products. Therefore, those reviewers who allocate and post more than 10% of reviews can be considered spammers.	$PRA = \begin{cases} spam, & if\ (Ratio\ of\ reviews) \geq 0.3 \\ non-spam, & if\ (Ratio\ of\ reviews) < 0.3 \end{cases}$
Number of extrem rates of reviewer (NER)	The ratio of the number of times the reviewer gives an extreme positive/negative rating	If most of the reviews of spammers have an extremely positive or negative rating, it can be a good indicator that the reviewer is an active spammer.	$NER = \begin{cases} spam, & if\ (Ratio\ of\ times) > 0.9 \\ non-spam, & if\ (Ratio\ of\ times) \leq 0.9 \end{cases}$
Release time burstiness (RTB)	Whether the review is released in time or not	If the review is posted coinciding with the release time of the product (i.e., based on the first review of the product) and the TW is BTW, it cannot be confidently considered as a spam review.	$RTB = \begin{cases} spam, & if\ (Review_{time} \in product_{burst\_time}) \\ & and\ (Review_{time} \notin product_{release\_time}) \\ & non-spam, \\ & if\ (Review_{time} \in product_{burst\_time}) \\ & and\ (Review_{time} \in product_{release\_time}) \end{cases}$
Similar group rating reviews (SGR)	Spammers who operate in a group provide similar ratings to the product	Group spammers post many reviews with similar ratings in burstiness TW(s). Therefore, extracting these reviews helps to establish the spammer's group.	$SGR = \begin{cases} spam, & if\ (ratio\ of\ similar\ rate) > 0.9 \\ non-spam, & if\ (ratio\ of\ similar\ rate) \leq 0.9 \end{cases}$
Empty reviews (ER)	Number of empty reviews by the reviewers	Due to time limitations, the length of most reviews of spammers tend to be less than 1, since they try to change the popularity trend of reviews by assigning the rating.	$ER = \begin{cases} spam, & if\ (ratio\ of\ empty\ reviews) > 0.9 \\ non-spam, & if\ (ratio\ of\ empty\ reviews) \leq 0.9 \end{cases}$

inability of features' in changing and reconstructing. Relation-based features, their spamming signals and references are presented in Table 3.

***New proposed features*:** Through examining the literature, it can be seen that there remains a great deal of useful information which can be extracted from the review's content, metadata or entities' relations. Therefore, ten new features have been proposed to reveal the hidden relations that exist among entities, which lead to detecting more precise spamming behaviours from different perspectives. These limited features that can provide accurate results are discussed in Section 5.4, where they can be used as complementary feature sets to enhance spamming detection. In addition, the computational cost of these features is mostly less compared to other types of features, which presents the motivation to exploit them. Our new proposed features in this study, along with their descriptions and spamming condition, are presented in Table 4.

Domain independence is one of the most significant characteristics of the MGSD model due to exploiting the robust cross-domain feature sets.

4.2. Graph structure

The improvement in current opinion spam detection techniques can be achieved by exploiting graph-based structures. In this section, the details of the proposed MGSD model are described.

As shown in Fig. 2, there are three main entities, including the reviewer, review, and product. Aside from these three main entities, there are some entities which cannot be explicitly detected from the graph, since they need to be identified based on their

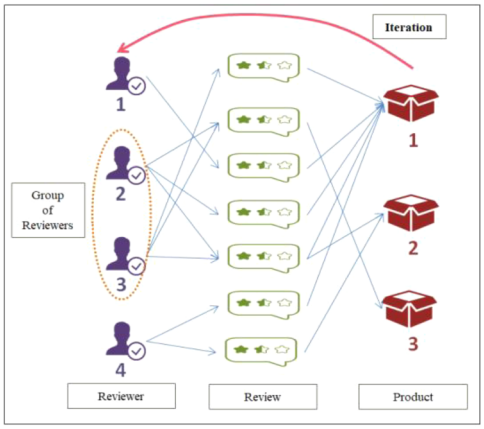

Fig. 2. Graph-based Model.

relationships. For example, in Fig. 2, *reviewer2* and *reviewer3* wrote similar and different reviews for product 1. In this case, if the MGSD model detects them as spammers, and they consist of many common characteristics, they can be considered as a group of spammers which target *product1*.

In addition, existing graph-based methods, such as Shehnepoor et al. (2017), can only calculate a spamming score based on prior knowledge and the path between entities. Whereas, in contrast, MGSD model finds that the spamicity of each entity needs to be computed based on the spamicity of its neighbours. The significance of this work is on updating the spamicity of entities during multiple iterations by applying the novel multi-iterative algorithm incorporated with the graph structure.

4.3. Multi-iterative spamicity algorithm

After representing all entities and their relations, schematically, it is required to determine their spamicity and to rank entities based on the calculated spamicity. There are different ranking algorithms applied in the information retrieval field. For instance, PageRank (Brin & Page, 1998) and HITS (Hyperlink-Induced Topic Search) (Kleinberg, 1999) algorithms are two well-known ranking algorithms applied for ranking web pages and their variations. The most commonly used algorithm for ranking pages is the PageRank algorithm which exploits the link structure of the web to calculate the importance of web pages. The page is important if it obtains a high rank calculated based on the sum of the ranks of its backlinks. Similarly, the HITS algorithm, as a popular and effective ranking algorithm, is used for rating and ranking websites and documents based on the link information of web pages. Moreover, it performs a series of iterations which consist of two fundamental steps: Authority update rule and the Hub update rule. Both PageRank and HITS are iterative algorithms.

Although, PageRank cannot be an effective algorithm for opinion spam detection given it can only detect those products which have many reviews and have been launched for an extended period. Additionally, this algorithm only considers the link relevancy that exists among entities and ignores the content of reviews and other meta-data related to the product and reviewer, which can affect the link's weighting. Furthermore, it has a rank sink problem during the iterations; when in a graph structure, an entity gets into infinite link cycles.

Even though the PageRank algorithm does not consider many features, it is also a very slow algorithm due to the complicated computation process involved. This algorithm works based on the popularity of the product, (i.e., if there are many links among entities, it cannot detect whether these entities are spam or not) and they only rank and cannot detect and classify spam and non-spam entities. On the other hand, as HITS is a query dependent algorithm, the computational speed is low, given it uses a traditional search engine to extract the set of relevant information. The main goal of opinion spam research is in detection rather than ranking, whereas these two algorithms rank the spam entities.

Therefore, to overcome these existing problems and limitations in this study, a multi-iterative algorithm was developed to update the spamicity of entities based on their neighbours' spamicity scores. For example, if most of the reviews written by a reviewer are spam reviews, the spamicity score of that reviewer should be updated and added to the current spamicity score. The details of the algorithm are illustrated in Algorithm 1 below.

In step 1, the spamicity values of all entities' samples are initialised to zero. As at the beginning of this stage, all samples are assumed not to be spam. The basic spamicity score of each sample is calculated, in step 2, based on its normalised weighted features. In line 5, f_m is the value and w_m is the weight assigned to feature m based on its importance. The *Basic Spamicity* is calculated based on the ratio of summation of all weighted features. In step 3, as different entities can affect each other, the spamicity of each sample needs to be updated. After updating the sample's spamicity, those samples connected to it need to be updated again, which causes an iterative procedure.

