Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2020.013382
Article

Tech Science Press

Spam Detection in Reviews Using LSTM-Based Multi-Entity Temporal Features

Lingyun Xiang[1,2,3], Guoqing Guo[2], Qian Li[4], Chengzhang Zhu[5,*], Jiuren Chen[6] and Haoliang Ma[2]

[1]Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Changsha University of Science and Technology, Changsha, 410114, China
[2]School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, 410114, China
[3]Hunan Provincial Key Laboratory of Smart Roadway and Cooperative Vehicle-Infrastructure Systems, Changsha University of Science and Technology, Changsha, 410114, China
[4]Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, 2007, Australia
[5]Academy of Military Sciences, Beijing, 100091, China
[6]Science and Technology on Test Physics and Numerical Mathematic Laboratory, Beijing, 100094, China
*Corresponding Author: Chengzhang Zhu. Email: kevin.zhu.china@hotmail.com
Received: 04 August 2020; Accepted: 12 September 2020

Abstract: Current works on spam detection in product reviews tend to ignore the temporal relevance among reviews in the user or product entity, resulting in poor detection performance. To address this issue, the present paper proposes a spam detection method that jointly learns comprehensive temporal features from both behavioral and text features in user and product entities. We first extract the behavioral features of a single review, then employ a convolutional neural network (CNN) to learn the text features of this review. We next combine the behavioral features with the text features of each review and train a Long-Short-Term Memory (LSTM) model to learn the temporal features of every review in the user and product entities. Finally, we train a classifier using all of the learned temporal features in order to predict whether a particular review is spam. Experimental results demonstrate that the proposed method can effectively extract the temporal features from historical activities, and can further jointly analyze the activity trajectories from multiple entities. Thus, the proposed method significantly improves the spam detection accuracy.

Keywords: Spam detection; LSTM; temporal feature; text feature; behavioral feature

1 Introduction

Due to the development of mobile communication technology and the widespread use of smartphones, it is now normal for people to share their reviews of products or the services they have purchased. As a result, many customers therefore consider the user ratings of a product before making a purchasing decision. For merchants, the quality of online evaluation is consequently closely related with their profit level [1]. It has therefore become common practice for online retailers to hire professional spammers to leave bad reviews or ratings for their competitors' products. According to one survey[1], about 25% of the reviews on Yelp.com are fake reviews of this kind.

[1] http://www.bbc.com/news/technology-24299742

With this in mind, researchers have conducted extensive studies in the challenging field of spam detection. Many of them take advantage of the machine learning's effectiveness, which have been widely used in dealing with optimization problems [2–5]. Some works in the literature have studied how to find clues to characterize spam and extract effective text features and behavioral features. Once the reviews are represented by these features, statistical models can then be applied. In addition, researchers mostly conduct grammatical and semantic analysis through the use of text features, which are among the most widely used feature types. Adopting a grammatical analysis perspective, some researchers have extracted part-of-speech tags [6], multi-grammatical features [7], and a wide range of other lexical and syntactic features [8] in order to infer the intent and purpose of the text. Moreover, semantic analysis-based methods perform abstract representation and feature extraction of a review, and further use these features to distinguish normal reviews from fake ones. These types of methods tend to follow research directions such as sentiment analysis [9], semantic representation learning [10,11] semantic similarity [12,13], etc.

Previous studies have shown that many fake reviews cannot even be identified by humans due to the highly imperceptible nature of fake reviews. Thus, the use of text features does not perform well on real data; however, the behaviors of the spammers contain a large number of clues that reveal suspicious patterns. As a result, many researchers have explored the extraction of behavioral features from the historical traces left by the reviewers. These traces include registration time, posting a review, giving a score for a product, etc. From these traces, certain behavioral features can be extracted: these include the maximum number of reviews posted by the reviewer in a day, the proportion of positive reviews among all reviews made by the user, the rank of these reviews among all reviews of the product, the average product score, etc. The work in [14] shows that users' behavioral features are more effective for this purpose than reviews' text features.

Unfortunately, behavioral features extraction is more reliant on expert knowledge, and is not always supported by rich trace information. The use of behavioral features can thus only improve spam detection performance to a certain extent. Therefore, many researchers choose to employ both text features and behavioral features when performing spam detection tasks, and consequently achieve better results. For example, the work in [14] proposed several new effective behavioral features and combined them with binary grammatical features on the Yelp dataset to improve the spam detection performance.

The aforementioned works, using either text features or behavioral features, tackle the spam detection problem as a binary (i.e., either fake review or normal review) classification problem. A classifier is trained on features extracted from current published reviews in order to predict whether a new review is fake or normal. However, most of these works do not consider the temporal features of the reviews; these are features related to the time interval between the related reviews of the same entity. Here, 'related reviews' are reviews posted by the same user, or for the same product, within a certain period of time. If the opinion in these related reviews changes abnormally, these reviews may contain fake opinions. Accordingly, the leveraging of temporal features may significantly contribute to spam detection. Moreover, existing studies also extract features from the relationships between users and reviews, or between products and reviews, independently. It would be preferable to adopt a holistic approach that considers the relationship among users, reviews and products to extract behavioral and text features.

Accordingly, in this paper, we propose a multi-entity temporal feature-based spam detection model (MTFSD). After obtaining the text and behavioral features of reviews, MTFSD uses LSTM to automatically learn temporal features for each review by considering the latest reviews for the user entity or product entity. Finally, MTFSD merges these learned features to mine the relationships between different entities so as to distinguish between normal and fake reviews. Experimental results show that MTFSD achieves better spam detection performance compares with other methods.

In summary, the main contributions of this paper include the following:

- The proposed method effectively learns the dynamic temporal features via LSTM. These temporal features can effectively capture the relevance among the related reviews and the internal characteristics of a review.

- The proposed method provides an effective way to learn comprehensive fusion features from the perspective of multiple entities in order to detect fake reviews. These comprehensive fusion features enable significantly better fake review detection performance.

2 Related Work

With the development of the internet technology and social networks, network and information security have attracted widespread attention [15–20]. The research on spam detection in reviews began in 2008 with the work of Jindal et al. [21]. These authors collected millions of product reviews from Amazon.com. After analyzing these reviews, they divided fake reviews into three types: untruthful reviews, reviews of the brand more generally and not the product specifically, and non-reviews. Moreover, due to the lack of labels in the data, they marked duplicate reviews as fake reviews, then adopted three kinds of features: review-centered, reviewer-centered, and product-centered. Finally, they used a logistic regression model to achieve fake review detection. Since then, a number of researchers have devoted significant effort to the development and exploration of extracting effective text and behavioral features.

