Hindawi
Complexity
Volume 2018, Article ID 5321280, 12 pages
https://doi.org/10.1155/2018/5321280

WILEY | Hindawi

Research Article

Research of Deceptive Review Detection Based on Target Product Identification and Metapath Feature Weight Calculation

Ling Yuan, Dan Li, Shikang Wei, and Mingli Wang

School of Computer Science, Huazhong University of Science and Technology, Wuhan 430074, China

Correspondence should be addressed to Dan Li; lidanhust@hust.edu.cn

Received 28 December 2017; Revised 20 March 2018; Accepted 10 April 2018; Published 11 June 2018

Academic Editor: Xiuzhen Zhang

It is widespread that the consumers browse relevant reviews for reference before purchasing the products when online shopping. Some stores or users may write deceptive reviews to mislead consumers into making risky purchase decisions. Existing methods of deceptive review detection did not consider the valid product review sets and classification probability of feature weights. In this research, we propose a deceptive review detection algorithm based on the target product identification and the calculation of the Metapath feature weight, noted as *TM-DRD*. The review dataset of target product is modeled as a heterogeneous review information network with the feature nodes. The classification method of graph is used to detect the deceptive reviews, which can improve the efficiency and accuracy of deceptive review detection due to the sparsity, imbalance of deceptive reviews, and the absence of category probability of feature weight calculation. The *TM-DRD* algorithm we proposed is validated on the real review dataset *Yelp* and compared with the *SpEagle*, *NFC*, and *NetSpam* algorithm. The experiment results demonstrate that the *TM-DRD* algorithm performs better than the other method with regard to the accuracy and efficiency.

1. Introduction

With the rapid development of E-commerce, traditional concepts and methods of consumption are rapidly changing. People are increasingly inclined to consume online because it is simpler, faster, and more convenient. Many shopping sites or platforms offer their own online review platforms, such as *Yelp* and *Amazon*, allowing consumers to comment on products.

Product reviews are widely used in individuals and organizations. A survey by Cone, Inc. (http://www.conecomm.com/contentmgr/showdetails.php/id/4008), states that 67% of consumers will read the relevant comments before purchase, where 82% of these consumers conclude that product reviews will affect their final purchase decisions and about 80% of them will change their purchase intentions after reading negative reviews. Evaluation of the products or services quality will directly affect the buying behavior. If a product has a lot of praise, the user will show a greater tendency to purchase. Deceptive detection and prevention are complicated by lack of standard online deception detection, a computationally efficient method for detecting deception in large online communities, and social media developers looking to prevent deception [1]. The deceptive reviews are fake reviews deliberately posted by a few illegal users. The reviews websites or platforms become the target of these deceptive users. Deceptive reviews control the viewpoint of target products and mislead consumers.

In recent years, there have been a large number of effective methods for detecting deceptive reviews [2], but there are still some problems to be solved in this field.

(1) Method Based on the Review Texts. The feature extraction of such methods has serious reliance on the field of review data. The scalability of the model is poor. Moreover, for different fields of the review data, the dataset needs to be regained and marked, while the deceptive review dataset is difficult to obtain. It has also become a major issue for deceptive review detection based on the review texts.

(2) Method Based on Abnormal Behavior. The main drawback of this kind of method is that most reviewers do not have the relevant information to conduct behavioral analysis, which results in limited ability to identify abnormal behavior. What is more, the professional deceptive users are good at hiding their abnormal behavior, making their behavior similar to the normal users.

In order to improve the efficiency and accuracy of deceptive review detection, this paper proposes a deceptive review detection algorithm based on the target product identification and the calculation of the metapath feature weight, noted as *TM-DRD,* involving two research contents.

(1) In order to identify the target product of deceptive review, we propose a method based on abnormal score, noted as *AS-TPI*. Firstly, we analyze the different states of deceptive reviews and then calculate the difference between the actual product rating scale and the standard score ratio. Finally, the distribution of the score in time is estimated by using the kernel density.

(2) We define the features separately from the reviews and reviewers, combine the target products and related review datasets identified by *AS-TPI*, and then construct the heterogeneous review information networks. We propose a method to calculate feature weights based on the metapath to calculate the deceptive degree probability of reviews to determine the final category of reviews, noted as *MFW-DDP*.

The related work is described in the Section 2. The prelimialires for the proposed *TM-DRD* algorihm are illustrated in Section 3. The proposed methodology is presented in Section 4. The experiments about the proposed algorithm are illustrated in Section 5. Section 6 concludes the whole paper.

2. Related Work

There are two directions of the current research on the deceptive review detection [3–5]: one is based on the reviews, and the other is based on the reviewers. For these two directions, there are the following research methods.

(1) Method Based on the Content of Reviews. The method detects the deceptive reviews based on the similarities and linguistic features of the reviews. It extracts relevant features from features of vocabulary, consistency of content, consistency of review style, and semantic consistency to identify deceptive reviews. By analyzing the tendencies of sentiment, semantics, we can find the deceptive reviews deviating from the normal reviews.

Ott et al. [6] used crowdsourcing platform (AMT) to construct datasets and used comprehension method of natural language to acquire linguistic features from multiple perspectives. They trained many kinds of classifiers and compared their performance. But the test results were not very well on real business datasets. Li et al. [7] created deceptive reviews datasets manually and used naive Bayesian machine learning algorithm for deceptive reviews detection. A two-sided cotraining semisupervised learning method was proposed to mark a large number of unlabelled reviews. And they used it as follow-up deceptive reviews test datasets. Rout et al. [8] also used semisupervised learning approaches to improve the F-score metric in classification, and they incorporated new dimensions in the feature vector to obtain better results. Feng et al. [9, 10] proved that deep syntactic information of texts is very effective in deceptive reviews detection. They used probabilistic context-free syntax PCFG. The deep syntactic features of the reviews texts are generated by the generative rules of the PCFG syntax analysis tree and the SVM classifier is trained to identify the deceptive reviews. Li et al. [11] proposed a method of deceptive detection based on the LDA model named as TopicSpam, which can classify the reviews by detecting the probability of the deceptive index by detecting the slight difference between the distribution of the keywords of the real reviews and the deceptive reviews.

Due to the concealment, the behaviors of reviewers who publish deceptive reviews are getting closer and similar to those of normal users, and deceptive strategies they use are also getting better and more diversified.

(2) Method Based on Behavior. In this method, most of the features are extracted based on the metadata of the reviews (time of reviews, frequency of reviews, information of the first reviewers of the product, etc.), such as the research of [12–14]. They analyze the temporal or spatial information of reviews. If conditions permit, they can also use some privacy data of the site such as IP address, MAC address, and location reviews published, which are very useful to extract behavioral features. Then they mathematicize the features, construct user behavior models, and classify reviews by models.