The number of iterations and the spamicity threshold are the two main parameters in the graph iterative algorithm. The results show that the stopping criteria would be satisfied in two conditions. In the first condition, the algorithm stops after a finite number of iterations, which in this study was assumed to be 10 iterations. This algorithm has been implemented on different sized datasets and

Table 5
Detection accuracy of applying different spamicity threshold.

Threshold	TP	TN	FP	FN	Accuracy
0.7	492/600	483/600	117/600	108/600	81.25%
0.8	565/600	589/600	11/600	35/600	96.17%
0.9	453/600	436/600	164/600	147/600	74.09%

*TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative .

has been found that for large datasets usually the algorithm is converged between 8 and 11 iterations and after that, the changes in the entity's spamicity will be less.

Therefore, in order to reduce the computational cost, the number of iterations considered was set to 10. The second condition is usually satisfied when the size of the dataset is small, and as such, the algorithm will be stopped when the difference between the current and previous spamicity scores of all samples is <0.05. The detection results demonstrate that by considering this threshold value, 98.3% of spam entities can be detected and the computational cost can be reduced due to the constraint of the algorithm's implementation.

However, if the convergence criteria are not satisfied, the algorithm continues the updating procedure for the next iteration, (line 13), until the convergence condition is achieved. Finally, in step 4, the final sample's spamicity is utilised to determine whether the sample is spam or not (line 19). Table 5 depicts the accuracy of spam detection by considering the different threshold of spamicity. As will be explained in Section 5.1, 600/600 positive and 600/600 negative spam/non-spam reviews were collected as the crowdsourced dataset. A different spamicity threshold was also applied to determine the best threshold that our algorithm could use to detect most of the spam entities.

As shown in Table 5, it can be seen that if the threshold is selected as 0.7, the number of false-positives (FP) and false-negatives (FN) will be high since many non-spam samples are incorrectly detected as spam. However, in contrast, if the threshold is set as 0.9, the rate of true positive (TP) and true negative (TN) will be lower since many real spam entities are ignored. The best result is achieved when the threshold is set as 0.8. With this threshold value, the algorithm can detect more real spam and non-spam and will not incorrectly detect non-spam reviews as spam reviews.

5. Experimental analysis

In this paper, we implemented and evaluated a novel multi-iterative graph-based model for opinion spam detection. To our knowledge, this is the first opinion spam detection model which considers all entities in unified structures by applying a comprehensive set of domain-independent features along with a new set of proposed features.

5.1. Dataset description

Three types of datasets were applied in the MGSD model. The real-word dataset (Amazon.com reviews He & McAuley, 2016) was used in implementing the model, while our crowdsourced dataset and Ott's (Ott et al., 2013) dataset were exploited in evaluating the model. Given the original Amazon dataset had many incorrectly categorised products, we needed to rearrange the categories. Here, three subcategories were considered, including “Software”, “Apps for Android” and “Appstore for Android” under the main category named “Software”. The main reason for selecting these three subcategories was due to the diverse number of reviews in each one. As such, we could evaluate the efficiency of our proposed model for small, medium and large datasets. The statistics of the rearranged dataset is presented in Table 6.

In the evaluation phase, MGSD was analysed by exploiting two different datasets. The first dataset was collected using the synthetic crowdsourced method where we asked 15 PhD students, who had had online shopping experience, in five groups to write 40 positive spam reviews for 5 products with a rating of 1 and 40 negative spam reviews for 5 products with a rating 5. Similar to Ott et al. (2013), 600 positive and 600 negative truthful reviews were added to balance the crowdsourced dataset. The second dataset, Ott's dataset (Ott et al., 2013) was used as a benchmark. This dataset contained spam reviews generated by using AMT and truthful reviews collected from TripAdvisor.com for 20 popular hotels in the Chicago area of the United States (US). However, the main problem with Ott's dataset was that there was no prior information concerning the targeted hotels provided to Turkers. Therefore, their reviews could not reflect the real behaviour of spammers. In our crowdsourced dataset, the PhD students were provided with the product's Amazon web page. They had a better understanding of the product's description and its reviews, so the

Table 6
Statistics of the software dataset.

Sub-category	Products/ Before pre-processing	Reviews/ Before pre- processing	Products/After pre-processing	Reviews/After pre-processing
Software	17,718	293,421	5799	268,881
Apps for Android	61,550	2,660,635	21,359	2,576,966
Appstore for Android	152	14,227	57	13,999

Table 7
Statistics of crowdsourced datasets.

Dataset	Target	Positive spam and non-spam reviews	Negative spam and non-spam reviews
Collected dataset	10 products	1200	1200
Ott dataset	20 hotels	800	800

dataset reflected real-world spamming activities. Regarding the size of the dataset, the dataset contained a larger number of reviews along with group spamming activities. The statistics of these two datasets are presented in Table 7.

5.2. Data pre-processing

Pre-processing was implemented for both the structure and data content of our real-world dataset. In structure pre-processing, the products with no reviews and inactive products (i.e., products with less than 5 reviews, with the last review posted more than one year ago) were removed to improve the quality of detection. This implies that the product is not popular, and no spamming activity has occurred. The results are presented in columns 4 and 5 in Table 6.

Regarding data pre-processing, unlike some existing methods which change all the words to lowercase, we maintained the words' style, since it could reflect the sentiment strength and emotions of the reviewers. In the next step, the punctuation marks, stop-words and numerical characters were removed as these can influence the results of any text processing approach. By removing them, the main focus was on the important words (Etaiwi & Naymat, 2017). Moreover, for similarity and sentiment analysis, stemming was also implemented using WordNet. These steps were performed for both the synthetic and real-world datasets.

5.3. Evaluation metric

Precision, recall and F-measure are the most popular metrics used in opinion spam detection. Accuracy is also a common metric for model evaluation as it provides an accurate result if the numbers of instances of both classes (e.g., spam and non-spam) are equal, and unless the accuracy has deviated to the majority class. We evaluated the effectiveness of the MGSD model by applying the following formulas for both spam and non-spam entities.

$$Accuracy = \frac{number\ of\ correct\ classifications}{total\ number\ of\ data\ samples} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Precision = \frac{number\ of\ correct\ predictions\ of\ each\ class}{total\ number\ of\ predictions\ of\ each\ class} = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{number\ of\ correct\ predictions}{total\ number\ of\ predictions} = \frac{TP}{TP + FN} \tag{5}$$

$$F - measure = \frac{2\ \times recall \times precision}{recall + precision} \tag{6}$$

5.4. Feature selection approach

Feature selection is the process of extracting desired information in the form of features from available documents. Features have different effects on the accuracy of spam detection, and as such, their importance cannot be considered as equal. Further, there are various feature weighting algorithms, such as Term frequency-inverse document frequency (Tf-idf), Point Mutual Information (PMI) and Information Gain (IG).

