Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 7562567, 12 pages
https://doi.org/10.1155/2020/7562567

Research Article

An Adaptive Deep Transfer Learning Model for Rumor Detection without Sufficient Identified Rumors

Meicheng Guo,[1] **Zhiwei Xu**,[2,3] **Limin Liu**,[2] **Mengjie Guo**,[3] **and Yujun Zhang**[3]

[1]*College of Information Engineering, Inner Mongolia University of Technology, Hohhot 100080, China*
[2]*College of Data Science and Application, Inner Mongolia University of Technology, Hohhot 100080, China*
[3]*Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China*

Correspondence should be addressed to Zhiwei Xu; xuzhiwei2001@ict.ac.cn

Received 27 April 2020; Accepted 17 June 2020; Published 17 July 2020

Academic Editor: Sotiris B. Kotsiantis

With the extensive usage of social media platforms, spam information, especially rumors, has become a serious problem of social network platforms. The rumors make it difficult for people to get credible information from Internet and cause social panic. Existing detection methods always rely on a large amount of training data. However, the number of the identified rumors is always insufficient for developing a stable detection model. To handle this problem, we proposed a deep transfer model to achieve accurate rumor detection in social media platforms. In detail, an adaptive parameter tuning method is proposed to solve the negative transferring problem in the parameter transferring process. Experiments based on real-world datasets demonstrate that the proposed model achieves more accurate rumor detection and significantly outperforms state-of-the-art rumor detection models.

1. Introduction

With the rapid development of mobile Internet technology, online social networking (OSN), a novel information publishing and sharing platform, has become an essential part of our daily life. Some OSN platforms, such as Facebook, Twitter, Weibo, WeChat, and other social networking platforms, have triggered a media revolution with the interactivity, immediacy, and diversity, which have profoundly affected all aspects of our society and economy. The existence of false information makes it difficult for OSN users to obtain credible information on OSN platforms. Rumors are the most common false information, which are false messages that spread among a large amount of people and have mislead these people [1]. Due to easy access to social media, rumors can spread extensively on social media, bringing huge harm to society and causing a lot of economic losses. For example, if there is a rumor about a bomb event in a hotel, the income of the hotel will suffer from the propagation of this rumor on social media platforms. Even worse, malicious rumors may seriously violate the opinions of OSN users, cause social panic, and even lead to a crisis of confidence. The rumors on OSN have become a serious social problem. However, it is unrealistic to rely on manual methods to identify and filter rumors, and the average accuracy of three human judges is only 57.33% [2]. Therefore, effective detection of rumors in OSN platforms is highly desired. The research studies on automatic detection of rumors have received increasing attention.

Most existing rumor detection methods employed learning algorithms that incorporated a wide variety of features to take rumor detection as a binary classification task [3]. These rumor detection models incorporated a wide variety of features of the text content [4], user characteristics [5], and diffusion patterns of the OSN messages [6] or simply exploited the patterns used in regular message to discover rumors [7]. These approaches aim at extracting distinctive features to describe rumors faithfully. However, these traditional machine learning methods fail to obtain an effective classification model when the features are sparse. Ma et al. [8] proposed an adaptive model and for the first time used the recurrent neural networks to achieve microblog rumor

detection. Recurrent neural network (RNN) models achieve significant improvements over state-of-the-art learning algorithms that rely on hand-crafted features. Chen et al. [3] introduced CallAtRumors, a novel recurrent neural network model based on soft attention mechanism to automatically carry out early rumor detection by learning latent representations from the sequential messages in OSN. Singh et al. [9] used a convolutional neural network (CNN) model to mine the semantic features of review texts, which was then used to identify false reviews. However, these neural network-based models require a large amount of training data, and the size of training datasets affects the accuracy of the model when the training data are insufficient.

Transfer learning (TL) is a branch of machine learning (ML) algorithms, which leverages the knowledge stored within a source domain and provides a method to transfer the knowledge of the source domain to a target domain [10]. At the same time, transfer learning has benefited many real-world applications where labelled data are abundant in source domains but scarce in the target domain [11]. Existing studies have already provided the evidences for applying TL on neural features. For the instance of image processing, the deep neural networks exhibit an interesting phenomenon that the model always tends to learn first-layer features that resemble either Gabor filters or color blobs [12, 13]. Donahue et al. [14] suggested that high-level layers are also transferable in general visual recognition. Mou et al. [15] further studied the transferability of neural layers. Semwal et al. [10] reported the results and conclusions obtained from extensive empirical experiments using a CNN and tried to uncover primary rules to ensure a meaningful transfer operation. The application of transfer learning in text classification provides new ideas for solving the problems when labelled texts are insufficient to support the training processes of models. However, without the knowledge about the difference between the source and the target domains, negative transferring [16] occurs when knowledge is transferred from different domains. Negative transferring refers to the phenomenon that instead of improving the classification accuracy of the models, transfer learning from other domains degrades the classification accuracy on the target domain. Despite the fact that how to avoid negative transferring is a very important issue, little research has been done on this research field.

The deep neural network incorporates the domain knowledge into the parameters of their nodes during the training process. We can transfer the related knowledge embedded in neural networks to the rumor detection domain by reusing the parameters of the neural networks. This paper proposes a deep transfer model based on CNN to approach an accurate rumor detection scheme. In detail, we propose a learning rate adaptive update method to solve the negative transferring problem in the transfer process.

The main contributions are listed as follows:

(1) A novel deep transfer model based on CNN for rumor detection is proposed, which can effectively identify rumors without sufficient training data. We evaluate that the knowledge related to large-scale datasets in the field of e-commerce reviews has similar features with the knowledge about the characteristics of rumors, which is used to train a model whose parameters is transferred to the rumor detection model.

(2) We propose a learning rate adaptive update method to solve the negative transfer problem during the parameter transfer process. In detail, based on the stochastic gradient descent algorithm, we achieve an adaptive learning rate updating method for fine-tuning the rumor detection model obtained in the transfer process.

(3) We implement the proposed detection scheme on an open source deep learning platform, TensorFlow [17]. The experiments based on real-world datasets demonstrate that under the interference of the common expressions frequently appearing both in rumors and regular messages, the proposed scheme achieves more accurate rumor detection compared with the existing rumor detection approaches.

The rest of the paper is organized as follows. In Section 2, we analyze the related work. Section 3 gives details of our proposed rumor detection model, and Section 4 provides the performance evaluation based on TensorFlow. Section 5 concludes our paper.

2. Related Work

In this section, we focus on providing a brief review of the work most closely related to effective and efficient rumor detection. We outline related research approaches in three fields: rumor detection, deep learning, and transfer learning.