Lim et al. [15] focused on the behavior of reviewers to find the deceptive reviews. They considered that it was better to study reviewers than reviews because the information obtained from the reviewers' behavior was far more than the information obtained from the reviews themselves. So they proposed a method to detect the deceptive reviewers based on the score of reviewers. They constructed a model from the multifaceted behaviors of reviewers, and designed a deceptive degree scoring function to calculate whether the reviewers are deceptive. Xie et al. [16] proposed a multi-time scale detection method and found time windows that concentratedly distributed deceptive reviews through time series analysis. They considered that the singleton review in such time windows is highly likely to be deceptive, where singleton review means that the reviewer of the review posted only this one review. Their method that makes use of features such as the release date of the review and the historical record of the reviewer is an unsupervised learning method. Mukherjee et al. [17] proposed an unsupervised model of hypothetical reviewers named ASM. They considered the distribution of different behaviors of deceptive reviewers and normal reviewers and set falsehood as an implicit variable and reviewers' behavior as an observation variable. They used a clustering algorithm to identify fake reviewers to identify deceptive reviews. Zhang and Lu [18] investigated the top Weibo accounts whose follower lists duplicate or nearly duplicate each other (hereafter called near-duplicates) and proposed a novel fake account detection method that is based on the very purpose of the existence of these accounts: they are created to follow their targets en masse, resulting in high-overlapping between the follower lists of their customers. The implementation is based on the estimation of Jaccard similarity

using random sampling. Unlike traditional fast algorithms for Jaccard similarity, they estimated the Jaccard similarities without the access to the entire data.

Compared with the method based on the content of the reviews, the behavior-based approach analyzes the characteristics of cheating behaviors from different perspectives and does not require a lot of textual analysis such as viewpoint mining and sentiment analysis. At present, deceptive reviews detection methods based on user behavior are analyzed from several common cheating behaviors. With the constant change of behavior of deceptive reviewers, new cheating behaviors need to be further extracted and analyzed to improve detection accuracy.

(3) Method Based on the Relationship. The method builds a relational model by studying the complex relationships among reviewers, critics, products, or stores. It uses the associations or some graph-based methods in the diagram to sort the reviews or mark the categories, with establishing a network diagram of relationships among the three.

Wang et al. [19] considered that it was not enough to only use behavior-based heuristic rules. Therefore, for the first time, a graph-based approach is proposed to detect the deceptive reviewers. This method can detect cheating behaviors that some original detection methods cannot detect. Li et al. [20] used a vector representation of products and reviewers related to reviews through the tensor decomposition method and combined it with the feature of bag bigram and then used SVM to detect the deceptive review. In their method, all reviewers and products related to reviews are characterized by a matrix, and then the tensor decomposition technique is used to translate each user and product into a corresponding vector representation. The advantage of this method is the vectorization of the global features, effectively improving the detection performance. There have been a large number of the deceptive reviewers who often work collaboratively to promote or demote target products, which severely harm the review system [21, 22]. Xu et al. [21] proposed a KNN-based approach based on the similarity of reviewers and the relevance of reviewer groups. They proposed a graph model of collusion reviewer based on Pairwise Markov Network, which was used to infer the classification of critics. Fei et al. [23] found that the reviewers and reviews appearing in sudden periods often showed the trend that the deceptive reviewers cooperate with each other and real reviewers are usually presented together. They established Markov random MRF network model for critics who appeared in different periods of emergency and proposed an evaluation method to evaluate the inference results. Their method has higher accuracy and recall rate for burst reviews detection. In the case of deceptive reviewers groups, Wang et al. [22] introduced a top-down computing framework to detect the deceptive reviewers groups by exploiting the topological structure of the underlying reviewer graph which reveals the coreview collusiveness. A novel instantiation is designed by modeling deceptive reviewers groups as biconnected graphs. Ye and Akoglu [24] proposed a two-stage approach to identify the deceptive reviewer groups and target products of deceptive reviews that they attack. They used GroupStrainer and a hash-clustering algorithm based on similarity in the graph model to detect the deceptive reviewer groups. For big reviews dataset, Dhingra and Yadav [25] proposed a novel fuzzy modeling based solution to the problem and defined novel FSL deduction algorithm generating 81 fuzzy rules and Fuzzy Ranking Evaluation Algorithm (FREA) to determine the extent to which a group is suspicious and used Hadoop for storage and analyzation.

3. Preliminaries

3.1. Product Rating Difference Calculation. The original review dataset is statistically processed in the product scoring stage to obtain each product and its corresponding scoring dataset. Then it is used as input to a target product recognition algorithm based on the differences in the grade scoring.

In order to describe the target product identification algorithm based on the difference of the grade scores, we present two assumptions and the definitions of related concepts used in the algorithm.

Definition 1 (score distribution, D_p). Each product p corresponds to a score distribution $D_p = \{n_i, 1 \le i \le 5\}$,where n_i indicates the number of reviews with score i, as shown in

$$D_p = \{n_1, n_2, n_3, n_4, n_5\}. \tag{1}$$

For example, there are 10 reviews of product p with 1 point, 20 reviews with 2 points, 30 reviews with 3 points, 40 reviews of with 4 points, and 50 reviews with 5 points. The score distribution of product p is $\{10, 20, 30, 40, 50\}$.

Definition 2 (rating scale, $R_{p,i}$). Given a product p and a rating level i, $i \in [1, 5]$, we gather the reviewers set $R_{p,i}$ of the product p rating for i. For $\forall r \in R_{p,i}$, the proportion $s_{p,i}$ of the reviews with rating i is defined as the product rating scale, as shown in (2). The value range is $[0, 1]$.

$$s_{p,i} = \frac{\left|\left\{v_{r,p} \mid e_v = i\right\}\right|}{\left|\left\{v_{r,p}\right\}\right|}, \tag{2}$$

where $v_{r,p}$ is the review of reviewer r on product p and e_v is the score associated with review v.

The ratio range $[0, 1]$ is divided into 10 equidistant intervals, and the proportion corresponding to each equidistant interval in turn is $\varphi_1 = 10\%, \varphi_2 = 20\%, \ldots, \varphi_{10} = 100\%$. The distribution of the score i of the product p in proportion is shown in

$$D_{p,i} = \left\{m_{i,j},\ 1 \le i \le 5,\ 1 \le j \le 10\right\}, \tag{3}$$

where $m_{i,j}$ is the number of ratings. The proportion of i-level reviews falls within the range of $[\varphi_{j-1}, \varphi_j)$.

Definition 3 (standard rating scale, s_i). For all the products with a rating of i, $i \in [1, 5]$, we calculate the proportion s_i of reviews for all reviews with a rating of i. s_i is defined as the

standard rating scale. The range and division criteria for s_i are similar as above. Standard rating scale is defined as shown in

$$s_i = \frac{\left|\{v \mid e_v = i\}\right|}{|\{v\}|}, \tag{4}$$

where v is any review and e_v is the rating of the v.

We can calculate the proportional distribution of the number of scores for all products rated as i, defining it as the Standard Rating Scale distribution, as shown in

$$SD_i = \left\{sm_{i,j},\ 1 \le i \le 5,\ 1 \le j \le 10\right\}. \tag{5}$$

Definition 4 (rating scale difference, $DIF_{p,i}$). The rating scale difference is the difference between the product rating scale and the standard rating scale. The rating scale difference in grade i on product p is defined as shown in (8).

$$S_i = \sum_{j=1}^{10} m_{i,j}, \tag{6}$$

$$SS_i = \sum_{j=1}^{10} sm_{i,j}, \tag{7}$$

$$DIF_{p,i} = \sum_{j=1}^{10} \left|\frac{m_{i,j}}{S_i} - \frac{sm_{i,j}}{SS_i}\right|. \tag{8}$$

Assumption 5. The criteria for a normal reviewer are fixed; that is, the same rating scale indicates the same tendencies to reviews on all products in its review, so the distribution of normal product ratings amount (the number of reviews or the number of reviewers) on each level should be consistent with a certain law.

Assumption 6. According to the majority voting principle, it is assumed that if there are three or more $DIF_{p,i}$ fallings within the range of nonconfidence intervals, the product is the target product.