Tf-idf weight is a weight used in information retrieval and text mining and is a statistical measure applied to evaluate the importance of a word within a document. The importance increases proportionally to the number of times a word appears in the document, which is offset by the frequency of the word in the corpus. In this study, Tf-idf was used in the MGSD model for similarity detection. However, since it only counts the number of the word's occurrence in a document and is limited to the review content, it cannot be used to weight all sets of features, as explained in Section 4.1. Similar to Tf-idf, PMI is based on word occurrence in the document, which is not a useful method for feature weighting in opinion spam detection. Likewise, simple frequency is not the best measure of association between words. One problem is that raw frequency is extremely skewed and is not very discriminative.

In addition, the weight of the feature calculated by IG is based on the class of the feature. Here, the higher the weight of a feature, the more relevant it is considered. Although IG is usually a good measure for deciding the relevance of a feature, it is not perfect as a notable problem occurs when it is applied to the feature that can take on a large number of distinct values. Moreover, this method assigns a high weight into the features of spam detection, which do not have important information, (e.g., the reviewer's ID). Therefore, there is no general algorithm that can be used for all types of data mining applications. Instead, it is necessary to implement different classifications, ML, or data mining methods to analyse and compare their performance and to select the best one. One of the strengths of the proposed algorithm is that it considers this effect and assigns the proper weight to each feature.

As mentioned in Section 4.1, several types of features were extracted and applied in the MGSD model selected via two main steps. First, a new set of features was proposed, and the popular ones were collected from the literature. Second, the most effective features were selected by applying three well-known classifiers, which included SVM, NB and DT (Decision Tree) on both crowdsourced datasets. The accuracy of these classifiers can help to assign the weight to each feature based on its importance during the detection process.

NB is a probabilistic classifier in the development of ML based on Bayes theorem by assuming the independency of features to predict the class label of samples. Here, the class probability of each sample is calculated by multiplying the class probability of each feature belonging to that sample by using Formula (5) below.

$$P(C_k|x) = P(C_k|x_1)P(C_k|x_2)P(C_k|x_3)...P(C_k|x_n) \tag{7}$$

where x is the sample and x_i is feature i, which belongs to class C_k. The probability of each feature, as mentioned in Formula (5), is defined in Formula (6).

$$P(C_k|x) = \frac{p(C_k)p(x|C_k)}{p(x)} \tag{8}$$

SVM is a supervised binary classifier that is applied in opinion spam detection (i.e., spam and non-spam). This algorithm classifies entities by determining the maximum margin hyperplane, which utilises a maximum distance between the hyperplane and the nearest sample from either the spam or non-spam class.

As a training dataset, $(\vec{x_1}, y_1)$, ..., $(\vec{x_n}, y_n)$ consists of n points, where y_i can be either -1 (non-spam) or 1 (spam) class label in which each sample $(\vec{x_n})$ belongs to it. The hyperplane is defined in Formula (7).

$$\vec{w}.\vec{x} - b = 0 \tag{9}$$

where, $\vec{w}$ is the vector of the hyperplane, $\vec{x}$ is the set of points (samples) and b is the offset of the hyperplane from its origin.

DT classifier, as a generative model, produces a hierarchical parsing of the training dataset. Here, each node represents the feature's value, and the connection between nodes is weighted according to the occurrence of the features in that document. At the lowest level, the leaves of the tree represent the class labels. Additionally, this classifier is designed based on the presence/absence of the features in the dataset.

Accordingly, the above classifiers on both crowdsourced datasets were implemented in exploiting the features to train them. Based on the accuracy of each classifier after applying the feature, and the feature's popularity, (as found in the literature), we assigned the weight (i.e., importance) to that feature, calculated by using Formula (8).

$$Weight_{f_n} = \frac{\sum_{i=1}^{3} acc_{c_i} + p_n}{4} \tag{10}$$

where, acc_{c_i} is the accuracy of classifier i when feature n is used to classify the testing dataset. p_n determines the publicity of the feature based on the number of techniques applied to that feature, thus achieving reliable results. The weight was normalised in the range of (0, 1). The result of the features' weighting is presented in Table 8.

As understood from referencing Table 8, some features provided prominent results while others were not good indicators for opinion spam detection. RPS, EXR and NER achieved high weights as all spammers attempted to write reviews for the products to enhance their popularity. Indeed, the main goal of spammers is to bias the average rating of products. RD, RTC, RAP and SGR imply their attempt to change the product's rating. Spammers are not typically actual or real customers, so they fail to have sufficient knowledge to write convincing reviews. Consequently, they usually perform a rating abuse task, given it saves time and is a more effective technique. Spamming activities attempt to change the popularity trend of a product within a short period and causes burstiness in reviews which can be highlighted using features such as BTW, RBTW and GBTW.

In some cases, spammers use the same user ID to write reviews, so MRP can be an excellent feature to capture this activity. In addition, either due to the limitation of time or deceiving spam detection methods, spammers post reviews in which the sentiment of the content is different from the rating (SRD). Feature weighting algorithms were applied to the crowdsourced dataset, as explained in Section 5.1. The comparison of the accuracy achieved by applying different feature weighting schemes is presented in Table 9.

The weights assigned by the ML classifiers were found to enhance the detection accuracy of MGSD model, which proves their weight calculation was more accurate. IG achieved an acceptable accuracy, although the accuracy of Tf-idf and PMI was very low due to the limited scope.

Even though applying all features can achieve high accuracy given they consider the entities' behaviours from different perspectives, the available computational resources are not always sufficient. In this situation, it is required to select the most effective features from a large number of features. Feature fusion is one of the feature selection techniques that can be applied to determine the best combination of features.

To analyse the most effective combination of features, the crowdsourced dataset, explained in Section 5.1, was used with 2400 spam and non-spam reviews. The three classifiers mentioned above were implemented on the testing datasets by exploiting a different combination of features via two phases. In the first phase, the classifiers' accuracy was analysed by applying features from different categories and also all features. The results are illustrated in Fig. 3.

In the second phase, those features with a weight greater than 0.6 from each category were selected, as they could give prominent results while reducing the performance cost. We also analysed the results by applying all sets of features where their weight was

Table 8
Features' weight of entities.

Type	Name	Weight
Content-based	Review polarity strength (RPS)	0.8
	Number of opinionated terms (OT)	0.68
	Personal pronoun (PP)	0.52
Behaviour-based	Rating deviation (RD)	1
	Extremity of rating (EXR)	0.89
	Reviewer rating average (RRA)	0.77
	Helpful feedback of review (HF)	0.56
	Helpful rate of reviewer (HRR)	0.51
	Review in burstiness TW (RBTW)	0.97
	Max. number of reviews per day (MRD)	0.82
	Time gap of first and last reviews (FLG)	0.65
	Anonymous review (AR)	0.42
Relation-based	Singleton reviewer (SR)	0.63
	Group rating deviation (GRD)	0.85
	Burstiness time window (BTW)	0.96
	Number of Positive/Negative reviews in TW (PNTW)	0.77
	Group burstiness time window (GBTW)	0.95
	Group content similarity (GCS)	0.89
	Rating abused product (RAP)	0.91
	Multiple reviews for product (MRP)	0.93
Proposed features	Sentiment-Rate difference (SRD)	1
	Review group agreemnet (RGA)	0.84
	Length deviation (LD)	0.25
	Rate for trend change (RTC)	0.95
	Reviewer's similar products' rates (SRP)	0.21
	Product's reviews allcoation (PRA)	0.89
	Number of extrem rates of reviewer (NER)	0.9
	Release time burstiness (RTB)	0.51
	Similar group rating reviews (SGR)	0.98
	Empty reviews (ER)	0.26

Table 9
Comparison of feature weighting techniques.

