Applied Intelligence (2023) 53:2189–2234
https://doi.org/10.1007/s10489-022-03427-1

Deceptive opinion spam detection approaches: a literature survey

Sushil Kumar Maurya[1] · Dinesh Singh[1] · Ashish Kumar Maurya[1]

Accepted: 22 February 2022 / Published online: 5 May 2022

Abstract

Nowadays, a large number of customers purchase products and services online. Customers can write their opinions in reviews to express the value and quality of purchased goods and services. These opinions are used to make purchase decisions by customers and design market strategies by sellers. The trustiness of online reviews highly affects a company's reputation and economic benefit. That is why online sellers hire people to write deceptive opinions to recommend their products or defame competitors' products. Detecting deceptive opinion spam has emerged as a challenging task. The article describes the publicly available review datasets and explores their deficiencies in finding deceptive opinion spam. This literature systematically unwinds prominent features and approaches that have been introduced to extricate the problem of deceptive opinion spam detection. Our primordial objective is to confer a solemnity analysis of recent papers on deceptive opinion spam detection that describes methodologies' features, strengths, and constraints. Finally, this work presents some crucial challenges and shortcomings of existing methodologies and introduces promising directions for further works. This paper presents a comprehensive review of recent research in deceptive opinion spam detection, which may prove helpful to researchers' best knowledge.

Keywords Deceptive opinion · Machine learning · Deep learning · Spammer · Spam detection

1 Introduction

With the increasing popularity of the internet, people nowadays use e-commerce websites to buy and sell products online. Online reviews are now playing a significant role in purchasing and selling products of customers and vendors, respectively. Based on these reviews, sellers can also design their additional marketing strategies and future planning of production. Unfortunately, this significance of reviews offers an excellent temptation for spamming, which incorporates malicious fable positive or negative opinions. Many companies appoint skilled people called spammers to write fictitious opinions to encourage their goods or services and gain economic benefits. Fictitious reviews are also called Deceptive Opinion Spam, which may be positive (hype) or negative opinions (defame). Some professional websites such as Yelp[1] and Amazon[2] have already taken the necessary steps to identify deceptive opinion spam to combat this problem. Nevertheless, deceptive opinion spam detection techniques still have much room for improvement.

Jindal and Liu [38] were the first researchers who discovered opinion spam in Amazon reviews using a supervised learning method. They described three types of opinions (reviews): deceptive opinions (Type 1, e.g., hype or defame), opinions on brands only (Type 2), and non-opinions (Type 3, such as advertisement, question-answering, and comments). Type 2 and Type 3 are called disruptive opinions, easily identified through human reading, as represented in Table 1. Using the bigram shingle method, they have detected duplicates and near-duplicates (similarity score of review pairs $> 90\%$) as opinion spam. They then labeled 470 deceptive reviews of Type 2 and Type 3 and applied supervised learning to detect them with 98.7% area under curve (AUC) value.

✉ Sushil Kumar Maurya
sushilbbiet@gmail.com

Dinesh Singh
dinesh_singh@mnnit.ac.in

Ashish Kumar Maurya
ashishmaurya@mnnit.ac.in

[1] Motilal Nehru National Institute of Technology Allahabad, Prayagraj 211004, India

[1] www.yelp.com

[2] www.amazon.com

Table 1 Types of reviews

Type	S. No.	Review Text	Description
I	1.	"Bought this for my Blackberry Phone and I have to say, this is pretty cool USB CORD:) I like the light in cord as it puts off a cool glowing effect in my room at night. Thanks for the great product!"	Look like Genuine Review
II	1.	"I don't trust HP and never bought anything from them"	Brand only opinion (HP Specific)
II	2.	"I like Apple mobile and never trust another mobile"	Brand only opinion (Apple Specific)
III	1.	"Buy this product at compuplus.com and Buy one get one free!"	Advertisement (Non-opinion)
III	2.	"Today 40% discount on Dell Lapy. Hurry Up!! Offer valid for limited period!"	Advertisement (Non-opinion)

Nowadays, spammers are experts in writing fake reviews (Type 1), which is hard to label manually. Therefore, checking the legitimacy of deceptive online reviews becomes a difficult challenge. For example, two hotel reviews were considered, which are given below [78].

- Opinion 1: "We recently stayed at the Intercontinental Hotel for a week. The hotel was in a wonderful location, and the service was great. We found all of the staff very helpful and prompt. We highly recommend the Intercontinental for any travel needs, whether it be for business or pleasure."
- Opinion 2: "This is a great place to stay while visiting Chicago. My husband and I went over the holidays to see my family and we stayed at this hotel. We could not have asked for nicer people! Everyone was always smiling and very helpfull! We usually stay at the Ramada...but will never stay anywhere other than the InterContinental Chicago hotel again!"

The first opinion is truthful, and the second is deceptive in the above two opinions. The second opinion contains the brand name (InterContinental Chicago), frequent use of first-person singular words (I, My, We), many exclamation marks (!), and focus on with whom (husband) were they. These words are strong deceptive indicators. Researchers have done many works for detecting deceptive opinion spam/spammers using various features and algorithms in different domains (i.e., hotel, restaurant, and product reviews).

1.1 Contribution in this literature survey

We have found several shortcomings in the existing literature survey, such as a lack of new methodologies developed in recent years and a lack of systematic summary of existing data resources. Our principal contributions in this article are as follows:

- Systematically summarize the various datasets and discuss some of their crucial deficiencies, such as lack of labeled datasets, limited features in the datasets, and imbalanced datasets.
- Categorize almost all the deceptive opinion spam detection features and divide these features into three major parts: content features, meta-data features, and behavioral features.
- The article describes the combination of textual and behavioral features to detect deceptive opinion spam efficiently and make a hybrid classification scheme. Here, we analyze the behavioral features that yield a good result as a hybrid. We discuss the revised weighting scheme on features proposed by various researchers and describe the effectiveness of the features used in the different deceptive opinion spam detection models in terms of matrices.
- The article presents deceptive opinion spam detection approaches that use various methodologies such as generative adversarial networks, recurrent neural network models, convolution neural networks, group spam detection approaches, and machine learning approaches, along with comprehensive evaluation. These methodologies and approaches have not yet been discussed in the existing literature.
- The paper points out some challenges and shortcomings related to opinion spam detection and introduces promising future research directions.

1.2 Comparison of related literature survey

Ren and Ji [86] have discussed different types of datasets, various features, and machine learning techniques in deceptive opinion spam detection. According to construction, they classified the dataset into four categories: based on rules, based on the humans, based on web filtering, and based on Amazon Mechanical Turk. The features have been analyzed in-depth by dividing them into textual and behavioral features. Moreover, they have presented machine learning techniques such as supervised, unsupervised, semi-supervised, and neural network models. Our article describes the deficiencies of datasets and explored novel features of group spamicity. Moreover, the article explains various feature dimensions and methodologies such as Machine learning-based, graph-based, pattern-based, and rule-based.

Vidanagama et al. [106] have discussed various approaches to find deceptive opinion spam in this field. These approaches are machine learning, network-based model, pattern-mining, some alternative methods such as big-data analytic. We discuss various types of content features and behavioral features in existing recent work. Besides, we describe novel generative adversarial network models like FakeGAN, spamGAN, bfGAN, attention-driven conditional GAN.

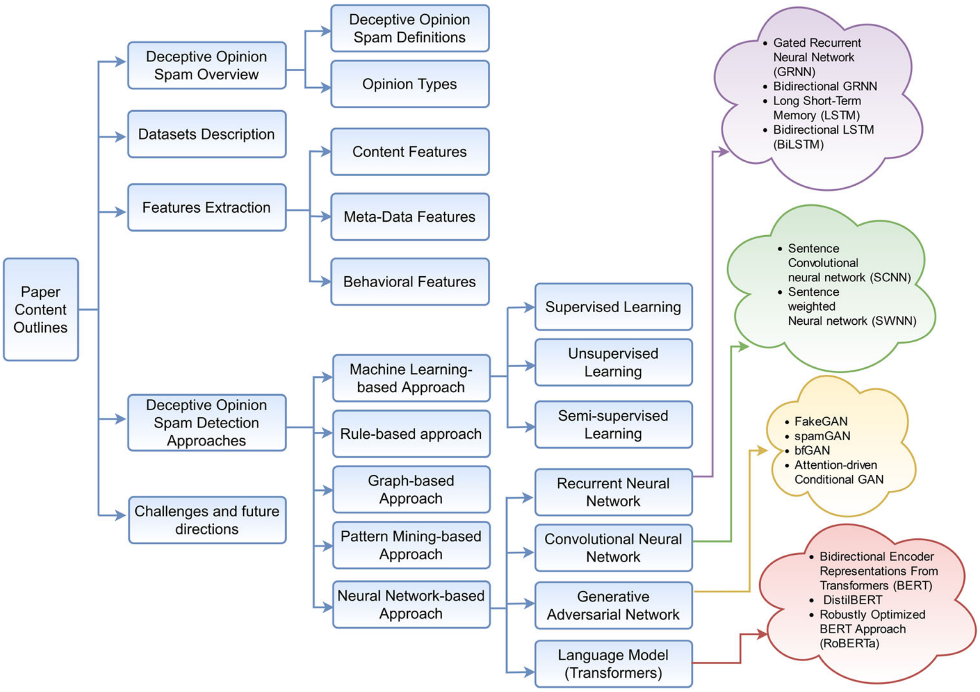
Hussain et al. [33] have explained the preprocessing steps (i.e., removing stop words or punctuation, stemming, and POS tagging) for a linguistic text; and discussed the different types of feature extraction techniques. They categorized deceptive opinion spam detection techniques into two parts, i.e., machine learning techniques and lexicon-based techniques. These techniques are primarily used in review text, as it is challenging to detect deceptive opinion spam by perceiving product reviews in chronological order. Conversely, there is crucial information gained from the reviewers' review and the relationship among reviewers when many reviewers review the targeted products in a short interval of time. Therefore, our paper includes several new models which are based on group behavior features.

The improvements of recently developed approaches addressed almost all research problems in deceptive opinion spam detection. Therefore, future studies need to include a survey that organizes all previous and current methods of identifying deceptive opinion spam. As a result of this comprehensive literature analysis, we believe that additional, alternate methods could be discovered to reduce the damage caused by deceptive opinion spam. For this purpose, we emphasize the need for a distributed researcher-enabled information system by presenting a synthesized survey from reliable sources. Therefore, this research work is valuable in comparison to its counterparts due to the considerations mentioned above.

1.3 Structure of the literature survey

This survey includes deceptive opinion spam detection fundamentals, methods & technologies, research gaps, and future directionas, as shown in Fig. 1. The existing papers' contributions are included in the survey regarding spam review detection and the distribution of the research article published in the last couple of years shown in Fig. 2. A lot of work has been carried out in the last decade in deceptive opinion spam detection. The remainder of this work is structured as follows: Section 2 discusses the various benchmark datasets and their deficiencies. Section 3 introduces the different types of features extraction and representation techniques. Section 4 discusses the various feature dimensions used in related research papers. Sections 5 and 6 describe various approaches and evaluation metrics, including the complexity of spam detection techniques, respectively. In Section 7, the paper describes many challenges in detecting

Fig. 1 Overview of literature survey


Paper Content Outlines
Deceptive Opinion Spam Overview
Datasets Description
Features Extraction
Deceptive Opinion Spam Detection Approaches
Challenges and future directions
Deceptive Opinion Spam Definitions
Opinion Types
Content Features
Meta-Data Features
Behavioral Features
Machine Learning-based Approach
Rule-based approach
Graph-based Approach
Pattern Mining-based Approach
Neural Network-based Approach
Supervised Learning
Unsupervised Learning
Semi-supervised Learning
Recurrent Neural Network
Convolutional Neural Network
Generative Adversarial Network
Language Model (Transformers)
• Gated Recurrent Neural Network (GRNN)
• Bidirectional GRNN
• Long Short-Term Memory (LSTM)
• Bidirectional LSTM (BiLSTM)
• Sentence Convolutional neural network (SCNN)
• Sentence weighted Neural network (SWNN)
• FakeGAN
• spamGAN
• bfGAN
• Attention-driven Conditional GAN
• Bidirectional Encoder Representations From Transformers (BERT)
• DistilBERT
• Robustly Optimized BERT Approach (RoBERTa)

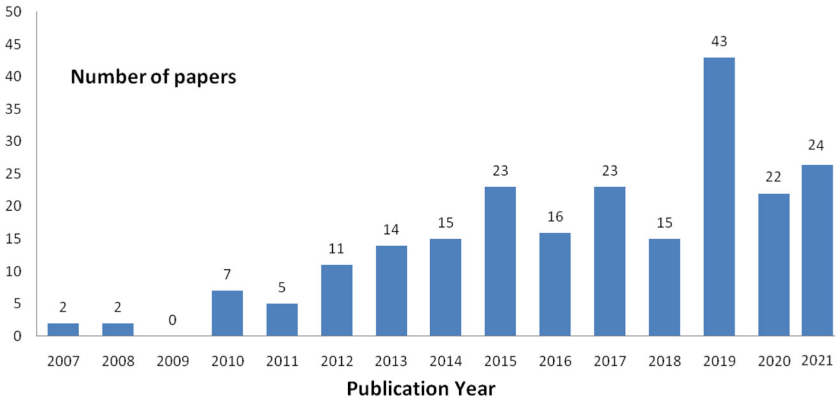


Fig. 2 Distribution of published papers

deceptive opinion spam and introduces some promising directions for future research. Finally, Section 8 discusses the conclusions of the literature survey.

2 Dataset used in deceptive opinion spam detection

This section describes various review datasets of products, stores, hotels, restaurants, and movies by analyzing 200 existing research papers—distribution of the datasets is shown in Fig. 3. Jindal and Liu [39] first crawled product reviews from the most successful e-commerce site amazon.com, and extracted 6.7 million products, 2.14 million reviewers, and 5.8 million reviews for their experiments. They found that 10% of reviewers wrote more than one review on at least one product; in 40% of the cases, reviews were posted within a day with the same contents (duplicates), and in 8% of the cases, a user wrote more than two reviews on the same product. In 200 papers, around 29% of papers, the researchers used Amazon data for their experiments. In contrast, approximately 27% of researchers have used Amazon Mechanical Turk (AMT) dataset (also known as Gold standard deceptive opinions by paying anonymous online workers). Ott et al. [78] gathered 400 truthful five-star reviews from 20 hotels (Chicago area) on TripAdvisor with 400 positive fake (deceptive) reviews on the same 20 hotels from AMT to complete his task. Using word bag of features, they claimed accuracy of 89.6%. Further, [79] mined negative reviews from six famous online review communities like Orbitz, TripAdvisor, Priceline, Hotels.com, Expedia, and Yelp and got 86% accuracy.

Mukherjee et al. [73] analyzed that AMT reviews are not genuinely fake because they don't have sufficient domain knowledge and the same psychological state of mind while writing a bogus review, such as the expert authors write real deceptive reviews. They crawled filtered (deceptive) and unfiltered (truthful) reviews from Yelps real-life data (YelpChi: reviews for some popular restaurants and hotels in the Chicago area) and found 67.8% accuracy using n-gram features. To improve the accuracy, they proposed a set of behavioral features of the user and their review. Li et al. [53] constructed a cross-domain gold-standard dataset in different domains, i.e., Restaurant, Hotel, and Doctor. The given dataset consists of three types of opinions, i.e., truthful customer reviews (by customers), domain-expert

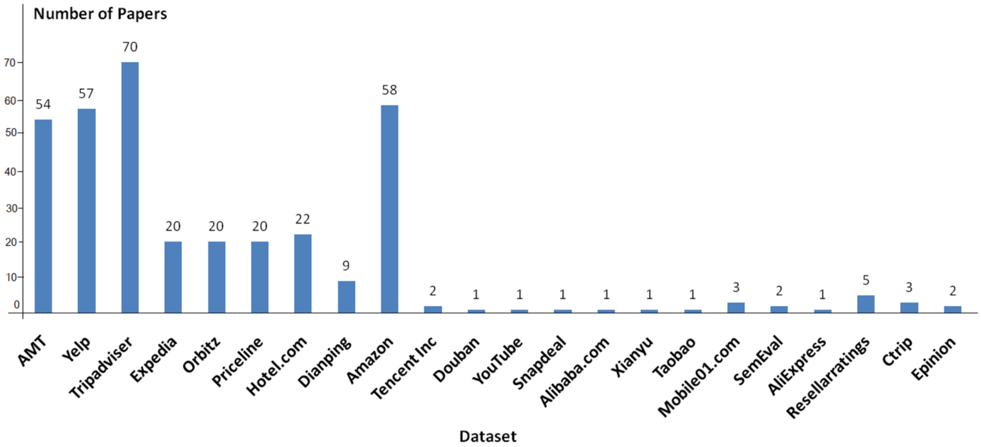


Fig. 3 Distribution of datasets

fake opinion spam (by the employee), and crowd-sourced fake opinion spam (by Turker). Wang et al. [108] used online user opinions on the store from Resellerratings[3], containing meta-data such as user id, opinions, and store star ratings with posting date-times and links of stores. They collected 408470 reviews written by 343603 users on 14561 stores and proposed a heterogeneous graph-based iterative model to capture the connections among stores, reviews, and users, to identify suspicious users.

Li et al. [47] have crawled 60 K product reviews from Epinions[4] and evaluated helpfulness scores for detecting deceptive reviews. Li et al. [48] used Chinese reviews from popular review hosting site Dianping[5] (the Chinese equivalent of yelp.com). Dianping has its filtering algorithm to detect deceptive reviews. This filtering algorithm has high precision but unknown recall indicating that remaining reviews may not be genuine. They created a balanced dataset of 3476 deceptive (positive labeled) reviews and 3476 unknown (negative labeled) reviews and applied PU Learning Algorithms to that dataset.

Rayana and Akoglu [85] used three datasets YelpChi (67,395 reviews, 201 restaurants & hotels), YelpNYC (contains 359,052 reviews for 923 restaurants located in NYC), YelpZip (contains 608,598 reviews for 5,044 restaurants located in NY state with different zip-code NJ, VT, CT, and PA) collected from Yelp.com. To find deceptive opinion spam and fraud reviewers, they proposed a user-product bipartite graph model such as FraudEagle (unsupervised) and SpEagle (semi-supervised). Yang [131] has built a labeled dataset (hotel reviews of the English version) from ctrip.com using expert human judges. Chen and Chen [13], Wang et al. [107], and Yuan et al. [135] used the mobile review data (fake reviews, regular reviews, and replies) collected from the Taiwan website Mobile01[6].

This section shows that researchers use various datasets. Some of them are benchmark datasets used by other researchers to do their experiments using different technologies and methodologies. Researchers may face some open issues and deficiencies in using these datasets, described in the next section.

2.1 Deficiencies in dataset

This literature found that there are still several deficiencies in datasets for deceptive opinion spam detection. Significant deficiencies are elaborated on below.

2.1.1 Lack of labeled dataset

The lack of a labeled dataset makes detecting deceptive opinion spam a challenge. Jindal and Liu [39] stated that only type 2 (brands only review) and type 3 (advertisement and comments) reviews could label easily. Type 1 review is hard to do labeling because the spammers can fabricate their opinions just like any other genuine opinion. Ott et al. [78] collected negative opinions from AMT and positive opinions from TripAdvisor, but this dataset has only a textual feature. To build a supervised classification model, researchers need standard labeled datasets for training to identify deceptive or truthful opinions.

2.1.2 Limited features in dataset

Some essential features like the spammer's IP address and the location when posting the review; can improve the model's accuracy. Many researchers used the dataset by crawling, with limited features or lack of essential features. Some gold-standard datasets [53, 63, 78] contain limited features (only textual features) for classification. Thus, the unavailability of multidimensional datasets is a big challenge in this area.

2.1.3 Drifting in spamming features

Nowadays, many reviews are posted every day on commercial websites like Amazon, Yelp, Tripadvisor, etc. Datasets are growing rapidly, creating difficulty in semantic analysis (SentiWordNet and WordNet) for review and requiring more computation power. It can be noticed that the spam features fluctuate substantially after the features are extracted, while the non-spam features have held steady [60]. Consequently, the classifier trained by the previous data will not be used to detect spam in the current dataset as the spam features still drift over time.

2.1.4 Multilingual words used in reviews

A review can be written in more than one language; handling multilingual is quite challenging. Some researchers have proposed solutions for languages other than English like Chinese [32, 128] and Arabic [92]. Many researchers used linguistic inquiry and word count (LIWC[7]) features with many English words into 80 psychologically meaningful dimensions. Thus, in-depth research needs to be done on detecting deceptive opinion spam in multilingual reviews.

[3] www.resellerratings.com

[4] www.epinions.com

[5] www.dianping.com

[6] www.mobile01.com

[7] liwc.wpengine.com

2.1.5 Imbalanced dataset

Many researchers found a common issue, i.e., imbalance class in the datasets that are used for classification. Imbalanced datasets are a case for classification problems where the class distribution is not uniform among the classes. For example, most reviews do not spam in a deceptive opinion spam detection dataset, and very few classes are labeled as spam. Therefore, an extra effort for balancing datasets is required. Many works of literature use many techniques to resolve this problem, like re-sampling (over-sampling and under-sampling) and ensembling (bagging and boosting).