2.1. Rumor Detection. Rumor is a powerful, pervasive, and persistent force that misleads people and groups [1]. Rumor detection has been a popular research topic in recent years. Rumor is a research subject in psychology and social cognition for a long time [18]. It is often viewed as an unverified account or explanation of events circulating from person to person and pertaining to an object, event, or issue in public concern [19]. The challenges of rumor detection, such as the veracity and accuracy of rumor with small data size, are discussed in [20–22]. Derczynski et al. [20] used journalism use case dataset [23] to accomplish support/rumor stance classification and veracity prediction. Reference [23] contains 330 conversational threads (297 in English for 8 events in total, and 33 in German), which includes 4,222 reply tweets. Ma et al. [21] created tweet dataset and microblog dataset by crawlers and other ways. For the Twitter data, they contain 498 rumors and 494 nonrumors. For Weibo data, they collected nonrumor and rumor data in Chinese. However, these datasets contains small-size data. In 2019, interest in automated claim validation has greatly increased. Gorrell et al. [22] extended the dataset compared with [23] in that the dataset is substantially expanded to include Reddit as well as Twitter data, and additional languages are also included. In addition, we consider fake news [24, 25], and

there are no agreed upon benchmark datasets for the fake news detection problem. Datasets mentioned in [26] cannot provide all possible features of interest, and these datasets also have specific limitation that make them challenging to use for fake news detection. BuzzFeedNews only contains headlines and text for each news piece and covers news articles from very few news agencies. LIAR includes mostly short statements, rather than the entire news content. BS Detector data are collected and annotated by using a developed news veracity checking tool, and the labels have not been properly validated by human experts. The tweets in CREDBANK are not really the social engagements for specific news articles. To address the disadvantages of above fake news detection datasets, Shu et al. [26] have an ongoing project to develop a usable dataset, called FakeNewsNet, for fake news detection on social media. It includes all mentioned news content and social context features with reliable ground truth fake news labels. However, the free data site is no longer available or the original fake news is not public. Therefore, the dataset available for rumor detection is insufficient.

Early exploration started from two special studies on rumor propagation during natural disasters like earthquakes and hurricanes [19, 27]. Castillo et al. [28] selected four types of features, namely, message-based features, user-based features, topic-based features, and propagation pattern-based features, and then used the J48 decision tree to detect rumors in Twitter. Zhang et al. [29] considered heterogeneous network and analyzed the structure of the information diffusion graph of mobile social network (MSN) to learn the latent factors of each piece of information and proposed a diffusion model to explain the spread of information in MSN. In [30], the major difference between rumors and nonrumors was discussed. The existing rumor detection methods mainly include the rumor detection methods based on traditional machine learning [4, 5, 7] and the more accurate rumor detection methods based on neural network models [3, 6, 8, 31]. Tian et al. [32] proposed to learn user attitude distribution for Twitter posts from their comments and then combined it with content analysis for early detection of rumors based on huge models when data for information sources or propagation are scarce. However, these existing rumor detection methods rely on a large amount of labelled training data or huge models for modeling. The number of the labelled rumors is always insufficient to train any of the existing neural network-based detection models to cover the characteristics of various rumors and cannot accurately detect rumors.

2.2. Deep Learning. Deep learning models simulate the human brain's thinking patterns to discover various characteristics of texts. Therefore, the accuracy of deep learning models is often higher than that of the traditional rumor detection technology. Recently, deep neural networks are emerging as the prevailing technical solution to almost all fields in natural language processing (NLP). Word embedding is the basis for deep learning to solve many natural language processing problems [33]. Liu et al. [31] used a CNN model to abstract textual and temporal information in social media and exploited postlevel textual information to generate group embedding for further analysis. Kim [34] applied CNN in text categorization. Experiments have shown that the CNN text categorization model can obtain higher accuracy than other machine learning models. However, these deep learning models are not easy to converge due to the huge number of their parameters.

2.3. Transfer Learning. Transfer Learning can alleviate the lack of labelled training data for training a deep learning model [10]. As illustrated in Figure 1, if the training data are insufficient, the characteristics of the features in the source domain cannot be identified by the deep layers, and transfer learning can be used to improve the accuracy of models in a domain by transferring knowledge from the related domains [10]. In [15], the evidence has been discovered, which shows that TL in NLP applications is more sensitive to the text semantics. While TL has produced positive results within the domain of image processing, its usage in NLP applications still remains a fairly unexplored research area. Yang and Zhang [33] proposed a transfer learning algorithm called automatic transfer learning (AutoTL) for short text mining. Johnson and Zhang [35] accomplished a semisupervised framework to improve the text classification accuracy by integrating knowledge from word vectors learned on unlabelled data. Do and Gaspers [36] achieved a considerable improvement in the accuracy of a language understanding task by initializing the parameters with an additional unlabelled dataset.

In view of the excellent performance of transfer learning for constructing a deep learning model without sufficient training data, this paper proposes a scheme for rumor detection based on transfer learning in the next section.

3. Rumor Detection Model Based on Deep Transfer Learning

After the deep learning model completes its training process, the domain knowledge will be fixed into the model parameters. When the training data are insufficient, an effective training model cannot be obtained, as shown in Figure 2. In view of this essential characteristic, we propose a rumor detection scheme based on parameter transferring.

In this section, we propose a deep transfer model, namely, TL-CNN. It achieves accurate rumor detection by using review evaluation knowledge in the e-commerce domain. In detail, based on the stochastic gradient descent algorithm, we propose an adaptive learning rate updating method for fine-tuning of the model obtained in the transfer process. The overall framework of the model is illustrated in Figure 3. A basic detection model has the same structure as the model used in the rumor detection process. Firstly, the basic detection model transfers its model parameters obtained in the training process on the polarity review data to the rumor detection model. The basic convolutional neural network-based detection model is proposed in Section 3.1, which is illustrated in Figure 4. In addition, we adapt the

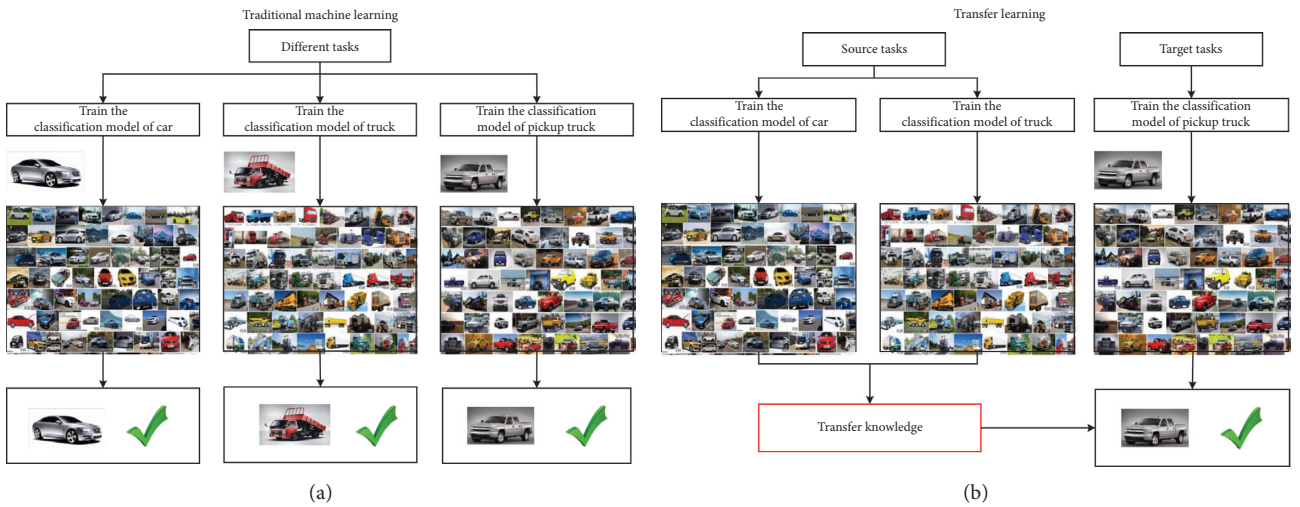


Figure 1: Difference between transfer learning and traditional machine learning.