Algorithm	Accuracy
Tf-idf	74.3%
PMI	78.1%
IG	84.6%
Machine learning classifiers (SVM, NB, and DT)	93.7%

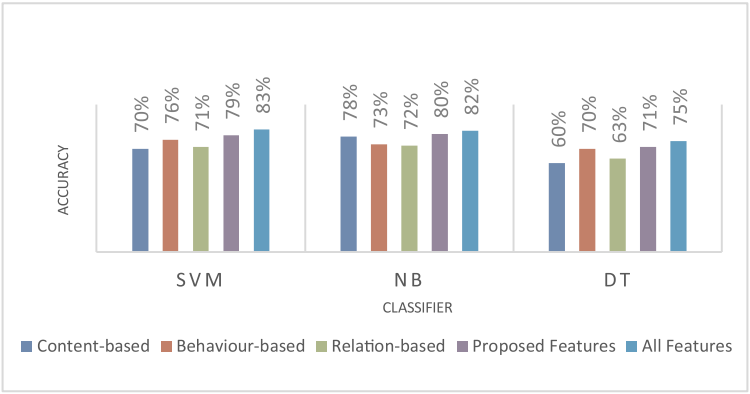

Fig. 3. Classifiers' accuracy by applying various set of features.

greater than 0.6. The results are presented in Fig. 4.

Figs. 3 and 4 illustrate that applying all sets of features provides prominent results. As spammers attempt to make their spam reviews look similar to truthful reviews, content-based features cannot be good indicators when they are applied alone. Accordingly, there is still no accurate algorithm for NLP applications which can effectively analyse the content of reviews. Among all the

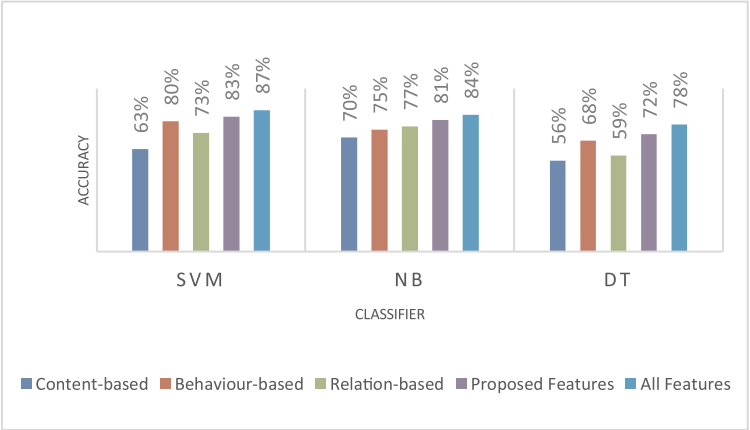

Fig. 4. Classifiers' accuracy by applying a various set of features (weight > 0.6).

classifiers, SVM provided more accurate results. Also, exploiting the proposed set of features enhanced the classifiers' accuracy, which proves their effectiveness in opinion spam detection. These features analysed the significant characteristics of the content, behaviour and relation of entities, simultaneously. They also achieved acceptable accuracy even though it was not as high as applying all features.

Therefore, in the case that there is a limitation of computational resources, the proposed features can be applied to detect spamming activities since they consider spamming from a different perspective. Although there are some cases where the existing features cannot capture them. For example, spammers typically do not spend a significant amount of time to write proper content, or they copy unrelated reviews as they wish instead to abuse the given rating(s). In this case, if the difference of sentiment of the review content and the rating is high, it will be a signal that the SRD feature has captured and detected spamming. This was explained in Section 4.1. This difference cannot be captured by using only content-based features or behaviour-based features. According to the results, all sets of features in the MGSD model's implementation were applied. Furthermore, the accuracy of the model was evaluated with existing techniques which used the crowdsourced dataset and also applying state-of-the-art graph-based techniques.

Also, due to the lack of either available information in some of the opinionated websites/documents or computational resources, the most effective features having a weight >0.9, can be applied in the MGSD model, eliminating the remaining features. Notably, the rating and time of the review are two common concepts that can be extracted from an opinionated document.

In the next step, the features presented in Table 10 were exploited using three classifiers. The accuracy of the classifiers is illustrated in Fig. 5.

The MGSD model is a flexible and adjustable model which can be implemented using a limited amount of information of reviews. In situations where a large number of features are not available or are systematically not easily obtainable, the features from Table 10 can be extracted and applied in the MGSD model. As shown in Fig. 5, the classifiers can provide an acceptable result, even though the accuracy is not high as in applying all set of features. Therefore, it can be concluded that by applying a limited number of features, where there is a lack of resources, the MGSD model can still be employed for opinion spam detection.

The main goal of the analysis, as presented in Figs. 3–5, was to determine the best and most effective set of features which could detect spamming activities with high accuracy. The flexibility of the MGSD model allows it to work with any set of features available for the user. Since the applied features are domain-independent, the user can also choose the best combination of features and to implement the model based on the necessity and availability of information.

Table 10
List of critical features for the training of classifier.

Name	Weight
Rating deviation (RD)	1
Review in burstiness TW (RBTW)	0.97
Burstiness time window (BTW)	0.96
Group burstiness time window (GBTW)	0.95
Rating abused product (RAP)	0.91
Multiple reviews for product (MRP)	0.93
Sentiment-rate difference (SRD)	1
Rate for trend change (RTC)	0.95
Number of extreme rates of reviewer (NER)	0.9
Similar group rating reviews (SGR)	0.98

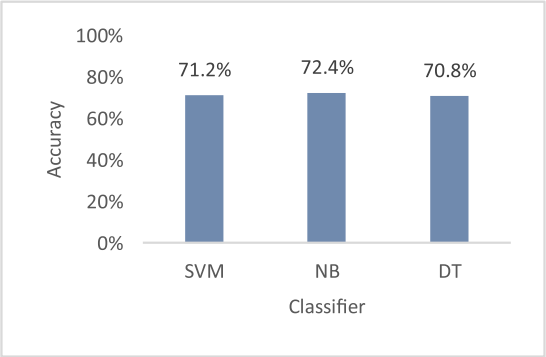

Fig. 5. Classifiers' accuracy by applying the most common important features (weight > 0.9).

5.5. Experimental results

The experimental results focus on the evaluation of the MGSD model with both the Amazon and synthetic crowdsourced datasets, as described in Section 5.1. The main objective of this evaluation was to analyse to what extent the model accurately detects spammed entities. The MGSD model was applied to the Amazon dataset in order to classify the entities into spam and non-spam based on their spamicity scores. The results of the evaluation are displayed in Table 11.

As the MGSD model is an unsupervised graph-based approach, and the real-world dataset is unlabelled, we needed human evaluators to interpret the proposed model's output.

- Human evaluation:

Employing human judgement is one of the well-established IR-based evaluation techniques that has been used by researchers in opinion spam detection when there is no labelled dataset to validate the credibility of the model (Xie et al., 2012). Three human evaluators (PhD computer science students with online shopping experience) were invited to analyse and classify the entities into spam and non-spam. Their results were then compared with the model's output to evaluate the effectiveness of the model.