3.2. Target Product Identification. The products involved in the real review data set are mainly divided into the following three groups:

(1) Type one: such products are usually not popular products with a very small number of reviews. Their sales and commentary information are relatively small, such as products in some small self-employed stores. The impact of reviews for such products is small.

(2) Type two: such products are usually popular products with a very large number of reviews but a very small number of deceptive reviews. These products generally come from shops with high reputation and high recognition, such as Tmall's official flagship store. The most reviews of these products are real reviews and therefore it is not enough to mislead consumers about the purchase decision.

(3) Type three: such products are defined as target product. They are usually popular products with a very large number of reviews and a very large number of deceptive reviews. It is not easy to tell whether the review is deceptive or not. It is easy to mislead consumers to make objective and correct judgments about the products and make risky purchase decisions. What is more serious is disruption of the equitable and orderly nature of the E-commerce market. Therefore, it is of significance to conduct in-depth analysis and research on this type of products and related reviews. Target products identification with research significance from the mass data can reduce the scope of the review data involved, and the detection efficiency and accuracy can all be improved.

After identifying the target product in the original product scoring dataset, the remaining unidentified product and its scoring dataset are used as the input of the target product identification algorithm based on the kernel density estimation in this section to identify the target product.

For a target product that is staged attacked by a large number of deceptive reviews, the sudden increase of the number of good reviews or bad reviews in some time windows leads to the sudden increase or decrease of the average rating of the products, so that the average scores and the number of reviews show a positive or negative correlation.

Since the probability density curve estimated by the kernel density is smoother, we consider the review published time as the sample point for the density function curve estimation. Since the probability density function of the kernel density estimated by the smoothed kernel is also smooth, we can use the Gaussian kernel function here, as shown in

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}. \tag{9}$$

Definition 7 (product review sequence V_p). The product review sequence $V_p = \{v_1, v_2, \ldots, v_m\}$ is all the reviews of the product p, which are sorted in turn by review published time, where m is the total number of reviews of the product p, v_i is the ith reviews of the product p, and the range of i is $[1, m]$.

Definition 8 (product review time sequence T_p). The product review time sequence is $T_p = \{t_1, t_2, \ldots, t_m\}$, where t_i is the time when the ith review is published.

Definition 9 (product time window I_i). The product time window is a time interval of a review. The time window is defined as shown in

$$I_i = \left(a_{i-1}, a_i\right], \quad a_i = i * \Delta t,\ 1 \le i \le k, \tag{10}$$

where Δt is the size of specified time window, $T = t_m - t_1$ is the length of time, k is the number of time windows, $k = T/\Delta t = (t_m - t_1)/\Delta t$, a_{i-1} is the left boundary of time window I_i, and a_i is the right boundary.

Definition 10 (time window review collection H_i). The time window review collection refers to the review collection

whose published time falls within a certain time window, and it is defined as shown in

$$H_i = \left\{ v_j \mid t_j \in (a_{i-1}, a_i], \ i \in [1, k] \right\}, \tag{11}$$

where v_j is the jth review of the product, and the corresponding publication time is t_j.

3.3. Metapath Feature Selection. The identified product-related review datasets are modeled as a heterogeneous review information network with feature nodes. In order to reflect the final impact of feature weight on the probability of deceptive review, the feature weight calculation algorithm is introduced into the calculation of the probability of the final deceptive degree of the review.

Definition 11 (heterogeneous information network G). A heterogeneous information network is a graph containing a types of nodes and b types of edges ($a > 1$ or $b > 1$), defined as $G = (N, E)$, where N is a set of all types of nodes and E is a collection of all types of edge. Any $v \in N$ or $\varepsilon \in E$ belongs to a particular type.

Definition 12 (network mode T_G). Given a heterogeneous information network graph $G = (N, E)$, we obtain a network pattern graph $T_G = (A, \Gamma)$, in which there exists a mapping relationship from heterogeneous information networks to network patterns $G = (N, E) \rightarrow T_G(A, \Gamma)$, involving the mapping relationship $\tau : N \rightarrow A$ and mapping relationship $\phi : E \rightarrow \Gamma$. The network pattern $T_G = (A, \Gamma)$ is a graph defined on a collection A of node types and a collection Γ of associated types that describes a new graph structure.

Definition 13 (metapath). The metapath is a path P in the network pattern diagram $T_G = (A, \Gamma)$. The corresponding metapaths of the two nodes A_1 and A_n in T_G are denoted as $A_1(\Gamma_1)A_2(\Gamma_2)\cdots A_{n-1}(\Gamma_{n-1})A_n$. The metapath extends the definition of associations to describe the association between two types of nodes that are not directly connected.

The features extracted from the research on the deceptive reviews are classified into three categories: related features of the review contents, relevant features of the reviewers, and related features of the network resources. The symbolic representations of related concepts and their meanings are illustrated in Table 1.

Features of the reviews include the following: the content features of review, the viewpoints features of review, and the metadata features of review. It is impossible to effectively distinguish the deceptive reviews from normal reviews simply by the features of language semantics, such as content features and viewpoints features, because the deceptive reviewers can mimic normal users' behavior so that they are not easily discoverable. Thus, more effective related features of reviewers are needed. The reviewer related features could be as follows: the feature of the reviewer and the feature of the reviewer's behavior.

With the comparative analysis, all the extracted features are classified according to four division strategies: the reviewers based on the behavior or semantic and the reviews based on the behavior or semantic. Table 2 shows the distribution of these features of reviews and reviewers.

Table 1: Symbol definition table.

Symbol	Definition
r	Reviewer
V_r	The collection of all the reviews published by Reviewer r
v	Review
v_i	The ith review
$V_{r,i}$	The collection of all the reviews published by Reviewer r on the ith day
e_v	The score of review v
$e_{r,p}$	The score of reviewer r on the product p
E_p	The collection of all the rating scores on the product p

Table 2: Features extraction in different strategy.

Features	Reviewers	Reviews
Based on behavior	MNRD RPR RNR BST ERD BDS RD RWR	RRD ETF
Based on semantic	ACS	RPP ROW

MNRD: max number of reviews daily, RPR: ratio of positive reviews, RNR: ratio of negative reviews, BST: burstiness, ERD: entropy of ratings distribution, BDS: brand deviation score, RD: rating deviation, RWR: ratio of weekend reviews, RRD: review rating deviation, ETF: early time frame, ACS: average content similarity, RPP: ratio of 1st and 2nd person pronouns, and ROW: ratio of objective words.

As the range of different features is inconsistent, which brings inconvenience to the measurement of the index, the above features need to be normalized, and the range of each feature is set to be limited to $[0, 1]$. The larger or smaller the value of different features indicates the abnormal performance.

Theoretically there are infinite examples of metapaths in the network, but we can abandon long metapath instances by selecting the path length [26]. According to the small-world phenomenon [27] and the third-degree influence theory [28], it can be inferred that the metapath with a length greater than 3 reflects a very weak association, so we can consider only the metapath whose path length is not greater than 3. Therefore we select the metapaths as shown in Table 3.

4. Our Method

The research on deceptive review detection has mainly focused on improving the accuracy of the results without considering the validity of the test objects. Therefore, we propose a deceptive review detection algorithm based on the target product identification and the metapath feature weight

Table 3: Metapath results.