The study of text features can be divided into two categories: Those based on grammatical analysis and those based on semantic analysis. From the grammatical analysis perspective, text features are primarily extracted from the word frequency statistics of words or phrases, which are typically processed using a bag-of-word (BOW) model or an N-gram model. The BOW model regards the content of each review as a 'bag' and assumes that each word in the bag is independent, while the word order, grammar and syntax in the text are ignored; the reviews are then classified according to the words contained in the bag. For its part, N-gram uses a window of size n to slide on the text content, forming a word fragment sequence of length n. Each of these word segments is called a 'gram'. Subsequently, the frequencies of all grams are counted and filtered in order to obtain the vector feature space of the text. Unigram, bigram, and trigram methods are commonly used in bag-of-words features [7,22]. However, the detection effect of bag-of-word features differs across different datasets [6,23].

To overcome the disadvantages associated with BOW features, subsequent works have focused on extracting text features from the part-of-speech feature analysis, which is achieved by tagging the part-of-speech of the text and counting the frequency of occurrence. While various part-of-speech features were employed in some early works [6–8,23], these were not systematically analyzed until the work of Li et al. [9] in 2014. Li and colleagues found that fake reviews fabricated by domain experts tend to contain more nouns, adjectives and qualifiers than real reviews, along with fewer verbs and adverbs than real reviews. Thus, through further analysis of these differences in language usage, deceptive opinion spam can be more precisely identified.

Not only in spam detection tasks, but also in most other text classification tasks, grammatical analysis is a simple and powerful tool; however, it also relies heavily on expert knowledge. Moreover, the performance of grammatical analysis is largely dependent on the datasets being employed, whose semantics cannot be intuitively understood with the aid of grammatical analysis. The work in [24] studied a frame-based deep semantic analysis of spam, which was concerned with the different distributions of the semantic frameworks of fake reviews as well as normal ones. Furthermore, the work in [25] noticed that both the extraction of grammatical features and the use of semantic frameworks for spam analysis lack a global perspective from which to effectively represent semantic information throughout the reviews. This work employs a CNN model to automatically learn text features for spam detection purposes, an approach that outperforms both RNNs and traditional hand-crafted linguistic features.

Behavioral information can also provide clues for use in identifying review spam. Through data screening, the work in [26] extracted several behavioral features that indicate abnormal reviewer activity, such as overall score bias and early time bias. The work in [27] further noticed the relationship between ratings and review posting time. By considering the reviews posted in different periods, their average

ratings, and the overall proportion of unique reviews, this work concluded that the ratings of normal reviews are unrelated to time, while the number of extreme ratings tends to increase explosively over a short period. Moreover, the work in [23] experimentally demonstrated that the text-feature-based approach adopted in [6] is not very effective on the Yelp dataset. Furthermore, the work in [28] conducted extensive statistical analysis to propose a series of effective behavioral features, subsequently proving experimentally that the behavioral features of reviews are more effective than text features.

Most of the above works analyze users (who write reviews), reviews, and products separately. The work in [29] proposed several behavioral features from the graph structure perspective, exploring the direct network connection utility between users and products, and successfully applied this approach in online applications. Based on the previous studies, the work in [30] analyzed the behavior patterns of spam in both the time and spatial dimensions, and consequently achieved better spam detection results.

However, not all reviews contain rich behavioral information; in fact, this phenomenon is relatively common in the datasets. Therefore, the fusion of text features and behavioral features to detect fake reviews has become the trend among most researchers. For example, the work in [31] comprehensively utilized the text features, behavioral features, metadata, and graph structure information of the reviews, then applied a Markov random field model to detect spam.

3 The Proposed Method

3.1 Framework Overview

In this subsection, we first define three important entities in a review system: namely, the review, user and product.

- **Review entity:** The review entity refers to the review text posted by a user regarding a product. It includes attributes such as review content, posting time, publisher, evaluation object, etc.
- **User entity:** The user entity contains all users along with their posted reviews. As with the review entity, these reviews also have many attributes, but these pertain instead to the reviewers (users).
- **Product entity:** The product entity includes all objects (e.g., hotel, restaurant, movie, etc.) in the review system that can be given reviews or ratings, along with all reviews these products have obtained. All reviews are connected to their corresponding reviewed objects.

Examining the definitions of these three entities, it can be seen that an individual review will be associated with these three entities from different perspectives. In the user and product entity, a review will have strong relevance to other reviews posted within the same time period. Regarding the review entity, the text and behavior of a review will contain clues that indicate whether or not the review is fake. However, reviews in the review entity are independent of each other while their relevance is commonly ignored. Regarding the user and product entities, the relevance of reviews and the behavior of users and products may reflect spam patterns, which are able to be captured by temporal features. Therefore, extracting temporal features of both text and behavior from the user and product entity is a subject we deem worthy of significant attention. Here, we use the LSTM model to automatically learn deeper temporal features from the extracted CNN-based text features and hand-crafted behavioral features of reviews in the user and product entities. Finally, we integrate the features derived from different entities and input these features into a classifier in order to determine whether or not the reviews are fake.

The framework of the proposed method is outlined in Fig. 1. This method comprises five main steps, which are as follows:

Initialization:

It is widely understood that users mostly write reviews subjectively; therefore, reviews are affected not only by the product itself, but also by the users' opinions and emotions. To effectively learn the temporal

features of a review in the user and product entity, other reviews in the same entity should be employed in order to capture their relevance with the learned review.

In fact, for the review in the review entity to be detected, we opt to consider only the most recent reviews made by the same user. Examining a user's most recent reviews rather than all the reviews can not only facilitate analysis of the user's recent status, but also avoids any influence being exerted from the user's status too long ago on the current review.

On the other hand, reviews for the same product in a given period always tend to cluster around a similar type of opinion (positive or negative). When a product is attacked by a spammer, the reviews it receives during the period of attack will fluctuate significantly. Therefore, the proposed method takes the latest several reviews into account in order to learn the temporal features of a review in the product entity.

As to the analysis above, the proposed method initially organizes the reviews in the user and product entity according to the time at which they were posted. As a result, each review within the same entity can be assigned a definite location.