Two-hundred spam and 200 non-spam reviews detected in the MGSD model along with 53 spammers and 184 non-spammers who had written these reviews extracted from 10 products (6 spam products and 4 non-spam products) were selected. It was found that 14 groups of spammers had attacked these spam products, as detected by the model. The profile of the spammers and non-spammers, along with the URL of the products were given to the evaluators (i.e., judges). If needed, they could obtain additional information, especially by checking the profile of reviewers.

In addition, various spam detection signals were presented to the judges in order to follow the standard evaluation instructions to facilitate more accurate analysis and results. These signals included the following:

- A reviewer with many opposite (i.e., opposing) review opinions;
- Reviews on brand;
- Reviews with no adjective and adverb;
- Reviews having all positive/negative opinions;
- Non-reviews (advertisement, promotion, hyperlink);
- Irrelevant reviews;
- A large number of reviews written in a short period;
- General reviews;
- A contradiction between the review rating and content;
- Duplication or near duplication in the review content;
- A reviewer with too many five/one-star reviews;
- Using ALL CAPS in the review;
- A reviewer who reviews only products of one manufacturer;
- A review with a repeated name and model of the product; and

Table 11
Statistics of the MGSD model implementation on the software category.

Sub-category	Spam reviews	Spammers	Products	Group of spammers
Software	**24,195**/268,881	**1855**/217,232	**248**/5799	**154**
Apps for Android	**318,769**/2,576,966	**184,896**/1,301,556	**3501**/121,359	6832
Appstore for Android	**984**/13,999	**265**/13,826	**12**/57	**19**

Table 12
The three human judges comments.

Entity	Judge	JUDGE 1 spam/non-spam	JUDGE 2 spam/non-spam	JUDGE 3 spam/non-spam
Review	JUDGE 1	163/174	–	–
	JUDGE 2	**145/151**	180/191	–
	JUDGE 3	**134/145**	**160/170**	174/185
Reviewer	JUDGE 1	41/178	–	–
	JUDGE 2	**30/153**	45/169	–
	JUDGE 3	**24/140**	**28/134**	36/150
Product	JUDGE 1	2/2	–	–
	JUDGE 2	**2/2**	4/4	–
	JUDGE 3	**2/2**	**3/4**	3/4
Group of reviewer	JUDGE 1	8	–	–
	JUDGE 2	**7**	11	–
	JUDGE 3	**6**	**8**	9

- Long explanation of the product.

The results obtained from the three judges for detecting the spammed entities are presented in Table 12.

Table 12 displays the number of spam and non-spam entities evaluated by the three human evaluators. The numbers in the diagonal cells indicate the judge's individual spam detection, while the numbers presented in the off-diagonal cells show the overlapping entities between the information received from each pair of judges. As interpreted from the table, the judges were predominantly consistent in their detection of spam and non-spam entities. Cohen's kappa values (Landis & Koch, 1977) were used to calculate the inter-evaluator consistency of the judges as follows.

$$kappa = \frac{p_o - p_e}{1 - p_o} \tag{11}$$

where p_o is the proportion of the frequency (i.e., number of times) the evaluators agreed with each other, and p_e is the proportion of frequency that they would be expected to agree by chance, as calculated using Formula (10).

$$p_e = \frac{1}{N^2} \sum_k n_{k1} n_{k2} \tag{12}$$

where N is the number of the entity's samples and n_{ki} is the number of times evaluator i detects category k. The three evaluators agreed on 130/142 spam and non-spam reviews, 22/131 spammers and non-spammers, 2/2 spam and non-spam products and 6 groups of spammers respectively. The Cohen's kappa values of pairs were between 0.64 and 0.85, indicating substantial inter-evaluator agreement. Therefore, this proves that the MGSD model is able to classify spam and non-spam reviews with an acceptable level of accuracy.

6. Evaluation results

In this section, analysing the evaluation undertaken on both the existing gold-standard datasets and state-of-the-art techniques is discussed.

6.1. Evaluation of synthetic crowdsourced dataset

A total of 600 positive and 600 negative spam reviews were collected in which were combined with 600 positive and 600 negative truthful reviews, detected by the model as non-spam reviews, in order to attain a balanced labelled dataset. The MGSD model was employed to ascertain to what extent the model can differentiate from those collected spam reviews from the non-spam reviews. Table 13 displays the number of correctly classified reviews.

Detecting the negative spam reviews was more challenging compared to detecting the positive spam reviews as interpreted from the results shown in Table 13. By analysing the review content, it was found that there were two main differences between positive

Table 13
Performance scores of crowdsourced datasets.

Sample type	Positive	Negative	A	P	R	F
Spam	512/600	480/600	90.5% (Pos)	95.3% (P_S) 93% (N_S)	85.3% (P_S) 80% (N_S)	90% (P_S) 86% (N_S)
Non-spam	575/600	564/600	87% (Neg)	86.7% (P_NS) 82.5% (N_NS)	95.8% (P_NS) 94% (N_NS)	91% (P_NS) 87.9% (N_NS)

*A: Accuracy, P: Precisions, R: Recall, F: F-Score, P_S: Positive Spam, N_S: Negative Spam, P_NS: Positive Non-spam, N_NS: Negative Non-spam.

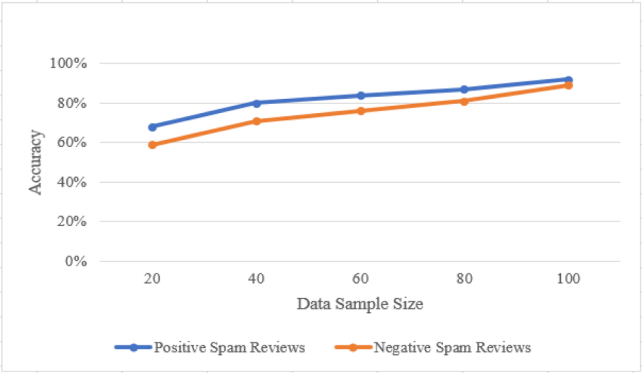

Fig. 6. The MGSD model's accuracy applied to the proposed crowdsourced dataset using various sample sizes.

and negative spam reviews. First, negative emotional words were detected more frequently in negative spam reviews compared to positive emotional words used in positive spam reviews. Secondly, the first-person singular pronoun was found to be more frequent in positive spam reviews compared to negative reviews. Accordingly, this could possibly be due to greater emphasis on having self-experience in positive spam reviews to convince potential customers while the tone used in the negative spam reviews caused the spammer to distance themselves from their negative statements from a psychological standpoint. The model was also evaluated based on various sized datasets.

As shown in Figs. 6 and 7, the MGSD model provided significant results applied to the small and large-sized data samples. Moreover, it outperformed Ott et al. (2011, 2013) for both positive and negative spam reviews of the proposed crowdsourced and Ott datasets. The main reason was possibly that those spam reviews applied in Ott et al. (2011, 2013) were written by AMT spammers, with limited knowledge regarding the targeted hotels and typically wrote simple reviews. In contrast, the crowdsourced dataset was collected using people who had good knowledge concerning the products and who wrote the spam reviews in such a way which appeared like real spammer's reviews.