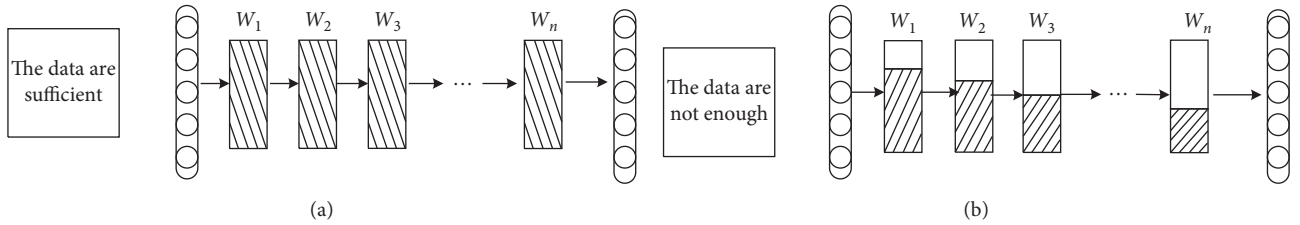


Figure 2: The effect of the training dataset size.

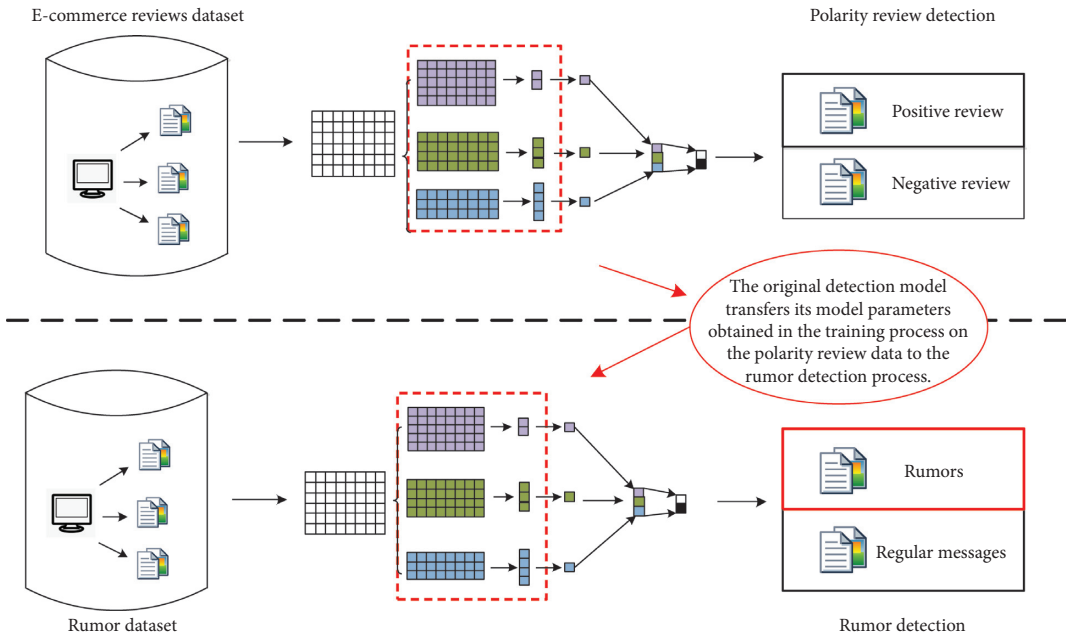


Figure 3: The framework of TL-CNN.

parameters of the basic detection model to the rumor detection process by performing a fine-tuning operation in Section 3.1. In this way, we can obtain an effective rumor detection model based on transfer deep learning.

3.1. Basic Detection Model. The convolutional neural network (CNN) model is originally proposed in computer vision and is proven to be effective in natural language processing, semantic analysis, and other traditional NLP

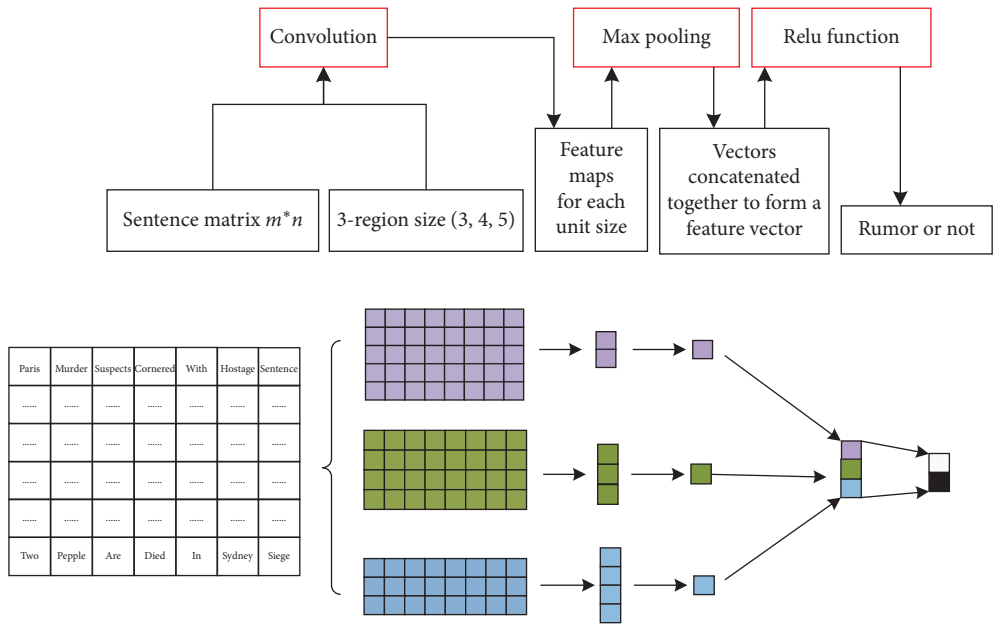


Figure 4: Basic detection model.

tasks [34]. It has been extensively applied in text classification. As a common detection model, a basic CNN-based detection model is presented in this section. It is feasible to collect review texts with polarity data on e-commerce platforms, and the polarity data provide a guideline for us to evaluate whether the corresponding texts are rational. Therefore, we can use review texts with polarity data to train the basic detection model, and the parameters of this model can be transferred to the rumor detection model to handle the problem of training data insufficiency. The basic detection model consists of five components: an embedded layer, a convolutional layer, a pooling layer, a fully connected layer, and an output layer. Among them, the convolutional layer, the pooling layer, and the fully connected layer are used to collect and mine features in the training data, the parameters of which can be used in transfer learning process in order to construct an accurate rumor detection model. The basic detection model is illustrated in Figure 4, and the configuration of this model is depicted in Table 1.

3.1.1. Embedding Layer. The embedding layer is the first layer of the basic detection model, which is used to preprocess the raw data. In detail, the embedding layer formulates the original input data as a matrix. For a task of text classification, a sentence will be represented with a vector of the identifiers of words, which is named as a word vector. All the input data of the model consist of a $n \times m$ matrix, namely, input matrix, where n is the number of sentences and m is the dimension of the word vector. The process of text preprocessing can be formulated with formula (1), where X represents the input matrix.