Symbol	Definition
V - V (RPP)	The reviews with the same ratio of 1st and 2nd person pronouns.
V – V (ROW)	The reviews with the same ratio of objective words
V - V (RRD)	The reviews with the same review rating deviation
V - V (ETF)	The reviews with the same early time frame
V - R - R - V (ACS)	The reviews published by the reviewers with the same average content similarity
V - R - R - V (MNRD)	The reviews published by the reviewers with the same max number of reviews daily
V - R - R - V (RPR)	The reviews published by the reviewers with the same ratio of positive reviews
V - R - R - V (RNR)	The reviews published by the reviewers with the same ratio of negative reviews
V - R - R - V (BST)	The reviews published by the reviewers with the same burstiness
V - R - R - V (ERD)	The reviews published by the reviewers with the same entropy of ratings distribution
V - R - R - V (BDS)	The reviews published by the reviewers with the same brand deviation score
V - R - R - V (RD)	The reviews published by the reviewers with the same rating deviation
V - R - R -V (RWR)	The reviews published by the reviewers with the same ratio of weekend reviews

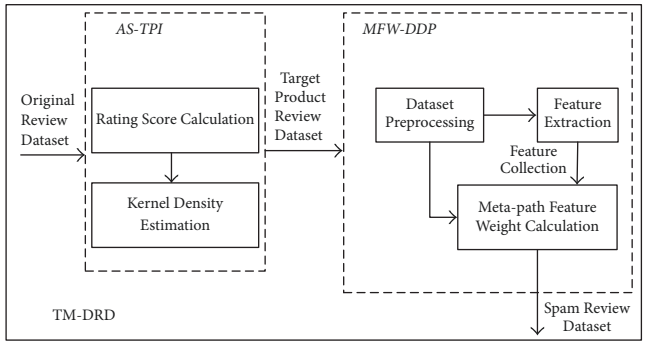

Figure 1: Framework of *TM-DRD*.

calculation *(TM-DRD)* for the valid product review dataset. The overall framework is shown in Figure 1.

4.1. AS-TPI Method. In order to identify the target product of deceptive review, we propose a target product identification method based on abnormal score, noted as *AS-TPI*. The original review dataset is statistically processed in the product scoring stage to obtain each product and its corresponding scoring dataset as input to *AS-TPI*.

AS-TPI is divided into two parts. The first part is based on the rating score calculation, which statically identifies the product for the number distribution of reviews on each rating level. The second part is based on the estimation of the kernel density to analyze the sudden abnormalities of reviews from the time dimension to dynamically identify the products.

Input: Product Set P, Review Set V.
Output: Target Product Set P_t
(1) **for** *each rating score* **do**
(2) *calculate* SD_i
(3) **for** *each product* p **in** P **do**
(4) *calculate* D_p, $D_{p,i}$, $DIF_{p,i}$, μ_i, δ_i
(5) **if** $DIF_{p,i}$ *not* **in** *the confidence interval* **then**
(6) Add($DIF_{p,i}$) to DD_i
(7) Add(DD_i) to DD
(8) **for** *each product* p **in** P **do**
(9) **for** *each rating score* **do**
(10) **if** $DIF_{p,i}$ **in** DD **then**
(11) Count(p)++
(12) **if** Count(p) > 2 **then**
(13) Add(p) to P_t
(14) **return** P_t

Algorithm 1: StaticTargetProductDetection(P, V).

Algorithm 1 is named as *StaticTargetProductDetection*, the number of reviews on each rating level of the product is counted to obtain D_p, then $R_{p,i}$ and s_i, according to the distribution of the number of reviews of the current product with the current rating scale. $DIF_{p,i}$ is calculated by comparing with the result of s_i. According to the Law of Large Numbers, $DIF_{p,i}$ follows a normal distribution. Finally, we set a confidence interval (a significance level) to find the grade difference index that does not satisfy the confidence interval in the normal distribution corresponding to the product grade difference. The pseudocode of static target product detection is shown in Algorithm 1.

In Algorithm 1, lines (2)–(4) calculate the rating score and other related parameters, lines (5)–(7) determine $DIF_{p,i}$ which does not meet the confidence interval, and add to the distribution of differences in the proportion of collection, lines (8)–(13) add p to the suspicious target product set where $DIF_{p,i}$ appear more than two times in the set, and line (14) returns target product set. The time complexity of the algorithm is $O(i * N)$, where i is the rating grades and N is the number of products in the review dataset to be detected.

Algorithm 2 is named as *DynamicTargetProduct-Detection*. In Algorithm 2, review sequence and other related parameters are calculated in lines (2)–(4); lines (5)-(6) calculate the set of extreme points of *KDE* and filter the extreme points and then add the time window which contains the extreme points to candidate burst time window set; lines (9)–(14) calculate the average score of each time window in the set of candidate time windows and then calculate the difference between the average of the ratings and the average of the overall score of the product. If the difference exceeds the threshold, the count of time windows increases by 1, and if count exceeds $k/2$, we add the product to the target product set. Line (15) returns the target product set.

Input: Product Set P, Review Set V.
Output: Target Product Set P_t
(1) **for** *each product* p **in** P **do**
(2) *calculate* V_p, T_p
(3) **for** *each rating score* **do**
(4) *calculate* w_i
(5) *calculate* Xp
(6) Add(Xp) to Xp'
(7) **for** x_{pj} **in** Xp' **do**
(8) Add(x_{pj}) to I_p
(9) **for** I_i **in** I_p **do**
(10) *calculate* $\mu_{p,i}$
(11) **if** $|\mu_{p,i} - \mu_p| > \tau$ **then**
(12) Count(p)++
(13) **if** Count(p) > $k/2$ **then**
(14) Add(p) to P_t
(15) **return** P_t

Algorithm 2: DynamicTargetProductDetection(P).

The time complexity of Algorithm 2 is $O(m * N)$, where m is the maximum among the number of time windows, the number of extreme points, and the number of candidate time windows. N is the number of products in the review dataset to be detected.

4.2. MFW-DDP Method. With the above *AS-TPI* method, we can obtain the target product review dataset. Combining this dataset with a given feature list as input, we propose a method to calculate the metapath based feature weights to calculate the deceptive degree probability of reviews to determine the final category of reviews, noted as *MFW-DDP*. This method is mainly divided into four steps: feature-based prior probability calculation, feature-based network pattern creation, metapath generation, and classification marking.

Step 1 (feature-based prior probability calculation). The following equation is used to calculate the prior probability s_u of deceptive degree and initialize all the review nodes in the information network graph:

$$s_u = \frac{1}{L}\sum_{l=1}^{L} f\left(x_{lu}\right), \tag{12}$$

where $f(x_{lu})$ represents the a priori probability of the deceptive degree of the review u calculated from feature l.

Step 2 (feature-based network pattern creation). Given a set of feature lists F, constructing a heterogeneous review information network graph $G = (N, E)$, and according to graph G, we can obtain the network pattern $T_G = (A, \Gamma)$.

When the list of features is {*ACS, MNRD, RPP, RRD*}, the network pattern is shown in Figure 2. We can figure out that network pattern only contains one different type of node.

Step 3 (metapath generation). As shown in Figure 3, the two dotted lines, respectively, represent the instances of two

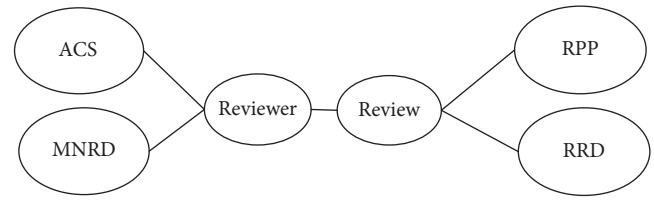

Figure 2: Network pattern based on the feature list {ACS, MNRD, RPP, RRD}.