In the user entity, the latest k reviews posted by user u before posting review r (the review to be analyzed) can be expressed as follows:

$$u_r = \left\{r_{uk}, r_{u(k-1)}, \cdots, r_{u2}, r_{u1}, r\right\} \tag{1}$$

In the product entity, moreover, the latest m reviews received by product p before review r can be expressed as follows:

$$p_r = \left\{r_{pm}, r_{p(m-1)}, \cdots, r_{p2}, r_{p1}, r\right\} \tag{2}$$

For convenience, we add r into u_r and p_r, and conceive of u_r as the user-related review set of review r, while p_r is the product-related review set of review r. Moreover, the reviews in u_r and p_r are referred to as the user-related and product-related reviews of review r.

Feature Extraction:

For each single review in the review, user, and product entity, we next perform feature extraction from two aspects (text and behavior) to facilitate the subsequent operations.

In previous studies, behavioral features have been considered more indicative than text features of whether a review was real or fake. With the help of previous works and expert knowledge, the proposed method extracts the behavioral features of a single review from the three entities; this process is described in more detail in the next subsection, Behavioral Feature Extraction.

Existing studies have demonstrated that CNN can be effectively applied to text classification tasks [32,33]. Thus, in order to effectively capture the text characteristics of a single review, we employ CNN to learn the feature representation from the content of the review [34,35]; to a certain extent, this can be used to differentiate a normal review text from a fake one. The subsection 3.3xsd provides a more detailed description of the text feature extraction process utilized in the proposed method.

Feature Fusion:

To comprehensively represent a single review as a vector for use in distinguishing fake reviews from normal ones, it is necessary to integrate all the extracted features for each review. Different feature types reflect different aspects of review characteristics, as outlined in [36].

Generally speaking, a review includes the review text along with other attributes such as author, product, rating, etc. Thus, the feature representation of a given review should include not only the text features extracted from the review text, but also the behavioral features extracted from other attributes.

Accordingly, we concatenate the CNN-based text features and behavioral features of each review to form the joint feature representation used for subsequent operations.

Temporal Feature Learning:

The reviews in the user entity are often written for different products. Their statistical characteristics always reflect changes in the user's personal sentiment and writing style. By contrast, those in the product entity are mostly written by different users, and are closely related to the real-time status of the product.

In the user and product entity, each review will most likely be influenced by the previous reviews in its user-related review set and product-related review set, which can be regarded as the states of these entities at different times. In addition, the closer together the posting/obtaining times for reviews in the same entity, the more relevant they are to each other. As existing researches [37–41] show that LSTM can effectively capture the relevant input data at different time points by combing historical information, the proposed method employs LSTM to learn the temporal features for each review in the user or product entity.

By considering the feature vector of each review as a moment on the time axis, each entity is represented can be a time series comprising several moments. Thus, with the help of LSTM, the temporal feature representation of a review in entities at a deeper level can be automatically learned from the fused features of its user-related and product-related reviews.

Classification:

Through Feature Extraction, the review to be detected in the review entity is represented as a joint feature vector made up of its behavioral feature and text feature. Moreover, after the Temporal Feature Learning step is complete, the review to be detected is represented as two learned temporal feature vectors (for both the user and the product entity). Finally, the three feature vectors of reviews from different entities are merged and input into a classifier, which facilitates determining whether or not it is a fake review.

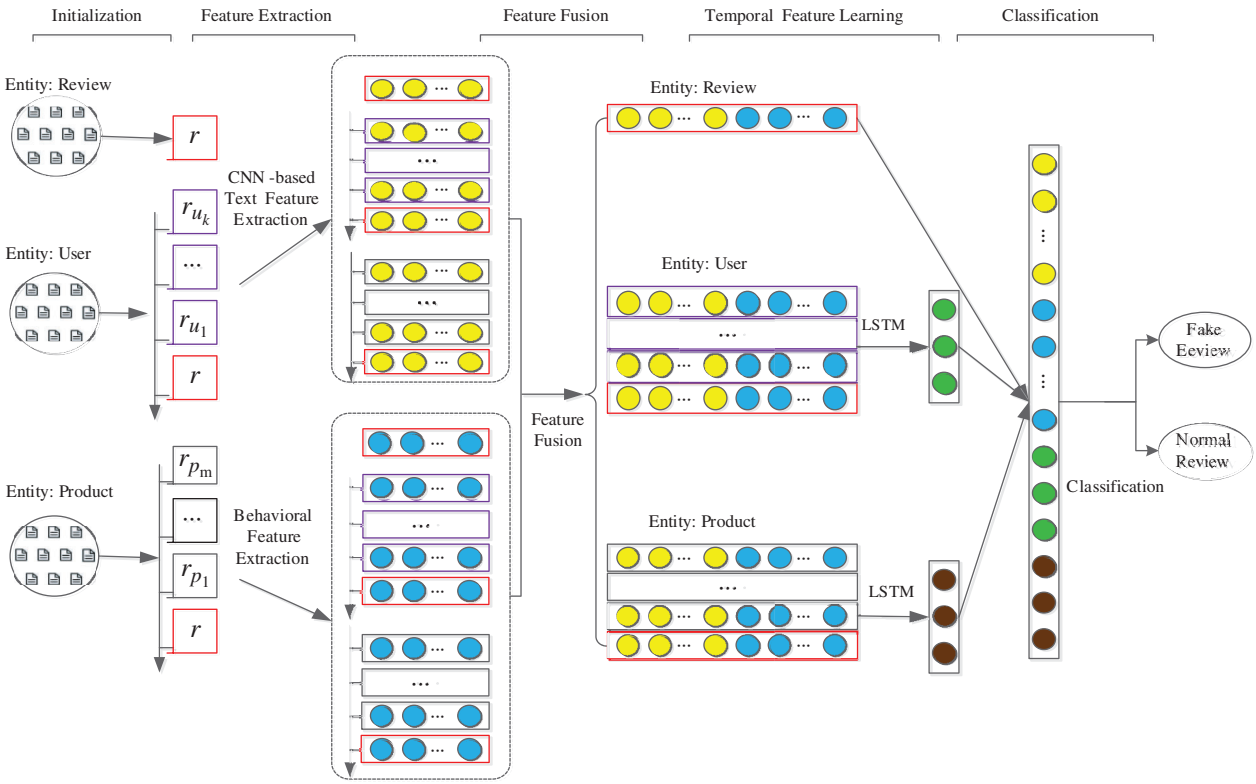

Figure 1: The framework of the proposed model

3.2 Behavioral Feature Extraction

Regarding the attributes of a review in the review entity, researchers have analyzed a large body of data and consequently identified many clues that could indicate the presence of a fake reviews. For example, the first review on a product usually attracts the most attention, as people tend to pay attention to the top reviews with the highest ratings. Accordingly, spammers often try to ensure that their reviews are placed as high on the list as possible. Thus, how high a given review is on the list can represent a clue as to whether this review is fake.