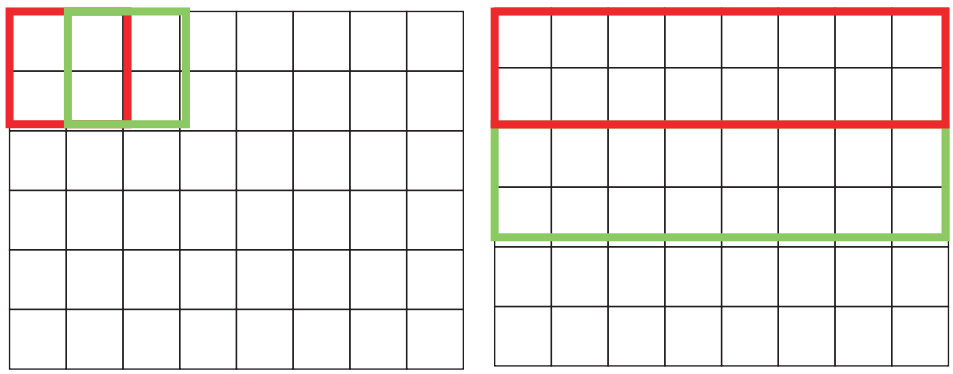
$$X = \left(\overrightarrow{x_1}, \overrightarrow{x_2}, \ldots, \overrightarrow{x_n}\right)^T,$$
$$\overrightarrow{x_i} = \left(x_{i,1}, x_{i,2}, \ldots, x_{i,m}\right). \quad (1)$$

Table 1: Parameter configuration.

Attribute	Value
Convolutional units	3, 4, 5
Feature maps	100
Activation function	Softmax
Pooling	1-max pooling
Dropout rate	0.5
l2 norm constraint	3
Batch_size	50

3.1.2. Convolutional Layer. The convolutional layer is the elemental layer of a convolutional neural network. The convolutional layer consists of several convolutional units, and the parameters of each convolutional unit are optimized by a backpropagation algorithm. Each convolutional unit can cover a part of the input matrix.

The difference between our convolutional layer and the existing convolutional layer is illustrated in Figure 5. It is worth nothing that the existing convolutional neural network model traverses all types of convolutional units for mining the features of the inputted texts (Figure 5(a)). In our convolutional layer (Figure 5(b)), since each line of the input matrix represents a sentence of a text, the width of convolutional units is configured according to the width of the input matrix. The purpose of the convolution operation is to extract different local features of the inputted sentences. For obtaining the processing result of each convolution unit, we use $x_{i,j}$ to represent the word at row i and column j, use $w_{m,n}$ to denote the weight of the word at row m and column n, and use b to denote the bias of this convolutional unit. Each convolution result for different convolution units consists a matrix, namely, feature map. The element at row i and column j of the feature map, $a_{i,j}$, is obtained with an activation function f.



(a) (b)

Figure 5: Embedding layer. (a) Traditional convolutional neural network. (b) Our convolutional neural network.

Ultimately, the corresponding eigenvalue of the convolution unit as well as the corresponding eigenvector is obtained. In detail, we use formula (2) to calculate the convolution units, where the results for the convolution units of size 3, 4, and 5 are indicated by $a_{i,j}^{(3)}$, $a_{i,j}^{(4)}$, and $a_{i,j}^{(5)}$, respectively.

$$\begin{aligned} a_{i,j}^{(3)} &= f\left(\sum_{k=0}^{2} w_{m,k} x_{i+m,j+k} + b\right), \\ a_{i,j}^{(4)} &= f\left(\sum_{k=0}^{3} w_{m,k} x_{i+m,j+k} + b\right), \\ a_{i,j}^{(5)} &= f\left(\sum_{k=0}^{4} w_{m,k} x_{i+m,j+k} + b\right). \end{aligned} \tag{2}$$

The feature values obtained on the convolution units of a specific size comprise the corresponding feature matrix.

$$\begin{aligned} A^{(3)} &= \begin{Bmatrix} a_{1,1}^{(3)} & \cdots & a_{1,n}^{(3)} \\ \vdots & \ddots & \vdots \\ a_{m,1}^{(3)} & \cdots & a_{m,n}^{(3)} \end{Bmatrix}, \\ A^{(4)} &= \begin{Bmatrix} a_{1,1}^{(4)} & \cdots & a_{1,n}^{(4)} \\ \vdots & \ddots & \vdots \\ a_{m,1}^{(4)} & \cdots & a_{m,n}^{(4)} \end{Bmatrix}, \\ A^{(5)} &= \begin{Bmatrix} a_{1,1}^{(5)} & \cdots & a_{1,n}^{(5)} \\ \vdots & \ddots & \vdots \\ a_{m,1}^{(5)} & \cdots & a_{m,n}^{(5)} \end{Bmatrix}. \end{aligned} \tag{3}$$

3.1.3. Pooling Layer. To remove trivial eigenvalues from the feature map, a max pooling-based layer is used to reduce the number of features, which simplifies the computational complexity of CNN and reduces the overfitting rate. Max pooling operation is the most popular pooling operation, which will take the maximum values in the feature map after performing a dot product on the weight matrix and the feature map. The weight matrix is valuable for obtaining the most important features included in the feature map. This paper uses the maximum pooling operation to process the results of convolution layer and obtains a brief semantic representation of the inputted texts, A, as shown in the following formula:

$$A = \begin{cases} a^{(3)} = \max\left(a_{i,j}^{(3)}\right), \\ a^{(4)} = \max\left(a_{i,j}^{(4)}\right), \\ a^{(5)} = \max\left(a_{i,j}^{(5)}\right), \\ \cdots. \end{cases} \tag{4}$$

3.1.4. Fully Connected Layer. The fully connected layer is a regular hidden layer of a multilayer neural network which makes higher order decisions. It receives inputs from the pooling layer. To avoid suffering from overfitting, a Softmax-based fully connected layer is included in the basic detection model. Softmax operation randomly discards some inputs from the pooling layer.

The Softmax operation pooling obtains the most important features that contribute to the classification process, connecting the overall features of the inputted texts.

3.1.5. Output Layer. The output layer is responsible for obtaining the final detection result. The weight matrix of this layer is highly related to the characteristics of the detection targets and is invaluable for transfer learning. The output layer is defined as follows:

$$y = WA + b. \tag{5}$$

Normalize y using the Softmax function to get the probability whether a text D belongs to a specific category.

$$P(g \mid D) = \frac{\exp\left(y_g\right)}{\exp\left(y_0\right) + \exp\left(y_1\right)}, \tag{6}$$

where g is equal to 0 or 1, y_0 means D does not belong to this category, and y_1 means D belongs to this category.

3.2. Rumor Detection Model Based on Parameter Transferring. The transfer learning algorithms reuse the existing knowledge to the target domain in order to solve the problem of training data (i.e., labelled data) shortage. The hierarchical architecture of the deep neural network model is very suitable for transfer learning [15]. We use the parameter transfer method to solve the problem of training data shortage in the rumor detection domain. In detail, we construct an original rumor detection model by reusing the parameters of the aforementioned basic detection model (Figure 6(a)). In order to avoid the negative transferring under limited amount of labelled rumors, we further fine-tune the original rumor detection model with a layerwise scheme (Figure 6(c)).