3 Features or indicators used in deceptive opinion spam detection

In the detection model, features are independent indicators whose value indicates the review's degree of spamicity. Selecting the more relevant features is crucial to improve the model's accuracy. It also decreases the model's complexity as we avoid the least significant/unnecessary feature data. Jindal and Liu [39] classified features into three types based on information related to a review: product-centric, reviewer-centric, and review-centric. Several features have been presented in the literature regarding deceptive opinion spam detection, as shown in Fig. 4.

3.1 Review content features

Reviews' content features play a significant role in detecting deceptive opinion spam by using Machine learning classifiers and Neural networks. Analyzing the efficacy of features, i.e., which features are the strongest dominant indicators of spamicity, is one of the most crucial aspects of deceptive opinion spam identification. In content features, we need to transform review text into a vector. Here some of the important content features extraction techniques are discussed, which have been used by various researchers.

3.1.1 Bag of words (BOW)

The BOW is a way to extract features from text data. These features are called n-grams, selected from a certain sequence by choosing n contiguous words. These bag of words are referred as a uni-gram (n=1), bi-gram (n=2) and tri-gram (n=3). BOW features give 89.6% accuracy on AMT gold standard dataset [78], while it gives only 67.8% accuracy on the Yelp dataset [73]. However, [22] noted that in supervised learning, the use of n-gram alone has proven insufficient because the features selected are not available in real-world deceptive reviews. Saeed et al. [92] have used n-gram features on Arabic text review to determine sentiment words.

3.1.2 Term frequency (TF)

Term frequency (TF) is commonly used in text mining to identify how many times a word or term appears in a document. The TF score for a word 't' in a document 'd' is computed as follows:

$$TF(t, d) = \frac{count\ of\ term(t)\ in\ document\ (d)}{total\ words\ in\ d} \tag{1}$$

Ott et al. [78] and Jindal and Liu [39] have used term frequency to convert uni-gram or bi-gram word into the vector form for supervised classifiers.

3.1.3 Term frequency-inverse document frequency (TF-IDF)

The TF-IDF indicates the importance of a word/term 't' for a document 'd' in a corpus or document set 'D'. It is numerically calculated as follows:

$$TF - IDF(t, d, D) = TF(t, d) * IDF(t, D) \tag{2}$$

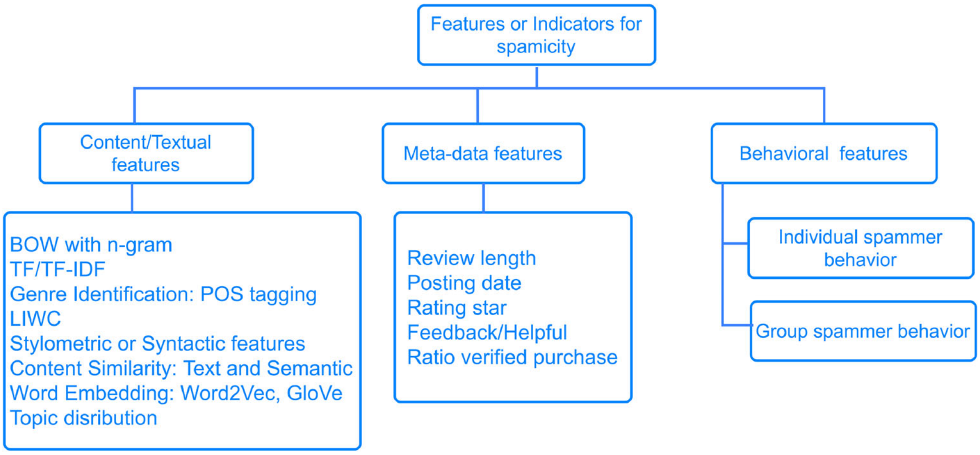


Fig. 4 Features classification

where inverse document frequency (IDF) is a quantitative weight utilized in a text document collection to quantify the importance of a word, the IDF value for a word 't' in a document set 'D' is defined as follows:

$$IDF(t, D) = log\left(\frac{D}{DF(t) + 1}\right)$$

where $DF(t)$ denotes the number of documents containing a word 't'.

3.1.4 Linguistic inquiry and word count (LIWC)

Linguistic inquiry and word count (LIWC) is a word counting tool used for text analysis to learn how our emotions, thoughts, motivation, personality, and social context are revealed by the words we use in daily language. Pennebaker et al. [82] introduced the famous LIWC software to create psychological features for the reviews. Existing studies [34, 78] have automatically combined LIWC features with n-grams to detect deceptive reviews using supervised machine learning. Li et al. [53] explored LIWC into three categories: sentiment, spatial detail, and the first person singular pronouns.

3.1.5 Part of speech (POS)

Frequency distribution of part-of-speech tags (POS tags or Grammatical tags) in a text (or corpus) often differentiates between informative and insubstantial writing. In POS tagging, it is determined that which words are acted as nouns, pronouns, adverbs, verbs, etc. POS tagger in NLTK library maps certain words to a specific tag. For example, NN for a singular noun, NNS for a plural noun, VB for the base verb, RB for an adverb, etc. [78] explored the POS distribution of the reviewed text and used the POS tag frequency as a vector for detecting deceptive opinion spam. The work concludes that spammers can write imaginative opinions using more adverbs, verbs, or pronouns. In contrast, genuine users write informative reviews using more nouns or adjectives to share their experiences. They found POS feature is less effective than BOW in the same domain. However, [53] have stated that POS and LIWC features are more robust in cross-domain settings when using a sparse-additive-generative model.

3.1.6 Stylometric

Stylometric is the statistical analysis of variance in literary style for evidence of the author's identity and authenticity. There are various stylometric-based features such as lexical features (word-based features and character-based features) and syntactic features (part of speech, punctuation, emotion words, and function words) that are used in opinion spam detection [99]. Feng et al. [23] used probabilistic context-free grammars (PCFGs) to identify authorship. A PCFG is a list of production rules, and each rule is associated with a probability.

3.1.7 Maximum content similarity and semantic similarity

A strong indicator of spamicity is duplicate reviews on a reviewer's targeted products. The first well-known characterizations of review deception were presented by [39], in which the author described duplicate reviews as spam reviews across the Amazon dataset. They used Jaccard similarity to calculate duplicate reviews. Algur et al. [4] introduced a conceptual-level similarity to find deceptive opinion spam using product features that have been commented on in the reviews. Lau et al. [45] proposed a semantic-language model, a text mining-based computational model, to identify deceptive opinion spam. Kim et al. [42] used text semantic representation to find spam reviews.

3.1.8 Word embedding (Word2Vec and GloVe)

One of the most common representations of text documents is word embedding. Many researchers used word embedding techniques to represent every review into the vector, such as Word2Vec, GloVe, etc. Mikolov et al. [68] introduced a Word2Vec model for representing words into the vectors using shallow neural networks. In this model, each word is defined in many dimensions that describe relation with other words and capture more semantic information. Trained vectors encode many linguistic regularities and patterns represented as linear translations. This model is based on two techniques: skip-gram and a continuous bag of words (CBOW). This method of representation causes terms with a similar meaning to have a similar representation, which will increase the classifier's efficiency. Word2Vec is a predictive model and cannot capture the global word co-statistics of a given corpus. To address this issue, [83] discovered a count-based GloVe model that differs from the Word2Vec model. Global vector (GloVe) incorporates word-word co-occurrence matrices from a corpus to obtain word vectors. Liu et al. [62] implemented multi-modal using Keras and pre-train 300-dimensional GloVe embedding to a text representation.

3.1.9 Topic distribution

The distribution of words/terms for every latent topic is known as topic-word distribution. A generative LDA (latent Dirichlet allocation) topic model called TopicSpam was built by [52] to detect deceptive opinion spam. LDA topic model [10] has been utilized in machine learning to find a latent topic in a document collection. This model gives the mysterious disparities between the topic-word distributions of genuine opinion and deceptive opinion spam. Jia et al. [37] have given a method to extract features based on LDA, and they compared different combinations of LDA and classifiers to achieve reasonable accuracy. Dong et al. [19] have given a probabilistic model unsupervised-topic-sentiment-joint (UTSJ) for mining sentiment and topic pairs that are based on LDA. They have used two parameters (Gibbs sampling iterations and topics) in their model. They found that the UTSJ model's perplexity is often lower than the JST model [58] and the LDA model [52]. Cao et al. [12] have explored a method that combined two granular implicit semantic features: coarse-grained (such as sentence, topic, and document) and fine-grained (such as term or word). The coarse-grained features from the topic distribution of reviews by concatenating LDA and backpropagation (BP) neural network and fine-grained features are obtained from the word2vec representation of reviews using deep learning (like LSTM, BiLSTM, and TextCNN) models. They found that the LDABP model combined with the TextCNN model achieved a good result than other models.

3.2 Meta-data features

In addition to its actual content, information about a review is called meta-data, such as review posting time, star-rating given by the reviewer, reviewer's identity, Geo-location of posting time (IP address of computer or mobile). The existing works have used meta-data features to capture the unusual spammer's behavior and create more effective deceptive opinion spam models to calculate spamicity scores. For example, suppose any user frequently writes multiple positive (with the highest rating) or negative (with the lowest rating) reviews of a particular product using different user-ids on the same computer. In that case, it may be a suspicious review. In hotel reviews, one can sometimes see positive reviews from places around the hotel; These reviews may not be authentic; However, hotel reviewers usually write their opinions only when they are away from that hotel's geographical location. The essential meta-data features are described below.

3.2.1 Review posting date and rating

It was observed that the posting date and ratings of spammer's review differ from legitimate reviewers. Xie et al. [125] used the posting time and ratings of the review to discover correlated temporal patterns of spammer's reviews. They constructed a multidimensional time series model to detect singleton review spam attacks based on aggregate statistics. Lim et al. [57] used early-deviation and general rating deviation to calculate behavior scores. Thus, more significant aggregate rating changes can help find dishonest reviewers. Many researchers [22, 38, 57, 72] have used Rating-related features to make it a good indicator of deceptive opinion.

3.2.2 Verified customer

In verified customer reviews, the customers can post their positive or negative opinion about the services or products on the e-commerce website only if they have taken those services or products from the same website. It is recognized that the unverified review ratio can be used as a benchmark to determine if a review is genuine [69].

3.2.3 Ratio of verified purchase

Verified purchase ratios are the indication of genuine reviews. The verified purchase ratio of a customer is the ratio of the number of verified purchases and the total number of reviews. Higher the ratio means more trustworthy reviews. The most potent indicator used in the article [22] is the ratio of verified purchases of Amazon. It helps to optimize the precision of methods in any detection model deeply.

3.2.4 Helpful or useful

Review helpfulness is one of the essential characteristics associated with online customer reviews [64]. Helpfulness has been generally used as the essential method of estimating how a customer evaluates a review. The customers can select reviews by their number of helpfulness. Wang et al. [112] investigated the determinant factors (like review length, review volume, review readability, and timeliness) that may influence the supportiveness of the review, which depends on the characteristics of the review.

3.3 Behavioral features

This section outlines the spamming behavior indicators used to estimate the spamming ranking of the suspicious reviewers. Since spammers can be detected by observing their various behavioral patterns, then many authors used unsupervised learning to identify these behavioral patterns. Two main behavioral features are Individual spammer behavioral features and Group spammer behavioral features.

3.3.1 Individual spammer behavioral features

Most of the existing researches [22, 36, 57, 72, 103, 108] have been done on the identification of fake reviews and individual spammers. They have used various types of features extracted from the reviewer's behaviors or individual reviewer's behaviors. For building a detection model, various types of individual spammer behavioral features are needed; these are described as:

Content similarity Spammers generally choose a similar word for writing a new review every time, without spending their own time [39]. Thus, the review content may be similar. The content similarity of the review pair can be calculated by the cosine method.

$$f_{CS}(a) = max_{r_i, r_j \in R_a,\ i<j}\ cosine(r_i, r_j) \tag{3}$$

where R_a is the collection of all reviews given by the author 'a'; r_i, r_j are i^{th}, j^{th} reviews respectively, written by the author 'a'.

Percentage of positive reviews: The high percentage of positive reviews are more likely to be spam reviews because more than 80% reviews are written as positive reviews by a spammer [39, 103].

Maximum number of reviews Posting several reviews within a day often indicates uncommon behavior of reviewer [36, 57, 103]. It is computed by dividing the maximum number of reviews posted by an author in a day by the maximum value for that data.

$$f_{MNR} = \frac{MaxReview(a)}{max_{a \in A}(MaxReview(a))} \tag{4}$$

where 'A' is the set of authors, and $MaxReview(a)$ is the total number of reviews posted by an author 'a' within a single day.

Reviewing burstiness or activity window Reviewing Burstiness is defined as a difference between the last posting date and the first posting date in short time intervals [22, 103]. The Reviewing burstiness can be calculated as:

$$f_{BST}(a) = \begin{cases} 0 & if\ LP(a) - FP(a) > \tau, \\ 1 - \frac{LP(a) - FP(a)}{\tau} & otherwise \end{cases} \tag{5}$$

where LP(a) and FP(a) are the last and first posting date of review by an author 'a', respectively. The Time interval τ may be 1 or 2 months.

Burst review ratio (BRR) The BRR of an author 'a' is the number of reviews divided by the total number of reviews in burst time. A reviewer is more likely to be a spammer if he/she posts many reviews in bursts time [22].

$$BRR(a) = \frac{|B_{a*}|}{|V_{a*}|} \tag{6}$$

where B_{a*} denotes the set of reviews posted by an author 'a' in review burst time and V_{a*} denotes the set of all reviews that author 'a' wrote towards all products.

Ratio of first reviews Spammers would try to be written the first review on the target product as this enables them to influence the opinion and sentiment. Mukherjee et al. [72] is calculated as the ratio of first reviews to the total number of reviews for each author.

$$f_{RFR} = \frac{|\{r \in R_a : r\ is\ a\ first\ review\}|}{|R_a|} \tag{7}$$

where R_a is total reviews of an author 'a'.

Extreme rating One or five stars are extreme ratings on a rating scale of 5-star. Spammers are likely to give extreme ratings to damage/advertise products [36, 72]. The formula to calculate extreme rating is given as:

$$f_{ExtR} = \begin{cases} 1 & if\ *(r_a, p(r_a)) \in \{1, 5\}, \\ 0 & otherwise \end{cases} \tag{8}$$

where $*(r_a, p(r_a))$ is star rating of r_a given by an author 'a' on product 'p'.

Rating deviation Spammers are likely to give different ranting from other reviewers to promote or damage products [22, 36, 103]. It is defined by

$$f_{Dev} = \begin{cases} 1 & if\ \frac{|*(r_a, p(r_a)) - E[*(r_{a' \neq a}, p(r_a))]|}{4} > \beta, \\ 0 & otherwise \end{cases} \tag{9}$$

where $*(r_a, p(r_a))$ is star rating of review r_a given by an author 'a' on product 'p', and β is a threshold value.

Early time frame Lim et al. [57] stated that spammers frequently review early to deceive customers as the early reviews have a larger effect on customers' opinions on a product.

$$f_{ETF}(r_a) = \begin{cases} 1 & if\ ETF(r_a, p(r_a)) > \beta, \\ 0 & otherwise \end{cases} \tag{10}$$

where $\beta = 0.69$ and $ETF(r_a, p)$ indicate how early an author reviewed the product 'p'. It is calculated as:

$$ETF(r_a, p(r_a)) = \begin{cases} 0 & if\ LP(a, p) - LD(p) > \delta, \\ 1 - \frac{LP(a,p) - LD(p)}{\delta} & otherwise \end{cases}$$

where LD(p) is launch date of product 'p', LP(a,p) is the last review posting date by an author 'a' for that product, and $\delta = 7$ is the threshold value, i.e., 7 months.

3.3.2 Group spammer behavioral features

Group spamming is a group of spammers who collectively write fabricated opinions to encourage or denigrate targeted products. Group spammer can be defined as a single spammer with more than one user ids, or multiple spammers, or a combination of both. The following indicators have been used for detecting group spammers by many researchers [36, 50, 57, 71, 117, 127].

Group time window (GTW) Members of the spammer's group are more likely to write reviews together for certain target products during a given time interval [71]. Therefore, the active GTW is defined as:

$$GTW(g) = max_{p \in P_g}(GTW_p(g, p))$$
$$where\ GTW_p(g, p) = \begin{cases} 0 & if\ LD(g, p) - ED(g, p) > \tau \\ 1 - \frac{LD(g,p) - ED(g,p)}{\tau} & otherwise \end{cases} \tag{11}$$

where P_g is the total products reviewed by group 'g' and threshold, τ is 2.87. $LD(g, p)$ and $ED(g, p)$ are the lastest date and earliest date of posted reviews for the product $p \in P_g$, respectively.

Group average time window (GATW) GATW is calculated using the standard deviation of review dates by considering the average time distribution of the review [36, 117]. Therefore, the GATW is defined as:

$$GATW(g) = avg_{p \in P_g}(TW_p(g))$$
$$where\ TW_p(g) = \begin{cases} 1 - \frac{SD_p}{T} & if\ SD_p < \tau \\ 0 & otherwise \end{cases} \tag{12}$$

where SD_p is the standard deviation of review dates. P_g is a set of targeted products reviewed by group 'g' and τ is threshold value may be 30 days.

Group deviation There is a considerable variation between the spammer group's ratings and the genuine reviewers' ratings on the targeted product. The higher the deviation, the more skeptical the group is. The group deviation in 5-star rating scale [36, 71] is formulated as:

$$GDev(g) = max_{p \in P_g}(Dev(g, p)) \tag{13}$$

where $Dev(g, p)$ is the deviation of rating given by the group on a product 'p' as follows:

$$Dev(g, p) = \frac{|r_{p,g} - \overline{r}_{p,g}|}{4}$$

where $r_{p,g}$ and $\overline{r}_{p,g}$ is average rating given by group members 'g' and the average rating given by other members (genuine reviewers) not in 'g', for product 'p', respectively.

Group content similarity and group member content similarity Group content similarity occurs when group spammers copy reviews among themselves. The group content similarity can be calculated as:

$$GCS(g) = max_{p \in P_g}(RCS_G(g, p))$$
$$RCS_G(g) = \underset{m_i, m_j \in g, i<j}{avg}\left(cosine\_sim(rc(m_i, p), rc(m_j, p))\right) \tag{14}$$

where $RCS_G(g, p)$ represents an pairwise contents similarity of reviewer group 'g' on a product 'p', the $rc(m_i, p)$ and $rc(m_j, p)$ are the content of reviews posted by i^{th} and j^{th} members of a group 'g', respectively.

Group member content similarity occurs when the members of a group 'g' copy their previous reviews. Suppose a large number of group members write duplicate reviews, more likely to be spam [39]. The group member content similarity can be calculated as:

$$GMCS(g) = \frac{\sum_{m \in g} RCS_M(g, m)}{|g|}$$
$$RCS_M(g, m) = \underset{p_i, p_j \in P_g, i<j}{avg}\left(cosine\_sim(rc(m, p_i), rc(m, p_j))\right) \tag{15}$$

where $RCS_M(g, m)$ represents an average pairwise content similarity posted by a member 'm' of group 'g' on products $p \in P_g$. For a member 'm' of group 'g', $rc(m, p_i)$ and $rc(m, p_j)$ are the content of review on the i^{th} and j^{th} products, respectively.

Group early time frame (GETF) If group members are among the first reviewers to write a review for a product, they significantly impact sentiment on the product [57]. It is defined as:

$$GETF(g) = \underset{p \in P_g}{max}(GTF(g, p)) \tag{16}$$

where $GTF(G, P)$ is the time frame, which defined how early a group 'g' posted the reviews on product 'P'. It is calculated as follows:

$$GTF(g, p) = \begin{cases} 0 & ifLP(g, p) - LD(p) > \beta \\ 1 - \frac{LP(g,p) - LD(p)}{\beta} & otherwise \end{cases}$$

where $LD(g, p)$ and $LP(g, p)$ is launch date of product 'p' and last posting date of review on that product by group 'g', respectively, β is a threshold value (nearly 6 months).

Group size ratio (GSR) Group size ratio (GSR) is the ratio of the number of reviewers in a group to the total number of reviewers for a product. The GSR [71] is defined as:

$$GSR(g) = \underset{p \in P_g}{avg}(GSR_P(g, p))$$
$$GSR_P(g, p) = \frac{|g|}{|M_P|} \tag{17}$$

where $|g|$ is the size of group and $|M_P|$ is the total number of reviewers for a product 'p'.

Group Review Tightness (GRT): Group review tightness is used to calculate the group member's degree who collaborate in writing deceptive reviews on targeted products [36, 117]. It is formulated as:

$$GRT(g) = \frac{|V_g|}{|R_g||P_g|} \tag{18}$$

where $|V_g|$ is the total number of reviews (genuine or deceptive reviews) posted by group 'g'. $|R_g|$ total reviewers of the group 'g', and $|P_g|$ is the total number of products reviewed by group 'g'.