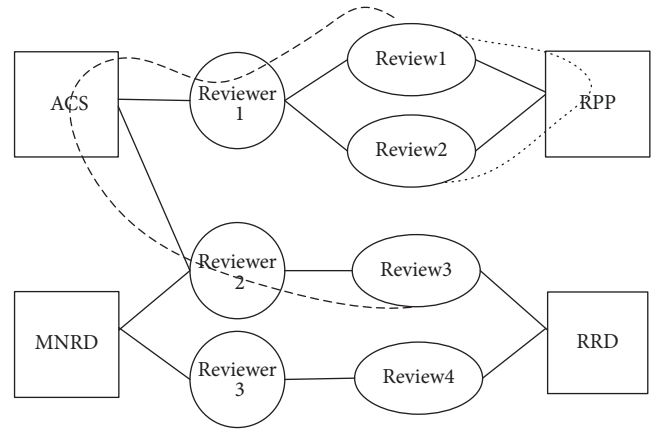

Figure 3: Metapath generation example.

metapaths. If the *Review* node and another *Review* node are associated with the feature *RPP* and their *RPP* values are equal, a metapath is generated, the symbol of which is denoted as *Review-RPP-Review*. If the *Review* node and another *Review* node are associated with the feature *ACS* and their *ACS* values are equal, a metapath is generated with the symbol *Review-Reviewer-ACS-Reviewer-Review*.

Step 4 (classification marking). Classification marking includes two steps: feature weight calculation and classification marking. The weight calculation determines the importance of identifying each feature of the deceptive review. The classification marking calculates the final deceptive probability of each review. To help consumers seek credible information, most current work apply mainly qualitative approaches to investigate the credibility of reviews or reviewers [29]. We adopt the probability of deceptive degree for the review node to quantify the credibility of reviewers.

The weight is calculated as shown in (13). The classification marking is defined as (14). The probability of deceptive degree for the current review node is estimated according to (15).

$$W_{pi} = \frac{\sum_{u=1}^{n}\sum_{v=1}^{n} mp_{u,v}^{pi} \times s_u \times s_v}{\sum_{u=1}^{n}\sum_{v=1}^{n} mp_{u,v}^{pi}}, \tag{13}$$

$$P_{u,v} = 1 - \prod_{i=1}^{L}\left(1 - mp_{u,v}^{pi} \times W_{pi}\right), \tag{14}$$

Input: Review Set V, Reviewer Set R, Feature Set F
Output: Deceptive review degree probability set P, feature weight set W
(1) **for** *each reviews* u **in** V **do**
(2) *calculate* s_u
(3) Define the network pattern *schema*(A, Γ)
(4) **for** $u, v \in V$ **do**
(5) **for** $p_l \in$ *schema* **do**
(6) *calculate* mp_u^{pl}, mp_v^{pl}
(7) **if** $mp_u^{pl} = mp_v^{pl}$ **then**
(8) $mp_{u,v}^{pl} = mp_u^{pl}$
(9) **Add** u, v **to** V'
(10) **for** $p_l \in$ *schema* **do**
(11) calculate w_{pl}
(12) **for** $u, v \in V'$ **do**
(13) *calculate* $P_{u,v}$
(14) *calculate* P_u
(15) **return** P, W

Algorithm 3: *TM-DRD*(V, R, F).

$$P_u = \frac{\sum_{v=1}^{n} P_{u,v}}{n}. \tag{15}$$

According to the above calculation, we can obtain the deceptive probability set P of all the review nodes.

4.3. TM-DRD Algorithm. With the result of target product identification method based on abnormal score *(AS-TPI)* and the calculation method of deceptive degree probability of reviews based on the metapath feature weights *(MFW-DDP)*, we can determine the final category of reviews. Our proposed deceptive review detection algorithm based on the target product identification and the metapath feature weight calculation *(TM-DRD)* is shown in Algorithm 3.

In Algorithm 3, lines (1)-(2) calculate the a priori probability for each review. Line (3) defines the network pattern. Lines (4)–(9) calculate the probability of each feature associated value of the metapath corresponding to two review nodes. The weight of two review nodes associated with each feature is calculated in lines (10)-(11). The probability of the final deceptive degree of the review node is calculated in lines (12)–(14). Line (15) returns the degree probability of deceptive review set and the feature weight set.

The time complexity of Algorithm 3 is $O(|V| * |M|)$, where $|V|$ represents the number of review nodes in the heterogeneous review information network and $|M|$ represents the number of feature sets (constant).

5. Experiment

5.1. Experimental Setup. The experimental environment is described in Table 4. To verify the validity of the proposed algorithm, a real, reliable, and accurate dataset plays a crucial role in the deceptive review detection. Therefore, we try to test on the review datasets in real environment. In the experiment, we use the review datasets *YelpChi* and *YelpNYC* from *Yelp*, a famous travel website, provided by [30]. The *YelpChi* [30] covers about 67,395 reviews, 38,063 reviewers, and 201 products for hotels and restaurants in the Chicago area from October 12, 2004, to October 8, 2012. The *YelpNYC* [30] covers about 359,053 restaurants related reviews, 160,225 reviewers, and 923 products in the New York City area from October 20, 2004, to July 1, 2015. The specific distribution of reviews, production, and reviewers is shown in Table 5. Six attributes extracted for structured processing are saved to the database. The reviews in this dataset contain the deceptive markups (fake or not) of each review. The annotation results are generated with the *Yelp* filtering algorithm [31].

Table 4: Experimental environment table.

Item	Content
CPU	Intel Core i5 3.30 GHz dual-core
RAM	2 GB
Hard disk	500 GB
Operating system	Microsoft Windows 7 32-bit
Development environment	Python 2.7.3
Development tools	Matlab R2014a + Eclipse
Database	MySQL5.6.26

Table 5: The distribution table of reviews, products, and reviewers in *Yelp*.

Dataset	Reviews number	Reviewers number	Products number
YelpChi	67395	38063	201
YelpNYC	359053	160225	923

5.2. Evaluation Index. In order to assess the performance of the target product identification and deceptive review detection methods, we should utilize the accepted assessment methods and evaluation criteria.

We adopt the widely used accuracy as an evaluation index to the behavior of *AS-TPI*. The accuracy λ is defined as the ratio of the number of target products M to the number of suspicious target products N identified by Algorithms 1 and 2, as shown in

$$\lambda = \frac{M}{N} \times 100\%. \tag{16}$$

There are two kinds of evaluation indexes to evaluate the recognition results of the algorithm comprehensively. The *TM-DRD* algorithm would adopt the second one.

The first evaluation index is the classification model evaluation indicators: Precision rate, Recall rate, and accuracy computed from Precision and Recall rate. $F1$ value is the reconciled average of Precision and Recall rate. False positive FPR and true positive TPR rates characterize the recognition accuracy and recognition range. The second evaluation index is the ranking model to evaluate the performance of the algorithm, including the PR curve, the ROC curve, and the area covered by the curve, corresponding to Average Precision (AP) and Area under Curve (AUC), which indicate

the trade-off evaluation index of test results in the Precision and Recall rate, as shown in (17) and (18).

$$\mathrm{AP} = \sum_{i=1}^{n} \frac{i}{I(i)}, \tag{17}$$

where n represents the number of reviews, i represents the position of the review in the sorted set of reviews, and I represents the position set of the review in the sorted set of reviews.

$$\mathrm{AUC} = \sum_{i=2}^{n} (\mathrm{FPR}(i) - \mathrm{FPR}(i-1)) * (\mathrm{TPR}(i)), \tag{18}$$

where FPR(i) represents the false positive rate of the ith review and TPR(i) represents the true positive rate of the ith review.