Notwithstanding, the accuracy of the MGSD model for lower-bound data samples was not as high as the upper-bound data samples. This could be due to the relation-based features which may not be so useful when the number of data samples is small since they cannot capture group spamming activities which therefore produce a good signal to detect spamming.

6.2. Evaluation of the proposed model with benchmark techniques

In this section, MGSD model was implemented on Ott's dataset, as explained in Section 5.1, and the result compared with the results reported for the existing state-of-the-art techniques which exploited the same dataset.

As shown in Table 14, the MGSD model outperformed the existing techniques which exploited the Ott dataset as they focused on content-based features which cannot detect spamming activities accurately. Also, in these techniques, either spam reviews or spammers were considered, and therefore could not be generalised in applying to other domains and to cover all spamming activities appropriately.

6.3. Evaluation of the proposed model with benchmark graph-based techniques

The performance measure of the MGSD model was next compared to the state-of-the-art graph-based techniques (as revealed in

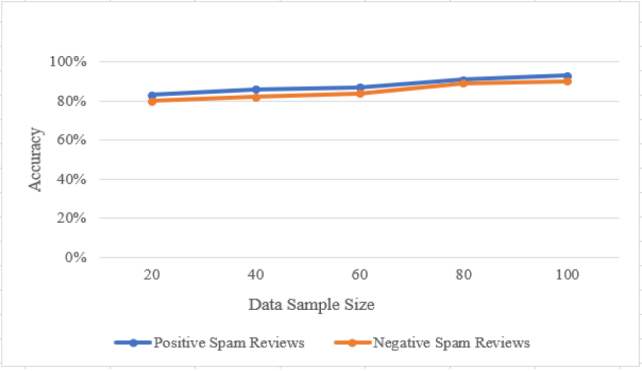

Fig. 7. The MGSD model's accuracy applying Ott's dataset using various sample sizes.

Table 14
Comparison of techniques based on Ott's dataset.

Technique	Method	Features	Performance scores			
			A	P	R	F
Deep semantic frame-based model (Kim et al., 2015)	SVM, NB	Content-based	87.6%	87.3%	88%	87.6
Semantic similarity model (Sandulescu & Ester, 2015)	Topic modeling (bag-of-words, bag-of-opinions)	Content-based	86%	–	–	–
Deep level linguistic model (Chen, Zhao, & Yang, 2015)	Rule-based	Content-based	91.4%	–	–	–
Psychological_Linguistic Integration model (Ott et al., 2011)	SVM, NB	Content-based	89.8%	–	–	–
Negative review spam detection model (Ott et al., 2013)	SVM	Content-based	86%	–	–	–
MGSD model	Graph-based	Content, behaviour and relation-based	93.2%	91.8%	90.2%	91%

*A: Accuracy, P: Precisions, R: Recall, F: F-score.

the literature) which produced the best results (refer to Table 15). These techniques were implemented on the same Amazon dataset, as used with the MGSD model. Here, the classification step as implemented in Shehnepoor et al. (2017) for review and reviewer was applied based on different meta-path types, for the samples of the same entity, and edges for the samples of different entities which are innovative in the spam detection domain. All meta-paths were simulated in this evaluation for the applied Amazon dataset, as explained in Section 5.1.

In Fayazbakhsh and Sinha (2012), three entities (review, reviewer and product) were considered, and all α, β, γ, and λ parameters were set to 1/3 giving equal weights to all contributing features. α, β, γ, and λ were non-negative, and real numbers in which α, β reflect the importance of suspicion scores of reviews and reviewers, and γ and λ reflect the importance of suspicion scores of reviews and products, respectively. In the experiments, the number of iterations of the algorithm was set to 20 iterations. The RFG model and message propagation of Lu et al. (2013) was simulated in this evaluation process, and the iterations were continued until the model arrived at the global convergence state. Whereas, in Wang, Xie, Liu, and Yu (2012), they detected the spamming activity of the review, reviewer and the store. For the evaluation, we manipulated the store features and replaced them with the corresponding product's features. The comparison highlighted the performance of each technique based on the set of applied features and graph/network structure.

The lowest precision was achieved by Wang et al. (2012) given they did not use any additional information concerning the reviews and reviewers. In this method, they considered the relation among the reviews, reviewers, and stores, while the relationships inside these entities were neglected using this approach. Moreover, as they focused on store-related features, their model could not be generalised to other domains. Furthermore, they also ignored a group of spammers and reviews during their detection process.

Similar to Wang et al. (2012), in Shehnepoor et al. (2017) the detection of group spammers' was not addressed, and their method instead exploited a limited number of features. Whereas, Fayazbakhsh and Sinha (2012) focused only on content-independent features which were found to reduce the accuracy of the proposed model and detected a limited range of spamming activities. The factor graph-based model in Lu et al. (2013) applied a complicated math formula, and they ignored the relation among the group of reviewers. Interestingly, they considered helpfulness as a highly accurate feature to detect spammers, although, spammers can spam this feature.

In summary, the MGSD model achieved remarkable results compared to existing graph-based, supervised and unsupervised techniques and focused on the explicit and implicit relationships that exist among entities. Since more current studies have considered limited types of entities, they could not capture all behaviours and clues related to spamming activities. Additionally, after analysis about the structure of data related to opinionated websites, it is evident that the review, reviewer and product can affect each other. As such, we cannot readily ignore this effect. Rather, the best model which can capture this concept is the graph-based model which addresses all entities along with their inter- and intra-relationships. Therefore, the MGSD model is considered to have the most effective features, in addition to proposing a novel set of features to improve the accuracy of opinion spam detection. These features have not been used in other models, but have been shown to monitor and track more precise characteristics associated with entities, as presented in Figs. 3 and 4.

Table 15
Comparison of state-of-the-art graph-based techniques.

Technique	Features	Performance scores			
		A	P	R	F
NetSpam model (Shehnepoor et al., 2017)	Content and behaviour-based	83.8	–	–	–
Network-based method (Fayazbakhsh & Sinha, 2012)	Content-based	82%	–	–	–
Factor graph-based model (Lu et al., 2013)	Content and behaviour-based	88.2%	–	–	–
Review graph-based model (Wang et al., 2012)	Content and behaviour-based	–	49%	–	–
MGSD model	Content, behaviour and relation-based	91.2%	90.8%	90%	91%

*A: Accuracy, P: Precisions, R: Recall, F: F-score.

Algorithm 1
Multi-iterative spamicity algorithm.