Group size (GS) In a large group, members likely to be work together are fewer. A larger group of spammers are more damaging to the product review. The normalized group size [36, 71] is formulated as:

$$Narmalized\ group\ size\ GS(g) = \frac{|g|}{max(|g_i|)} \tag{19}$$

where $max(|g_i|)$ is the maximum group size.

GS is calculated by a logistic function [117] as follows.

$$GS(g) = \frac{1}{1 + e^{-(|R_g|-3)}} \tag{20}$$

where $|R_g|$ represents the total reviewers in group 'g'.

Group support count (GSUP) It is the total number of products to work together. GSUP [71] is calculated as:

$$GSUP(g) = \frac{|P_g|}{max(|P_{g_i}|)} \tag{21}$$

where $|P_g|$ represents the total number of products targeted by group 'g' and $max(|P_{g_i}|)$ is the maximum number of product targeted by all recoverable groups.

Group Co-Activities (GCA) Group members collectively post reviews in a short period may be viewed as a suspected co-active spam expedition. GCA [36, 117] has been normalized by logistic function as follows:

$$GCA(g) = \frac{1}{1 + e^{-|CA_g|}} \tag{22}$$

where $|CA_g|$ represents the number of suspected spam expedition of group 'g'.

Group Co-Active Review Ratio(GCAR) GCAR is described as the ratio of reviews posted by group members $|R_{CA_g}|$ to the total number of reviews $|V_g|$ in the co-active time interval of group g. It is given as follows [36, 117]:

$$GSUP(g) = \frac{|R_{CA_g}|}{(|V_g|)} \tag{23}$$

GSRank (Group Spam Rank) Mukherjee et al. [71] have proposed a relation (among individual reviewers, groups, and products) based model called GSRank. This model achieved good results in identifying the potential spamming groups compared to the frequent itemset mining-based model. They evaluate GSRank by three relational models: group Spam-Member model, group Spam-Products model, and member Spam-Product model. The experimental results proved that GSRank is a good learner for ranking algorithms than supervised classification.

4 Feature dimensions for deceptive opinion spam detection

The online shopping environment is seriously impacted by deceptive opinion spam and damages the reputation of a target product or brand. Identifying the best features that better characterize the spamming activity of fraudulent reviewers is a main challenging part of opinion spam detection.

4.1 Content-based features

In the reviews' content, the words used by spammers represent their sentiments, thoughts, and feelings. When someone writes a deceptive review in the language used, some indications of linguistic deception can be seen. Content-based approaches are widely used for detecting deceptive opinion spam where supervised learning classifiers such as decision tree (DT), random forest (RF), logistic regression (LR), and support vector machine (SVM) require labeled datasets for training.

Jindal and Liu [39] used content features to find duplication and near-duplication of reviews and got 63% accuracy. They built a logistic regression model using the duplicate reviews as labeled spam training samples and the rest of the reviews as labeled non-spam training samples. Ott et al. [78] used linguistic features with a combination of POS tagged features to build an SVM classifier. They have collected synthetic datasets in their experiments and achieved an accuracy of 89.6%. The lack of experimentation on the real dataset is an essential drawback of the proposed model [73]. Feng et al. [23] used shallow syntax and deep syntax features as a probabilistic context-free grammar (PCFG) rule-based encoding TF-IDF with uni-gram to improve the performance, but these patterns are not so valuable for a real dataset.

Some existing works have used lexicon to calculate sentiment score with orientation (negative and positive words). The lexicon can be used in two ways: corpus-based and dictionary-based. In the dictionary-based technique, the opinion word (e.g., good, bad) search from WordNet or SentiWordNet dictionary for synonyms and antonyms [43, 99, 134]. In the corpus-based technique, large corpora are required for training data to find syntactic patterns of opinion words. This technique is based on domain-specific orientation, while the dictionary-based technique is not based on domain-specific orientation. Zhang [138] have worked on aspect-based sentiment analysis in Chinese reviews data using the corpus-based technique.

Li et al. [53] have built generative Bayesian-based sparse-additive-generative (SAGE) models using LIWC and POS features to detect review spam in three domains, i.e., hotel, restaurant, and doctors. They found in their experiments that LIWC and POS features are more robust in a cross-domain setting. There is still a need to add more vigorous features to the cross-domain setting. Shojaee et al. [99] worked on SVM with minimal sequential optimization (SMO) for detecting deceptive opinion spam by using stylometric features (syntactic and lexical features). They used a manually tagged dataset rather than a real-word dataset in their experiment. Unfortunately, they are unable to compare syntactic features and n-gram features in terms of performance. With the small numbers of labeled data and many unlabeled data, authors [26, 88, 90, 136] built a semi-supervised learning model such as co-training, self-training, and PU-learning, using n-gram features.

Dong et al. [19] have proposed an unsupervised-topic-sentiment-joint (UTSJ) probabilistic model based on latent Dirichlet allocation (LDA) that uses the combination of topic and sentiment features. In this model, perplexity is low, achieving better performance than LDA and JST models.

Li et al. [54] used POS and person pronoun features combined with sentence weighted neural network (SWNN) to improve deceptive opinion spam detection performance in the mixed domain. They utilized a combination of POS tag and person pronoun features with SWNN to obtain more robust results. Barushka and Hajek [9] have proposed a content-based approach using a deep feed-forward neural network (DNN) model. They used the skip-gram model with the new version of the soft-max function to generate word embedding and trained a DNN model to identify spam reviews. Liu et al. [61] used a bidirectional long-short-term memory (BiLSTM) based on a recurrent neural network for learning document representation. They achieved better performance than other neural network models in the cross-domain and mix domain. Cao et al. [12] introduce a model by combining fine-grained and coarse-grained features to detect deceptive opinion spam. The model worked on two parts together. In the first part, reviews have been transformed into the word vectors and used for training the deep neural network model to get the fine-grained features. The second part concentrates on learning coarse-grained features, which combines the LDA model and a 2-layered back propagation neural network.

Analysis: Most research has commonly used supervised learning in the content-based approaches. This approach shows that the SVM learning technique outperforms the NB, LR, KNN, and decision tree, while NB performs best for small datasets. An F-measure score could be high by applying SVM with minimal sequential optimization (SMO) over combined lexical and syntactic features while considering the review content's features. The probabilistic language model and the KL divergence are assumed adequate to identify deceptive reviews. Some researchers have used linguistic and psychological features by applying the LIWC tool to understand review text better. LIWC and POS features were found to be more robust in cross-domain settings when using SAGE Model.

To eliminate the limitations of BoW, the author proposed a word embedding representation model. This type of representation model, based on neural networks, can encode some semantic and syntactic properties of words. In some cases, the word embedding technique is directly used to train other non-neural models to obtain an optimal classifier. In most studies, lexicon-based techniques are used for aspect-based sentiment analysis, improving opinion spam detection performance. Experimental results show that LDA-based topic modeling is better to perform than SVM. The primary issue in the content-based approach is less availability of labeled datasets. In this domain, real-world labeled datasets are hard to obtain, and most of the datasets available are synthetically generated. However, it is unreliable to construct models based on synthetically generated datasets. Therefore, the researchers have used semi-supervised learning on a limited labeled dataset instead of supervised learning. This learning technique might ultimately alleviate the problem of the labor-intensive issue of identifying and labeling data. Recently, deep learning has been used to learn global semantic representations for text documents, yielding highly competitive performance. Convolutional neural networks (CNNs) can extract prominent n-gram features from an input text document to represent informative semantic for a document. But CNN is unable to preserve sequential information and long-term dependency of the text content.

Therefore, the author has moved toward recurrent neural networks (RNNs) for sequence modeling tasks. However, in practical terms, RNN suffers from the vanishing gradient problem, making the model reduce the learning ability. Thus, to solve this problem, the author used various RNN models such as gated recurrent units (GRUs), bi-directional GRU, long short-term memory (LSTM), bi-directional LSTM (BiLSTM), and attention-based LSTM. One of the most recently used neural networks is a generative adversarial network (GAN) that achieved better results and created samples pretty similar to actual data.

Table 2 summarizes the significant contribution in detecting deceptive reviews using content-based features. The table encompasses many of the performance parameters along with the results and limitations.

Table 2 Summary of detecting deceptive opinion spam using content-based features

Author (year)	Datasets used	Features used	Techniques used	Results	Strength/Limitations
Supervised Learning					
[43]	Amazon.com (707,664 reviews from Five Product categories)	Syntactical, lexical, and stylistic	Kullback-Leibler (KL) divergence (truthful reviews) and SVM (non-review)	1-AUC= 0.1416% (Untruthful reviews) 1-AUC= 1.8059% (Non-reviews)	In order to identify fake reviews, KL divergence and probabilistic language modeling are effective than the LR model.
[78]	Tripadviser and AMT: Gold-standard dataset	POS, LIWC, n-gram	Supervised (SVM, NB)	Accuracy (LIWC + BIGRAMS) = 89.8% (SVM)	Fake reviews by AMT are not a real fake reviews. Yelp dataset gives only 67.8 % accuracy.
[25]	Tripadviser and AMT: Gold-standard dataset by (from Ott, 2011)	Alignment compatibility features, n-gram, syntax features	Supervised (SVM)	Accuracy =90.1%, Accuracy=91.3% using deep syntax features	By integrating the profile alignment compatibility features with the n-gram and deep-syntax features achieved better performance.
[79]	400 Unfavorable (1-star or 2-star) genuine reviews from Priceline, Expedia, Orbitz, Hotels.com, TripAdvisor, and Yelp. 400 Unfavorable deceptive reviews are received from AMT	n-gram features	Supervised SVM	Accuracy =86%	The data (Synthetic dataset) being used is not real-world data.
[99]	800 Truthful reviews from Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor and Yelp and 800 fake reviews from AMT	Stylometric features or Lexical and syntactic features	Support vector machine (SVM) with sequential minimal optimization (SMO), Naive Bayes (NB)	F measure = 84%	Needs optimal set of features to achieve high accuracy.
[52]	Dataset from Ott, 2011	BOW (unigram, bigram, topic)	Supervised SVM, Bayesian approach: Latent Dirichlet Allocation (LDA) topic models (TopicSpam, TopicTD, TopicTDB)	Accuracy (TopicSpam)= 94.8% Accuracy (TopicTD)= 88.8% Accuracy (TopicTDB)= 93.1% Accuracy (SVM-Unigram)= 88.4% Accuracy (SVM-Bigram)= 89.6% Accuracy (SVM-Unigram with removing hotel-specific topics)= 89.5% Accuracy (SVM-Unigram with removing hotel-specific topicsand all back ground)= 82.2%	TopicSpam model effective than SVM.
[53]	Domain expert deceptive opinion spam (Employee, AMT, dataset from Ott, 2011), truthful Customer reviews (Customer) , cross-domain (i.e., Hotel, Restaurant and Doctor) gold-standard dataset	n-gram, LIWC and POS	Bayesian generative approach: Extended SAGE Model, One-Versus-Rest (OvR)	Accuracy = 65% (three-class classifier) Accuracy = 76% (two-class classifier) Accuracy (LIWC) = 64.7% (doctor in cross domain)	LIWC and POS features found to be more robust in cross domain setting when using SAGE Model.

Table 2 (continued)

Author (year)	Datasets used	Features used	Techniques used	Results	Strength/Limitations
[42]	Deceptive reviews from AMT (400), Employee (140), Not recommended Yelp (3361) and truthful reviews from Tripadvisor (400), recommended Yelp(3869)	Frame and Bi-frame Features, semantic frame features, Normalized Frame Rate (NFR), Bi-frame Rate (NBFR)	Supervised (SVM, NB)	Accuracy (AMT dataset)=0.914 (combination of Frm5, Bi-Frm20 and full Bi+ with a Naive Bayes model), Accuracy (Employee dataset)= 0.924 (Frm5, Bi-Frm15 and Uni full features using an SVM model)	Frame features outperformed the baseline model by 4.34% for the AMT dataset.
[89]	Data from TripAdvisor (After preprocessing and annotating 712 spam reviews in all 3000 reviews)	Lexicon (unigram, bigram), Shallow syntax (POS,stylometry), Deep syntax (Probabilistic context free grammar (PCFG)), Psycholinguistic features (LIWC)	Uses the minor-set (one-subset) to identify mislabeled samples from the major-set , Supervised Learning (SVM, LR, NB)	F measure = 66.4%	Perform noise identification in multiple round
[2]	Dataset (from Ott, 2011) and Yelp dataset	n-gram, punctuations	Active and Supervised learning: Ensemble method	Precision = 95%	Naïve Bayes outperform all other classifiers. The hybrid data set developed by the Active Learning System fits very well in the supervised training model.
[102]	Amazon gold standard: Obtained 100 products where each product includes 8 fabricated reviews and 12 real reviews	Bigrams, tri-grams	Bagging Model: Product word composition classifier (PWCC-CNN), tri-grams(SVM) classifier, and BIGRAMS(SVM) classifier	F measure = 77.2 % (Bagging) F measure = 74.9% (PWCC) F measure = 71.4 % (bi-gram SVM) F measure = 72.2% (tri-grams SVM)	Bagged with PWCC to produce more robust results.
[6]	Dataset (from Ott, 2011, 2013)	Word and Character N-grams	Feature Attributes Reduction using PCA, Supervised (REG,SVM, SVMlib, PSVM, NBC, and aNBC)	F measure (for Positive opinion)=89.35% (uni-grams + bi-grams + tri-grams + 4 characters feature), F measure (for Negative opinion)=89.35% (unigrams + bi-grams + tri-grams + 1 characters feature)	A combination of word and character n-grams achieved a good result.
[95]	Amazon (39,382 online product reviews)	Product Rating, Comment Sentiment, Review Sentiment, Number Comment, Avg Cosine Similarity Helpful Votes, and Rating Deviation	Supervised Learning (RF, GB, SVM) combination with ADASYN and SMOTE (both are over-sampling techniques for balance data)	F measure =0.91 (best result for RF with ADASYN oversampling technique)	Highly depends on comments of the review. Only those reviews are considered which have at least 5 comments, other reviews are not considered. Author have experimented on very few data.

Table 2 (continued)

Author (year)	Datasets used	Features used	Techniques used	Results	Strength/Limitations
[75]	Dataset (from Ott, 2011, 2013)	LIWC, POS tags, n-gram, Sentiment score	Supervised (DT, NB, SVM, k-NN, RF, and LR)	Accuracy (using logistic regression) = 86.25% (Unigram, LIWC, sentiment score)	Sentiment score gives good accuracy and precision.
[91]	Yelp dataset (64445 reviews from 99 restaurants) from mukherjee, 2013b	Semantic based feature (latent-topic), Relational based features (binary features of frequent itemsets, maximum content similarity, rating and rating deviation, length of review)	Supervised (SVM, Xgboost), Latent topic distribution using LDA	Accuracy (by SVM)= 61.2% (Semantic), Accuracy (by XGboost)= 62.9% (Semantic), Accuracy (by XGboost)= 75.6% (Semantic+ Relational), Accuracy (by SVM)= 74.2% (Semantic+ Relational)	Semantic-based features give more unsatisfactory results than relational based features because the Yelp dataset contains some noisy data.
[37]	Yelp Filter Dataset (64195 reviews across 85 hotels and 130 restaurants)	Features from dataset (Date, reviewID, reviewerID, reviewContent, rating, usefulCount, coolCount, funnyCount, flagged, restaurantID), Tf-Idf, word2vec, Latent topic distribution (Skip-gram OR CBOW)	Supervised (SVM, LR, MLP), Latent Topic Distribution: LDA	Accuracy= 81.3% (LDA+ Logistic Regression)	LDA with linguistic features given better performance than [74]'s accuracy 68.1%. Means LDA is effective for features extraction.
[80]	Dataset (from Ott, 2011, 2013; Sun, 2013; Mukherjee, 2013b), twitter	Linguistic Inquiry and Word Count	Spiral cuckoo search clustering method	Accuracy (of different datasets Spam review, Synthetic spam review, Yelp hotel review, Yelp restaurant review respectively) = 64.82%, 71.63%, 70.92%, 71.42%	Higher accuracy with spiral cuckoo search other than PSO, DE, GA, CS, ICS feature selection method.
[93]	Dataset (from Ott, 2011, 2013) and Yelp dataset (2600 truthful and 2600 deceptive/filtered reviews)	Unigram, bigram, POS, Quantitative Feature, Psychological and Linguistic Features, Readability Features	Supervised (k-NN, NB and SVM) , Multi-view Ensemble Learning (MEL), Rough Set Based Optimal Feature Set Partitioning (RS-OFSP) algorithm	Accuracy high (RS-OFSP) than (RFSP) and (OFSP).	RS-OFSP algorithm has better performance than basic RFSP and OFSP.
[84]	Dataset (from Ott, 2011, 2013)	TFIDF	Hybrid approach of iBPSO and cuckoo search algorithm (for feature selection), Supervised (NB, KNN)	Accuracy = 96.97%	Optimised feature set improves the performance. But the author used a synthetic dataset, which is not a real-world dataset. It might degrade the performance if using real data.

Table 2 (continued)

Author (year)	Datasets used	Features used	Techniques used	Results	Strengh/Limitations
Semi-supervised Learning					
[48]	dianping.com (Chinese reviews)	unigram, bigram, Average number of reviews per day (ANR), Maximum content similarity (MCS)	Learning from positive and unlabeled(LPU): Spy Algorithm, EM or SVM	F measure(Spy+SVM)= 67%	PU learning improves the performance over SVM.
[88]	Dataset (from Ott, 2011)	Similarity weights of spy examples	Semi-supervised mixing population and individual property PU learning (MPIPUL) based on LDA and SVM	Accuracy = 83.91% (20% positive example) Accuracy = 86.69% (40% positive example)	Mixing population and individual property and LDA (capture the deeper informationand) to build a more accurate classifier.
[26]	Dataset (from Ott, 2011, 2013)	unigrams and bigrams	Semisupervised: PU Learning	F measure=79.6% (positive review), F measure=69.9% (negative review)	No large labeled data required.
[136]	Dataset (from Ott, 2011 and Li, 2014b)	BOW, POS	1. Semi-supervised (Co-training for Spam review identification CoSpa-C and CoSpa-U) 2. Rule-based (Probabilistic Context-Free Grammars: deep syntax)	PCFG with CoSpa-U achieved highest performance.	CoSpa-C and CoSpa-U (in lexical terms) techniques outperform the traditional SVM. The performance of these techniques in actual datasets has been degraded.
[90]	Dataset (from Ott, 2011, 2013)	Bigram, POS, linguistic, word count and sentimental features	Semi-supervised: Co-training algorithm, expectation-maximization algorithm, label propagation and spreading, positive unlabeled learning for KNN, LR, RF, SGD	F measure=78% (Co-training algorithm), F measure=83% (expectation-maximization algorithm), F measure= 83% (label propagation and spreading), F measure=84% (PU)	Meta-data features are not used. Textual content with multimedia content can improve the performance.
Other techniques					
[4]	Amazon.com (5 camera reviews)	Concepts word as features	Conceptual level similarity measures: vector space model	Accuracy (Pros)= 57.29 % Accuracy (Cons) = 30.00 %	Accuracy is very low.
[45]	Amazon.com	Syntactical, lexical, and stylistic, semantic relationships captured from WordNet	Unsupervised (Semantic language model) /Supervised (SVM, KNN)	1-AUC= 0.1346% (Untruthful reviews) 1-AUC= 2.5112% (Non-reviews: SVM)	The language model approach is not as effective as the Semantic language model. Legitimate reviews was misclassified as spam. Unable to detect Singlton reviews.

Table 2 (continued)

Author (year)	Datasets used	Features used	Techniques used	Results	Strentgh/Limitations
[54]	Dataset from Li, 2014b : Reviews for Hotel, Restaurant, Doctor	Unigram, bigram, pos, swnn	Neural network based models : sentence weighted neural network (SWNN) model	Accuracy= 80.1%	SWNN is more efficient than other neural network based models. Fixed-weight model, and hard-alignment is the drawback of this model. Liberal alignment can be done by memory network based model.
[19]	Yelp dataset (Hotel: 780 untruthful reviews and 5078 truthful reviews, Restaurant: 8308 untruthful reviews and 58716 truthful review)	Topic and sentiment features	Unsupervised-topic-sentiment-joint probabilistic model (UTSJ) based on LDA	F measure (in balanced dataset)=83.92%, F measure (in unbalanced dataset)=77.82%	Perplexity is low that show better performance than LDA and JST models. Performance depends on number of topic selected.
[9]	Dataset (from Ott, 2011, 2013)	n-gram, skip-gram	Deep feed-forward neural network (DNN)	Accuracy = 84.38% (NB) Accuracy = 84.50% (SVM) Accuracy =89.75 % (DNN) Accuracy = 84% (RF) Accuracy = 85.75% (CNN)	DNN method performs well on both features.
[12]	Datasets from Li, 2014b and Yelp restaurant reviews (Mukherjee, 2013)	Coarse-grained and fine-grained features	2-layered BP neural network	Accuracy = 85.9% (LDA-BP+TextCNN)	LDA-BP+TextCNN model better performs than three derived models LDA+BPLSTM, LDA+BPBi-LSTM, and LDA+BPTextCNN.
[30]	Four benchmark datasets, namely the hotel, restaurant, doctor and Amazon	BoW (n-gram), Skip-Gram word embeddings, lexicon-based emotion indicators	Deep feed-forward neural network (DFFNN), CNN	Accuracy (hotel by DFFNN) = 89.56%, Accuracy (restaurant by CNN) = 89.80%, Accuracy (amazon by DFFNN) = 82.80%	Limited features used in reviewer-based and product based. Domain Specific causes ignore sentance weight.