5.3. Experimental Results and Analysis

5.3.1. Target Product Identification Experiment. The experiment uses the *YelpNYC* [30] review dataset. The purpose of the experiment is to identify the target product attacked by a large number of deceptive reviews. The original dataset is filtered, the threshold value τ is set to 0.3, and the time window is two weeks. The target product identification method based on abnormal score was used for screening to obtain the collection of suspicious target products. We invite 3 online shopping experienced college students as judges to manually evaluate the collection. In order to reduce the impact of subjective factors or other random factors on the evaluation results, we consider the marking results of most evaluators as the final mark according to the voting principle.

Then, a time window Δt is set, and, for each time window size, $\tau \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$, each τ is used as a time window score mean difference parameter in the target product identification algorithm based on the nuclear density estimation. Differentiate the mean score of the review burst time window and calculate the collection of suspicious target products. Then we observe the influence of the change of τ on the target product recognition rate.

The marking results of 3 judges are shown in Table 6. According to the confirmation of the final marker, there are 35 true target products finally determined in the evaluation target products in the experiment; that is, the recognition rate was $\lambda = 35/42 * 100\% = 83.33\%$. It shows that the target product identification method based on abnormal score has high recognition accuracy. The target product-related review collection only accounted for 15.99% of the original review dataset. It shows that a large number of meaningless review data exist in original review dataset. If we detect deceptive review directly, it will lead to the decline in detection efficiency. Therefore, the target product identification method solves the overall sparseness and imbalance of deceptive reviews.

As shown in Figure 4, under the setting of time window size Δt of 7 days and 14 days, respectively, the recognition rate curve decreases with the increase of threshold parameter τ. The recognition rate drops to 0 until $\tau = 0.7$ and then remains unchanged at 0. The target product recognition rate obtains the highest value when $\tau = 0.1$. It is usually short-term behavior that a large number of fake reviews are published by fake reviewers periodically, so when the smaller appropriate value of the time window is set, we can capture burst situation of reviews, so there is higher recognition rate when the time window is set to 7 days.

Table 6: Artificial premark results for the target product.

Judge	The number of premark target products
Judge 1	34
Judge 2	35
Judge 3	34

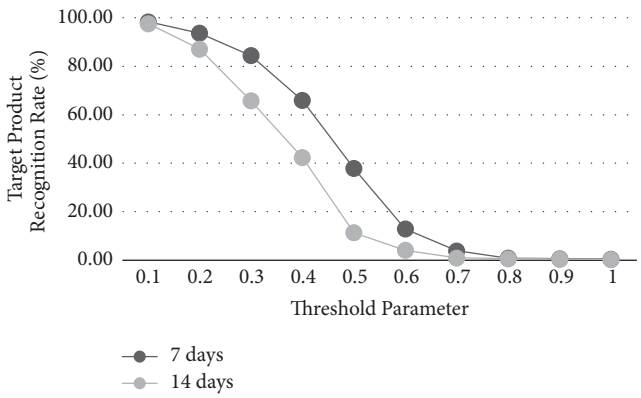

Figure 4: The influence of threshold parameters on the recognition rate of target products.

5.3.2. Comparative Experiment of Deceptive Review Detection Related Methods. The experiments in this section will compare the performance of the *TM-DRD* and the *NFC* [24], *SpEagle* [30], and *NetSpam* [32] on accuracy indices such as AP and AUC. We verify the impact of the target product review dataset and feature weight calculation on the detection efficiency of *TM-DRD* and the accuracy of the test results.

The experiment uses four review datasets: *YelpChi* [30], *YelpChiOP*, *YelpNYC* [30], and *YelpNYCOP*. The datasets *YelpChiOP* and *YelpNYCOP* are, respectively, related review datasets on the target product identified by the fusion algorithm based on the anomaly scores proposed in chapter 4 from the original data sets *YelpChi* and *YelpNYC* [30]. Next, we will compare the performance of *TM-DRD* and *NFC* [24], *SpEagle* [30], and *NetSpam* [32] in AP and AUC, respectively, on the above 4 review datasets. We analyze the impact of feature weights on the accuracy of deceptive review detection.

In order to verify the impact of feature weight on accuracy and find out whether there is a relationship between weight and accuracy, AP index is used here to measure the accuracy. The equation based on ranking difference set is adopted here, as shown in the following:

$$\rho = 1 - \frac{6 \sum_{i=1}^{N} d_i^2}{N(N^2 - 1)}, \tag{19}$$

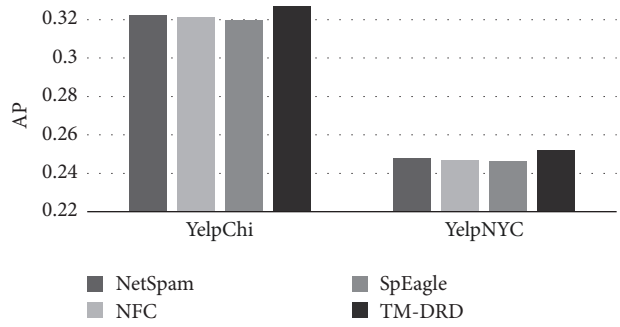

Figure 5: The AP for TM-DRD and SpEagle, NFC, and NetSpam in different datasets.

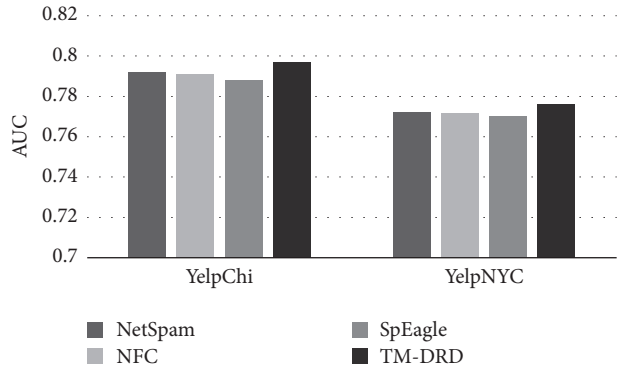

Figure 6: The AUC for TM-DRD and SpEagle, NFC, and NetSpam in different datasets.

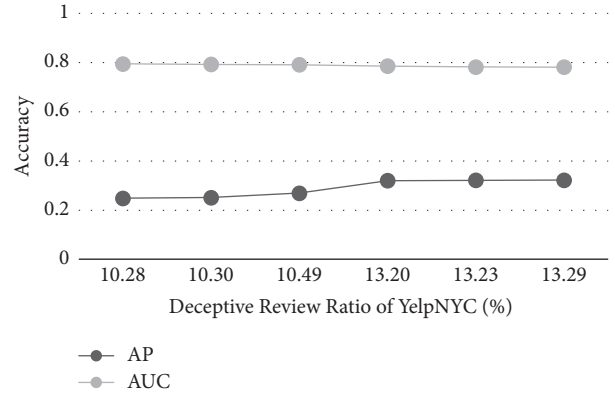

Figure 7: The AP and AUC of *TM-DRD* in different deceptive review ratios.

