

An Opinion Spam Detection Method Based on Multi-Filters Convolutional Neural Network

Ye Wang¹, Bixin Liu², Hongjia Wu¹, Shan Zhao¹, Zhiping Cai^{1,*}, Donghui Li^{3,*}
and Cheang Chak Fong⁴

Abstract: With the continuous development of e-commerce, consumers show increasing interest in posting comments on consumption experience and quality of commodities. Meanwhile, people make purchasing decisions relying on other comments much more than ever before. So the reliability of commodity comments has a significant impact on ensuring consumers' equity and building a fair internet-trade-environment. However, some unscrupulous online-sellers write fake praiseful reviews for themselves and malicious comments for their business counterparts to maximize their profits. Those improper ways of self-profiting have severely ruined the entire online shopping industry. Aiming to detect and prevent these deceptive comments effectively, we construct a model of Multi-Filters Convolutional Neural Network (MFCNN) for opinion spam detection. MFCNN is designed with a fixed-length sequence input and an improved activation function to avoid the gradient vanishing problem in spam opinion detection. Moreover, convolution filters with different widths are used in MFCNN to represent the sentences and documents. Our experimental results show that MFCNN outperforms current state-of-the-art methods on standard spam detection benchmarks.

Keywords: Opinion spam detection, deceptive reviews, deep learning, convolutional neural network, activation function.

1 Introduction

In recent years, online shopping has been favored by consumers because of its convenience and fast speed. When purchasing commodities on the Internet, instead of relying on intuitive feelings, consumers more likely to make their purchasing decisions based on the seller's reputation, the commodities sales, and the comments of products. It can be considered that these are the essential elements of meeting customers' online shopping preferences. In this case, sellers and customers are paying increasing attention

¹ College of Computer, National University of Defense Technology, Changsha, 410073, China.

² Academy of Military Science, Beijing, 100091, China.

³ College of Information and Intelligence, Hunan Agricultural University, Changsha, 410128, China.

⁴ Faculty of Information Technology, Macau University of Science and Technology, 999078, Macau.

* Corresponding Author: Donghui Li. Email: doner@hunau.edu.cn.

Received: 21 January 2020; Accepted: 11 May 2020.

to the content of reviews, which triggers the motivation of generating spam comments [Jansen (2010)]. Numerous biased comments on the comment page deliberately promote or maliciously lower the rating of a certain product. These comments are usually powerfully deceptive and misleading [Jindal and Liu (2007)]. Sellers may counterfeit unreal praises for their products by hiring astroturfers to get higher ratings. Besides, they may maliciously lower relating ratings to their competitors in the same way. The abuse of cheating and spam reviews will lead to adverse effects on the entire online-shopping industry. In order to break this black market and protect consumers' legitimate rights, we designed an opinion spam detection method in this paper.

Researches on traditional spam such as spam email and advertisement have received considerable attention, while recognition of spam opinion is neglected. Existing spam recognition works mainly focus on identifying destructive spam opinions, such as advertisements, non-commentary texts, or other irrelevant texts, or Jindal et al. [Jindal and Liu (2008)]. Although destructive spam opinions do cause some disruption to customers, the actual impact is not significant because customers can choose to ignore them. There is also a stealthier type of spam opinion in the commentary, namely deceptive spam opinion (deliberately written to sound authentic). Deceptive spam opinion detection can then be considered as an exercise of taking a review and determining whether it is a spam or not. Therefore, in this paper, we focus on how to identify deceptive spam opinion. To summarize, our main contributions are as follows:

- 1) We introduce a CNN-based algorithm with a simple structure and powerful representation ability to get good spam opinion representation.
- 2) We propose the LDA structure, which can obtain fixed-length inputs and a new activation function, combines the advantages of traditional methods and makes up for the shortcomings.

2 Related work

Over the last decade, the researches on spam opinion detection are in two aspects: spam opinion detection based on comment content and commentator's behavior.