4.2 Behavior-based features

In a behavioral-based approach, authors have extracted unusual review patterns of the spammers. Research observations suggest that spammers show behavior abnormalities that differ from that of ordinary consumers, such as they admire or condemn the product of a particular brand; in a short period of time, they post the highest possible number of reviews; they always give extreme high or low rating; they frequently post duplicate reviews; and so on.

Most of the research used behavioral indicators to find individual spammers [3, 22, 57, 72, 108] and group spammers [71, 85, 115–118]. Based on the algorithms used in detecting spammer and spammer groups, existing related works can be categorized into frequent item-set mining (FIM) [71, 127] and graph-based approaches [3, 85, 108, 115–118].

Lim et al. [57] concentrated on extracting reviewer-centric features based on the reviewer's profile and behavior instead of review characteristics. They worked on review patterns and star ratings to find various types of spamming behaviors (such as duplication of reviews, i.e., a reviewer writes multiple opinions on a target product). They framed four models, i.e., target products (TP), target groups (TG), early rating deviation (ERD), and general rating deviation (GRD), for calculating spamicity scores. Jindal et al. [40] stated that unexpected rules and unexpected groups indicate abnormal behaviors of spammers. These unusual behaviors are a prime indication of spam activities. Mukherjee et al. [71] proposed a new model GSRank for ranking group spam using a relation-based iterative model. They used frequent item-set mining (FIM) to generate candidate reviewer groups by computing group deviation (GD), time window (TW), member content similarity (MCS), group content similarity (GCS), the ratio of group size (RGS), group size (GS), early time frame (ETF), and support count (SC). Mukherjee et al. [72] have proposed an unsupervised author spamicity model (ASM) on behavioral footprints to detect spammers and achieved an accuracy of 74.6%. Their model is expensive and painstaking for broad-scale machine learning and analysis. Moreover, they ignored the usable review text generality information and did not consider the item's abnormal features. Fei et al. [22] employed a loopy belief propagation (LBP) based on a message-passing algorithm to solve approximate inference problems. They proposed a kernel density estimation (KDE) based algorithm for detecting review-burst and used several indicators in review burst to find review spammers. Ji et al. [36] proposed a group spam detection using a review burst (GSDB) model to discover candidate spammer groups. They used individual and group behavior features to build the GSDB model.

Analysis: Fully content-based classifiers have some disadvantages. For instance, spammers can rewrite the review to avoid detection of deceptive opinion spam. Secondly, content-based models have domain dependence and can not be applied to various domains. Moreover, content-based models require labeled datasets, which is hard to prepare manually. Therefore, the researchers have focused on a behavioral-based approach. Existing research revealed that the various individual and group behavior indicators are useful in discriminating between spam and genuine reviews. The article shows that the GSRank model achieves high success than supervised learning techniques like SVM and regression. Many samples are required to observe behavioral features, and hence the detection process becomes more costly. It is not easy to extract compelling behavioral features for new users if they have only posted one review. Thus, the behavioral-based approaches are suffering from the cold start problem. Therefore, to solve this issue, authors need a lot of reviews for singleton users. It may be done by jointly utilizing reviewers' behavioral and textual features.

Table 3 shows the summary of performance results and limitations of behavioral-based features.

4.3 Hybrid features

Hybrid features (a combination of textual and behavioral features) came into studies to increase deceptive opinion spam detection efficiency. Noekhah et al. [77] discussed a graph-based unsupervised approach using three types of entities and their relations (behaviors). They exploited an iterative algorithm to calculate the spamicity (degree of spam) by updating the feature's weight and obtained 93% accuracy on the Amazon dataset. Wu et al. [123] introduced a hybrid PU-learning-based spammer detection model (hPSD). Three essential parts of hPSD, including dependable negative set extraction, feature vectorization, and hybrid learning schemes, are emphasized. Rout et al. [90] have given a semi-supervised spam detection method by integrating various textual features (i.e., parts of speech, linguistic, sentiment, and word count) and behavioral features.

Additionally, [7] have given a hybrid model using the combination of spam-related, spammer-related, and product-related features with a revised weighting scheme. They found that the model achieved an improvement in opinion spam detection performance. In the hybrid approach, four main contributions are required.

- Revised Features Selection: Identifying the hybrid set of features that are strongly related to spamicity.
- Revised Feature Weighting: Calculate each entity's spamicity score (sample) using the revised feature's

Table 3 Summary of detecting deceptive opinion spam using behavioral-based features

Author (year)	Datasets used	Features used	Techniques used	Results	Strentgh/Limitations
Supervised Learning					
[24]	crawled from TripAdvisor.com (Hotel), Amazon.com	Reviewer's rating distribution	Supervised (LIBSVM)	Accuracy=74.4 (rule)	High accuracy than the human judged dataset.
[126]	Rating data of restaurants New York City NYC(12733 number of rating, 79 restaurants), Chicago CHI (11175 number of rating, 76 restaurants), Phoenix PHX (8381 number of rating, 77 restaurants) and restaurant ranking data from Zagat.com as ground truth	Ranking and Rating features	Bi-level framework (Ranking based model)	Significant result	Macroscopic methods can save the expensive human effort.
Unsupervised Learning					
[57]	Datset from Amazon.com	Various rating behavioural feature	Unsupervised/Supervised spammer detection: combined behavior score formulation	513 (4.65%) reviewers having spam scores>= upper-whisker of the distribution (0.23)	Cohen's kappa b/w 0.48 to 0.64 (moderate). Unable to identify singleton spam review.
[72]	Datset from Amazon.com	Author Features (CS, BST, MNR, RFR) and Review Features (DUP, EXT, Rating Dev.,ETF, Rating Abuse)	Unsupervised: Author Spamicity Model (ASM)	Accuracy (on top 5%)= 74.6% (Rankboost)	Better accuracy score than judge evaluations. Expensive and painstaking for large scale machine learning and analysis. Moreover, they did not consider the item abnormal features and dismissed the useful generality of text analysis knowledge.
[22]	Datset from Amazon.com	Behavior Features (RAVP, RD, BRR, RCS, Reviewer Burstiness)	Unsupervised (K-mean), Supervised (SVM), Kernel Density Estimation (KDE) for Burst Detection, Markov Random Field (MRF), Loopy belief propagation (LBP)	Accuracy = 71.2% (LBP with prior and local observation) Kappa (spam) = 0.71(with local) Kappa (Non-spam) = 0.84(with local)	Ignored textual feature which could improve the accuracy of spammer detection. Singleton reviews are not considered.
[36]	Amazon Books review data from 1993 to 2014	individual and group behaviour features	Candidate group generation by Kernel Density Estimation, Unsupervised	GSDB method better performance than GSBC	Weighting scheme are average, can be improve by automatic learner.

weight (importance of feature), then rank samples based on these scores.

- Iterative Algorithm for Spamicity: Updating the spamicity score of an entity by iteration process until convergence on spamicity score of other related entities.
- Integrated Approach: A unified approach (the combination of effective weighted features) based on a normalized scoring technique is needed.

Analysis: As it may be challenging to recognize spammers by focusing only on textual features or behavioral features, it could be more effective to use a hybrid set of features to detect fraudulent reviews or reviewers effectively. The authors suggested a kernel density estimate (KDE) method identify review spam, taking into account the bursty reviews, the reviewer's behavior patterns, and the review features. However, heuristic rules are widely adopted by existing unsupervised methods; therefore, the exploratory space these methods are very massive in the detection model. Many current approaches for opinion spam detection focus on traditional machine learning models. In this model, the authors have performed various features extraction manually. In deceptive opinion spam detection models, feature engineering techniques perform a huge influence.

The studies revealed a number of the techniques, such as Gini index, information gain, X^2-statistic, in the text classification; these are helpful for selecting features. Besides, most of the common features like behavior patterns, IP address, MAC address, Geo-location, and profile information can be shared by the spammer. The recent papers focused on the relation-based user-product network structure to better understand opinion spamming. Most of the studies used generative adversarial networks (GAN) to generate synthetic features for improving detection opinion spam performance. Semi-supervised GAN can solve the problem of the labeled dataset's scarcity. The summary of the hybrid approach, along with strengths and limitations, is shown in Table 4.

5 Deceptive opinion spam detection approaches

Many approaches for improving spam identification have been proposed by existing research as shown in Fig. 5. In this literature, deceptive opinion spam detection approaches are classified into several categories such as Machine learning-based, Rule-based, Graph-based, Pattern mining-based, and Neural network-based. These approaches are discussed below:

5.1 Machine learning-based approach

Detecting deceptive opinion spam is a binary classifying problem divided into two classes spam or not-spam. In the beginning, researchers applied a Machine learning-based approach to handling such classification problems. These approaches are widely used in detecting deceptive opinion spam and can be divided into Supervised machine learning, Semi-supervised learning, and Unsupervised learning.

5.1.1 Supervised learning

The model learns from the labeled training data in supervised machine learning and predicts the correct label for newly presented input data. Jindal and Liu [39] incorporated supervised learning (i.e., LR, SVM, and NB) for deceptive opinion spam detection and got 63% accuracy. They have trained a model using similar reviews as positive training samples (spam) and remaining reviews as negative training samples (non-spam).

- **Support vector machine (SVM):** The SVM algorithm analyses data and determines decision boundaries by providing hyper-planes. The hyper-plane divides the data points into two classes. The data points near to hyper-plane are support vectors used to maximize the classifier's margin [100]. SVM has various kernel function that converts the non-separable problem into the separable problem. The SVM technique is widely applied for the classification of textual data as well as image data [67]. The authors [78, 79, 89, 99] proposed an SVM model using various content features (n-gram, shallow syntax, deep syntax, LIWC) and behavioral features (review rating, extreme rating, number of reviews per day). The performance of spam detection has been improved by using probabilistic context-free grammar derived from deep syntax analysis.
- **Logistic regression (LR):** A LR classifier is a statistical method that forecasts an outcome from one or more response variables X for a binary variable Y. According to the hypothesis of logistic regression, the cost function tends to limit between 0 and 1.

  $$0 \leq h_\theta(x) \leq 1 \tag{24}$$

  where $h_\theta(x)$ is the cost function.

  Jindal and Liu [39] used a logistic regression model on the Amazon dataset to find deceptive opinion

Table 4 Summary of detecting deceptive opinion spam using hybrid features

Author (year)	Datasets used	Features used	Techniques used	Results	Strengh/Limitations
Supervised Learning					
[39]	Amazon dataset	Review, reviewer and product centric features	Supervised (LR)	AUC = 78%	Very hard to manually label training examples. Only duplicate reviews as positive training examples. This feature is not good for training because some repeated reviews of same product can be genuine.
[73]	Real-life data from Yelp and the AMT data (from Ott, 2011)	TF, and TF-IDF features, POS, unigram, bigram, and behavioral features	Supervised (linear kernel SVM)	Accuracy (Yelp)= 67.8% (bigram) Accuracy(Yelp)=83.8%(BF) Accuracy (Yelp, Hotel)= 85.1% (bigram+BF) Accuracy (Yelp, Restaurant)= 86.5% (bigram+BF)	Behavioral features alone perform better than n-gram features in yelp data set. In balanced data (50:50) precision is good, but in natural distribution (N.D.) precision is low.
[109]	Yelp's real-life data (66887 reviews, 35102 reviewers)	Unigrams, bigrams, topic probabilistic, Readability features (Automated Readability Index ARI, Coleman-Liau Index CLI), Behavioral features (Restaurant Number, Date Interval, Percentage of Positive Reviews, Review length)	Supervised (LR, KNN, NB, SVM)	Accuracy (by LR)= 97.2% (with all features)	Semantic features (readability+behavioral) achieved better performance than base line (unigram,bigram, tf-idf) features.
[134]	Hotel review from TripAdvisor	Sentiment lexicon, Strength words, negation woords, meta-data features	Aspect-rating local outlier factor model (AR-LOF), Supervised	Accuracy = 79.6%	AR-LOF uses fixed aspects, and domain dependent.
[8]	Yelp dataset	Review centric (POS, word2vec,Tf-Idf, etc.) and User centric features (Personal, Social, Reviewing Activity, and Trusting features	Supervised (Random Forest, Ada Boost classifiers) on Fake Feature Framework (F3)	F measure= 82%	Reviewer-centric features performed slightly better than review-centric features in the F3 Framework.

Table 4 (continued)

Author (year)	Datasets used	Features used	Techniques used	Results	Strentgh/Limitations
[41]	Yelp from Mukherjee, 2013b and dataset (from Ott, 2011, 2013)	Linguistic (TF-IDF; Number of words, characters, and special characters) and structural features (Time, Rating, Useful-count, Cool-count, Funny-count)	Supervised Ensemble Learning: Three classifiers Discriminative Multinomial Naive Bayes DMNB, J48, and LibSVM in tier 1, LR in tier 2. Used full features set and some feature selection techniques	F measure (Yelp) = 83.2% (on full set of features), F measure (Ott's dataset)=82% (on full set of features), F measure (Yelp) = 84.1% (using ChiSq), F measure (Ott's dataset)=81.7% (using ChiSq)	Ensemble approach better perform than individual classifier. Used imbalance dataset.
[94]	Dataset from Amazon (200 online reviews of LED TV)	Response, Useful Profile, Template, Star Rating, Reply, Thick	Supervised (Decision tree) classifier and Information Gain	efficiency =96% success rate	Identify most significant features.
Unsupervised Learning					
[121]	TripAdvisor dataset (26,903 opinions from 21,440 distinct customers of 741 Ireland hotels)	Reactive-Positive-Singletons (RPS), Concentration of Positive-Singletons (CPS), Proportion of Positive-Singletons (PPS), Review-Weighted-Rating (RWR), Truncated-Rating (TR), Contribution-Weighted-Rating (CWR), Positive-Review-Length-Difference (PRLD), Sentiment Shift (SS)	SingularValueDecomposition (SVD), Unsupervised-Hedge Algorithm (UH)	Suspiciousness scores = .27	SVD outperforms UH in aggregation. Only singleton reviews are considered.
[71]	Amazon dataset (53,469 customers, 109,518 opinions and 39,392 products)	Group spam features, Individual spam features, Linguistic features	Unsupervised (GSRank) /supervised(SVM, LR)	Spam = 38% (at threshhold=0.5) Spam = 29% (at threshhold=0.7) Spamicity= 86% (at threshhold = 0.5) Spamicity= 88% (at threshhold=0.7)	GSRank has a statistically significant advantage over other approaches. Not handle the problem of singleton review, because candidate groups need three or more than three reviews.

Table 4 (continued)

Author (year)	Datasets used	Features used	Techniques used	Results	Strengh/Limitations
[59]	Amazon dataset	Personal content similarity, Similarity on a product's reviews, Similarity on different product's reviews, Reviewer's posting frequency, Review frequency of a product, The repeatability measure	Unsupervised (spam score with a threshold value) /supervised (SVM, LR)	F measure=92.30% (SVM) F measure = 85.60% (LR) F measure = 85.1% (Unsupervised with 0.55 threshold value)	The investigation of similarities amongs all reviews are time-consuming process and, in most cases, a huge number of assessments are needed. Similarity-Review methods are not semantic-based.
[32]	Dianping (489989 Chinese reviews, 204954 reviwers, 265 restaurants)	Positive and negative sentiment words, rating score, Posting Frequency	Frequency based model FD, sentiment strength based model CSD (Computing Sentiment Strength using HowNet (Dictionary based))	Cohen's kappa betwween 0.6 to 0.8 substantial agreement, FD performs better than CSD	Experiment on large real dataset and FD+CSD performance is better than Lim et al.(2010)'s model.
[70]	AMT Dataset (from Ott, 2011), Amazon Dataset (Mukherjee, 2012) 2000 deceptive (filtered by Yelp) and 2000 truthful (unfiltered) opinions by 601 customers from Yelp.com across 50 Boston restaurants	Review features and reviewer features	Unsupervised: (1) Latent spam model LSM + Uninformed Priors (LSM-UP), (2) LSM + Hyperparameter Estimation (LSM-HE)	F measure (LSM-UP) =74.6%, 69.2%, 58.4% (AMT Dataset, Mukharjee, 2012's dataset, Yelp respectively) F measure (LSM-HE) = 75.9%, 74.3%, 61.6%(AMT Dataset, Mukharjee, 2012's dataset, Yelp respectively)	In terms of entropy, purity, and F-score, LSM-UP and LSM-HE outperform the basic clustering technique significantly.
[20]	Amazon review data set (7950 reviews, 815 unique users): Users who wrote less than 20 reviews or greater than 500 reviews, were filtered	review content features, User's behavioral features	Stochastic decision tree model: Autoencoder (Auto-encoder is a neural network) and random forest.	Accuracy=95.85%	Achieved good accuracy compare to other author's model Mukherjee et al., Zhang et al, Heydari et al., Xu and Zhang, Rout et al. But author selected users who wrote reviews between 20 to 500.
[17]	Yelp dataset	Geolocation features (Review location, Center point, Radius review location and the center point)	SpamTracer model, Hidden Markov Model.	Accuracy = 71%, and recall= 76%	Low accuracy compare to other supervised learning.
Semi-supervised Learning					

Table 4 (continued)

Author (year)	Datasets used	Features used	Techniques used	Results	Strength/Limitations
[47]	Obtained from Epinions	Content Features, Sentiment Features, Product Features, Meta-data Features, Reviewer related Features, Behavior Features	Supervised (Naive Bayes) / Semi-supervised (co-training)	F measure=58.3% (NB) F measure = 63.1% (Co-Training with Agreement)	Only positive and unlabeled data contributes to disappointing performances. Requires more information from the reviewer.
[123]	Movie and Amazon dataset: MovieLens and Netflix datasets	Entropy, DegSim, LengthVar, RDMA, FMTD, GFMV and TMF, popularity rank (PopRank), average distance with other users (DistAvg), and category entropy (CatEnt)	PU-learning-based Spammer Detection model (hPSD)	F measure=97.6%, 98%(MovieLens, Netflix respectively) kappa statistic = 54.8%	hPSD has the overwhelming performance advantages over other detectors.
[14]	AliExpress dataset (612 spam reviews of 2321 reviews)	Review features and reviewer features	Semi-supervised: co-training and tri-training	F measure= 0.65 (co-training) F measure= 0.70(tri-training)	Tri-training gives better performance than co-training.
[16]	Crawled from JD.com (after preprocessing 30568 comments)	Meta-data Feature: length of the review, Review Content Feature: BOW	Semisupervised learning (PU learning, K-means), Autoencoder to reduce dimension	Accuracy= 89.3% (700-Labelled,100-Test Data0	Eliminated small comments.
[124]	Dataset from Amazon China (9424 users, 19185 products, 469393 reviews)	User_ID, Product_ID (ASIN), review title, star rating, and posting date	Semi-supervised: hybrid PU-learning-based spammer detection (Features Discretization, Reliable Negative Set Extraction, Hybrid Semi-supervised learning)	F measure (MovieLens)=97.6%,	F measure (Netflix)=98% hPSD achieves the good score.

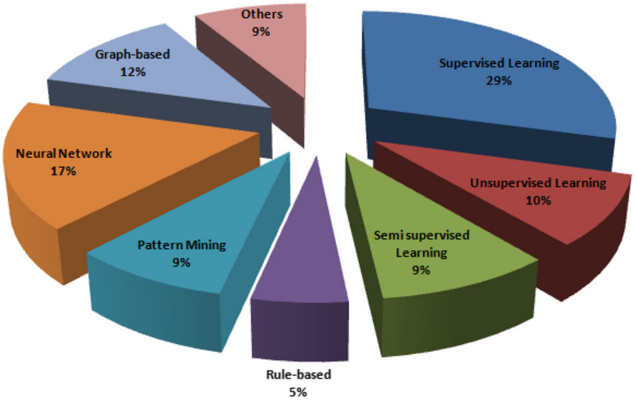


Fig. 5 Distribution of Approaches

spam with a maximum AUC of 78%. Wang et al. [109] achieved better performance by LR model using semantic features and behavioral features on Yelp's real-life dataset. Narayan et al. [75] used integrated unigram features, LIWC, and sentiment score on LR. They found better accuracy than other supervised models. Latent topic distribution with linguistic features gave an excellent performance by LR model on Yelp dataset [37].