3.2.1. Original Rumor Detection Model Based on Parameter Transferring. As illustrated in Figure 6, we initialize an original rumor detection model (TL-CNN-Strawman) by reusing the parameters of the basic detection model, where the rumor detection model has the same structure as the basic detection model proposed in Section 3.1. The accuracy rates of the basic detection model and TL-CNN-Strawman are listed in Table 2, where accuracy rate is equal to the ratio of the messages correctly classified to the total number of messages. In a straightforward way, the basic detection model is trained on a review dataset YELP-2, and TL-CNN-Strawman is initialized by reusing the parameters of the basic detection model. After cloning the parameters of the basic detection model, TL-CNN-Strawman is additionally trained on a small rumor dataset (FBN).

As depicted in Table 2, the accuracy of the basic detection model on a review dataset YELP-2 is 89.71%, which demonstrates that the basic detection model can effectively detect the reviews with different polarities. However, negative transferring occurs when the parameters of the basic detection model are transferred to TL-CNN-Strawman, the accuracy of which is only 67.85%, since the labelled rumors of FBN are insufficient for fine-tuning TL-CNN-Strawman. Without an effective fine-tuning method, the rumor detection accuracy of TL-CNN-Strawman is lower. Negative transferring occurred during reusing the parameters of the basic detection model in TL-CNN-Strawman.

In order to solve this problem, we need to fine-tune the hyperparameters in the model training process to avoid negative transferring, instead of reusing the parameters of the basic detection model in a straightforward way. As depicted in Figure 6(b), the parameters of TL-CNN-Strawman could be left unstable if the labelled data are insufficient for fine-tuning the model. In Figure 6(c), the parameters of a frozen layer will be skipped during the learning process, and we will focus on training other parameters. Since less parameters will be learned, the fine-tuning process will converge in a short time. By applying this layerwise fine-tuning mechanism in the training process of our rumor detection model, we can tune different layers more efficiently and handle the negative transferring problem.

3.2.2. Adaptive Layerwise Fine-Tuning of Learning Rate. To achieve an effective layerwise tuning scheme, we analyze the effect of the hyperparameters in the training process. Among them, the learning rate is the most important hyperparameter that is related to the efficiency of the model training process and the accuracy of the model training results. With an applicable learning rate, we can obtain an accurate model as soon as possible. If the learning rate is too high, the model will miss the optimal point and need multiple iterations to reach convergence. On the other hand, a low learning rate always is related to a longer training process and causes the model to fall into a local optimal point. Different layers in the neural network can acquire different types of features, and thus the parameters of these layers should be tuned with different learning rates. With the limited amount of training data in rumor detection domain, we apply discriminative fine-tuning to configure each layer with different learning rates, instead of using the same learning rate for all layers of the rumor detection model.

To discover the optimal learning rate for each layer, we propose a learning rate updating scheme to update the learning rate in a reasonable way. In detail, stochastic gradient descent (SGD) [37] is applied to adapt the learning rate to the training process of a specific layer l. In this way, loss function $L(w)$ can reduce more rapidly, and the layerwise training can converge only with a limited number of labelled rumors.

The adaptive learning rate updating rule is as follows:

(1) Updating learning rate μ for the t-th iterate.

$$\nu_t = \beta\nu_{t-1} + (1-\beta)\nabla L\left(w_{t-1}; x_{t-1}\right)^2, \tag{7}$$

where ν_t is the moving average of uncentered variance over past first-order gradient of the loss function $\nabla L(w_{t-1}; x_{t-1})$, β is the decay rate for computing ν_t, w_{t-1} is the parameter vector at time t, and x_{t-1} is the input from the last layer in the $t-1$-th literate.

$$\mu_t = \frac{\mu_{t-1}}{\sqrt{\nu_t + \varepsilon}}, \tag{8}$$

where μ_t is the learning rate in the t-th literate and ε is the small hyperparameter for obtaining the stable convergence. The learning rate, μ_t, is divided by magnitude $\sqrt{\nu_t}$ of the past first-order gradient of the loss function. Intuitively, if the parameter vector has large value of $\nabla L(w_{t-1}; x_{t-1})$ in terms of the magnitude in the past, the next literate yields a small learning rate because $\sqrt{\nu_{i,t}}$ in equation (8) is large.

(2) Updating the weight of this layer l.

$$w_t = w_{t-1} - \frac{\mu_{t-1}}{\sqrt{\nu_t + \varepsilon}}\nabla L\left(w_{t-1}; x_{t-1}\right). \tag{9}$$

The detailed updating process is introduced in Algorithm 1. We first initialize the number of Batch_size b, decay rate β, weight w, gradient ν, learning rate μ, and hyperparameter ε. The loop from line 2 to line 9 is

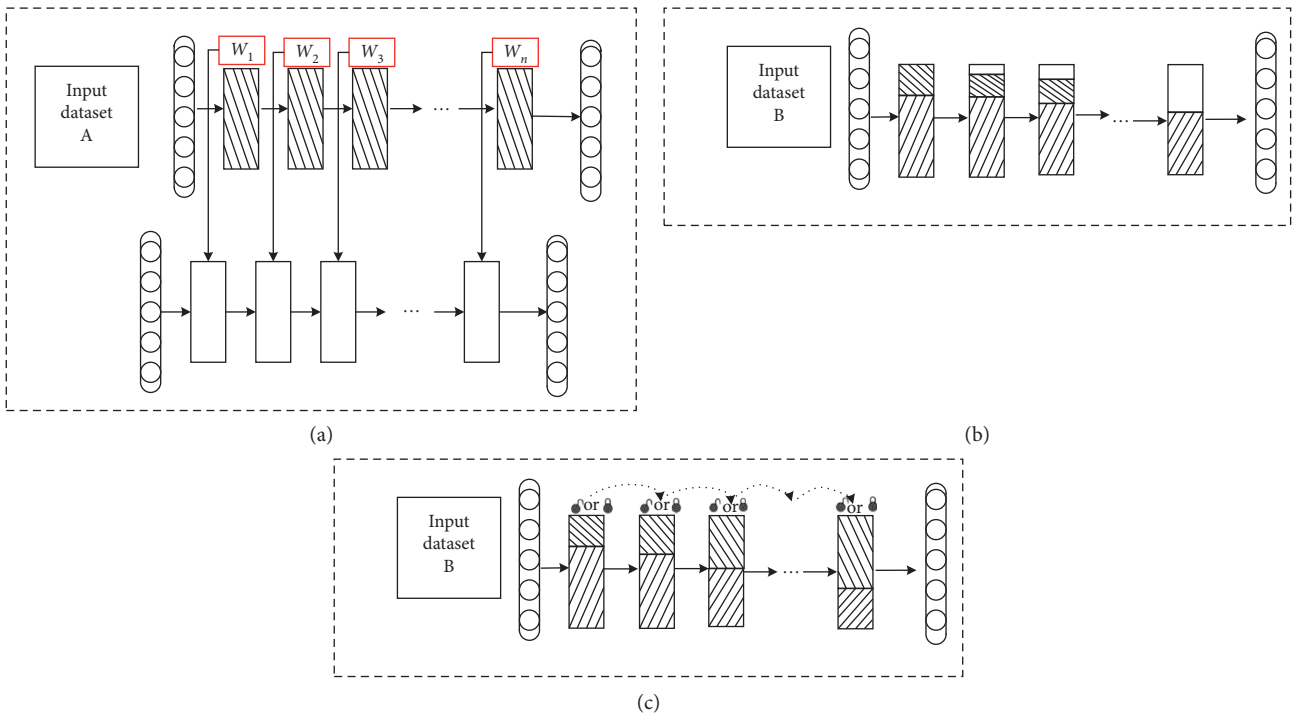


Figure 6: Transfer learning process.