Jindal et al. [Jindal and Liu (2008)] firstly proposed the concept of spam opinion detection using a Logistic regression algorithm to identify repetitive reviews, non-commentary reviews, and deceptive reviews. A few years later, Jindal et al. [Jindal, Liu and Lim (2010)] used frequent itemset mining to detect spam comments. Ott et al. [Ott, Choi, Cardie et al. (2011)] combined linguistic features and psychological features as a classifier to effectively identify spam opinion. Chen et al. [Chen, Zhao and Yang (2015)] used deep-level language features to identify spam comments by analyzing information in the sentence and inter-sentence information.

There have been lots of works which detects features outside the comment content itself. Mukherjee et al. [Mukherjee, Liu and Glance (2012)] obtained spam comments by grouping reviewers with similar behaviors. Li et al. [Li, Huang, Yang et al. (2011)] comprehensively considered the content of comments and the characteristics of commentators and obtained the conclusion that the emotional characteristics of comments have less influence on the identification of deceptive comments. Akoglu et al. [Akoglu, Chandy and Faloutsos (2013)] grouped commentators and comment content into a network,

used network effects to rate commentators and comments, and marked low-rated content as spam opinion. Asghar et al. [Asghar, Ullah, Ahmad et al. (2019)] improved performance by combining spam-related features with a rule-based weighting scheme.

Apart from classical machine learning methods, many more NLP tasks have used neural network in classification. Le et al. [Le and Mikolov (2014)] introduced paragraph vector to learn document representations in order to get more information. Yin et al. [Yin and Schütze (2015)] learned multi-granular sentence representations using a convolutional neural network (CNN) and model interaction features at each level. Ren et al. [Ren and Ji (2017)] used CNN and a gated recurrent neural network to identify deceptive opinion spam. Jain et al. [Jain, Sharma and Agarwal (2019)] used CNN and long short term neural network (LSTM) to represent the words and improves the performance. Sedighi et al. [Sedighi, Ebrahimpoor-Komleh, Bagheri et al. (2019)] used an attention-based neural network to detect spam reviews automatically.

3 MFCNN framework

We propose a Multi-Filters Convolutional Neural Network (MFCNN) with LDA structure input and an improved activation function (SRLU) based on the research of Kim [Kim (2014)]. Moreover, we also refer to the ideas from other research fields [Chen, Yu, Liu et al. (2019); Liu, Guo, Cai et al. (2019); Liu, Tang, Li et al. (2019); Luo, Wang, Cai et al. (2019); Yu, Cai, Wang et al. (2019)]. Fig. 1 illustrates our CNN-based document representation model structure.