- **Naive bayes (NB):** A NB classifier is a probabilistic classifier based on Bayes's theorem. In the Bayes theorem, the posterior probability is computed as $P(c|x)$, from $P(c)$, $P(x)$, and $P(x|c)$.

$$P(c|x) = \frac{P(x|c) \times P(c)}{P(x)} \tag{25}$$

where $P(x)$ is the prior probability, $P(x|c)$ denotes likelihood, and $P(c)$ is the class prior probability.

Li et al. [47] used an NB classifier and semi-supervised co-training method to detect deceptive opinion spam by integrating two groups of features, i.e., textual and behavioral features. Khurshid et al. [41] proposed an ensemble learning model in two tiers. They used multinomial NB, J48, and LibSVM in the first tier; and LR in the second tier.

- **Decision tree (DT):** A DT is a hierarchical structure where internal nodes denote a test on the features/attributes, leaf nodes denote the class labels, and edges denote interflows of the attributes leading to those class labels. At each step of the tree-building process, information gain is used to decide which attribute to split on. The first split starts with the maximum information gain as the root node of the decision tree. Inner nodes of DT are labeled with distinct attributes, and these attributes have less information gain than the root node. This process continues until all internal nodes have consistent data or information gain becomes zero. The core decision tree algorithm developed by Quinlan is ID3. Sanjay and Danti [94] used DT with information gain to find deceptive opinion 32 spam and achieved a 96% success rate on LED TV reviews in the Amazon dataset. Barbado et al. [8] used random forest and Ada-boost classifiers on the yelp dataset's fake feature framework and stated that reviewer-centric features give good performance than review-centric features.

5.1.2 Unsupervised learning

Unsupervised learning approaches find out the structure by observing the relationship among the data samples; this structure is known as a cluster. Clustering is the process of organizing sample data point into groups that have similar properties or patterns. Supervised learning models require labeled datasets for evaluations, which are hard to label manually. Thus, the researchers propounded neoteric unsupervised text mining by using semantic language models [43–45].

Several unsupervised graph-based algorithms [3, 45, 81, 85] have been proposed for detecting opinion spammers. Mukherjee et al. [72] introduced an unsupervised Bayesian framework, i.e., author spamicity model (ASM), by exploiting the behavioral footprints of authors. ASM is a rank-based model, where the top ranker is likely to be a spammer and the bottom ranker likely to be a non-spammer.

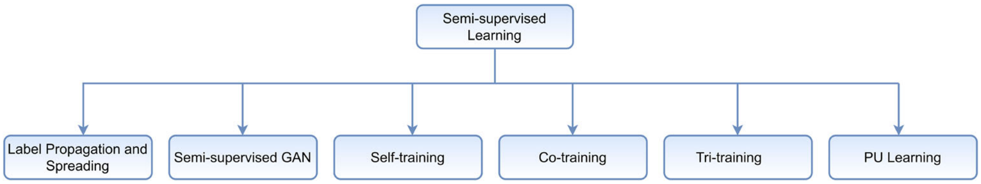
Guangyu et al. [121] collected a set of features and assigned score values to each feature to predict suspicious reviews. After that, they used two aggregation methods named unsupervised-hedge algorithm (UH) and singular-value-decomposition (SVD) to determine a single ranking score. Dong et al. [19] proposed an unsupervised-topic-sentiment-joint probabilistic model (UTSJ) based on latent Dirichlet allocation (LDA) model to extract the topic and sentiment words from the opinions for classifying deceptive opinion spam. This model was found to have less perplexity, due to which its performance was better than LDA and JST.

5.1.3 Semi-supervised learning

Semi-supervised learning uses labeled data (only a small volume) to classify a large amount of unlabeled data. It uses a combination of labeled and unlabeled data to predict the sample class. The current work based on the semi-supervised learning model is mainly divided into various learning methods, i.e., Label propagation and spreading, Semi-Supervised GAN, Self-training, Co-training method, Tri-training, and Positive unlabeled (PU) Learning (Fig. 6).

Label propagation and spreading Label propagation and the spreading algorithm is a graph-based algorithm that is used in a semi-supervised learning model. Each node has some tagged/labeled or untagged data in the graph,

Fig. 6 Semi-supervised learning classification



and the edges define the similarities among these tagged and untagged data. Both label propagation and spreading are employed in classification and regression. The labeled data can be propagated across the graph to the unlabeled data until convergence in label propagation. Label spreading minimizes a loss function with regularization properties, making it more noise resistant. Giasemidis et al. [28] used graph-based semi-supervised learning to classify message stance, namely label propagation and label spreading. Yang and Shafiq [132] have proposed a robust label propagation algorithm for large data whose language style frequently changes.

Semi-Supervised GAN In the semi-supervised GAN, the discriminator is concurrently trained in two modes, i.e., unsupervised and supervised mode. The most prominent generative algorithm in text categorization is multinomial naive Bayes associated with expectation-maximization (EM). In this model, three types of samples, i.e., labeled training samples, unlabeled training samples, and fake samples generated by the generator, are passed to the discriminator. The unlabeled data classification process is repeated until it converges into a stagnant classifier and a labeled set. Mukherjee et al. [70] have propounded an unsupervised generative model to detect deceptive opinion spam. They exploit linguistic and behavioral features in their experiment. Stanton and Irissappane [101] proposed a semi-supervised GAN model, called spamGAN, to detect deceptive opinion spam. They used an ADAM optimizer to train the generator and discriminator. Experimental results show that the model performs better than other models on the TripAdvisor dataset.

Self-training It is generally used for semi-supervised learning. In self-training, the limited volume of labeled data is first trained by a classifier, and then the classification model is used to classify unlabeled data. The classified data are called pseudo-label data added to the labeled dataset. The classifier is re-trained on the combined pseudo-labeled and labeled training data. After that, the trained classifier is used to predict the labeled test dataset. This process repeated until the convergence. Navastaraet al. [76] used self-training semisupervised learning on 600 manufactured product reviews (200 labeled and the rest 400 unlabeled) such as a printer, monitor, laptop, music player, CD. Ligthart et al. [56] have analyzed that the self-training model can obtain better results than the supervised models if they use a limited labeled dataset.

Co-training The co-training method is a bootstrapping method that is iteratively updated with unlabeled data. It trains two classifiers on two different features and enlarges the labeled data with high confidence by each classifier to the training set. This technique is mainly based on a probably approximately correct (PAC) learning model, first proposed by [11]. Li et al. [47] have built a model to train two classifiers using two groups of different features on limited labeled samples and predicted the class label of the unlabeled samples. After expanding the labeled samples, the final classifier was trained by combining two types of features used to predict the class labels of unlabeled samples in the test set. To build this model, they used Naive Bayes classifiers. Zhang et al. [136] proposed co-training approaches CoSpa-C and CoSpa-U to improve the performance using SVM as the base classifier. They used two types of representations: the lexical terms obtained from the textual features and the probabilistic context-free grammar rule obtained from deep syntax analysis.

Tri-training The tri-training is semi-supervised learning that uses three classifiers on a labeled dataset. In this mechanism, two classifiers are randomly selected to classify unlabeled instances, and if both classifiers agree on assigned labels, these instances are used to learn the third classifier. These steps are repeated until all unlabeled samples are labeled. Chengzhang and Kang [14] proposed a three-view semi-supervised model, tri-training, to utilize the massive amount of unlabeled samples. They experimented that tri-training has better performance than the co-training method.

PU learning PU learning is another type of semi-supervised learning which learns from a small positive example and large unlabeled dataset. It is an iterative process that recognizes reliable negative instances from the unlabeled instances. Various techniques exist, including variants of the expectation-maximization (EM) algorithm or SVM, to transform supervised classifiers into a PU learning setting. Li et al. [48] have used the PU learning technique to increase the effectiveness of the supervised classifier. Ren et al. [88] proposed a novel PU learning technique on the small set of deceptive/truthful review samples and a large set of unlabeled review samples, named mixing population and individual nature PU learning technique, to detect deceptive review spam. Narayan et al. [75] have used six classifiers (DT, NB, SVM, KNN, RF, and LR) in the PU-learning algorithm to detect deceptive opinion spam from their dataset and got 78.12% accuracy. [104] proposed a ramp one-class SVM (Ramp-OCSVM) model based on robust and non-convex semi-supervised learning. The author got in their experiment that Ramp-OCSVM did not affect the outliers because the ramp loss function has a non-convex property.

In addition, [90] explained how semi-supervised learning methods were used for identifying deceptive opinion spam. They proposed four types of semi-supervised learning techniques: co-training, EM, PU learning (for KNN, LR, RF, SGD), and 'label propagation and spreading' using bi-gram, POS, LIWC, and sentimental features. They found in their experiment that PU learning-based classification has a good F-measure than the other three methods. Ligthart et al. [56] have experimented on four different semi-supervised learning algorithms: self-training, co-training, transductive SVM, and label propagation and label spreading. They found that self-training with naive Bayes classifier achieved 93% accuracy on the AMT dataset and 73% accuracy on the Yelp dataset.

The summary of identifying deceptive reviews with performance results and limitations using traditional machine learning techniques on different features is shown in Tables 2, 3, and 4.

5.2 Rule-based approach

Some of the researchers used rule-based approaches for classifying spam or non-spam reviews. A rule might be based on product-centric features, review-centric features, or reviewer-centric features. Jindal et al. [40] proposed a domain-independent unexpected rule and rule groups using class association rules (CAR) to represent spam reviewers' unusual behaviors. Gao et al. [27] proposed a rule-based method to detect the corresponding cause events from the Chinese micro-blog dataset using the emotional lexicon and other linguistic features. They claimed 65% accuracy in their experiment results. Peng [81] integrated a time series with discriminating rules to efficiently detect the spam review and spam score. Saeed et al. [92] proposed an ensemble method by combining a rule-based classification model with machine learning techniques for detecting Arabic spam reviews. They constructed a set of rules for classifiers and achieved an almost 28% increase in inaccuracy.

Analysis: A rule can represent the reviewer's abnormal behavior pattern. The main issue is how to build a rule to find spamacity patterns. The existing research paper has built various rules, i.e., unexpected rule, sentiment pattern rule, POS rule, and various discriminating rules. The rule-based classification model depends on the collection of pre-defined rules. A rule is built from combining various n-gram features, word count, unique word percentages, and review rating deviation in content features. The experimental results show that the rule-based classification model performs better than the machine learning model. Some researchers used heuristic rules(i.e., if A then B) to detect deceptive opinion spam. Rule-based approaches have been beneficial for a massive dataset, reducing the burden of the content-based approach. The existing studies concluded that rules become powerful (such as flexibility, convenience, and robustness of pattern matching) using a regular expression. After some time, certain features can be out of date, which builds a rule, and thus, the weight of importance of features should be dynamically changed.

5.3 Graph-based approach

Imitating linguistic and behavioral patterns is often simple for spammers but rather difficult to imitate the graphical structure of legitimate reviewers. In a graph-based approach, we analyze the relation among reviewers' opinions, reviewers, and products or stores by the heterogeneous graph to detect spammers and spammer groups, as shown in Fig. 7. The reviewers' simple behavioral include fewer spamming data and can contribute to a high false-positive rate. Wang et al. [108] introduced a tri-partite graphical approach on a store-based review dataset (resellerrating.com). They used three features, i.e., review's honesty, reviewer's trustworthiness, and store's reliability, to make relationships among reviewers, reviews, and stores using

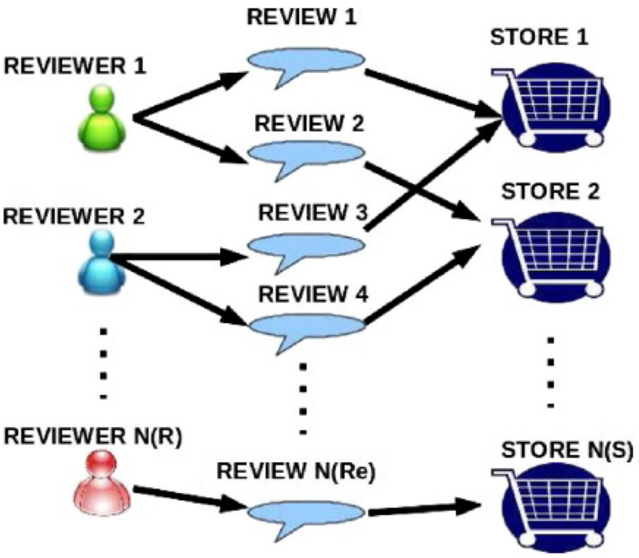


Fig. 7 Reviewer-Review Graph [108]

heterogeneous graphs. They cannot detect singleton review spam because they have insufficient information to find the reviewer's trustworthiness. Noekhah et al. [77] proposed an iterative (weight of each edge) algorithm that can improve spam detection accuracy. They have integrated the linguistic and behavioral features to detect spam entities. Li et al. [48] have used a tri-partite graphical structure for exploring the relationship among reviewers' opinions, reviewers, and IP/MAC addresses to find unusual behavior patterns of reviewers.

FraudEagle [3] and SpEagle [85] models, which rank individual review spammers, are based on reviewerproduct bipartite and reviewerreview-product tripartite graph, respectively. Edges of these graphs can be marked as positive or negative using Markov random field. FraudEagle does not use any prior knowledge of nodes and keeps linear time complexity. SpEagle comparatively performed better than FraudEagle, and it takes more effort to calculate prior knowledge. Wang et al. [119] proposed a pairwise MRF graph-based model, ColluEgale, in which nodes indicate reviewers and edges indicate the co-review relationships. They considered two types of prior knowledge, prior neighbor tightness (NT) and prior all, to improve the group spammer detection. Wang et al. [120] proposed an unsupervised graph embedding-based collective opinion spammer detection (COSD) model to find the co-review correlation among spammers. They applied jointly two types of relevance embeddings direct (strong pairwise collusive behavior) and indirect (positive-based random-walk). The experimental result shows that the COSD outperforms the baseline graph-based approaches like colluEagle and FraudScan. Spam detection algorithms of graph-based approaches are needed for effectiveness, efficiency, and scalability. Therefore, most of the studies have concentrated on graph-dependent methods for identifying deceptive opinion spam or spammers separately; it may be more effective to detect both spammer and spammer's opinions on a unified framework.

Analysis: The existing work on detecting opinion spam in this approach has been summarized by three components: identification of review spam, individual review spammers, and collusive review spammer's groups of graphical frameworks. The earliest studies have very little information about spamming behavior in the graph-based model, leading to a high false-positive rate. Therefore, The researchers have proposed various graph-based approaches(i.e., FraudEagle and SpEagle) to find relationships among reviewers' opinions, reviewers, and products. FraudEagle is unable to capture additional prior knowledge in the graphical network. Therefore, the author enhances the performance of FraudEagle as a SpEagle using some additional features like review text, timestamp, and rating. Both models, FraudEagle and SpEagle, work on individual spammers and cannot capture group spamming. The MRF-based co-review pairs model ColluEagle has been proposed to detect individual review spammers and spammer groups. Experimental results have shown that ColluEagle can significantly increase the detecting precision. The literature concludes that spammers use multiple identities to post their reviews, and it is often hard to recognize them by simple spam detection techniques. The authors have proposed two techniques, semantic similarity, and topic modeling similarity, to recognize a spammer's reviews written under different user's Ids. Existing classification systems would not work for a long time because of the ever-changing behavior of spammers. Thus, researchers need to propose efficient spamming filters to monitor opinion spammers' dynamic behavior to deter potential fraud.

The summary of the graph-based approach, along with strengths and limitations, is shown in Table 5.

5.4 Pattern mining-based approach

Meta-data features like review ID, helpful vote or feedback, posting date, store ID, IP address, and star rating help find reviewers' unusual patterns or abnormal behavior. When a reviewer is identified as someone who writes deceptive reviews, nearly all reviews related to that reviewer can be classified as spam reviews. Xie et al. [125] discovered unusual temporal patterns on singleton reviews because a significant portion (larger than 90%) of the reviewers write only one review (singleton review) and can influence the rating trend of a product. They find the time windows using temporal curve fitting and Longest Common Sub-sequence algorithms in each dimension of the time series, where spam attacks have happened. An emphatic increase in the number of (singleton) reviews with rating increases or decreases strikingly in this time window. Li et al. [49] studied that spammers are mainly active on weekdays except Monday compared to genuine users. Genuine users generally share their real experiences as an authentic review on Sunday and Monday. They analyzed the pattern of timestamp (posting patterns over the weekdays and weekends) and spatial (registration pattern, average travel speed, and traveling sequence) behavior of reviewers on the Dianping dataset, which contains additional information such as IP address and location co-ordinate.

Analysis: The majority of reviewers write only one review. Graph-based approaches work better in a predicament where spammers write several reviews. Patterns mining helps to detect singleton review spam. The authors proposed a multi-scale multidimensional time series model to find the relation between singleton review arrival patterns and the average store's star rating. In addition, reviewers' various spatial features (i.e., IPs, MAC, and city locations) differentiate spammers from non-spammers in the hotel domain.

Table 5 Summary of graph-based approaches

Author (year)	Datasets used	Features used	Techniques used	Results	Strengh/Limitations
[108]	Store's reviews from www.resellerratings.com (343603 users who posted 408470 reviews on 14561 stores)	Meta-data features such as reviewer id, review id, review rating with posing time, and store id	Iterative Computation Framework is used to calculate the trustworthiness of reviewers, the honesty of reviews, and the reliability of stores (Unsupervised)	Pprecision = 49%, kappa=60.3% (among 3 evaluators)	No need of labeled dataset. Human evaluation is necessary. Precision is low. Unable to handle singleton reviews because these have insufficient information to find trustiness of reviewers.
[3]	SoftWare Marketplace (SWM) dataset (1, 132, 373 reviews of 15, 094 software product-app like games, movies, news, sports, etc. by 966, 842 unique users)	Userid, product-app id, rating	Unsupervised graph-based framework, FRAUDEAGLE (Honesty-and-Goodness scores for the nodes by iterative process: Loopy Belief Propagation)	11%(107K) user's spamicity score >= 90th quantile distribution (0.89), and 156K user's spamicity score >= 0.5 (threshold)	The time complexity is linear. Some parameter are required to calculate honesty of reviewers and difficult to estimate a priori.
[63]	Amazon dataset (gold-std labeled): 1,078 users, 6489 reviews, 851 product after preprocessing	Review related feature (second person pronouns, sentiment, product average rating, product popular rating, absolute rating difference, helpful feedback rate and number, the first product review, similarity score), Reviewer related features (minimum time interval, reviewer rating difference, etc.), Review group features (group rating feature)	Supervised learning graph-based framework: review factor graph (RFG) model, SVM, LR	F measure=39.61% Accuracy=88.20% (RFG on review dataset) F measure=51.33% Accuracy=90.13% (RFG on reviewer dataset)	RFG achieved better accuracy than SVM and LR. Some features (like high rating of reviews, positive / negative trend) they used can not be good indicators of spam.
[77]	Data was prepared by crawling in Amazon website	Behavioral features and content features of review, reviewers, group of reviewers, and targeted products	Graph-based method, Iterative algorithm: Update each entity's spamicity degree based on the spamicity degree of other entities from the previous iteration.	Accuracy=93%	This model discarded singleton reviews.

Table 5 (continued)

Author (year)	Datasets used	Features used	Techniques used	Results	Strentgh/Limitations
[114]	Amazon Book review dataset (53,417 reviewers, 1,123,034 reviews, and 412,799 products)	Review, Reviewer, and Product related features	Graph-based method, Iterative Computation with Elimination ICE: reviewer's trustiness claculation by updating review's honesty and product's reliability.	Avg precision = 82.3%	No need of labeled dataset. Human evaluation is necessary. Unable to handle singleton reviews because these have insufficient information to find trustiness of reviewers.
[85]	Datasets collected from Yelp.com (YelpChi, YelpNYC, YelpZip)	Textual and bahavioral features	SpEagle (unsupervised), SpEagle+(semi-supervised) (YelpChi, YelpNYC, YelpZip repectively)	Avg precision (user ranking) = 0.3393, 0.2680, 0.3616 Avg precision (review ranking) = 0.3236, 0.2460, 0.3319 Avg AUC (user ranking) =0.6905, 0.6575, 0.6710 Avg AUC (review ranking) = 0.7887, 0.7695, 0.7942	SpEagle performance is comparatively good. But unable to detect group spammers.
[97]	Amazon product review data set (5,018,344 reviews of 570,606 products by 1,859,242 reviewers)	Rating behavior	Identification of spammers through consideration of majority opinion	AUC= 0.964 while AUC=0.940 for FraudEagle	Hyper-parameters are not required. Simple and less computations.
[98]	Yelp NYC (608,598 reviews), Three other datasets (Review-based, Item-based, and User-based) created from Yelp NYC, Amazon	Review-behavioral, User-behavioral, Review-linguistic, User-linguistic	Unsupervised-NetSpam: (1) weight calculation, and (2) Labeling	High performance AP, AUC	NetSpam (using weight calculation method and considering meta-path concept) outperforms SPeaglePlus specially when number of features increase . Group spammer features are not Considered.
[133]	YelpChi (67395 reviews, 38063 users, 201 hotel and restaurants), YelpNYC (359052 reviews, 260277 users, 923 hotel and restaurants), YelpZip (608598 reviews, 260277 users, 5044 hotel and restaurants))	Doc2vec and Node2vec, review ratings	Semi-supervised learning SPR2EP (SPam Review REPresentation with Node2vec and Doc2vec)	AUC (Yelpchi, YelpNYC, YelpZip respectively)=0.8071, 0.8129, 0.8318	Doc2vec features with Node2vec features improves the accuracy performances.