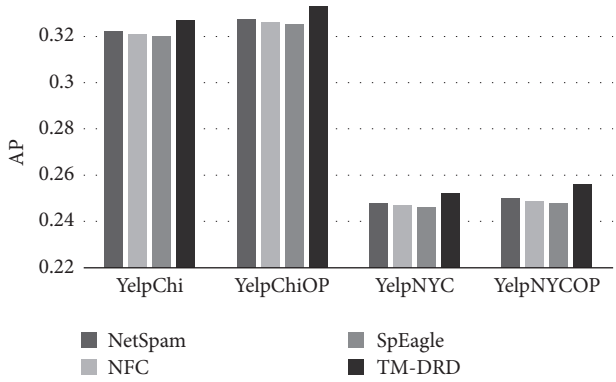

Figure 8: The AP for TM-DRD and SpEagle, NFC, and NetSpam in different datasets.

where $d_i = x_i - y_i$ represents the ith element in the ranking differential set d, x_i represents the ith element in the isometric rank of the X variable, similarly, y_i represents the ith element in the ranking of Y variables, and N represents the number of elements in the X-variable set or Y-variable set. The two are equal, and here N is 13, the number of features.

We use *TM-DRD* and *NFC* [24], *SpEagle* [30], and *NetSpam* [32], respectively, to calculate the deceptive degree probability of each review in the experimental review datasets above. We sort all the reviews according to the deceptive probability in descending to obtain a list of reviews. Next, AP and AUC values are calculated according to (17) and (18), respectively. The experimental results are shown in Figures 5, 6, 8, and 9. We observe and analyze the test results in the performance of those two indicators. At the same time, experiments on the impact of the proportion of deceptive reviews in the datasets on the accuracy of the test results are carried out, as shown in Figure 7. Figure 10 shows the distribution of the features weight in the *YelpNYC*. The results show that behavior-based features are assigned higher weights than semantic-based features. The features in reviewer-behavior classification strategy UB in experimental data sets have higher weight and better performance. The feature list {RPP, ROW, RRD, ETF, ACS, MNRD, RPR, RNR, BST, ERD, BDS, RD, RWR} is obtained according to the definition order of the features in Section 3.3.

As shown in Figures 5 and 6, the detection results of the *TM-DRD* on the same review datasets are superior to others on the indicators of AP and AUC. The results of deceptive review detection on the *TM-DRD* algorithm on different review datasets are very different in the AP index. The difference between the detection results of *YelpChi* and *YelpNYC* [30] is more than 0.05 in the AP index, but the difference in the AUC index is far below 0.05. As shown in Figure 7, with the increasing proportion of deceptive review in the datasets, the AP index of *TM-DRD* algorithm is increasing, but the AUC index is almost unchanged.

Since the experimental data are all annotated, the proportion of deceptive review in the *YelpChi* and *YelpNYC* [30] is, respectively, calculated to be 13.2% and 10.3%. The proportion of deceptive review in the *YelpChi* [30] dataset is 13.23% and 13.29%, respectively. The ratio of deceptive review on restaurants and hotels in the *YelpNYC* [30] is 10.49% and 10.28%. As the proportion of deceptive review in the datasets increasing, the probability of review being detected as deceptive review increases. More and more reviews are identified as deceptive review, while the AUC values are almost unchanged. It shows that the AUC index has nothing to do with the proportion of deceptive review, it depends on the list of reviews after sorting.

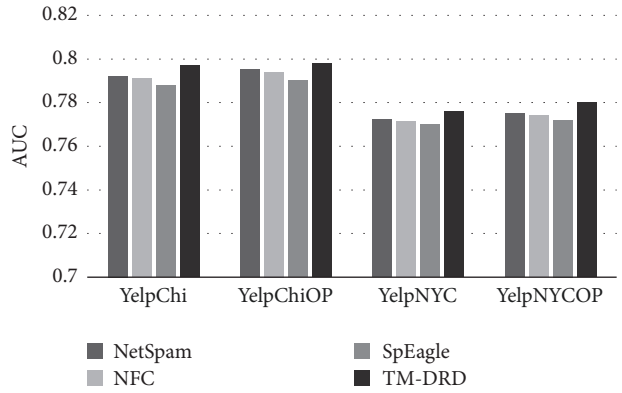

Figure 9: The AUC for TM-DRD and SpEagle, NFC, and NetSpam in different datasets.

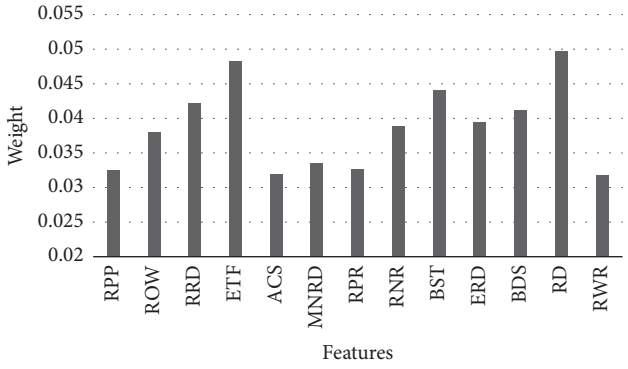

Figure 10: Features weight distribution of YelpNYC.

As shown in Figures 8 and 9, the performance of the AP and AUC indicators on the related review datasets *YelpChiOP* and *YelpNYCOP* are, respectively, better than the corresponding original review datasets *YelpChi* and *YelpNYC* [30]. The AP indicators and the AUC indicators improve on different review datasets.

As shown in Figure 11, the 13 levels of feature weights and their AP levels used in the experiment correspond to the coordinate points in the figure, respectively. From the figure, it can be seen that the overall trend of the accuracy rate increasing with the increase of weight level; that is, the higher the weight value, the higher the accuracy of the detection result. The feature weight is closely related to the accuracy of the final test result of the deceptive review detection. The feature weight calculated through the *TM-DRD* algorithm indicates the ability of the feature to distinguish the deceptive review, and the feature with the greater weight is more effective in the deceptive review detection. With the increase of weight, these features are accompanied by the corresponding increase of the test results on the AP, AUC, and other indicators, which shows that the feature weight calculation improves the accuracy of deceptive review detection test results.

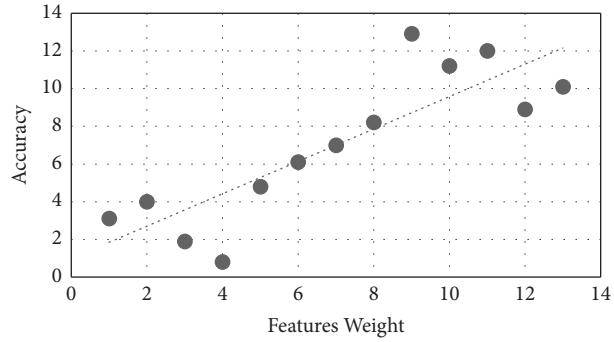

Figure 11: Relationship between features weight and accuracy.

6. Conclusion

In this paper, we analyze the existing research on the deception review detection and design a deceptive review detection algorithm based on target product identification and metapath feature weight calculation, *TM-DRD* algorithm. In this algorithm, we firstly analyze the different deceptive review states of the product type and then design the static target product detection algorithm based on the difference of the grade score and the dynamic target product detection algorithm based on the kernel density estimation for different states. Based on these proposed algorithms, we identify the target product. Then, we construct the related review datasets as a heterogeneous review information network and calculate the weight of the metapath feature of the target product. In the following, with the metapath based feature weights, we calculate the deceptive degree probability of reviews to determine the final category of reviews. Finally, we conduct several experiments to evaluate the accuracy and efficiency of the proposed *TM-DRD* algorithm. We analyze the experiment results, respectively, according to the target product identification and the deceptive review detection. In particular, comparative analysis of the performance of the proposed *TM-DRD* algorithm and the *NFC* [24], *SpEagle* [30], and *NetSpam* [32] on AP, AUC, and other evaluation indicators shows that the method of feature weight calculation is very helpful to improve the accuracy of the deceptive review detection.