Five behavioral features obtained from the attributes of reviews in the review entity are employed in the proposed method: namely, the order of the review (Rank), the absolute value of the score deviation rate (RD), the extremeness of the score (EXT), the score deviation rate with threshold (DEV), and whether the review is a singleton (ISR). Further details are presented in Tab. 1.

Table 1: Behavioral feature in review entity

Features	Meaning
Rank	The order of the review [21].
RD	The absolute value of score deviation rate [42].
EXT	Extremes of score. 1, if the score is 4,5; 0 otherwise.
DEV	Score deviation rate with threshold of β_1 [28], $x_{\mathrm{DEV}}(i) = \begin{cases} 1, if \ \dfrac{\lvert r_i - avg(r)\rvert}{4} > \beta_1 \\ 0, otherwise \end{cases}$, r_i indicates the rating of review, $avg(r)$ indicates the average score of all reviews, β_1 is learned by recursive minimum entropy partition.
ISR	If the user posts only one review, x_{ISR}= 0; 1 otherwise [31].

In addition to the behavioral information contained in reviews, spammers also tend to leave some traces of their corresponding behavior trajectories when they post fake reviews to attack other products. For example, according to the statistics in [23], 75% of spammers will write more than five reviews per day on average, while 90% of normal users typically write no more than one review per day. These pieces of behavioral information can be statistically analyzed at the user and product levels. Accordingly, some more behavioral features can be extracted from the reviews in the user and product entity.

The following six features based on user behaviors listed in Tab. 2 are adopted in the proposed method: the maximum number of reviews posted within a day (uMNR), the ratio of positive evaluation (uPR), the ratio of negative evaluation (uNR), the distribution entropy of user evaluation scores (uERD), the average deviation rate (uavgRD), and the overall burstiness (uBST). Moreover, five features are also extracted from the attributes of reviews in the product entity, which are listed in the Tab. 3. These features, which are similar to those in the user entity, are as follows: the maximum number of reviews posted within a day (pMNR), the ratio of positive evaluation (pPR), the ratio of negative evaluation (pNR), the average deviation rate (pavgRD) and the distribution entropy of the average valuation score obtained (pERD).

In summary, for each review r, five review-based behavioral features, six user-based behavioral features, and five product-based behavioral features are extracted using the methods described above to form a complete behavior feature vector. If we denote the value of the j-th behavioral feature as o_j, the behavior feature vector of r can be expressed as $q(r)$, where:

Table 2: Behavioral features in the user entity

Features	Meaning
uMNR	Maximum number of reviews that a user posted within a day [23].
uPR	The ratio of positive reviews (4–5 star) in all reviews posted by this user [23].
uNR	The ratio of negative reviews (1–2 star) in all reviews posted by this user [23].
uERD	Distribution entropy of user evaluation scores [31].
uavgRD	Average deviation rate [23].
uBST	Burstiness [23], $x_{BST}(i) = \begin{cases} 0, if L(i) - F(i) > \tau \\ 1 - \dfrac{L(i) - F(i)}{\tau}, otherwise \end{cases}$, where $L(i) - F(i)$ describes days between last and first review, and $\tau = 29$ (day).

Table 3: Behavioral features in the product entity

Features	Meaning
pMNR	Maximum number of reviews that a product received within a day [23].
pPR	The ratio of positive reviews (4–5 star) in all of the product's reviews [23].
pNR	Ratio of negative reviews (1–2 star) in all of the product's reviews [23].
pavgRD	Average deviation rate [23].
pERD	Distribution entropy of the average evaluation score obtained [31].

$$q(r) = \left[o_1, o_2, \cdots, o_j, \cdots, o_{16}\right] \tag{3}$$

3.3 CNN Based Text Feature Learning

The acquisition of text features depends on the text content of the review itself. For each review, the proposed method uses CNN to learn the features of the global semantic information from the review's text content. This learning process is illustrated in Fig. 2. First, each word in the review text is represented as a dense vector of fixed dimensions using a distributed word representation model, thereby converting each piece of review text into a vector matrix. Next, after multi-kernel convolution, pooling, and full-connection operations are implemented, the text feature output of each review is finally obtained.

Suppose that the review r to be analyzed contains n words in its text content; that is $r = \{w_1, w_2, \cdots, w_n\}$. Each word w_j is represented by a word representation language model) which can be word2vect [43], ELMo [44], glove [45], etc., creating a vector $e(w_i)$ of dimension d, after which the text of r is then converted into a vector matrix $E(r) \in R^{n \times d}$. We then use three convolution kernels with window sizes of $3 \times d$, $4 \times d$, and $5 \times d$ to perform the convolution operations. The number of each type of convolution kernel is set to 100. The process of one convolution for each kernel is described as follows:

$$c_i = W_c \cdot E_{i:i+h-1}(r) + b_c \tag{4}$$

where h is the length of the convolution window and i represents the i-th row of the matrix $E(r)$, while W_c is the weight to be learned. The convolution window slides on $E(r)$; each sliding interval is set to 1, such that $n - h + 1$ features will be obtained. After maximum pooling:

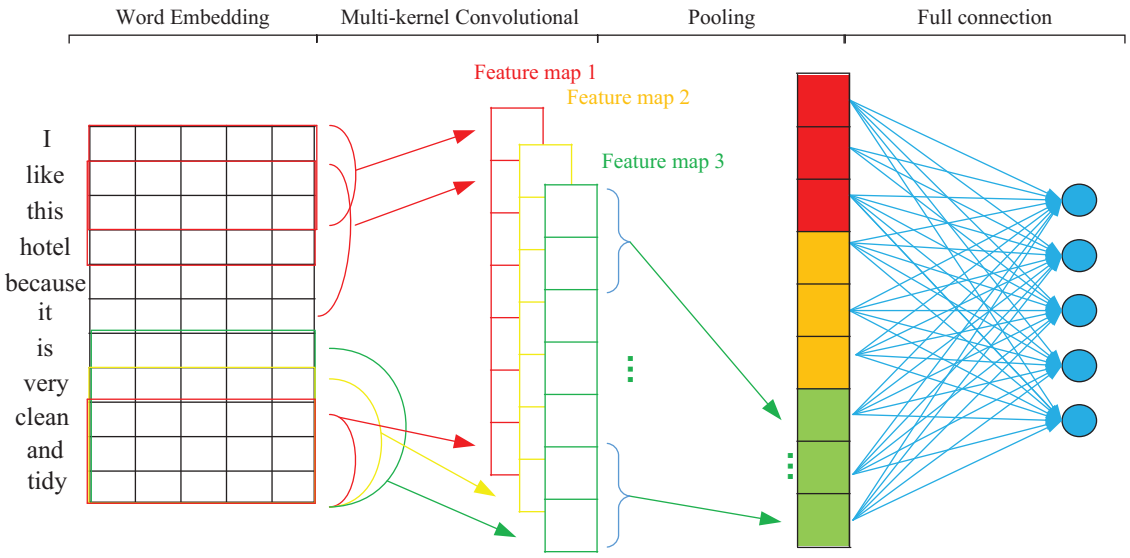

Figure 2: CNN-based feature learning of a single review

$$\hat{c} = \max(c_i), i \in (n - h + 1) \tag{5}$$

Each convolution kernel with a fixed h will be assigned an Eigenvalue $\hat{c}$, and $\hat{C}$ will be learned for three windows of different sizes. These features are then fully connected to obtain a feature vector $Te(r)$ of dimension l.

$$Te(r) = ReLu\left(W_{\hat{C}} \cdot \hat{C} + b_{\hat{C}}\right) \tag{6}$$

where $ReLu$ is the activation function.

3.4 LSTM-Based Temporal Entity Feature Extraction

Most reviews written by spammers are based on templates or existing reviews and the slightly modified. These highly similar reviews are posted continuously. By contrast, in the case of non-spammers, reviews are usually written with reference to the features of a specific product, and the differences among these normal reviews tend to be large. When a merchant hires spammers to either post positive reviews for their own products or to leave malicious fake reviews for competitors' products, the spammers will leave traces in the process that provide clues for the detection of fake reviews. This paper analyzes the historical traces of reviews from two aspects, namely users and products, and automatically learns the temporal features from reviews in the user and product entity through LSTM.

By performing feature extraction and feature fusion, the extracted text feature vector $Te(r)$ and behavioral feature vector $q(r)$ are fused to a joint feature representation x_r of the review r, which is expressed as follows:

$$x_r = Te(r) \oplus q(r) \tag{7}$$

Similarly, for each review in the user-related review set u_r and the product-related review set p_r of review r, a joint feature representation is obtained. For the review r_{Si}, $S \in \{u, p\}$, in S_r, after combining its text feature vector $Te(r_{Si})$ with its behavioral feature vector , its joint feature representation x_{Si} can be expressed as follows:

$$x_{Si} = Te(r_{Si}) \oplus q(r_{Si}) \tag{8}$$

Consequently, S_r can be converted to the following joint feature representation $v(S_r)$:

$$v(S_r) = [x_{S1}, x_{S2}, \cdots, x_{Sz}] \tag{9}$$

where z is the number of reviews contained in S_r. If $S = u$, i.e., S_r denotes u_r, then $z = k + 1$; otherwise, $z = m + 1$. Generally speaking, the number of reviews corresponding to the same product as the review r is much larger than the number of reviews corresponding to the same user as the review r. Therefore, when learning temporal features on a product entity, the number of reviews in association analysis is greater than the number obtained when extracting temporal features on a user entity, which is usually expressed as $m > k$.

The z reviews contained in S_r are sorted according to their time of occurrence. Each review is considered as a moment, which is represented by x_{Si}. $v(S_r)$ is a time series that is input into an LSTM model for the learning of temporal features. An LSTM is a type of Recurrent Neural Network (RNN) [46]; thus, it inherits the features of most RNN models, and further solves the Vanishing Gradient problem caused by the gradual reduction of the gradient back-propagation process. These models are widely applied in the time series analysis context. Each neuron of LSTM contains three gates: a forgetting gate, an input gate and an output gate. If the input at time i is x_{Si}, the forgetting gate f_i will decide to either discard or retain the information by using the following equation:

$$f_i = \sigma\left(W_f \cdot (h_{i-1}, x_i) + b_f\right) \tag{10}$$

here, the input gate I_i is used to update the state of neurons:

$$I_i = \sigma(W_I \cdot (h_{i-1}, x_i) + b_I) \tag{11}$$

moreover, the current neuron state C_i can be expressed as:

$$C_i = f_i \cdot C_{i-1} + I_i \cdot \tanh(W_C \cdot (h_{i-1}, x_i) + b_C) \tag{12}$$

additionally, the output gate O_i is used to determine the value of the next hidden state:

$$O_i = \sigma(W_O \cdot (h_{i-1}, x_i) + b_O) \tag{13}$$

finally, the neuron outputs h_i:

$$h_i = O_i \cdot \tanh(C_i) \tag{14}$$

where W_f, W_I, W_C and W_O are the weights that can be learned, b_f, b_I, b_C and b_O represent the bias, h_{i-1} represents the output of the previous moment, C_{i-1} denotes the state of neuron at the last moment, *tanh* is the activation function, and σ represents the sigmoid function.

As a result, by employing LSTM and the input of $v(u_r)$ and $v(p_r)$, we can learn a temporal feature vector $V(u_r)$ and $V(p_r)$ from the user and product entity at a deeper level in time series.

3.5 Multi-Entity Feature Fusion and Classification

Through the above operations, four types of features—namely the text features $Te(r)$, the behavioral features $q(r)$, the user-based temporal features $V(u_r)$, and the product-based temporal features $V(p_r)$ are obtained for given a review r. After these are cascaded, a new comprehensive feature vector $F(r)$ is formed, such that:

$$F(r) = Te(r) \oplus q(r) \oplus V(u_r) \oplus V(p_r) \tag{15}$$

Finally, a classification model is constructed using softmax for $F(r)$, enabling the final classification result y to be obtained:

$$y = \text{softmax}(\mathrm{W_F} \cdot \mathrm{F(r)} + \mathrm{b_F}) \tag{16}$$

where y is the probability distribution of the detected review being fake or normal, while W_F and b_F are the model parameters obtained during training.