Table 5 (continued)

Author (year)	Datasets used	Features used	Techniques used	Results	Strentgh/Limitations
[46]	Xianyu dataset (37,323,039 comments published by 9,158,512 users)	edge, user and item embedding	Graph convolutional networks (GCN), dubbed GCN-based Anti-Spam method (GAS)	AUC=0.9872, F measure=0.8217	Reduce the impact of adversarial actions. Data loading (fetching and parsing) take a lot of time.
[65]	Datasets from Yelp.com (Rayana, 2015)	Reviewer features (such as RNR, ARD, WRD, RBST, ERD, ETG, ARL) and Review features (such as EXT, RR, RL, ISR, RES)	Semi-supervised BeGP : Graph-partitioned approach, Ranking based approach	BeGP's NDCG@k performances is better than FraudEagle, and SpEagle	Few labeled data are required to indentify spammer. Temporal features have not considerd.
[55]	Yelp-hotel and Yelp-restaurant	Entities relation embedding, social relation mining and user behavior embedding	Socially-aware unsupervised user behavior representation method for cold-start fraud detection (SUPER-COLD)	F measure (Hotel)=0.66 F measure (Restaurant)=0.65	Solved the cold start problem in deceptive opinion spam detection.
[129]	YelpChi, YelpNYC and YelpZip datasets (Rayana, 2015; Mukherjee, 2013b)	Burstiness (BST), Maximum number of reviews (MNR), Average rating deviation (avgRD), Review tightness (RT), Product tightness (PT), Group rating deviation (GRD), Group size (GS)	Unsupervised: Graph based Clique Percolation Method (CPM)	Precision is good	Outperformed in term of precision on large dataset (YelpNYC and YelpZip). But in YelpChi dataset SpEagle acheived good precision. Able to detect group spammers.

Table 6 Summary of pattern mining-based approach

Author (year)	Datasets used	Features used	Techniques used	Results	Strength/Limitations
[125]	Dataset from www.resellerratings.com (408,469 opinions posted by 343,629 users for 25,034 stores)	Posting time and ratings of the reviews	Time-Series Construction, Detecting correlated unusual patterns in curve-fitting based Multi-Dimensional-Time-Series (using Bayesian change point detection algorithm and longest common substring), Multi-Scale Spam Detection Algorithm Human Evaluation	Recall=75.86%, precision=61.1%	Effective in detecting singleton review spams. Many abnormal patterns are irrelevant. Increases false positive rate.
[49]	Dataset Dianping (6126113 reviews, 1074604 users, 1331471 IPs, 108787 restaurants)	regMainsite, regTu2Tr, regDist2SH, ATS, weekendPcnt, pcPcnt, avgDist2SH, AARD, uIPs, ucookies, ucities	Temporal and Spatial Patterns, ATS measure, (SVM) with n-gram	Accuracy= 0.85, Precision = 0.83, Recall= 0.87, F measure= 0.85	High accuracy in large real life dataset.
[31]	crawled from Amazon.com	Authors' rating behavior, Authors' activeness	Knowledge discovery tasks: Capturing peak intervals in number of reviews, and calculate context similarity in time window size.	F measure=0.86	No expensive calculations are needed because they captured suspicious time intervals. But, Some number of spam reviews exist in non-captured intervals.
[122]	Two real-world datasets from Yelp (CLT: 8870 users, 3130 objects, 29640 reviews) and Phoenix (PHX: 24506 users, 6167 objects, 62803 reviews)	userID, business ID, review, timestamp	Probabilistic generative model—Reliable Fake Review Detection (RFRD): Maximization (EM) based learning algorithm using bipartite graph	Accuracy (CLT)=76.3%, Accuracy (PHX)=75.6%	More hyper-parameters are used.
[113]	Taobao (256,443 reviews)	Number of reviews per time period, user rating, posting time	NPAR model (Near point auto regressive model) based on AR model	Accuracy = 85.91%	NPAR algorithm is efficient and real-time.
[15]	Amazon Dataset	Number of Reviews (NR), Content Similarity (CS), Content Similarity in Burst (CSBu), Rating Deviation (RD), Bursty Activity (BuA), Number of Reviews per Product (NRP), Extreme Rating (EXR), Reviewer Burstiness (RBu)	Rank based method using reviewer reputation and burst pattern	F measure=74.9%	Improved the performance (reducing loss of information) by using burst pattern discovery.

The summary of pattern mining-based approach, along with strength and limitations, are shown in Table 6.

5.5 Neural network-based approaches

Neural network-based approaches have advantages compared to Machine learning approaches. It has non-linear fitting capabilities, automatically learns interior features from raw data without any human effort, and captures the semantic relationship between the context word by pre-trained word embedding (word2vec, GloVe). But it required high computational power and training time. In neural network models, the backpropagation learning method is an iterative process used to train the feed-forward neural network to minimize the total error or mean error of output by adequately tuning the weights (gradient descent method). A neural network consists of an input layer, an output layer, and one or more hidden layers. The deep learning (deep neural network) approach uses a multilevel neural network. Most of the researchers used deep neural network models such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), long short-term memory (LSTMs), and deep feed-forward neural networks (DNNs), which have become the center of attention in recent years.

5.5.1 Convolutional neural network (CNN)

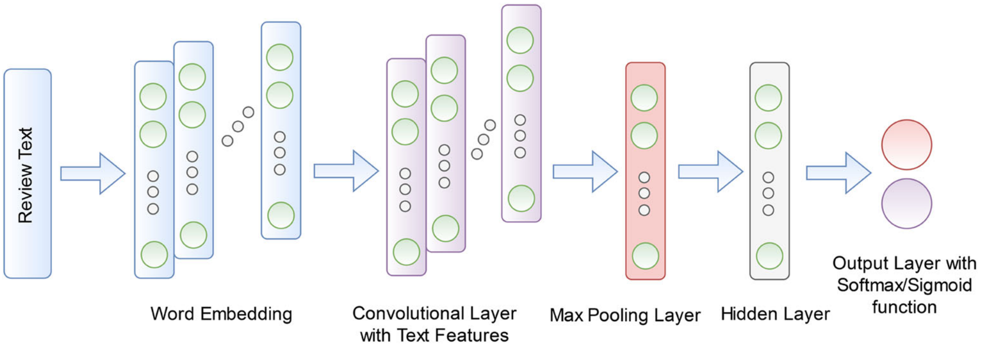
CNN performs a significant role in capturing essential features for classifying deceptive opinion spam, as shown in Fig. 8. First, the review text is converted into the word vectors as an input layer and passed to the convolutional layer that consists of multiple features with distinct dimensions. In this layer, the activation functions (relu, tanh) are applied to find convolution feature maps. Thereafter, max-pooling is applied to the feature map to capture the most important feature. Finally, these features are passed to the fully connected dense layer. In this dense layer, the activation function (softmax or sigmoid) is applied to classify reviews, and in the output layer, it shows deceptive or genuine labels.

Li et al. [54] proposed two convolutional neural network models, sentence convolutional neural network (SCNN) and sentence weighted neural network (SWNN), to represent the semantic meaning of sentences. They represented the review's whole document based on the word embedding on cross-domain datasets in their experiments and found that the SWNN model is more effective than other models. Ren and Zhang [87] used the various neural network model on the cross-domain dataset and found the CNN model is more effective than the RNN model. They also stated that attention-based bi-directional GRNN gives the best performances. Zhang et al. [137] proposed a Deceptive Review Identification approach by Recurrent Convolutional Neural Network (DRI-RCNN) combining max-pooling and ReLU filter. The first used a skip-gram model to create word embedding, then trained the contextual information by the recurrent neural network. The max-pooling is used for the review representation as a review vector, and the ReLU filter removes the negative element in the review vector. They found that DRI-RCNN achieves good results than other CNN models. Wang et al. [111] proposed a CNN model that jointly encodes the textual and behavioral information into the review embedding (RE), review's rating embedding (RRE), product's average rating embedding (PRE).

5.5.2 Recurrent neural network (RNN)

CNN is not able to preserve past information with current information. RNN has internal memories to preserve sequential information in a chain-like neural network architecture. But, RNN cannot be used for long sequences due to vanishing and exploding gradient at the backpropagation time. As a result, the researchers built various models such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Bi-directional LSTM to overcome the limitations of RNNs. Figure 9 shows the general architecture of LSTM and Gated RNN. At timestamp t, X_t is an input (i.e., word vector), H_{t-1} is a hidden state output from the previous timestamp. A new hidden state H_t and cell state C_t pass to the next timestamp.

Fig. 8 General Architecture of CNN for Deceptive Opinion Spam Detection


Review Text
Word Embedding
Convolutional Layer with Text Features
Max Pooling Layer
Hidden Layer
Output Layer with Softmax/Sigmoid function

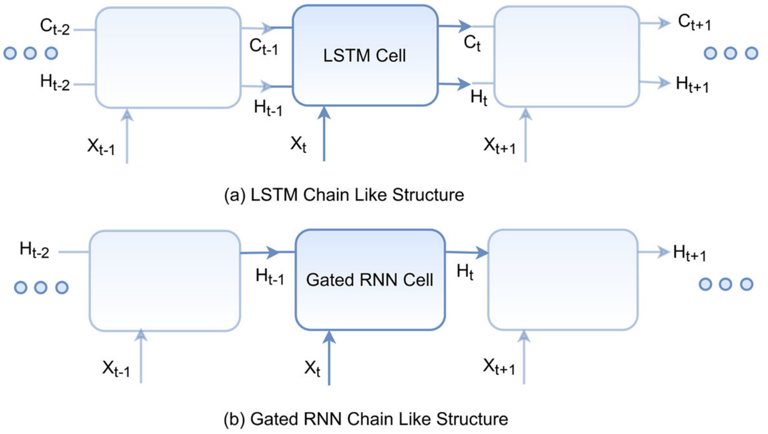


Fig. 9 General Architecture (a) LSTM and (b) Gated RNN

Wang et al. [110] introduced an attention-based LSTM neural network model using linguistic and behavioral features to detect deceptive opinion spam. Experimental results show that the attention module better performance than without attention. Wang et al. [107] have used LSTM with a deep neural network on Chinese word segmentation and achieved 89.4% accuracy. Saumya and Singh [96] proposed an unsupervised learning model LSTM and autoencoder to overcome the labeled dataset problem. They used different word embedding techniques such as OneHot, GloVe, Word2Vec to represent text data and found that OneHot encoding performed better than others. Liu et al. [61] proposed a BiLSTM model and features combination for learning document-level representation. They conducted two types of features first-person pronoun and POS with GloVe embedding in a cross-domain and mixed-domain dataset. The domain-independent experimental results show that the model performs better than the other models like SWNN, HAN, CLSTM, CNN-LSTM, and Deep CNN.

Jain et al. [35] proposed multiple deep learning-based models to solve the detection problem of variable review length. They found that the hierarchical CNN-GRU model achieved good performance than classical CNN and RNN models on all four benchmark datasets, i.e., Four-city, DRD, LMRD, and YelpZip. In this model, GRU is employed to learn long-range semantic dependency among extracted features obtained from CNN.

5.5.3 Generative adversarial networks (GAN)

Aghakhani et al. [1] proposed a generative adversarial network (GAN) called FakeGAN for identifying deceptive reviews. Standard GAN has two parts: generator and discriminator as shown in Fig. 10. The generator learns to create new data instances that resemble training data, and the discriminator learns to distinguish the generator's fake data from actual data. But, they used two discriminators to improve performance in their experiment. One discriminator tries to distinguish between genuine and

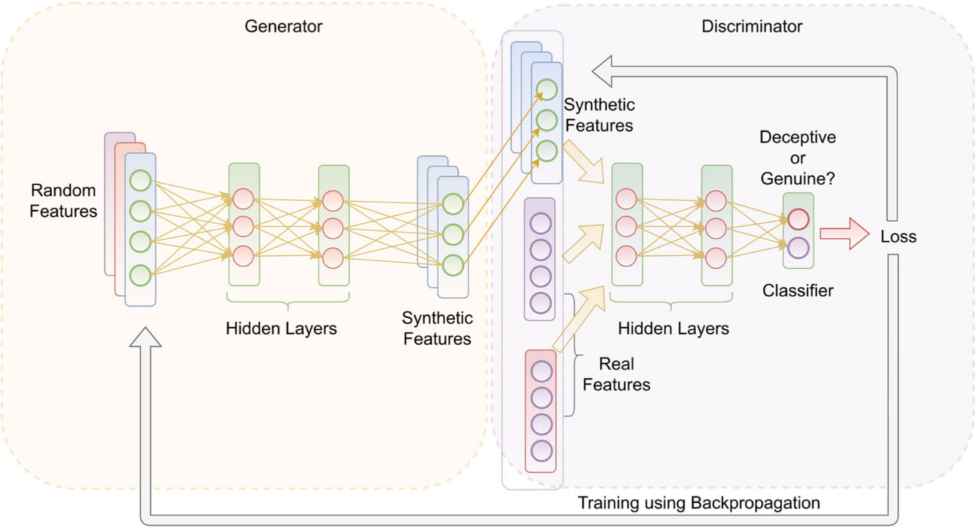


Fig. 10 Architecture of Generative Adversarial Network

deceptive reviews, whereas another distinguishes between reviews generated by the generator and samples from deceptive reviews distribution.

Stanton and Irissappane [101] proposed a semi-supervised generative adversarial network model (spam-GAN) on limited labeled data. The spam GAN model used three components: generator, discriminator, and classifier, which are trained using an ADAM optimizer. These components work together to identify deceptive opinion spams and create samples similar to the train set.

Tang et al. [103] have explored a behavior-based generative adversarial network (bfGAN) to handle the cold start problems. They extracted six real behavioral features RBFs such as AW, MNR, PR, RC, RD, and MCS for the regular users. The generators create synthetic behavior features SBFs for the new users from easily accessible features EAFs (exist in both new and regular users like text, rating, register, and posting timestamp). The discriminator gets EAFs and RBFs from the training data as input and SBFs from the generator for training the generator. They experimented on two Yelp datasets, and the model outperformed the state-of-art model with an accuracy 83% on hotel reviews and 75.7% on restaurant reviews.

5.5.4 Neural network-based language models (Transformers)

Statistical Language models such as n-grams, hidden Markov models (HMM), and specific linguistic rules perform to learn the probability distribution over words or sequence of words for the NLP task. The n-gram model requires more computational overhead for large documents and sparse representation of language. Recently, various NLP models like Word2Vec, GloVe, BERT, and RoBERTare being done on neural networks. Earlier models like Word2Vec and GloVe learn a single representation for each word without its context. Later these representation models scale up to sentences and documents. Today, the models learned word representation based on the word's context. If two models are developed to perform similar tasks, generalized learning knowledge can be transferred between them rather than training separately. Thus, transfer learning is the reuse of a pre-trained model on a new NLP task. Transformer works on encoder-decoder and multi-head self-attention mechanism, which promotes sequence-to-sequence learning.

Bidirectional encoder representations from transformers (BERT): A word like "Apple" may have two different meanings (fruits or company) in a different context. The word embedding like Word2Vec and GloVe gives the same vector for the word "Apple" in both contexts. BERT learns the word representation from both sides left and right context of the word simultaneously in the training phase [18]. The model is deeply bidirectional pre-trained on a large corpus so that every word here has only one meaning. It consists of 12 transformer blocks, 12 attention heads per block, 768 hidden layers, and 110 million parameters. Three embedding layers (position, segment, and token) are combined as an input layer. The classification token [CLS] and separate segment token [SEP] were put at the beginning and end of each statement, respectively. There is a need to pad all token lists to the fixed size. A fixed-length attention mask 0 indicates padded tokens, whereas 1 indicates unpadded tokens. The last transformer layer [CLS] output contains a prediction probability vector for classification.

Distilled version of BERT (DistilBERT) DistilBERT is based on knowledge distillation. Knowledge distillation is a compression strategy that involves training a tiny model to mimic the behavior of a bigger model. DistilBERT has nearly 40% fewer parameters than the BERT base model, making it faster. It consists of 6 transformer blocks, 12 attention heads per block, 768 hidden layers, and 66 million parameters. The model uses a triple loss system that includes a cosine embedding loss, masked language modeling loss, and student loss with random initialization. Token-type embeddings and the pooler are not present in the model. We need a lightweight and robust model with low latency on edge devices without compromising much on performance.

Robustly optimized BERT (RoBERT) RoBERT is pre-trained transformer learning built by the Facebook AI team similar to the BERT masking. It performed better than BERT in various NLP cases because of extensive training data. Byte-Pair encoding is used to tokenize the text data. It pre-trained on 160 GB corpus of Wikipedia and CCNews in the English language. It consists of 12 transformer blocks, 12 attention heads per block, 768 hidden layers, and 125 million parameters. The next sentence prediction has been removed in RoBERT; that's why it can solve the NSP loss in BERT.

Table 7 shows the summary of neural network-based techniques with performance results and limitations.

6 Evaluation metrics and complexity of detection models

6.1 Evaluation metrics

Various metrics such as accuracy, precision, recall, F-measure, AUC-ROC curve, etc., measure the model's

Table 7 Summary of neural network-based approach

Neural network models	Results	Datasets used	Features used	Author (year)	Strengh/Limitations
CNN, RNN, GRNN, bi-directional GRNN (Attention)	Accuracy (CNN)=75.9%, Accuracy (RNN)=63.2%, Accuracy (GRNN)=80.1%, Accuracy (Bi-directional GRNN)=83.6%, Accuracy (Bi-directional GRNN (Attention))= 84.1%	Dataset from Li, 2014b (Hotel, Restaurant and Doctor reviews by Customer, Employee , Turker)	Word embedding (CBOW)	Ren and Zhang [87]	Neural network model outperforms a state-of-the-art discrete baseline. Bi-directional GRNN with Attention gives best performances.
Convolutional neural network (CNN) using jointly encode the textual and behavioral features into the review embedding (RE), product's average rating embeddings (PRE), review's rating embeddings (RRE)	Accuracy (Hotel)= 65.3% (RE+RRE+PRE) Accuracy (Restaurant)= 62.0% (RE+RRE+PRE)	Yelp dataset	Textual and behavioral featuresyl	Wang et al. [111]	CNN achieves better performance than the traditional model and handle the cold-start problem.
CNN model with max-pooling	Accuracy = 88.50%	Dataset from ott et al.	GloVe Word embedding, Count vectors features, TF-IDF	Archchitha and Charles [5]	Scalable but require more computational time.
Attention-Based Neural Networks (CNN)	Accuracy (Hotel)=88.8%, Accuracy (Restaurant)=91.0%	Yelp dataset (5678 Hotel's reviews by 5124 users, 58517 Restaurant's reviews by 35593 users)	Linguistic and behavioral features	Wang et al. [110]	Attention-based achieved good performance than Mukharjee (2013c)'s model because it learn dynamic weights for each traininig sample of linguistic and behavioral features.
Recurrent convolutional neural network (RCNN)	Accuracy at 90% training = 88.15% (DRI-RCNN)	Datasets (from Ott, 2011; Li, 2014b)	Skip-gram	Zhang et al. [137]	DRI-CNN achieve good result than other CNN models.
Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN)	Accuracy = 0.894, F measure = 0.796	Mobile01.com (8363 posts: 458 deceptive review posts and 7905 regular review posts, 111065 responses: 5245 deceptive review responses and 105820 regular review responses)	Chinese word segmentation	Wang et al. [107]	Deep learning LSTM has achieved better performance than the conventional SVM approach of machine learning.