Table 2: Accuracy rates of the original detection models.

Model	Dataset	Accuracy (%)
Basic	YELP-2	89.71
TL-CNN-Strawman	FBN	67.85

performed repeatedly before convergence. In line 3, the output of the last layer, $x = \{x_1, x_2, \ldots, x_b\}$, is obtained. In line 4, the loss function for this layer, $L_i(w) = L_i(y_i, f(x_i, w))$, is calculated. In line 5, the moving average of uncentered variance over past first-order gradient, $\nabla L_i(w_{i,t-1}; x_{t-1})$, is obtained. As the iteration process continues, the value of the loss function continuously decreases, and the learning rate μ is updated in line 6. Additionally, we update the weights according to the gradient in lines 7 and 8. By applying the aforementioned fine-tuning method into the original rumor detection model (TL-CNN-Strawman), an effective rumor detection model (TL-CNN) is obtained.

4. Experiments

In order to evaluate the proposed rumor detection scheme, we implemented the proposed rumor detection scheme and baseline schemes on TensorFlow. TensorFlow is a machine learning system that operates at large scale and heterogeneous environments. It is the second generation of artificial intelligence learning system developed by Google. On the real-world datasets, we comprehensively compare the proposed scheme and the baseline schemes in terms of different accuracy metrics.

4.1. Datasets

4.1.1. Yelp Polarity (YELP-2). The Yelp review dataset was obtained from the 2015 Yelp dataset challenge [38]. The dataset contains 1,569,264 samples with review texts. Two tasks are performed on this dataset. The first one predicts the total number of stars given by the user, and the other predicts the polarity label by considering stars, 1 and 2 being negative and 3, 4, and 5 being positive. This dataset includes 130,000 training samples and 10,000 test samples for each state of ranking and has 280,000 training samples and 19,000 test samples for each polar state.

4.1.2. Five Breaking News (FBN). The FBN dataset is a small rumor dataset that is about all five events and includes 5,802 labelled tweets [39]. The five events are Ferguson unrest, Ottawa shooting, Sydney siege, Charlie Hebdo shooting, and Germanwings plane crash.

These two datasets are used, respectively, in the basic detection model and the rumor detection model obtained in the transfer learning process, which is named as D_s and D_t. The detailed information of these two datasets is listed in Table 3.

4.2. Implementation. We provide the relevant parameters used in the proposed model in Table 1. The size of convolutional units is configured to be 3, 4, and 5. Each convolutional unit relates to a feature map. The dropout rate of the fully connected layer is 0.5. The L2 regular term with a coefficient 3 is used in the Softmax. During the training process, the batch size is configured as 50.

Require:
Batch_size b;
decay rate β;
weight parameter w;
gradient parameter ν;
Learning rate μ;
Hyperparameter ε;
(1) initialize parameter w, ν, μ;
(2) **while** no convergence **do**
(3) Obtain $x = \{x_1, x_2, \ldots, x_b\}$ from the last layer.
(4) Calculate the loss function $L(w_{t-1}; x_{t-1})$;
(5) Calculate the moving average of uncentered variance over past first-order gradient of the loss function $\nu_t = \beta\nu_{t-1} + (1-\beta)\nabla L(w_{t-1}; x_{t-1})^2$
(6) Update the learning rate update: $\mu_t = (\mu_{t-1}/\sqrt{\nu_t + \varepsilon})$
(7) Calculate the past first-order gradient of weights: $\Delta w_{t-1} = -(\mu_{t-1}/\sqrt{\nu_t + \varepsilon})\nabla L(w_{t-1}; x_{t-1})$
(8) Update weights of this layer: $w_t = w_{t-1} + \Delta w_{t-1}$
(9) **end while**

Algorithm 1: Adaptive learning rate update algorithm.

Table 3: Statistics for the datasets.

Datasets	Domain type	Class	Size
YELP-2	D_s	Positive/negative review	1,569,264
FBN	D_t	Regular message/rumor	5,802

We initially train the basic detection model on YELP-2 dataset. After the transfer learning process, we fine-tune the obtained rumor detection model on FBN dataset. To extensively evaluate the performance of the rumor detection model, we implement the proposed scheme (TL-CNN) on TensorFlow as well as three state-of-the-art baseline schemes. The detailed information of these three baseline schemes is introduced as follows:

(i) VDCNN-based model:

VDCNN is a convolutional neural network-based model proposed by Gereme and Zhu [40]. As the depth of the model increases, the accuracy of the solution can also increase.

(ii) Char-CNN-based model:

Based on a character-level convolutional network, Char-CNN was proposed by Joo and Hwang [41] to perform classification tasks, such as the Yelp polarity dataset and the Amazon review dataset.

(iii) RCNN-based model:

RCNN is a model proposed by Fang et al. [42], which essentially incorporates RNN and CNN into text categorization tasks. First, it applies a RNN model to capture context information as much as possible while learning word representation. To capture key features of the text, this model additionally uses the maximum pooling layer to automatically determine which words play a key role in text categorization.

4.3. Evaluation Metrics. We use accuracy rate, precision rate, recall rate, $F1$-measure and accuracy gain to evaluate the effectiveness of the proposed scheme.

Accuracy rate is the ratio of the messages correctly classified as rumors to the total number of messages.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \tag{10}$$

where TP, FP, FN, and TN are the abbreviations of true positives, false positives, false negatives, and true negatives, respectively.

Precision rate is calculated as the ratio of all messages correctly classified as rumors (TP) to all messages classified as rumors (TP + FP).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{11}$$

Recall is the ratio of all messages correctly classified as rumors (TP) to all messages that should be classified as rumors (TP + FN).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{12}$$

$F1$-measure is the harmonic mean of precision and recall.

$$F1\text{-measure} = \frac{2\text{PR}}{P + R} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \tag{13}$$

Accuracy gain (α) is calculated as the ratio of A_e to A_c, which is used to evaluate the accuracy increment of the proposed scheme relative to baselines.

$$\alpha = \frac{A_e}{A_c}, \tag{14}$$

where A_e represents the accuracy rate of our model and A_c represents the accuracy rate of baselines.

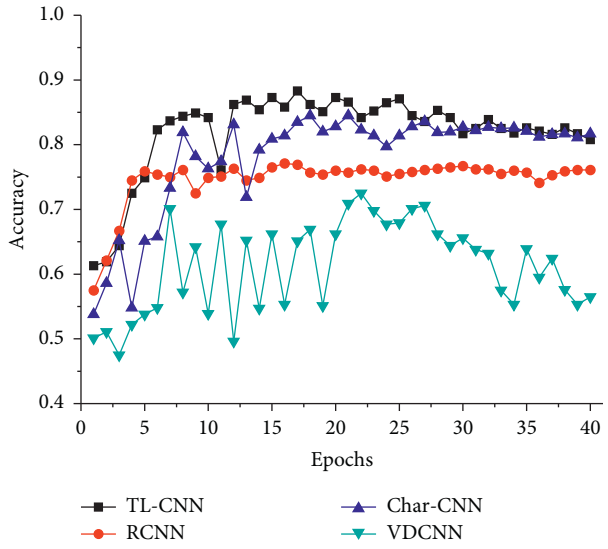


Figure 7: Trend of accuracy.