4 Experiments and Analysis

4.1 Datasets and Evaluation Metrics

4.1.1 Datasets

To verify the effectiveness of the proposed method, we select the Yelp dataset for our experiments. The Yelp dataset is a publicly available commercial website dataset that offers a good balance between commercial authenticity and ground truth, and has thus been widely used in the works of many predecessors. In this paper, we focus on hotels in the Yelp dataset for our experiments.

The Yelp Hotel dataset contains 6,883,290 reviews for 3,680,118 hotels from 66,599 reviewers. Of these, 5,679 reviews are labeled. Among the labeled reviews, 803 are fake reviews, while 4,876 are normal. Each review contains written time (date), content (reviewContent), the ID of the reviewer (reviewerID), the rating provided by the reviewer (rating), the hotel ID (hotelID), and the label (flagged), along with some other attributes. Moreover, the dataset records the username, registered address, registration time, and the number of reviews posted by each user. Relevant information about the hotel (such as its name, registration date, registered address, price, telephone, and other information) is also recorded.

We mainly use the attribute date, reviewContent, reviewerID, rating, hotelID, and flagged elements of a given review to extract features. Here, date and rating are employed to extract the behavior information of a single review. Moreover, the combination of reviewerID and date can be used to construct the reviewer's historical behavior trajectory, which in turn serves as the basis for extracting the temporal feature from the user entity. Similarly, hotelID associates reviews to different hotels and also incorporates the attribute date, making it possible to extract temporal features from the product entity.

4.1.2 Evaluation Metrics

We use Precision (P), Recall (R), F1-Score (F1) and Accuracy (A) as the metrics for evaluating the spam detection performance. These are defined as follows:

$$P = \frac{TP}{TP + FP} \tag{17}$$

$$R = \frac{TP}{TP + FN} \tag{18}$$

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{19}$$

$$A = \frac{TP + TN}{TP + FN + FP + TN} \tag{20}$$

Here, TP denotes the number of fake reviews correctly detected as fake reviews by the spam detection model; FN is the number of fake reviews incorrectly identified as normal reviews; FP represents the number

of normal reviews incorrectly identified as fake reviews; finally, *TN* indicates the number of normal reviews that are correctly identified as normal reviews.

4.2 Parameter Settings and Experimental Environment

We use a pre-trained word2vec model, which is the model introduced in [47], to obtain the word embeddings for the words contained in the reviews. word2vec takes a text *corpus* as input and produces a low-dimensional vector for each distinct word in the *corpus*. In our experiments, each word is mapped into a 300-dimensional vector, with the *corpus* GoogleNews used as the training set of the word2vec model.

In the process of learning various features, the number of convolution filters is set to 100, moreover, the number of feature dimensions finally output by the convolutional layer l is 30, the number of units in LSTM is set to 10, m is set to 10, while k is equal to 3. A learning rate of 0.00001 is set using the Adam optimizer, the number of epochs is set to 100, and we configure the focal_loss loss function with values of $\alpha = 0.25$, $\gamma = 2$. These parameters are set based on empirical experience.

We developed the proposed method by using the TensorFlow framework with three important libraries Numpy 1.18.5, Keras2.3.1 and Tensorflow-gpu1.14.0 in the Python programming. The implemented model is trained on a computer with windows operating system. Moreover, the computer has 32G memory, RTX2080 Super GPU and Intel Core i7-9700k Processor.

4.3 Comparison of Detection Performance

To validate the effectiveness of the proposed MTFSD, we compare the performance of four similar fake review detection methods with that of our method.

SPEAGLE$^+$, proposed by Rayana et al. [31], is a graph-based semi-supervised method with the labeling ratio set to 80%.

MK, proposed by Mukherjee et al. [14], has two variants: one based on behavioral features (MK_BF) and one based on the combination of behavioral and text features (MK_BF+Bigram). The parameters are set in a manner consistent with the original work.

W_BF+Bigram, proposed by Wang et al. [11], uses the tensor decomposition method to extract behavioral features, and further adds the bigram text features of the reviews as representations of the reviews.

Tab. 4 presents the detection results of different methods. As can be seen from the results, our proposed MTFSD, using text + behavioral + temporal features, achieves the best precision, F1 score and accuracy results among all methods. These experimental results therefore validate the effectiveness of our proposed method.

Table 4: Detection results of different spam detection methods

Method	P	R	F1	A
SPEAGLE$^+$(80%)	26.5%	56.0%	36.0%	80.4%
MK_BF	41.4%	74.6%	55.6%	82.4%
MK_BF + Bigram	46.5%	82.5%	59.4%	84.9%
W_BF + Bigram	48.2%	**85.0**%	61.5%	85.9%
MTFSD	**70.0**%	58.3%	**63.6**%	**91.8**%

4.4 Validity Analysis of Temporal Feature and Multi-Entity Feature Fusion

In addition to the above comparison experiments, we conduct further experiments to validate the impact of the different types of features we employ, particularly the temporal features and multi-entity fused features.

We first construct a spam detection model Te + Be, which uses only $Te(r)$ and $q(r)$ for classification; this means that this model detects spam based only on the text features and behavioral features extracted from the review of interest.

Secondly, an MU detection model is constructed by combining $Te(r)$, $q(r)$, $v(u_r)$ and $v(p_r)$ for review classification purposes. In Eq. (9), when $S = u$, then $v(S_r) = v(u_r)$, while $S = p$, then $v(S_r) = v(p_r)$. $v(u_r)$ is the joint feature representation from the review r and its user-related reviews, while $v(p_r)$ refers to the joint feature representation from the review r and its product-related reviews. As the features employed by MU model are from three entities (i.e., review, user and product), the MU model aims to capture the relevance of these different entities in order to improve the detection accuracy of the Te + Be model.

Compared to MU model, the proposed MTFSD uses LSTM on $v(u_r)$ and $v(p_r)$ temporal feature extraction, then performs multi-entity feature fusion. In short, MTFSD is a model that integrates $Te(r)$, $q(r)$, $V(u_r)$ and $V(p_r)$.

The spam detection results of the three models described above are listed in Tab. 5. As can be seen from the table, compared with Te + Be, MU achieves a certain degree of improvement in terms of detection recall, accuracy and F1 value by adding joint feature representation from related reviews. However, this model simply cascades the text and behavioral features of related reviews, and consequently does not capture their relevance. For its part, the proposed MTFSD learns temporal features from the related reviews using LSTM, and consequently achieves even higher recall, accuracy and F1 value. Accordingly, the results in Tab. 5 reveal that the temporal feature extraction and multi-entity feature fusion proposed in this paper represent effective methods of improving the fake review detection performance.