Table 7 (continued)

Neural network models	Results	Datasets used	Features used	Author (year)	Strength/Limitations
Long Short-Term Memory (LSTM) autoencoder	F1 score (OneHot embedding)=0.99 F1 score (Glove and Word2Vec)=0.49	Review (or comments) of five popular videos (Psy, KatyPerry, LMFAO, Eminem, Shakira) on YouTube	Sequences of words and sentences (representation of words like one-hot encoding vector, or Word2Vec , or GloVe)	Saumya and Singh [96]	OneHot embedding preserves the word sequence so it performs better than Glove and Word2Vec. Not a generalized model and fail to detect duplicate review as spam.
Neural network which is based on BiLSTMWF and features combination	F measure=87.6%, F measure = 82.4% (in cross domain)	Datasets (from Ott, 2011; Li, 2014b)	Glove embedding, POS embedding, First person pronouns embedding	Liu et al. [61]	On a cross-domain data collection, it outperforms baseline methods.
Sentence vector/twin-word embedding conditioned bi-LSTM, Dynamic knowledge graph-based method for fake-review detection (DKG-FRD)	Accuracy (Amazon)=93.41% Accuracy (Netflix)=92.65% Accuracy (Movielens)=94.38%	Amazon, Netflix, Movielens	ID, posted reviews, action times, ratings, and store links of the reviewer	Fang et al. [21]	ST-BLSTM model detected more mysterious spam review activities with higher precision than KNN and SVM. Not being able to manage multiple reviewer IDs.
Stochastic decision tree model: Autoencoder (Auto-encoder is a neural network) and random forest.	Accuracy=95.85%	Amazon review data set (7950 reviews, 815 unique users): Users who wrote less than 20 reviews or greater than 500 reviews, were filtered	User's behavioral features, review content features	Dong et al. [20]	Achieved good accuracy compare to other author's model(Mukherjee, Heydari, Xu and Zhang, Zhang, Rout). But the author selected users who wrote reviews between 20 to 500.
Topic modelling approach (LDA), Modified possibilistic fuzzy c-means, Selective memory architecture-based CNN	Accurcay = 92.83% (SMA-CNN with LDA-modified PFCM)	Amazon dataset (35 million reviews up to March 2013)	Topic word	Mandhula et al. [66]	SMA-CNN with LDA-modified PFCM achieved good accuracy than other classifiers (such as NB, RF, DT).
Multiple Deep Neural Network (DNN)	Accuracy of DOSC, Four city, YelpZip, LMRD, DRD respectively (using CNN-GRU)=91.9%, 84.7%, 66.4%, 88.9%, 83.8%	DOSC (from Ott, 2011, 2013), Four-city Dataset (Li, 2013), YelpZip Dataset (Rayana, 2015), Large Movie Review Dataset (LMRD from Maas, 2011), Drug Review Dataset (DRD from Gräßer et al., 2018)	pre-trained word embedding (CBOW)	Jain et al. [35]	On all four benchmark datasets, hierarchical CNN-GRU models outperform the classical RNN and CNN models. Handling text with broad and variable reviews.

Table 7 (continued)

Neural network models	Results	Datasets used	Features used	Author (year)	Strentgh/Limitations
Deep feed-forward neural network (DNN)	Accuracy (on Negative dataset)=88.38% (n-gram+skip-gram), Accuracy (on positive dataset)=89.75% (n-gram+skip-gram)	Dataset (from Ott, 2011, 2013)	n-gram , skip-gram	Barushka and Hajek [9]	DNN achieves a better performance than CNN, NB, and SVM.
Generative Adversarial Networks(GANs): FakeGAN (two discriminator models and one generative model)	Accuracy = 0.891	Datasets (from Ott, 2011)	Content features	Aghakhani et al. [1]	Discriminator can gets stuck in a local minimum. GAN can fail to converge at training time.
behavior feature GAN (bfGAN) model and SVM classifier	Accuracy (Hotel) =83% Accuracy (Restaurant)=75.7%	yelp (X. Wang, 2017 and Z. You, 2018 dataset)	Easily accessible features (EAFs), Real behavior features (RBFs), generated synthetic behavior features (SBFs)	Tang et al. [103]	bfGAN improved the performace for detecting cold start deceptive reviews. Still have a domain adaption problem.
Attention-driven Conditional GAN	Accuracy= 87.03%	Douban (Moview review)	Movie score, User rating, Region, Movie genres, Average director score, Average actor score	Gong et al. [29]	adCGAN model achieved best performance than LOF, OCSVM, MCSVM, MCD, iForest, VAE, and adCGAN_w/o_attention. It may not be well suited to product reviews. In this model, we have to reinvestigate feature words maually for other types of reviews.

performance. These metrics can be obtained from a confusion matrix. 2 × 2 matrices are used for the binary classification model, as shown in Fig. 11. The confusion matrix confers information about spam and non-spam predictions regarding true-positive (TP), true-positive (TN), false-positive (FP), and false-negative (FN).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{Reviews\ have\ been\ correctly\ predicted\ as\ deceptive\ or\ truthful}{Total\ reviews} \tag{26}$$

$$Precision = \frac{TP}{TP + FP} = \frac{Reviews\ have\ been\ correctly\ predicted\ as\ deceptive}{All\ reviews\ have\ been\ predicted\ as\ deceptive} \tag{27}$$

$$Recall = \frac{TP}{TP + FN} = \frac{Reviews\ have\ been\ correctly\ predicted\ as\ deceptive}{All\ reviews\ that\ have\ as\ actual\ deceptive\ class} \tag{28}$$

$$Specificity = \frac{TN}{TN + FP} = \frac{Reviews\ have\ been\ correctly\ predicted\ as\ truthful}{All\ reviews\ that\ have\ as\ actual\ truthful\ class} \tag{29}$$

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{30}$$

The area under the receiver operating characteristics curve (AUC – ROC) plots the true-positive rate (sensitivity or recall) and false-positive rate (1- specificity), as shown in Fig. 12. AUC under ROC is a metric that measures overall performance across all categorization thresholds.

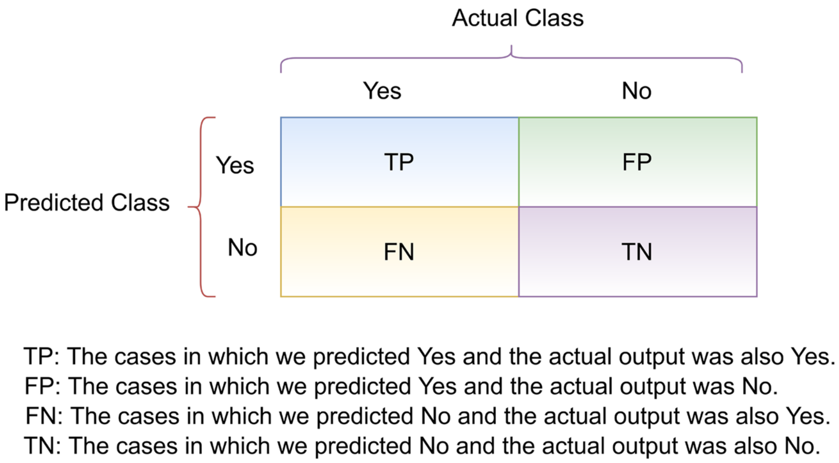
6.2 Complexity

In asymptotic analysis, machine learning algorithms' time and space complexity are estimated using $Big - O$ notation. The time used by an algorithm for the input dataset and the number of features is time complexity. Space complexity refers to the amount of memory used by the algorithm for the input dataset. Several researchers have calculated the complexity of their algorithm to detect deceptive opinion spam and spammers, which are given below (Table 7).

6.2.1 The complexity of graph-based models

The COSD model [120] has a time complexity $O(I \times |U| \times k \times K)$, where U is the embedding matrix, k is a negative sample, K is the dimension of representation vectors, and I is the number of iteration. Xu et al. [130] proposed a graph-based approach clique percolation method (CPM) to detect opinion spammer groups which time and space complexity are calculated as $O(R + E)$ and $O(R)$, respectively, where

Fig. 11 Confusion Matrix



		Actual Class	
		Yes	No
Predicted Class	Yes	TP	FP
	No	FN	TN

TP: The cases in which we predicted Yes and the actual output was also Yes.
FP: The cases in which we predicted Yes and the actual output was No.
FN: The cases in which we predicted No and the actual output was also Yes.
TN: The cases in which we predicted No and the actual output was also No.

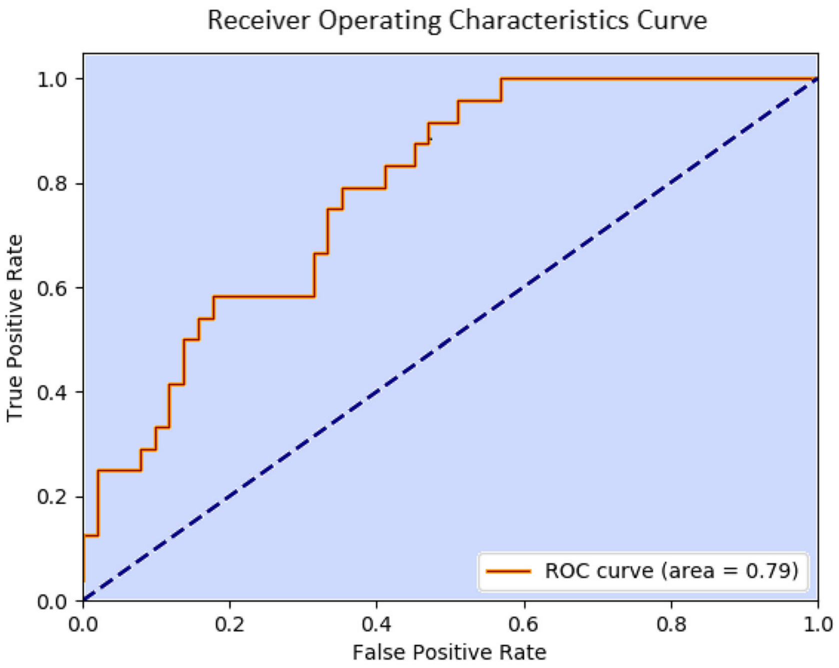


Fig. 12 Area Under the ROC curve

R is the number of reviewers and E is the number of edges.

6.2.2 The complexity of unsupervised models

Li et al. [52] introduced LDA having time complexity as $O(N_{terms} \times N_{topic} \times T)$, where Nterms is the number of terms, Ntopic is the number of topics, and I is the number of iterations. According to [51], frequency itemset mining (FIM) based algorithm having complexity $O(N_{item}^3)$ with support 3, where Nitem is the total number of items. Latent topic mining on the reviews, the time complexity can be calculated by $O(N \times N_{terms} \times N_{topic})$ where N is the number of reviews, N_{terms} is the number of terms, and Ntopic is the topic number. However, The time complexity of Grouped Spam Detection approach based on the Nominated Topics (GSDNT) model is $O(N^2)$, where N is the number of reviews.

6.2.3 The complexity of machine learning algorithms

Accurately estimating the complexity of a machine learning algorithm can be a difficult task, as it primarily depends on the implementation of the algorithm and training parameters. The attributes of any dataset may lead to other algorithms, due to which the overall complexity changes. The complexity of some supervised algorithms is given below. Let n is the total number of training samples, p is the total features, n_{trees} is the depth of trees, n_{sv} is the number of support vectors, c is the number of class labels (Table 8).

6.2.4 The complexity of neural network models

The training time complexity of a simple neural network is calculated as $O(mn \times (ij + jk + kl))$. Where '$n$' is the number of training samples, 'm' is the epoch number; i, j, k, l are the neurons of 4 layers, respectively. The time

Table 8 Complexity of Supervised Algorithm

Algorithm	Training time	Prediction time	Space complexity
Naive Bayes	$O(npc)$	$O(pc)$	$O(pc)$
SVM (Kernel)	$O(n^2 p + n^3)$	$O(n_{sv} p)$	$O(n_{sv})$
Logistic regression	$O(np)$	$O(p)$	$O(np)$
kNN	$O(knp)$	$O(kp)$	$O(np)$
Decision Tree	$O(n^2 p)$	$O(n_{trees})$	$O(p)$
Random Forest	$O(n^2 p n_{trees})$	$O(p n_{trees})$	$O(p n_{trees})$
Linear Regression	$O(p^2 n + p^3)$	$O(p)$	$O(p)$
Gradient Boosting	$O(n p n_{trees})$	$O(p n_{trees})$	$O(p n_{trees})$
PCA	$O(np \times min(n, p) + p^3)$	–	$O(n^3)$

complexity of the CNN mainly depends on the depth of the layers, the number of filters, and input channels. It can be calculated as $O(\sum_{l=1}^{d} .n_{l-1}.s_l^2.n_l.m_l^2)$ [105]. Where 'l' is the index layer, 'd' is the depth of the layers, n_{l-1} number of input channels of l^{th} layer, n_l is the number of filters in l^{th} layer, s_l is the spatial length of the filter, and m_l is the spatial size of the output feature map. The time of pooling layers is negligible in the model.

7 Challenges and future direction

This literature showed that there are still a variety of research gaps and open challenges in deceptive opinion spam detection research. Significant research challenges and directions are elaborated below:

Creating a labeled dataset is a significant challenge to detect opinion spam. Although labeled data is required to train classification models, it is an expensive and time-consuming process to construct labeled data. Moreover, we need specialists to correctly classify each review as genuine and deceptive. For researchers, creating an extensible and domain-independent ground-truth dataset for opinion spam identification is still an open challenge. Many reviews are added through numerous social media tools, facing more challenges in mining reviews and deriving truthful conclusions effectively. Consumers are free to write their opinion on the commercial or social website, and there is no control over the quality of reviews, including which deceptive reviews are considered worst. The availability of fake reviews also makes the problem more severe as it may impair consumer buying decisions. Further, it mentioned that sellers generate deceptive reviews to promote their products or unfairly tarnish their rivals' reputations.

Another difficulty in the identification of opinion spam is changing opinions over time. After using a particular product and being convinced about its quality, people change their opinion over time. Therefore, it becomes a critical challenge when individuals change their opinion after using the product. It is observed that people are different from each other in the writing style of reviews or opinions, which is associated with the perception that everyone expresses their opinion in a different manner poses a challenge on its own. In addition, people's writing styles typically do not follow grammatical rules and are not common in certain situations. It is also advised that sentiment analysis can be conducted to classify reviews, but it is a big challenge for analyzing sentiment words in cross-domain.

One of the major challenges in detecting deceptive reviews is finding the appropriate set of features that best define the reviewers' spamming activity. Most of the existing works have used textual features and behavioral features individually. Therefore, to improve the existing training model's efficiency, it needs to be fitted with an optimum hybrid features set that can improve the detection model. Cold-start Review detection is still a challenge. Finding effective spamming behaviors for singleton reviews (i.e., a review by new users who post only one review) is a challenging task. Many researchers are still working on the cold-start problem using a generative adversarial network. Reviews can be in more than one language, which is challenging to handle. Many dictionaries (WordNet, SentiWordNet) and tools (LIWC) are available to analyze English language reviews that can easily find the word's semantic meaning. Thus, in-depth research needs to be done on detecting deceptive opinion spam in multilingual reviews.

The device's IP address and MAC address can prove very useful to detect review spammers because the same user, using multiple identifications, post multiple reviews with similar opinions. Thus, the reviewer's IP address and the location where the reviewer logged in to post the review; can help detect the spammers. The unavailability of multidimensional data is a big challenge in this area. The models used in deceptive opinion spam detection based on the old dataset are less precise and inaccurate; therefore, reconstruct this model (concept drift). Thus, it is a challenge to create an appropriate model that can handle the target concept drift and be rapidly adapt to it. The literature indicates that detecting spammer groups is the more essential part of the research on deceptive opinion spam. Burst patterns are a more crucial behavior of spamming. Thus, in-depth research needs to be done on burst patterns of the spammer groups.

Most of the deep learning models are not suitable for deceptive opinion spam detection. Many researchers use as black-box of these models. So, there is a need for a deep learning model to expand declarative knowledge. Many researchers have worked on only one domain. A few researchers have used cross-domain in this area, which means they trained in one domain and tested in other domains. Experimental results show that the model's performance degrades in cross-domain compared to the same domain. Thus, there is a need to handle the cross-domain issues effectively by more research and investigation.

Still, the expected annotated data is not antecedently available for some languages. Creating new annotation data requires more time and costly exercise, which is challenging. Transfer learning using language models that have been pre-trained is a promising option. Much research has been done to work on opinion and sentiment analysis, but some important problems like domain dependency, sarcasm/slang, and non-English detection need to be solved in the future.

8 Conclusion

This literature presents a systematic review of almost all the approaches related to deceptive opinion spam detection and states the importance of all its features, which can be used in performance evaluation. We know that most of the researchers have focused on content-based features and used supervised learning, but it requires a large real labeled dataset that can not create easily. Even though many studies have developed their synthetic datasets, these datasets are not ground truth real-world reviews because these are written by Turkers or other people, not by real spammers. In this way, evaluating classifiers can be problematic. Therefore, the attention of the researchers moved towards semi-supervised models (Co-training and PU learning), which require few labeled data. But, the performance of these models is not very satisfactory and needs a more effective algorithm to make these models. Unsupervised learning models played an essential role in effectively detecting review spam and spammer.

Moreover, many researchers have discovered behavioral-based models to detect spammers and group spammers in two ways FIM-based and Graph-based. Neural network models or deep learning models (like CNN, RNN, gated RNN, LSTM, BiLSTM, GAN, BERT) are used to learn semantic representation in this area. Neural network models utilize one or more hidden layers to capture the most relative global semantic words and sentences. This model shifts the burden of manual feature engineering. We have discussed the overall description of recent research work and corresponding limitations. Apart from this, we analyzed the various dataset and research methodologies in this field and highlighted future research directions. Thus, this literature survey can help to enhance the knowledge of deceptive opinion spam detection.

References

1. Aghakhani H, Machiry A, Nilizadeh S, Kruegel C, Giovanni Vigna (2018) Detecting deceptive reviews using generative adversarial networks. In: 2018 IEEE security and privacy workshops (SPW), pp 89–95
2. Ahsan MNI, Nahian T, Kafi AA, Hossain MdI, Shah FM (2016) An ensemble approach to detect review spam using hybrid machine learning technique. In: 2016 19th International conference on computer and information technology (ICCIT). IEEE, pp 388–394
3. Akoglu L, Chandy R, Faloutsos C (2013) Opinion fraud detection in online reviews by network effects. ICWSM 13(2–11):29
4. Algur SP, Patil AP, Hiremath PS, Shivashankar S (2010) Conceptual level similarity measure based review spam detection. In: 2010 International conference on signal and image processing. IEEE, pp 416–423
5. Archchitha K, Charles EYA (2019) Opinion spam detection in online reviews using neural networks. In: 2019 19th International conference on advances in ICT for emerging regions (ICTer), vol 250. IEEE, pp 1–6
6. Aritsugi M et al (2017) Combining word and character n-grams for detecting deceptive opinions. In: 2017 IEEE 41st annual computer software and applications conference (COMPSAC), vol 1. IEEE, pp 828–833
7. Asghar MZ, Ullah A, Ahmad S, Khan A (2020) Opinion spam detection framework using hybrid classification scheme. Soft Comput 24(5):3475–3498
8. Barbado R, Araque O, Iglesias CA (2019) A framework for fake review detection in online consumer electronics retailers. Inform Process Manag 56(4):1234–1244
9. Barushka A, Hajek P (2019) Review spam detection using word embeddings and deep neural networks, Springer
10. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3(Jan):993–1022
11. Blum A, Mitchell T (1998) Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on computational learning theory, pp 92–100
12. Cao N, Ji S, Chiu DKW, He M, Sun X (2020) A deceptive review detection framework: Combination of coarse and fine-grained features. Expert Systems with Applications page 113465
13. Chen Y-R, Chen H-H (2015) Opinion spammer detection in web forum. In: Proceedings of the 38th International ACM SIGIR conference on research and development in information retrieval, pp 759–762
14. Chengzhang J, Kang D-K (2015) Detecting the spam review using tri-training. In: 2015 17th International conference on advanced communication technology (ICACT). IEEE, pp 374–377
15. Dematis I, Karapistoli E, Vakali A (2018) Fake review detection via exploitation of spam indicators and reviewer behavior characteristics. In: International conference on current trends in theory and practice of informatics. Springer, pp 581–595
16. Deng H, Zhao L, Luo N, Liu Y, Guo G, Wang X, Tan Z, Wang S, Zhou F (2017) Semi-supervised learning based fake review detection. In: 2017 IEEE International symposium on parallel and distributed processing with applications and 2017 IEEE International conference on ubiquitous computing and communications (ISPA/IUCC). IEEE, pp 1278–1280
17. Deng R, Ruan N, Jin R, Lu Y, Jia W, Su C, Xu D (2018) Spamtracer: manual fake review detection for o2o commercial platforms by using geolocation features. In: International conference on information security and cryptology. Springer, pp 384–403
18. Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805
19. Dong L-Y, Ji S-J, Zhang C-J, Zhang Q, Chiu DW, Qiu L-Q, Li D (2018) An unsupervised topic-sentiment joint probabilistic model for detecting deceptive reviews. Expert Syst Appl 114:210–223
20. Dong M, Yao L, Wang X, Benatallah B, Huang C, Ning X (2020) Opinion fraud detection via neural autoencoder decision forest. Pattern Recogn Lett 132:21–29
21. Fang Y, Wang H, Zhao L, Fengping Y, Wang C (2020) Dynamic knowledge graph based fake-review detection. Appl Intell 50(12):4281–4295
22. Fei AM, Liu B, Hsu M, Castellanos M, Ghosh R (2013) Exploiting burstiness in reviews for review spammer detection. ICWSM 13:175–184
23. Feng S, Banerjee R, Choi Y (2012a) Syntactic stylometry for deception detection. In: Proceedings of the 50th annual meeting