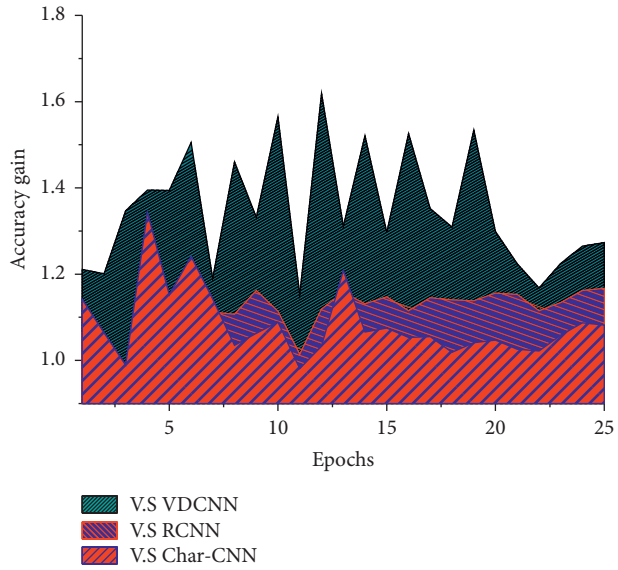


Figure 8: Trend of accuracy gain.

4.4. Performance Evaluation. To evaluate the accuracy of the proposed rumor detection model TL-CNN, we compare the rumor detection results of TL-CNN to the results of the baseline models. As depicted in Figure 7, TL-CNN obtains the best accuracy evaluation results. After the number of training epochs reaches twelve times, the accuracy rate of TL-CNN as well as the other three baseline models becomes stable. The maximum accuracy rate of TL-CNN is 87.28%, and the minimum accuracy rate is 79.91%. Compared with Char-CNN, RCNN, and VDCNN, the accuracy rate of TL-CNN has been improved by 4%, 7.7%, and 15.4%, respectively. Our model achieves more accurate rumor detection.

In Figure 8, we use accuracy gains to evaluate the improvement of accuracy rate compared with baselines. When the number of epochs is greater than 12, the accuracy gain values of TL-CNN with respect to Char-CNN and RCNN tend to be stable. From then on, although the accuracy gain value of TL-CNN relative to VDCNN fluctuates within a large range, all accuracy gains are higher than 1.10. Therefore, TL-CNN improves the accuracy of rumor detection.

A detailed result is listed in Table 4. The accuracy rate of VDCNN is only 71.88%. RCNN achieves a better result with an accuracy rate of 79.53%. The accuracy rate of Char-CNN is higher compared with the other two baselines, which is equal to 83.21%. Our model TL-CNN achieves the best results with an accuracy rate of 87.28%, which is 4% higher than Char-CNN. Similarly, the precision of VDCNN, RCNN, and Char-CNN is 60.59%, 76.92%, and 78.80%, respectively, and the precision rate of our model is 79.12%. In terms of recall rate, Char-CNN achieves the best results with a recall rate of 85.47%. The recall rate of our model is higher than the recall rates of both VDCNN and RCNN; however, it is lower than the recall rate of Char-CNN. The reason for this is that our model considers the lower false positives as the most important guiding principle during the rumor detection process. As a result, our model's recall rate is slightly lower than that of Char-CNN. TL-CNN achieves the best results in the $F1$-measure, which is the comprehensive metric for accuracy evaluation. The $F1$ value of our model is 0.825, while the $F1$ values of VDCNN, RCNN, and Char-CNN are 0.597, 0.799, and 0.819, respectively.

Table 4: Accuracy evaluation.

Model	Accuracy (%)	Precision (%)	Recall (%)	$F1$-measure
VDCNN	71.88	60.59	59.12	0.5973
RCNN	79.53	76.92	82.79	0.7997
Char-CNN	83.21	78.80	**85.47**	0.8195
TL-CNN	**87.28**	**79.12**	84.76	**0.8254**

Table 5: Efficiency evaluation.

Model	Training (min)	Test (data/s)
VDCNN	1482.58	3.69
RCNN	356.19	5.17
Char-CNN	275.83	6.78
TL-CNN	529.57	6.70

The proposed model was trained within 529.57 minutes on Yelp and FBN. It trains 6.6963 data per second and tests 2.4361 data per second in average. A detailed result is listed in Table 5. Although the training time of TL-CNN is longer than that of the baselines, it achieves more accurate detection results within similar testing time, even faster than VDCNN and RCNN.

5. Conclusion

In this paper, we present an effective deep transfer model based on convolutional neural network, TL-CNN, to detect rumors with limited amount of training data. To achieve that, considering the phenomenon of negative transferring

during the transfer learning process, we propose a learning rate adaptive tuning method to avoid negative transferring. The extensive experiments on the real-world datasets demonstrate that the proposed rumor detection model can significantly improve the accuracy of rumor detection, which can be applied to social media, e-commerce, and other fields.

Data Availability

Previously reported text data, Yelp and FBN, were used to support this study and are available at WOS: 000450913101042 and ArXiv:1610.07363v1. These prior studies (and datasets) are cited at relevant places within this paper as references [38, 39]. Among them, Yelp review dataset contains 1,569,264 items with review texts. The other dataset, FBN dataset, is a rumor dataset, which contains 5,802 labelled tweet messages, including five events, Ferguson unrest, Ottawa shooting, Sydney siege, Charlie Hebdo shooting, and Germanwings plane crash.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Key Research and Development Program of China (2018YFB1800403 and 2016YFE0121500), the National Natural Science Foundation of China (61902382, 61972381, 61672500, 61962045, 61502255, and 61650205), the Strategic Priority Research Program of Chinese Academy of Science (XDC0203500), and the Natural Science Foundation of Inner Mongolia Autonomous Region (2017MS(LH)0601 and 2018MS06003).

References

[1] L. Li, H. Xia, R. Zhang, and Y. Li, "DDSEIR: a dynamic rumor spreading model in online social networks," *Wireless Algorithms, Systems, and Applications*, vol. 11604, Springer, Berlin, Germany, 2019, Lecture Notes in Computer Science.

[2] 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-Volume 1*, Association for Computational Linguistics, Portland, OR, USA, pp. 309–319, June 2011.

[3] T. Chen, X. Li, H. Yin et al., "Call attention to rumors: deep attention based recurrent neural networks for early rumor detection," *Lecture Notes in Computer Science*, Vol. 11154, Springer, Berlin, Germany, 2018.

[4] D. Zimbra, M. Ghiassi, and S. Lee, "Brand-related twitter sentiment analysis using feature engineering and the dynamic architecture for artificial neural networks," in *Proceedings of the 2016 49th Hawaii International Conference on System Sciences (HICSS)*, pp. 1930–1938, IEEE, Koloa, HI, USA, January 2016.