Table 5: Detection results of different variants of the proposed method

Method	P	R	F1	A
Te+Be	91.7%	27.8%	42.7%	89.9%
MU	64.3%	39.7%	49.1%	90.5%
MTFSD	70.0%	58.3%	63.6%	91.8%

5 Conclusion

In this paper, an LSTM-based spam detection model is proposed that can effectively extract the temporal features of different entities and conduct fusion analysis of these features. The model obtains the temporal embedding representation of multiple entities by learning the correlation features from the perspective of users and products based on posting time, then uses a classifier to complete the spam detection task. Experimental results demonstrate that out proposed method effectively improves the accuracy of spam detection. At present, the extraction of behavioral features relies solely on expert knowledge; this suggests that it would be fruitful to apply machine learning techniques in order to automate the feature extraction process in future work.

Funding Statement: This project is supported by National Natural Science Foundation of China under Grant 61972057 and U1836208, Hunan Provincial Natural Science Foundation of China under Grant 2019JJ50655 and 2020JJ4624, Scientific Research Fund of Hunan Provincial Education Department of China under Grant 18B160 and 19A020, Open Fund of Hunan Key Laboratory of Smart Roadway and

Cooperative Vehicle Infrastructure Systems (Changsha University of Science and Technology) under Grant kfj180402, and "Double First-class" International Cooperation and Development Scientific Research Project of Changsha University of Science and Technology (No. 2018IC25).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Luca, *Reviews, Reputation, and Revenue: The Case of yelp.com*. Harvard Business School NOM Unit Working Papers. No. 12-016, 2016. Available at SSRN: https://ssrn.com/abstract=1928601 or http://dx.doi.org/10.2139/ssrn.1928601.

[2] L. Xiang, G. Zhao, Q. Li, W. Hao and F. Li, "TUMK-ELM: A fast unsupervised heterogeneous data learning approach," *IEEE Access*, vol. 6, pp. 35303–35315, 2018.

[3] W. Li, Z. Chen, X. Gao, W. Liu and J. Wang, "MultiModel framework for indoor localization under mobile edge computing environment," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4844–4853, 2019.

[4] Y. Chen, J. Tao, Q. Zhang, K. Yang, X. Chen *et al.,* "Saliency detection via improved hierarchical principle component analysis method," *Wireless Communications and Mobile Computing*, vol. 2020, 8822777, 2020.

[5] W. Li, H. Xu, H. Li, Y. Yang, P. K. Sharma , *et al.,* "Complexity and algorithms for superposed data uploading problem in networks with smart devices," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5882–5891, 2020.

[6] M. Ott, Y. J. Choi, C. Cardie and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics*, Portland, Oregon, USA, vol. 1, pp. 309–319, 2011.

[7] M. Ott, C. Cardie and J. T.Hancock, "Negative deceptive opinion spam," in *2013 Conf. of the North American Chapter of the Association for Computational Linguistics*, Atlanta, GA, USA, pp. 497–501, 2013.

[8] S. Somayeh, A. A. M. Masrah, A. Azreen and B. M. S. Nurfadhlina, "Detecting deceptive reviews using lexical and syntactic features," in *Int. Conf. on Intelligent Systems Design and Applications*, Salangor, Malaysia, pp. 53–58, 2014.

[9] J. W. Li, M. Ott, C. Cardie and E. Hovy, "Towards a general rule for identifying deceptive opinion spam," in *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 1566–1576, 2014.

[10] L. Y. Li, W. J. Ren, B. Qin and T. Liu, "Learning document representation for deceptive opinion spam detection, *Lecture Notes in Computer Science*, pp. 393–404, 2015.

[11] X. P. Wang, K. Liu, S. Z. He and J. Zhao, "Learning to represent review with tensor decomposition for spam detection," in *Proc. of the 2016 Conf. on Empirical Methods in Natural Language Processing*, Austin, Texas, USA, pp. 866–875, 2016.

[12] R. Y. K. Lau, S. Y. Liao, R. C. W. Kwok, K. Q. Xu, Y. Q. Xia *et al.,* "Text mining and probabilistic language modeling for online review spam detection," *ACM Transactions on Management Information Systems*, vol. 2, no. 4, pp. 1–30, 2011.

[13] W. Lu, X. Zhang, H. Lu and F. Li, "Deep hierarchical encoding model for sentence semantic matching," *Journal of Visual Communication and Image Representation*, vol. 71, 102794, 2020.

[14] A. Mukherjee, V. Venkataraman, B. Liu and N. Glance, "Fake review detection: Classification and analysis of real and pseudo reviews," *Technical Report*, vol. 80, no. 2, pp. 159–169, 2013.

[15] F. Yu, L. Liu, H. Shen, Z. Zhang, Y. Huang *et al.,* "Dynamic analysis, circuit design and synchronization of a novel 6D memristive four-wing hyperchaotic system with multiple coexisting attractors," *Complexity*, vol. 2020, pp. 5904607, 2020.

[16] L. Xiang, Y. Li, W. Hao, P. Yang and X. Shen, "Reversible natural language watermarking using synonym substitution and arithmetic coding," *Computers, Materials & Continua*, vol. 55, no. 3, pp. 541–559, 2018.

[17] Y. Luo, J. Qin, X. Xiang, Y. Tan, Q. Liu *et al.,* "Coverless real-time image information hiding based on image block matching and dense convolutional network," *Journal of Real-Time Image Processing*, vol. 17, no. 1, pp. 125–135, 2020.

[18] J. Wang, C. Yang, P. Wang, X. Song and J. Lu, "Payload location for JPEG image steganography based on co-frequency sub-image filtering," *International Journal of Distributed Sensor Networks*, vol. 16, no. 1, pp. 1–16, 2020.

[19] B. Qi, C. Yang, L. Tan, X. Luo and F. Liu, "A novel haze image steganography method via cover-source switching," *Journal of Visual Communication and Image Representation*, vol. 70, pp. 102814, 2020.

[20] Y. Tan, J. Qin, X. Xiang, W. Ma, W. Pan *et al.,* "A robust watermarking scheme in YCbCr color space based on channel coding," *IEEE Access*, vol. 7, no. 1, pp. 25026–25036, 2019.

[21] N. Jindal and B. Liu, "Opinion spam and analysis," in *Proc. of the Int. Conf. on Web Search and Web Data Mining*, New York, NY, USA, pp. 219–230, 2008.