Entities: Reviewer; Review; Product; Group of Reviewers.
1. Step 1: Initialize entities
2. *Initialize spamicity scores of entities' samples as zero;*
3. Step 2: Calculate basic spamicity for entities' samples
4. *For each entity sample i:*
5. *Calculate SampleBasicSpamicityScore:*

$$BasicSpamicity\ (entity_i) = \frac{\sum_m normalized\ (f_m * w_m)}{\sum_m (w_m)}$$

6. Step 3: Update spamicity scores iteratively
7. *While (Iteration < 10)*
8. *For R from 1 to i:*
9. *Current Spamicity of R(i) = avg ((AvgSpamicity of PR(i) + (AvgSpamicity of RWR(i)) + Previous Spamicity of R(i));*
10. *For RW from 1 to j:*
Current Spamicity of RW(j) = avg ((AvgSpamicity of PRW(j)) + (AvgSpamicity of RRW(j)) + Previous Spamicity of RW(j));
11. *For P from 1 to k:*
Current Spamicity of P(k) = avg ((AvgSpamicity of RP(k)) + (AvgSpamicity of RWP(k)) + Previous Spamicity of P(k));
12. *For GRW from 1 to z:*
Current Spamicity of GRW (z) = avg ((AvgSpamicity of Rs for GRW(z)) + (AvgSpamicity of RWs for GRW(z)) + (AvgSpamicity of Ps for GRW (z)) + Previous Spamicity of GRW(z));
13. *While (true)*
If (((Current - Previous Spamicity of R) > 0.05) or ((Current - Previous Spamicity of RW) > 0.05) or ((Current - Previous Spamicity of P) > 0.05) or ((Current - Previous Spamicity of GRW | >0.05)))
14. *repeat line 7–13 for 1 iteration;*
15. else
16. *break;*
17. Step 4: Determine spam and non-spam entities' samples
18. For *each entity sample i:*
19. *Define the Review is Spam or Not Spam:*

$$SpamClass\ (Entity_i) = \begin{cases} spam\ , & Spamicity > 0.8 \\ not\ spam\ , & Spamicity \leq 0.8 \end{cases}$$

*P: Product, R: Review, RW: Reviewer, GRW: Group of Reviewers, PR: Product of Reviews, RWR: Reviewer of Review, PRW: Product of Reviewer, RRW: Review of Reviewer.

7. Conclusion

By increasing the use of online reviews relating to the purchasing decisions of consumers, analysing market trends and opinion mining applications, the need to filter any opinionated content resulting from spamming activities is crucial. The accuracy of opinion spam detection in this study was shown to improve by employing a graph-based model, which considered the relationships of all entities simultaneously within a unified structure. Proposing a new set of features and applying an effective set of existing features also offered more reliable output. The dataset used to implement this model was the real-world Amazon review dataset. Here, an accuracy of 91.2% was achieved when the MGSD model used a set of novel features, while without using these features; the accuracy value was 87.6%.

Generally, the proposed new features applied in the MGSD model provided more efficient detection compared to existing techniques. In evaluating this model, a new crowdsourced dataset, nearer to the characteristics of real spammers was used and generated rich in content. The synthetic spam reviews of our crowdsourced and Ott's datasets were distinguished with an accuracy of 93% and 95.3%, respectively. As such, this model has significant potential to be utilised in both business and research domains for cleansing opinionated documents.

8. Future works

For future study, deep learning due to its significance and results in various domains could be applied to detect and predict spamming activities for opinionated websites in real-time applications. Given there are vast amounts of online reviews, different big data analysis techniques can be applied to identify spamming activities more precisely and to lower performance costs. Future investigation could also consider social network connections and their influence on opinionated spam activities to detect spammers based on their connections and activities on social media.

Acknowledgements

This work is supported by Ministry of Higher Education (MOHE) and Research Management Centre (RMC) at the Universiti Teknologi Malaysia (UTM) under Research University Grant Category (R.J130000.7828.4F719).

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ipm.2019.102140.

References

Al Najada, H., & Zhu, X. (2014). iSRD: Spam review detection with imbalanced data distributions. *Proceedings of the 2014 IEEE 15th international conference on information reuse and integration (IEEE IRI 2014)* (pp. 553–560). . https://doi.org/10.1109/IRI.2014.7051938.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems, 30*(1–7), 107–117.

Cagnina, L. C., & Rosso, P. (2017). Detecting deceptive opinions: Intra and cross-domain classification using an efficient representation. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 25*(Suppl. 2), 151–174. https://doi.org/10.1142/S0218488517400165.

Chen, C., Zhao, H., & Yang, Y. (2015). Deceptive opinion spam detection using deep level linguistic features. *Natural Language Processing and Chinese Computing,* 465–474. https://doi.org/10.1007/978-3-319-25207-0.

Chengzhang, J., & Kang, D.-. K. (2015). Detecting the spam review using tri-training,. *2015 17th international conference on advanced communication technology (ICACT) 2015* (pp. 374–377). . https://doi.org/10.1109/ICACT.2015.7224822.

Choo, E., Yu, T., & Chi, M. (2017). Detecting opinion spammer groups and spam targets through community discovery and sentiment analysis. *Journal of Computer Security, 25*(3), 283–318. https://doi.org/10.3233/JCS-16941.

Dong, L. Y., Ji, S. J., Zhang, C. J., Zhang, Q., Chiu, D. W., Qiu, L. Q., et al. (2018). An unsupervised topic-sentiment joint probabilistic model for detecting deceptive reviews. *Expert Systems with Applications, 114*, 210–223. https://doi.org/10.1016/j.eswa.2018.07.005.

Dong, M., Yao, L., Wang, X., Benatallah, B., Huang, C., & Ning, X. (2018). Opinion fraud detection via neural autoencoder decision forest. *Pattern Recognition Letters,* 1–9. https://doi.org/10.1016/j.patrec.2018.07.013.

Etaiwi, W., & Naymat, G. (2017). The impact of applying different preprocessing steps on review spam detection. *Procedia Computer Science, 113*, 273–279. https://doi.org/10.1016/j.procs.2017.08.368.

Fayazbakhsh, S. K., & Sinha, J. (2012). *Review spam detection: A network-based approach.* 1–10 Final Project Report: CSE 590.

Goswami, K., Park, Y., & Song, C. (2017). Impact of reviewer social interaction on online consumer review fraud detection. *Journal of Big Data, 4*(1), 1–15. https://doi.org/10.1186/s40537-017-0075-6.

He, R., & McAuley, J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. *Proceedings of the 25th international conference on world wide web - WWW '16* (pp. 507–517). . https://doi.org/10.1145/2872427.2883037.

Hernández Fusilier, D., Montes-y-Gómez, M., Rosso, P., & Guzmán Cabrera, R. (2015). Detecting positive and negative deceptive opinions using PU-learning. *Information Processing & Management, 51*(4), 433–443. https://doi.org/10.1016/j.ipm.2014.11.001.

Hutto, C. J., & Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Eighth international conference on weblogs and social media (ICWSM-14)* (pp. 216–225). .

Jindal, N., & Liu, B. (2007). Analyzing and detecting review spam. *Seventh IEEE international conference on data mining (ICDM 2007)* (pp. 547–552). . https://doi.org/10.1109/ICDM.2007.68.

Karami, A., Zhou, B., & Baltimore, M. (2015). Online review spam detection by new linguistic features. *iConference 2015 proceedings* (pp. 1–5). .

Kim, S., Chang, H., Lee, S., Yu, M., & Kang, J. (2015). Deep semantic frame-based deceptive opinion spam analysis. *Proceedings of the 24th ACM international on conference on information and knowledge management - CIKM '15* (pp. 1131–1140). . https://doi.org/10.1145/2806416.2806551.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM, 46*(5), 604–632.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *International Biometric Society Stable, 33*, 159–174.