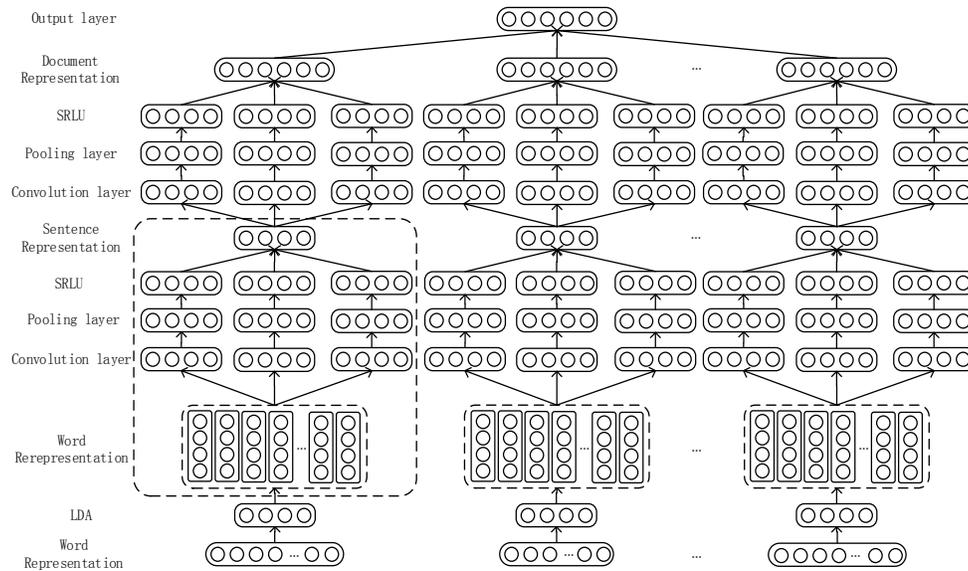


Figure 1: MFCNN-SRLU framework

As shown in Fig. 1, the CNN-based model proposed in this paper can generate a representation of a document (the part drawn by the dotted line is the sentence representation model). Taking the sentence representation model as an example, the model consists of four layers: input layer, convolution layer, pooling layer, and fully

connected layer. The input layer outputs the word vector matrix matched from the search matrix to complete the conversion from the word to the short text feature. The convolution layer is composed of three convolution kernels of different widths, and each convolution kernel can extract different local features. It can extract semantic information of different granularity better. The pooling layer adopts the maximum pooling, and comprehensively considers all the information. Finally, the fully connected layer completes the output of the sentence through the activation function.

3.1 Input layer

The role of the input layer is to convert a short text into input data and transfer the input into the convolutional neural network by connecting to the next layer. Let w_i indicates a word in a sentence, and a sentence in a document is represented as $s_i = w_1 \oplus w_2 \oplus \dots \oplus w_n$, where \oplus represents the connection between w_i and w_{i+1} .

However, the lengths of the sentences are inconsistent, and we need a sequence of consistent length to be entered into the input layer, so converting each sentence into a fixed-length sequence is necessary. Considering the sequence problem of the sentence and in order to get a more reasonable vector representation, we use the LDA model to get the final vector [Duan and Ai (2015)]. Unlike the previous method of directly intercepting the length of a sentence, the LDA model is used to extract the essential words and reconstruct the sentences in the previous order. For each word w_i , we use the lookup matrix E to obtain its word embedding $e(w_i) \in R^D$, where $E \in R^D$ is the model parameter, D is the word vector dimension, and V is the vocabulary. In a traditional neural network, E can be randomly initialized from a uniform distribution, or it can be pre-trained from a sizeable original corpus through an embedded learning algorithm [Socher, Perelygin, Wu et al. (2013); Mikolov, Sutskever, Chen et al. (2013)].

Since the convolutional neural network has initially been designed for image processing tasks, the difference between image and text is that image can be directly converted into a two-dimensional matrix, which is directly utilized by the input layer convolutional neural network. Therefore, we consider using word2vec to transform the text information so that the result becomes the inputs of the convolutional neural network.

The input layer of the sentence representation based on convolutional neural network proposed in this paper uses Google's word vector GoogleNews-vectorsnegative300 [Mikolov, Chen, Corrado et al. (2013)] which has been trained with word2vec. The word vector model trains a 3-billion-word corpus and obtains a 300-dimensional word vector model, so the input layer in this paper is a word vector matrix with a dimension of 300. Word2vec can pay more attention to the relationship between words and sentences in the text, which will affect the classification results.

3.2 Convolution layer

In order to capture semantic fragments of various granularities, three convolutional operations kernels are used to generate sentence representations. The convolution operation kernel is a linear list with shared parameters. Parameters sharing ensures that

only one parameter set learning is needed in the learning process, and no separate parameter sets learning is required for each position.

Let $D = \{D_1, D_2, \dots, D_n\}$ denotes the width of the convolution kernel. Taking the convolution kernel of width D_1 as an example. Let W_1 and b_1 be shared parameters of the convolution kernel, where W_1 is weight and b_1 is offset value. The input to the convolutional layer is a fixed-length vector matrix, denoted as $I_{1,i} = [e(w_i); e(w_{i+1}); \dots; e(w_{i+D_1-1})] \in R^{D \times D_1}$. Then the output of the convolutional layer is shown in Eq. (1):

$$H_{1,i} = W_1 \cdot I_{1,i} + b_1 \quad (1)$$

where $W_1 \in R^{D_n \times D \times D_1}$, and D_n is the output vector matrix sizes of the convolutional layer.