of the association for computational linguistics (volume 2: short papers), pp 171–175
24. Feng S, Xing L, Gogar A, Choi Y (2012b) Distributional footprints of deceptive product reviews. ICWSM 12(98):105
25. Feng VW, Hirst G (2013) Detecting deceptive opinions with profile compatibility. In: Proceedings of the sixth international joint conference on natural language processing, pp 338–346
26. Fusilier DH, Gómez MM-y, Rosso P, Cabrera RG (2015) Detecting positive and negative deceptive opinions using pu-learning. Inform Process Manag 51(4):433–443
27. Gao K, Hua X, Wang J (2015) A rule-based approach to emotion cause detection for chinese micro-blogs. Expert Syst Appl 42(9):4517–4528
28. Giasemidis G, Kaplis N, Agrafiotis I, Nurse JRC (2018) A semi-supervised approach to message stance classification. IEEE Trans Knowl Data Eng 32(1):1–11
29. Gong M, Gao Y, Xie Y, Qin AK (2020) An attention-based unsupervised adversarial model for movie review spam detection. IEEE Transactions on Multimedia
30. Hajek P, Barushka A (2019) A comparative study of machine learning methods for detection of fake online consumer reviews. In: Proceedings of the 2019 3rd international conference on E-Business and internet, pp 18–22
31. Heydari A, Tavakoli M, Salim N (2016) Detection of fake opinions using time series. Expert Syst Appl 58:83–92
32. Huang J, Qian T, He G, Zhong M, Peng Q (2013) Detecting professional spam reviewers. In: International Conference on advanced data mining and applications. Springer, pp 288–299
33. Hussain N, Mirza HT, Rasool G, Hussain I, Kaleem M (2019) Spam review detection techniques: a systematic literature review. Appl Sci 9(5):987
34. Hussain N, Mirza HT, Hussain I, Iqbal F, Memon I (2020) Spam review detection using the linguistic and spammer behavioral methods. IEEE Access 8:53801–53816
35. Jain N, Kumar A, Singh S, Singh C, Tripathi S (2019) Deceptive reviews detection using deep learning techniques. In: International conference on applications of natural language to information systems. Springer, pp 79–91
36. Ji S-J, Qi Zhang, Li J, Chiu DKW, Xu S, Yi L, Gong M (2020) A burst-based unsupervised method for detecting review spammer groups. Information Sciences
37. Jia S, Zhang X, Wang X, Liu Y (2018) Fake reviews detection based on lda. In: 2018 4th International conference on information management (ICIM). IEEE, pp 280–283
38. Jindal N, Liu B (2007) Analyzing and detecting review spam. In: Seventh IEEE International conference on data mining ICDM, vol 2007, pp 547–552
39. Jindal N, Liu B (2008) Opinion spam and analysis. In: Proceedings of the 2008 International conference on web search and data mining, pp 219–230
40. Jindal N, Liu B, Lim E-P (2010) Finding unusual review patterns using unexpected rules. In: Proceedings of the 19th ACM International conference on information and knowledge management, pp 1549–1552
41. Khurshid F, Zhu Y, Zhuang X, Ahmad M, Ahmad M (2018) Enactment of ensemble learning for review spam detection on selected features. Int J Comput Intell Syst 12(1):387–394
42. Kim S, Chang H, Lee S, Minhwan Y, Kang J (2015) Deep semantic frame-based deceptive opinion spam analysis, pp 1131–1140
43. Lai CL, Xu KQ, Lau RYK, Li Y, Jing L (2010a) Toward a language modeling approach for consumer review spam detection. In: 2010 IEEE 7th International conference on E-Business engineering. IEEE, pp 1–8
44. Lai CL, Xu KQ, Lau RYK, Li Y, Song D (2010b) High-order concept associations mining and inferential language modeling for online review spam detection. In: 2010 IEEE International conference on data mining workshops. IEEE, pp 1120–1127
45. Lau RYK, Liao SY, Kwok RC-W, Xu K, Xia Y, Li Y (2012) Text mining and probabilistic language modeling for online review spam detection. ACM Trans Manag Inform Syst (TMIS) 2(4):1–30
46. Li A, Qin Z, Liu R, Yang Y, Li D (2019a) Spam review detection with graph convolutional networks. In: Proceedings of the 28th ACM International conference on information and knowledge management, pp 2703–2711
47. Li FH, Huang M, Yang Y, Zhu X (2011) Learning to identify review spam. In: Twenty-Second International joint conference on artificial intelligence
48. Li H, Liu B, Mukherjee A, Shao J (2014a) Spotting fake reviews using positive-unlabeled learning. Computación y Sistemas 18(3):467–475
49. 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. In: Ninth International AAAI conference on web and social media
50. Li H, Fei G, Wang S, Liu B, Shao W, Mukherjee A, Shao J (2017a) Bimodal distribution and co-bursting in review spam detection. In: Proceedings of the 26th International conference on world wide web, pp 1063–1072
51. Li J, Lv P, Xiao W, Yang L, Zhang P (2021) Exploring groups of opinion spam using sentiment analysis guided by nominated topics. Expert Syst Appl 114585:171
52. Li J, Cardie C, Li S (2013) Topicspam: a topic-model based approach for spam detection. In: Proceedings of the 51st annual meeting of the association for computational linguistics (volume 2: short papers), pp 217–221
53. Li J, Ott M, Cardie C, Hovy E (2014b) Towards a general rule for identifying deceptive opinion spam. In: Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: long papers), pp 1566–1576
54. Li L, Qin B, Ren W, Liu T (2017b) Document representation and feature combination for deceptive spam review detection. Neurocomputing 254:33–41
55. Li Q, Wu Q, Zhu C, Zhang J, Zhao W (2019b) Unsupervised user behavior representation for fraud review detection with cold-start problem. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, pp 222–236
56. Ligthart A, Catal C, Tekinerdogan B (2021) Analyzing the effectiveness of semi-supervised learning approaches for opinion spam classification. Appl Soft Comput 101:107023
57. Lim E-P, Nguyen V-A, Jindal N, Liu B, Lauw HW (2010) Detecting product review spammers using rating behaviors. In: Proceedings of the 19th ACM international conference on information and knowledge management, pp 939–948
58. Lin C, He Y (2009) Joint sentiment/topic model for sentiment analysis. In: Proceedings of the 18th ACM conference on information and knowledge management, pp 375–384
59. Lin Y, Zhu T, Wu H, Zhang J, Wang X, Zhou A (2014) Towards online anti-opinion spam: Spotting fake reviews from the review sequence. In: 2014 IEEE/ACM International conference on advances in social networks analysis and mining (ASONAM 2014). IEEE, pp 261–264
60. Liu S, Zhang J, Xiang Y (2016) Statistical detection of online drifting twitter spam. In: Proceedings of the 11th ACM on asia conference on computer and communications security, pp 1–10
61. Liu W, Jing W, Li Y (2020) Incorporating feature representation into bilstm for deceptive review detection. Computing 102(3):701–715

62. Liu Y, Bo P, Wang X (2019) Opinion spam detection by incorporating multimodal embedded representation into a probabilistic review graph. Neurocomputing 366:276–283
63. Yuqing L, Zhang L, Xiao Y, Li Y (2013) Simultaneously detecting fake reviews and review spammers using factor graph model. In: Proceedings of the 5th annual ACM web science conference, pp 225–233
64. Malik MSI, Hussain A (2018) An analysis of review content and reviewer variables that contribute to review helpfulness. Inform Process Manag 54(1):88–104
65. Manaskasemsak B, Chanmakho C, Klainongsuang J, Rungsawang A (2019) Opinion spam detection through user behavioral graph partitioning approach. In: Proceedings of the 2019 3rd international conference on intelligent systems metaheuristics & swarm intelligence. ACM, pp 73–77
66. Mandhula T, Pabboju S, Gugulotu N (2019) Predicting the customer's opinion on amazon products using selective memory architecture-based convolutional neural network. The Journal of Supercomputing, 1–25
67. Maurya R, Singh SK, Maurya AK, Glcm AK (2014) Multi class support vector machine based automated skin cancer classification. In: International conference on computing for sustainable global development (INDIACom). IEEE, pp 444–447
68. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119
69. Kakhki AM, Kliman-Silver C, Iolaus AM (2013) Securing online content rating systems. In: Proceedings of the 22nd International conference on world wide web, pp 919–930
70. Mukherjee A, detection VV (2014) Opinion spam An unsupervised approach using generative models. Techincal Report UHx
71. Mukherjee A, Liu B, Glance N (2012) Spotting fake reviewer groups in consumer reviews. In: Proceedings of the 21st International conference on world wide web, pp 191–200
72. Mukherjee A, Kumar A, Liu B, Wang J, Hsu M, Castellanos M, Ghosh R (2013a) Spotting opinion spammers using behavioral footprints. In: Proceedings of the 19th ACM SIGKDD International conference on knowledge discovery and data mining, pp 632–640
73. Mukherjee A, Venkataraman V, Liu B, Glance N et al (2013b) Fake review detection: Classification and analysis of real and pseudo reviews. UIC-CS-03-2013. Technical Report
74. Mukherjee A, Venkataraman V, Liu B, Glance NS (2013c) What yelp fake review filter might be doing? ICWSM 2013:409–418
75. Narayan R, Rout JK, Jena SK (2018) Review spam detection using opinion mining. In: Progress in intelligent computing techniques: theory, practice, and applications. Springer, pp 273–279
76. Navastara DA, Zaqiyah AA, Fatichah C (2019) Opinion spam detection in product reviews using self-training semi-supervised learning approach. In: 2019 International conference on advanced mechatronics, intelligent manufacture and industrial automation (ICAMIMIA), pp 169–173
77. Noekhah S, Fouladfar E, Salim N, Ghorashi SH, Hozhabri AA (2014) A novel approach for opinion spam detection in e-commerce. In: Proceedings of the 8th IEEE International conference on E-commerce with focus on E-trust
78. Ott M, Choi Y, Cardie C, Hancock JT (2011) Finding deceptive opinion spam by any stretch of the imagination. arXiv:1107.4557
79. Ott M, Cardie C, Hancock JT (2013) Negative deceptive opinion spam. In: Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: human language technologies, pp 497–501
80. Pandey AC, Rajpoot DS (2019) Spam review detection using spiral cuckoo search clustering method. Evol Intel 12(2):147–164
81. Peng Q (2013) Store review spammer detection based on review relationship, Springer
82. Pennebaker JW, Francis ME, Booth RJ (2001) Linguistic inquiry and word count: Liwc 2001. Mahway: Lawrence Erlbaum Associates 71(2001):2001
83. Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
84. Rajamohana SP, Umamaheswari K (2017) Hybrid optimization algorithm of improved binary particle swarm optimization (ibpso) and cuckoo search for review spam detection. In: Proceedings of the 9th International conference on machine learning and computing, pp 238–242
85. Rayana S, Akoglu L (2015) Collective opinion spam detection: Bridging review networks and metadata. In: Proceedings of the 21th acm sigkdd International conference on knowledge discovery and data mining, pp 985–994
86. Ren Y, Ji D (2019) Learning to detect deceptive opinion A survey spam. IEEE Access 7:42934–42945
87. Ren Y, Zhang Y (2016) Deceptive opinion spam detection using neural network. In: Proceedings of COLING 2016, the 26th International conference on computational linguistics: technical papers, pp 140–150
88. Ren Y, Ji D, Zhang H (2014) Positive unlabeled learning for deceptive reviews detection. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 488–498
89. Ren Y, Ji D, Yin L, Zhang H (2015) Finding deceptive opinion spam by correcting the mislabeled instances. Chin J Electron 24(1):52–57
90. Rout JK, Dalmia A, Choo K-KR, Bakshi S, Jena SK (2017) Revisiting semi-supervised learning for online deceptive review detection. IEEE Access 5:1319–1327
91. Runa D, Zhang X, Zhai Y (2017) Try to find fake reviews with semantic and relational discovery. In: 2017 13th International conference on semantics, knowledge and grids (SKG). IEEE, pp 234–239
92. Saeed RMK, Rady S, Gharib TF (2019) An ensemble approach for spam detection in arabic opinion texts. Journal of King Saud University-Computer and Information Sciences
93. Saini M, Verma S, Sharan A (2019) Multi-view ensemble learning using rough set based feature ranking for opinion spam detection. In: Advances in computer communication and computational sciences. Springer, pp 3–12
94. Sanjay KS, Danti A (2019) Detection of fake opinions on online products using decision tree and information gain. In: 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC). IEEE, pp 372–375
95. Saumya S, Singh JP (2018) Detection of spam reviews A sentiment analysis approach. Csi Transactions on ICT 6(2):137–148
96. Saumya S, Singh JP (2020) Spam review detection using lstm autoencoder: an unsupervised approach. Electron Commer Res, 1–21
97. Savage D, Zhang X, Xinghuo Y, Chou P, Wang Q (2015) Detection of opinion spam based on anomalous rating deviation. Expert Syst Appl 42(22):8650–8657
98. Shehnepoor S, Salehi M, Farahbakhsh R, Netspam NC (2017) A network-based spam detection framework for reviews in online social media. IEEE Trans Inform Forensic Secur 12(7):1585–1595

99. Shojaee S, Murad MAA, Azman AB, Sharef NM, Nadali S (2013) Detecting deceptive reviews using lexical and syntactic features. In: 2013 13th International conference on Intellient systems design and applications. IEEE, pp 53–58
100. Singh VP, Maurya AK (2021) Role of machine learning and texture features for the diagnosis of laryngeal cancer. Machine Learning for Healthcare Applications, p 353
101. Stanton G, Irissappane AA (2019)
102. Sun C, Qiaolin DU, Tian G (2016) Exploiting product related review features for fake review detection. Math Probl Eng, 16
103. Tang X, Qian T, You Z (2020) Generating behavior features for cold-start spam review detection with adversarial learning Information Sciences
104. Tian Y, Mirzabagheri M, Tirandazi P, Bamakan SMH (2020) A non-convex semi-supervised approach to opinion spam detection by ramp-one class svm. Inform Process Manag 57(6):102381
105. Tsironi E, Barros P, Weber C, Wermter S (2017) An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition. Neurocomputing 268:76–86
106. Vidanagama DU, Silva TP, Karunananda AS (2020) Deceptive consumer review detection: a survey. Artif Intell Rev 53(2):1323–1352
107. Wang C-C, Day M-Y, Chen C-C, Liou J-W (2018a) Detecting spamming reviews using long short-term memory recurrent neural network framework. In: Proceedings of the 2nd International conference on E-commerce, E-Business and E-Government, pp 16–20
108. Wang G, Xie S, Liu B, Philip SY (2011) Review graph based online store review spammer detection. In: 2011 IEEE 11Th International conference on data mining. IEEE, pp 1242–1247
109. Wang X, Zhang X, Jiang C, Liu H (2018b) Identification of fake reviews using semantic and behavioral features. In: 2018 4th International conference on information management (ICIM). IEEE, pp 92–97
110. Wang X, Liu K, Zhao J (2017a) Detecting deceptive review spam via attention-based neural networks. In: National CCF conference on natural language processing and chinese computing. Springer, pp 866–876
111. Wang X, Liu K, Zhao J (2017b) spam detection by jointly embedding texts and behaviors. In: Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers), pp 366–376
112. Wang Y, Wang J, Yao T (2019a) What makes a helpful online review?a meta-analysis of review characteristics. Electronic Commerce Research 19(2):257–284
113. Wang Y, Zuo W, Wang Y (2019b) Research on opinion spam detection by time series anomaly detection. In: International conference on artificial intelligence and security. Springer, pp 182–193
114. Wang Z, Hou T, Li Z, Song D (2015) Spotting fake reviewers using product review graph. J Comput Inform Syst 11(16):5759–5767
115. Wang Z, Hou T, Song D, Li Z, Kong T (2016) Detecting review spammer groups via bipartite graph projection. Comput J 59(6):861–874
116. Wang Z, Gu S, Xu X (2018c) Gslda: Lda-based group spamming detection in product reviews. Appl Intell 48(9):3094–3107
117. Wang Z, Gu S, Zhao X, Xu X (2018d) Graph-based review spammer group detection. Knowl Inform Syst 55(3):571–597
118. Wang Z, Hu R, Chen Q, Gao P, Xu X (2019c) Collueagle: Collusive review spammer detection using markov random fields. arXiv:1911.01690
119. Wang Z, Runlong H, Chen Q, Gao P, Xiaowei X (2020) Collueagle: collusive review spammer detection using markov random fields. Data Min Knowl Disc 34:1621–1641
120. Wang Z, Wei W, Mao X-L, Guo G, Zhou P, Jiang S (2021) User-based network embedding for opinion spammer detection Pattern Recognition, p 108512
121. Guangyu W, Greene D, Smyth B, Cunningham P (2010) Distortion as a validation criterion in the identification of suspicious reviews, pp 10–13
122. Wu X, Dong Y, Tao J, Huang C, Chawla NV (2017) Reliable fake review detection via modeling temporal and behavioral patterns. In: 2017 IEEE International conference on big data (big data). IEEE, pp 494–499
123. Wu Z, Wang Y, Wang Y, Wu J, Cao J, Zhang L (2015) Spammers detection from product reviews: a hybrid model. In: 2015 IEEE International conference on data mining. IEEE, pp 1039–1044
124. Wu Z, Jie C, Wang Y, Wang Y, Zhang L, Wu J (2018) hpsd: A hybrid pu-learning-based spammer detection model for product reviews. IEEE transactions on cybernetics
125. Xie S, Wang G, Lin S, Yu PS (2012) Review spam detection via temporal pattern discovery. In: Proceedings of the 18th ACM SIGKDD International conference on knowledge discovery and data mining, pp 823–831
126. Xie S, Hu Q, Zhang J, Philip SY (2015) Economic bi-level approach to ranking and rating spam detection. In: An effective IEEE International conference on data science and advanced analytics (DSAA), p 2015
127. Xu C, Zhang J (2015) Towards collusive fraud detection in online reviews. In: IEEE International conference on data mining. IEEE, p 2015
128. Chang X, Zhang J, Chang K, Long C (2013) Uncovering collusive spammers in chinese review websites, pp 979–988
129. Xu G, Hu M, Ma C, Daneshmand M (2019) Gscpm: Cpm-based group spamming detection in online product reviews. In: ICC 2019-2019 IEEE International Conference on Communications (ICC). IEEE, pp 1–6
130. Xu G, Hu M, Ma C (2021) Secure and smart autonomous multi-robot systems for opinion spammer detection. Inf Sci 576:681–693
131. Yang X (2015) One methodology for spam review detection based on review coherence metrics. In: Proceedings of International conference on intelligent computing and internet of things. IEEE, p 2015
132. Yang Y, Shafiq MO (2018) Large scale and parallel sentiment analysis based on label propagation in twitter data. In: 2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (trustcom/bigdataSE). IEEE, pp 1791–1798
133. Yilmaz CM, Durahim AO (2018) Spr2ep: a semi-supervised spam review detection framework. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, pp 306–313
134. You L, Peng Q, Xiong Z, He D, Qiu M, Zhang X (2020) Integrating aspect analysis and local outlier factor for intelligent review spam detection. Futur Gener Comput Syst 102:163–172
135. Yuan C, Zhou W, Ma Q, Lv S, Han J, Hu S (2019) Product level information for spam detection. In: 2019 Learning review representations from user IEEE International Conference on Data Mining (ICDM). IEEE, pp 1444–1449
136. Zhang W, Bu C, Yoshida T, Zhang S (2016) Cospa: A co-training approach for spam review identification with support vector machine. Information 7(1):12

137. Zhang W, Yuhang D, Yoshida T, Dri-rcnn QW (2018) An approach to deceptive review identification using recurrent convolutional neural network. Inform Process Manag 54(4):576–592

138. Zhang W, Hua X, Wan W (2012) Weakness finder: Find product weakness from chinese reviews by using aspects based sentiment analysis. Expert Syst Appl 39(11):10283–10291

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

Sushil Kumar Maurya is currently pursuing a Ph.D. degree in Computer Science & Engineering with Motilal Nehru National Institute of Technology (MNNIT) Allahabad, India. He received the M.Tech. degree in Computer Science & Engineering from the Uttarakhand Technical University, Dehradun, India, and a bachelor's degree from the Uttar Pradesh Technical University Lucknow, India. His research interests include Machine Learning, Data Mining, and Sentiment & Opinion Analysis.

Dinesh Singh is currently an Assistant Professor in the Department of CSE, Motilal Nehru National Institute of Technology (MNNIT) Allahabad, India. He received his Ph.D. degree from MNNIT Allahabad, a Master's degree from the Indian Institute of Technology (IIT) Roorkee, India, and a Bachelor's degree from the Uttar Pradesh Technical University Lucknow, India. He has authored more than fifteen research papers in national/international conferences refereed journals. Dr. Singh's main areas of interest are Vehicular ad-hoc Network, Network Security, and Machine Learning.

Ashish Kumar Maurya is currently an Assistant Professor in the Department of CSE, Motilal Nehru National Institute of Technology (MNNIT) Allahabad, India. He received his Ph.D. degree from IIT BHU, India, a Master's degree from the Indian Institute of Technology (IIT) Roorkee, India, and a Bachelor's degree from the Uttar Pradesh Technical University Lucknow, India. He has authored more than ten research papers in national/international conferences refereed journals. Dr. Maurya's main areas of interest are Parallel & Distributed Computing, Cloud Computing, Design & Analysis of Algorithms, and Machine Learning.