Li, H., Chen, Z., Liu, B., Wei, X., & Shao, J. (2014). Spotting fake reviews via collective positive-unlabeled learning. *2014 IEEE international conference on data mining, 2015* (pp. 899–904). . https://doi.org/10.1109/ICDM.2014.47.

Li, H., Chen, Z., Mukherjee, A., Liu, B., & Shao, J. (2015). Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns. *Proc. ninth int. AAAI conf. web soc. media* (pp. 634–637). .

Li, J., Ott, M., Cardie, C., & Hovy, E. (2014). Towards a general rule for identifying deceptive opinion spam. *Proceedings of the 52nd annual meeting of the association for computational linguistics. 1. Proceedings of the 52nd annual meeting of the association for computational linguistics* (pp. 1566–1576). . https://doi.org/10.3115/v1/P14-1147.

Li, L., Qin, B., Ren, W., & Liu, T. (2017). Document representation and feature combination for deceptive spam review detection. *Neurocomputing, 254*, 33–41. https://doi.org/10.1016/j.neucom.2016.10.080.

Lim, E. P., Nguyen, V. A., Jindal, N., Liu, B., & Lauw, H. W. (2010). Detecting product review spammers using rating behaviors. *Proceedings of the 19th ACM international conference on information and knowledge management - CIKM '10* (pp. 939–948). . https://doi.org/10.1145/1871437.1871557.

Liu, Y., & Pang, B. (2018). A unified framework for detecting author spamicity by modeling review deviation. *Expert Systems with Applications, 112*, 148–155. https://doi.org/10.1016/j.eswa.2018.06.028.

Lu, Y., Zhang, L., Xiao, Y., & Li, Y. (2013). Simultaneously detecting fake reviews and review spammers using factor graph model. *Proceedings of the 5th annual ACM web science conference on - WebSci '13* (pp. 225–233). . https://doi.org/10.1145/2464464.2464470.

Mukherjee, A., Kumar, A., Liu, B., Wang, J., & Hsu, M. (2013). Spotting opinion spammers using behavioral footprints. *Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 632–640). . https://doi.org/10.1145/2487575.2487580.

Mukherjee, A., Liu, B., & Glance, N. (2012). Spotting fake reviewer groups in consumer reviews. *Proceedings of the 21st international conference on world wide web - WWW '12* (pp. 191–200). ACM. https://doi.org/10.1145/2187836.2187863.

Mukherjee, A., & Venkataraman, V. (2014). *Opinion Spam Detection: An Unsupervised Approach Using Generative Models.* Techincal Report, UH 7.

Narayan, R., Rout, J. K., & Jena, S. K. (2018). *Review spam detection using opinion mining. Progress in intelligent computing techniques: Theory, practice, and applications, 519,* 273–279. https://doi.org/10.1007/978-981-10-3376-6_30.

Noekhah, S., Salim, N. B., & Zakaria, N. H. (2018). A comprehensive study on opinion mining features and their applications. *International conference of reliable information and communication technology. 5. International conference of reliable information and communication technology* (pp. 78–89). Cham: Springer. https://doi.org/10.1007/978-3-319-59427-9.

Ott, M., Cardie, C., & Hancock, J. T. (2013). Negative deceptive opinion spam. *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies* (pp. 497–501). .

Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011). Finding deceptive opinion spam by any stretch of the imagination. *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies. 1. Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (pp. 309–319).

Ren, H., Ji, Y., Yin, D., & Zhang, L. (2015). Finding deceptive opinion spam by correcting the mislabeled instances. *Chinese Journal of Electronics, 24*(1), 52–57. https://doi.org/10.1049/cje.2015.01.009.

Ren, Y., & Ji, D. (2017). Neural networks for deceptive opinion spam detection: An empirical study. *Information Sciences, 385–38*, 213–224. https://doi.org/10.1016/j.ins.2017.01.015.

Sánchez-Junquera, J., Villaseñor-Pineda, L., Montes-y-Gómez, M., & Rosso, P. (2018). Character N-grams for detecting deceptive controversial opinions. *International conference of the cross-language evaluation forum for European languages* (pp. 135–140). .

Sandulescu, V., & Ester, M. (2015). Detecting singleton review spammers using semantic similarity. *Proceedings of the 24th international conference on world wide web - WWW '15 companion* (pp. 971–976). . https://doi.org/10.1145/2740908.2742570.

Savage, D., Zhang, X., Yu, X., Chou, P., & Wang, Q. (2015). Detection of opinion spam based on anomalous rating deviation. *Expert Systems with Applications, 42*(22), 8650–8657. https://doi.org/10.1016/j.eswa.2015.07.019.

Shehnepoor, S., Salehi, M., Farahbakhsh, R., & Crespi, N. (2017). NetSpam: A network-based spam detection framework for reviews in online social media. *IEEE Transactions on Information Forensics and Security, 12*(7), 1585–1595. https://doi.org/10.1109/TIFS.2017.2675361.

Sun, H., Morales, A., & Yan, X. (2013). Synthetic review spamming and defense. *Proceedings of the 22nd international conference on world wide web - WWW '13 companion* (pp. 155–156). . https://doi.org/10.1145/2487788.2487864.

Wang, G., Xie, S., Liu, B., & Yu, P. S. (2011). Review graph based online store review spammer detection. *2011 IEEE 11th international conference on data mining* (pp. 1242–1247). . https://doi.org/10.1109/ICDM.2011.124.

Wang, G., Xie, S., Liu, B., & Yu, P. S. (2012). Identify online store review spammers via social review graph. *ACM Transactions on Intelligent Systems and Technology, 3*(4), 1–21. https://doi.org/10.1145/2337542.2337546.

Wang, Z., Hou, T., Song, D., Li, Z., & Kong, T. (2016). Detecting review spammer groups via bipartite graph projection. *The Computer Journal, 59*(6), 861–874. https://doi.org/10.1093/comjnl/bxv068.

Wu, G., Greene, D., Smyth, B., & Cunningham, P. (2010). Distortion as a validation criterion in the identification of suspicious reviews. *Proceedings of the first workshop on social media analytics - SOMA '10* (pp. 10–13). . https://doi.org/10.1145/1964858.1964860.

Xie, S., Wang, G., Lin, S., & Yu, P. S. (2012). Review spam detection via temporal pattern discovery. *Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining - KDD '12* (pp. 823–831). . https://doi.org/10.1145/2339530.2339662.

Xu, Y., Shi, B., Tian, W., & Lam, W. (2015). A unified model for unsupervised opinion spamming detection incorporating text generality. *Twenty-fourth international joint conference on artificial intelligence* (pp. 725–732). .

Ye, J., & Akoglu, L. (2015). *Discovering opinion spammer groups by network footprintsLecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)9284*. *Discovering opinion spammer groups by network footprints* Cham: Springer267–282. https://doi.org/10.1007/978-3-319-23528-8_17.

Zhang, W., Du, Y., Yoshida, T., & Wang, Q. (2018). DRI-RCNN: An approach to deceptive review identification using recurrent convolutional neural network. *Information Processing & Management, 54*(4), 576–592. https://doi.org/10.1016/j.ipm.2018.03.007.