[22] N. Jindal, B. Liu and E. P. Lim, "Finding unusual review patterns using unexpected rules," in *Proc. of the 19th ACM Int. Conf. on Information and Knowledge Management*, Toronto, Canada, pp. 1549–1552, 2010.

[23] A. Mukherjee, V. Venkataraman, B. Liu and N. Glance, "What yelp fake review filter might be doing?," in *Proc. of the 7th Int. Conf. on Weblogs and Social Media*, Cambridge, Massachusetts, USA, pp. 409–418, 2013.

[24] S. Kim, H. Chang, S. Lee, M. Yu and J. Kang, "Deep semantic frame-based deceptive opinion spam analysis," in *Proc. of the 24th ACM Int. on Conf. on Information and Knowledge Management*, Melbourne, Australia, pp. 1131–1140, 2015.

[25] Y. F. Ren and Y. Zhang, "Deceptive opinion spam detection using neural network," in *26th Int. Conf. on Computational Linguistics*, Osaka, Japan, pp. 140–150, 2016.

[26] E. P. Lim, V. A. Nguyen, N. Jindal, B. Liu and H. W. Lauw, "Detecting product review spammers using rating behaviors," in *Proc. of the 19th ACM Int. Conf. on Information and Knowledge Management*, Toronto, Canada, pp. 939–948, 2010.

[27] S. H. Xie, G. Wang, S. Y. Lin and P. S. Yu, "Review spam detection via temporal pattern discovery," in *Proc. of the 18th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Beijing, China, pp. 823–831, 2012.

[28] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu *et al.,* "Spotting opinion spammers using behavioral footprints," in *Proc. of the 19th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Chicago, Illinois, USA, pp. 632–640, 2013.

[29] L. Akoglu, R. Chandy and C. Faloutsos, "Opinion fraud detection in online reviews by network effects," in *Proc. of the 7th Int. Conf. on Weblogs and Social Media*, Cambridge, Massachusetts, USA, pp. 2–11, 2013.

[30] H. Y. Li, Z. Y. Chen, A. Mukherjee, B. Liu and J. D. Shao, "Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns," in *Proc. of the 9th Int. Conf. on Web and Social Media*, Oxford, UK, pp. 634–637, 2015.

[31] S. Rayana and L. Akoglu, "Collective opinion spam detection: Bridging review networks and metadata," in *Proc. of the 21th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Sydney, NSW, Australia, pp. 985–994, 2015.

[32] L. Xiang, G. Guo, J. Yu, V. S. Sheng and P. Yang, "A convolutional neural network-based linguistic steganalysis for synonym substitution steganography," *Mathematical Biosciences and Engineering*, vol. 17, no. 2, pp. 1041–1058, 2020.

[33] J. Wang, J. H. Qin, X. Y. Xiang, Y. Tan and N. Pan, "CAPTCHA recognition based on deep convolutional neural network," *Mathematical Biosciences and Engineering*, vol. 16, no. 5, pp. 5851–5861, 2019.

[34] Z. Y. Xiong, Q. Q. Shen, Y. J. Wang and C. Y. Zhu, "Paragraph vector representation based on word to vector and cnn learning," *Computers, Materials & Continua*, vol. 55, no. 2, pp. 213–227, 2018.

[35] Y. T. Zhang, W. P. Lu, W. H. Ou, G. Q. Zhang, X. Zhang *et al.,* "Chinese medical question answer selection via hybrid models based on CNN and GRU," *Multimedia Tools and Applications*, vol. 79, no. 21–22, pp. 14751–14776, 2020.

[36] Y. T. Chen, J. J. Tao, L. W. Liu, J. Xiong, R. L. Xia *et al.,* "Research of improving semantic image segmentation based on a feature fusion model," *Journal of Ambient Intelligence and Humanized Computing*, 2020.

[37] Z. W. Qu, B. Y. Cao, X. R. Wang, F. Li, P. Q. Xu *et al.,* "Feedback LSTM network based on attention for image description generator," *Computers, Materials & Continua*, vol. 59, no. 2, pp. 575–589, 2019.

[38] Y. T. Shen, Y. Li, J. Sun, W. K. Ding, X. J. Shi *et al.,* "Hashtag recommendation using LSTM networks with self-attention," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1261–1269, 2019.

[39] B. X. Tang, G. Ya, Z. Yu, E. S. Pan and L. F. Li, "An ensemble framework based on convolutional bi-directional lstm with multiple time windows for remaining useful life estimation," *Computers in Industry*, vol. 115, pp. 103182, 2020.

[40] J. Deng, B. Schuller, F. Eyben, D. Schuller, Z. X. Zhang *et al.,* "Exploiting time-frequency patterns with LSTM-RNNs for low-bitrate audio restoration," *Neural Computing and Applications*, vol. 32, no. 4, pp. 1095–1107, 2020.

[41] B. W. Wang, W. W. Kong, H. Guan and N. X. Xiong, "Air quality forecasting based on gated recurrent long short term memory model in internet of things," *IEEE Access*, vol. 7, pp. 69524–69534, 2019.

[42] F. T. Li, M. L. Huang, Y. Yang and X. Y. Zhu, "Learning to identify review spam," in *IJCAI Int. Joint Conf. on Artificial Intelligence*, Barcelona, Catalonia, Spain, pp. 2488–2493, 2011.

[43] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space," in *1st Int. Conf. on Learning Representations*, Scottsdale, AZ, USA, 2013.

[44] E. P. Matthew, N. Mark, I. Mohit, G. Matt, C. Christopher *et al.,* "Deep contextualized word representations," in *2018 Conf. of the North American Chapter of the Association for Computational Linguistics*, New Orleans, Louisiana, pp. 2227–2237, 2018.

[45] J. Pennington, R. Socher and C. Manning, "GloVe: Global vectors for word representation," in *2014 Conf. on Empirical Methods in Natural Language Processing*, Doha, Qatar, pp. 1532–1543, 2014.

[46] J. Wang, Y. Zou, L. Peng, L. Wang, O. Alfarraj *et al.,* "Research on crack opening prediction of concrete dam based on recurrent neural network," *Journal of Internet Technology*, vol. 21, no. 4, pp. 1161–1170, 2020. https://jit.ndhu.edu.tw/article/view/2343

[47] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space," in *1st Int. Conf. on Learning Representations*, Scottsdale, AZ, USA, 2013.