[5] G. Liang, W. He, C. Xu, L. Chen, and J. Zeng, "Rumor identification in microblogging systems based on users' behavior," *IEEE Transactions on Computational Social Systems*, vol. 2, no. 3, pp. 99–108, 2015.

[6] L. Wu and L. Huan, "Tracing fake-news footprints: characterizing social media messages by how they propagate," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining—WSDM'18*, pp. 637–645, Angeles, CA, USA, February 2018.

[7] Z. Zhao, R. Paul, and Q. Mei, "Enquiring minds: early detection of rumors in social media from enquiry posts," in *Proceedings of the 24th International Conference on World Wide Web—WWW'15*, pp. 1395–1405, Florence, Italy, May 2015.

[8] J. Ma, W. Gao, P. Mitra et al., "Detecting rumors from microblogs with recurrent neural networks," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pp. 3818–3824, New York, NY, USA, July 2016.

[9] M. Singh, L. Kumar, and S. Sinha, "Model for detecting fake or spam reviews," *Advances in Intelligent Systems and Computing*, Vol. 653, Springer, Singapore, 2018.

[10] T. Semwal, G. Mathur, P. Yenigalla et al., "A practitioners' guide to transfer learning for text classification using convolutional neural networks," 2018, https://arxiv.org/abs/1801.06480.

[11] L. Ge, J. Gao, H. Ngo, K. Li, and A. Zhang, "On handling negative transfer and imbalanced distributions in multiple source transfer learning," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 7, no. 4, pp. 254–271, 2014.

[12] Y. Tamaazousti, H. Le Borgne, C. Hudelot, M. El Amine Seddik, and M. Tamaazousti, "Learning more universal representations for transfer-learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 99, 2019.

[13] G. Zongjiang, Y. Zhang, and Y. Li, "Extracting features from infrared images using convolutional neural networks and transfer learning," *Infrared Physics & Technology*, vol. 105, 2020.

[14] J. Donahue, Y. Jia, O. Vinyals et al., "DeCAF: a deep convolutional activation feature for generic visual, recognition," in *Proceedings of the International Conference on Machine Learning*, Atlanta, GA, USA, June 2013.

[15] L. Mou, Z. Meng, R. Yan et al., "How transferable are neural networks in NLP appications?," 2016, https://arxiv.org/abs/1603.06111.

[16] B. Cao, S. J. Pan, Yu Zhang, D.-Y. Yeung, and Q. Yang, "Adaptive transfer learning," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI'10)*, AAAI Press, Atlanta, GA, USA, pp. 407–712, July 2010.

[17] M. Abadi, P. Barham, J. Chen et al., "Tensorflow: a system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283, Savannah, GA, USA, November 2016.

[18] L. J. Postman, "The psychology of rumor," *Daonal Yhology*, SAGE Publications, Thousand Oaks, CA, USA, 1948.

[19] M. Mendoza, B. Poblete, and C. Castillo, "Twitter under crisis: can we trust what we RT?" in *Proceedings of the First Workshop on Social Media Analytics (SOMA'10)*, Association for Computing Machinery, New York, NY, USA, pp. 71–79, July 2010.

[20] L. Derczynski, K. Bontcheva, M. Liakata, R. Procter, G. Wong Sak Hoi, and A. Zubiaga, "SemEval-2017 task 8: RumourEval: determining rumour veracity and support for rumours," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada, August 2017.

[21] J. Ma, W. Gao, P. Mitra et al., "Detecting rumors from microblogs with recurrent neural networks," in *Proceedings of the IJCAI*, New York, NY, USA, July 2016.

[22] G. Gorrell, K. Bontcheva, D. Leon et al., "Determining rumour veracity and support for rumours," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, Minneapolis, MN, USA, June 2019.

[23] A. Zubiaga, M. Liakata, R. Procter, G. W. Sak Hoi, and T. Peter, "Analysing how people orient to and spread rumours in social media by looking at conversational threads," *PLoS One*, vol. 11, Article ID e0150989, 2016.

[24] C. Boididou, S. Papadopoulos, D. Dang-Nguyen et al., "Verifying multimedia use at mediaeval 2016," in *Proceedings of the Mediaeval Workshop*, Hilversum, The Netherlands, October 2016.

[25] O. Papadopoulou, G. Kordopatis-Zilos, M. Zampoglou et al., "Brenda starr at SemEval-2019 task 4: hyperpartisan news detection," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, Minneapolis, MN, USA, June 2019.

[26] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: a data mining perspective," *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, 2017.

[27] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi, "Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy," in *Proceedings of the 22nd International Conference on World Wide Web—WWW'13 Companion*, Association for Computing Machinery, New York, NY, USA, pp. 729–736, 2013.

[28] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the 20th International Conference on World Wide Web—WWW'11*, Association for Computing Machinery, New York, NY, USA, pp. 675–684, 2011.

[29] Y. Zhang, K. Bian, L. Chen, S. Dong, L. Song, and X. Li, "Early detection of rumors in heterogeneous mobile social network," in *Proceedings of the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, vol. 1, pp. 294–301, Guangzhou, China, June 2018.

[30] K. Sejeong, C. Meeyoung, and J. Kyomin, "Rumor detection over varying time windows," *PLoS One*, vol. 12, no. 1, Article ID e0168344, 2017.

[31] Y. Liu, X. Chen, Y. Rao et al., "Supervised group embedding for rumor detection in social media," *Web Engineering, Lecture Notes in Computer Science*, Vol. 11496, Springer, Cham, Switzerland, 2019.

[32] L. Tian, X. Zhang, Y. Wang, and H. Liu, "Early detection of rumours on twitter via stance transfer learning," *Lecture Notes in Computer Science*, In European Conference on Information Retrieval, pp. 575–588, Springer, Cham, Switzerland, 2020.

[33] L. Yang and J. Zhang, "Automatic transfer learning for short text mining," *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, no. 1, p. 42, 2017.

[34] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, October 2014.

[35] R. Johnson and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," *Advances in Neural Information Processing Systems*, vol. 28, pp. 919–927, 2015.

[36] Q. N. T. Do and J. Gaspers, "Cross-lingual transfer learning for spoken language understanding," 2019, https://arxiv.org/abs/1904.01825.

[37] F. Shang, K. Zhou, H. Liu et al., "VR-SGD: a simple stochastic variance reduction method for machine learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 1, pp. 188–202, 2020.

[38] X. Zhang, J. Zhao, and Y. Lecun, "Character-level convolutional networks for text classification," in *Proceedings of the Neural Information Processing Systems*, vol. 28, Montreal, Canada, 2015.

[39] A. Zubiaga, M. Liakata, and R. Procter, "Learning reporting dynamics during breaking news for rumour detection in social media," 2016, https://arxiv.org/abs/1610.07363.

[40] F. B. Gereme and W. Zhu, "Early detection of fake news"-before it flies high"," in *Proceedings of the 2nd International Conference on Big Data Technologies*, pp. 142–148, Jinan, China, August 2019.

[41] Y. Joo and I. Hwang, "Steve martin at SemEval-2019 task 4: ensemble learning model for detecting hyperpartisan news," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 990–994, Minneapolis, MN, USA, June 2019.

[42] Y. Fang, C. Zhang, C. Huang, L. Liu, and Y. Yang, "Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism," *IEEE Access*, vol. 7, pp. 56329–56340, 2019.