Conflicts of Interest

All the authors do not have any possible conflicts of interest.

Acknowledgments

This work was supported by National Natural Science Fund of China under Grant 61502185 and the Fundamental Research Funds for the Central Universities (no. 2017KFYXJJ071).

References

[1] M. Tsikerdekis and S. Zeadally, "Online deception in social media," *Communications of the ACM*, vol. 57, no. 9, pp. 72–80, 2014.

[2] S. KC and A. Mukherjee, "On the Temporal Dynamics of Opinion Spamming," in *Proceedings of the the 25th International Conference on World Wide Web, WWW 2016*, pp. 369–379, Montreal, Canada, April 2016.

[3] C. Xu, "Detecting collusive spammers in online review communities," in *Proceedings of the the sixth workshop on Ph.D. Students in Information and Knowledge Management, PIKM@CIKM 2013*, pp. 33–40, San Francisco, Calif, USA, November 2013.

[4] H. Li, G. Fei, S. Wang et al., "Bimodal Distribution and Co-Bursting in Review Spam Detection," in *Proceedings of the the 26th International Conference*, pp. 1063–1072, Perth, Australia, April 2017.

[5] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. Al Najada, "Survey of review spam detection using machine learning techniques," *Journal of Big Data*, vol. 2, no. 1, article no. 23, 2015.

[6] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT '11)*, vol. 1, pp. 309–319, Association for Computational Linguistics, Portland, Ore, USA, June 2011.

[7] F. Li, M. Huang, Y. Yang et al., "Learning to identify review spam," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 2488–2493, AAAI Press, Barcelona, Spain, 2011.

[8] J. K. Rout, A. Dalmia, K.-K. R. Choo, S. Bakshi, and S. K. Jena, "Revisiting semi-supervised learning for online deceptive review detection," *IEEE Access*, vol. 5, pp. 1319–1327, 2017.

[9] S. Feng, R. Banerjee, and Y. Choi, "Syntactic stylometry for deception detection," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, ACL 2012*, pp. 171–175, Jeju Island, Korea, July 2012.

[10] S. Feng, L. Xing, A. Gogar, and Y. Choi, "Distributional footprints of deceptive product reviews," in *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media, ICWSM 2012*, pp. 98–105, Dublin, Ireland, June 2012.

[11] J. Li, C. Cardie, and S. Li, "TopicSpam: A topic-model-based approach for spam detection," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013*, pp. 217–221, bgr, August 2013.

[12] T. Lappas, "Fake Reviews: The Malicious Perspective," in *Natural Language Processing and Information Systems*, vol. 7337 of *Lecture Notes in Computer Science*, pp. 23–34, Springer, Berlin, Germany, 2012.

[13] H. Li, Z. Chen, and A. Mukherjee, "Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns," in *Proceedings of the Ninth International Conference on Web and Social Media, ICWSM, 2015, University of Oxford*, pp. 634–637, Oxford, UK: the, 2015.

[14] J. Ye, S. Kumar, and F. Akoglu, "Temporal opinion spam detection by multivariate indicative signals," in *Proceedings of the Tenth International Conference on Web and Social Media*, pp. 743–746, Cologne, Germany, 2016.

[15] E. Lim, V. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, "Detecting product review spammers using rating behaviors," in *Proceedings of the the 19th ACM international conference*, p. 939, Toronto, Canada, October 2010.

[16] S. Xie, G. Wang, S. Lin, and P. S. Yu, "Review spam detection via temporal pattern discovery," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2012*, pp. 823–831, Beijing, China, August 2012.

[17] A. Mukherjee, A. Kumar, B. Liu et al., "Spotting opinion spammers using behavioral footprints," in *Proceedings of the the 19th ACM SIGKDD international conference*, pp. 632–640, Chicago, Ill, USA, August 2013.

[18] Y. Zhang and J. Lu, "Discover millions of fake followers in Weibo," *Social Network Analysis and Mining*, vol. 6, no. 1, article no. 16, 2016.

[19] G. Wang, S. Xie, B. Liu, and P. S. Yu, "Review graph based online store review spammer detection," in *Proceedings of the 11th IEEE International Conference on Data Mining, ICDM 2011*, pp. 1242–1247, Vancouver, Canada, December 2011.

[20] L. Li, W. Ren, B. Qin et al., "Learning Document Representation for Deceptive Opinion Spam Detection//Processing," in *Proceedings of the of 14th China National Conference on Chinese Computational Linguistics*, pp. 393–404, Guangzhou, China, 2015.

[21] C. Xu, J. Zhang, K. Chang, and C. Long, "Uncovering collusive spammers in Chinese review websites," in *Proceedings of the the 22nd ACM international conference*, pp. 979–988, San Francisco, California, USA, October 2013.

[22] Z. Wang, S. Gu, X. Zhao, and X. Xu, "Graph-based review spammer group detection," *Knowledge & Information Systems*, vol. 3, no. 2017, pp. 1–27, 2017.

[23] G. Fei, A. Mukheijee, B. Liu et al., "Exploiting burstiness in reviews for review spammer detection," in *Proceedings of the 7th International Conference on Weblogs and Social Media, ICWSM, 2013*, The AAAI Press, Cambridge, Mass, USA, 2013.

[24] J. Ye and L. Akoglu, "Discovering Opinion Spammer Groups by Network Footprints," in *Machine Learning and Knowledge Discovery in Databases*, vol. 9284 of *Lecture Notes in Computer Science*, pp. 267–282, Springer International Publishing, Cham, 2015.

[25] K. Dhingra and S. K. Yadav, "Spam analysis of big reviews dataset using Fuzzy Ranking Evaluation Algorithm and Hadoop," *International Journal of Machine Learning and Cybernetics*, 2017.

[26] M. A. Hasan, "Link Prediction using Supervised Learning," *Proceedings of SDM Workshop on Link Analysis Counterterrorism & Security*, vol. 30, no. 9, pp. 798–805, 2006.

[27] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, pp. 440–442, 1998.

[28] K. Walker Susan, "Connected: The Surprising Power of Our Social Networks and How They Shape Our Lives," *Journal of Family Theory Review*, vol. 3, no. 3, pp. 220–224, 2011.

[29] Y. Wang, S. C. F. Chan, H. Va Leong, G. Ngai, and N. Au, "Multi-dimension reviewer credibility quantification across diverse travel communities," *Knowledge & Information Systems*, vol. 49, no. 3, pp. 1071–1096, 2016.

[30] S. Rayana and L. Akoglu, "Collective opinion spam detection: Bridging review networks and metadata," in *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2015*, pp. 985–994, aus, August 2015.

[31] A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, "What yelp fake review filter might be doing?" in *Proceedings of the 7th International AAAI Conference on Weblogs and Social Media, ICWSM 2013*, pp. 409–418, usa, July 2013.

[32] S. Shehnepoor, M. Salehi, R. Farahbakhsh, and N. Crespi, "NetSpam: A Network-Based Spam Detection Framework for Reviews in Online Social Media," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 7, pp. 1585–1595, 2017.