3.3 Pooling layer

In the task of applying traditional methods to do text classification, traditional approaches mainly classify text according to the most representative words in each sentence. However, this method is hard to complete, and the accuracy is not satisfactory. The convolutional neural network model designed in this paper utilizes the maximum pooling method. The factors which have the most significant influence on classification results are extracted from the feature vectors output from the upper convolution layer to complete the process of extracting local features of text from input data.

The maximum pooling layer has the advantage of reducing the amount of calculation and limiting the network scale. Moreover, the pooling layer can extract the essential local features in the data for the convolutional neural network calculation.

Let $P = [P_1, P_2, \dots, P_n]$ denotes the output of the pooling layer. Taking P_1 as an example, let the size of the pooling function be P_d , then the calculation of $P_{1,1}$ is as shown in Eq. (2):

$$P_{1,1} = \max(H_{1,1}, H_{1,2}, \dots, H_{1,P_d}) \quad (2)$$

Then the maximum values are combined as a fixed length vector matrix to be the output matrix of the pooling layer.

3.4 Activation function

The activation function is to map the features through nonlinear functions and remove the redundancy in the data. It makes the neural network applicable with nonlinear problems. In this paper, the SRLU function is used as the activation function, and the SRLU function is shown in Eq. (3):

$$y = \lambda \begin{cases} x & x \geq 0 \\ \alpha(\log(1+e^x) - \beta) & x < 0 \end{cases} \quad (3)$$

where λ is the slope factor of positive interval. The larger the value of the slope factor λ is, the steeper the function of positive interval becomes. The setting of α is related to the saturation point of the negative value interval. The larger the value of α is, the lower the saturation point position will get. β represents the distance between the intersection of the function and the negative part of the y-axis and the origin. The larger the value of β is, the further the distance will become. Let $\beta > 0$, obviously, the negative part of SRLU is always less than zero. Compared with ReLU, the average value of SRLU tends to be 0, which lowers offset value.

The derivative of the SRLU function is shown in Eq. (4):

$$f'(x) = \lambda \begin{cases} 1 & x \geq 0 \\ \alpha \frac{e^x}{1+e^x} & x < 0 \end{cases} \quad (4)$$

Eq. (4) shows that the Sigmoid function multiplies constant $\lambda\alpha$ by a negative interval of the derivative of SRLU function. To ensure the function is derivable at zero, the parameter constraints are shown in Eq. (5):

$$\left\{ \begin{array}{l} f(x)|_{x=0^+} = 0 \\ f(x)|_{x=0^-} = \lambda\alpha(\log 2 - \beta) \\ f(x)|_{x=0^+} = f(x)|_{x=0^-} \\ f'(x)|_{x=0^+} = \lambda \\ f'(x)|_{x=0^-} = \frac{\lambda\alpha}{2} \\ f'(x)|_{x=0^+} = f'(x)|_{x=0^-} \end{array} \right. \quad (5)$$

In the backpropagation of a convolutional neural network, the change in the gradient is represented by a power function whose base is the derivative of the activation function, and the exponent is the number of nonlinear transformation layers [Trottier, Gigu, Chaib-draa et al. (2017)]. In order to avoid gradient vanishing or explosion problems during backpropagation, the slope factor should be set as shown in Eq. (6):

$$\lambda = 1 \quad (6)$$

The result is shown in Eq. (7) can be obtained by solving Eq. (5):

$$\begin{cases} \alpha = 2 \\ \beta = \log 2 \end{cases} \quad (7)$$

So, the exact mathematical definition of the SRLU function is shown in Eq. (8):

$$f(x) = \begin{cases} x & x \geq 0 \\ 2\log \frac{1+e^x}{2} & x < 0 \end{cases} \quad (8)$$

The improved activation function is as shown in Fig. 2. The improved activation function has sparsity and smoothness. It can avoid gradient vanishing problem in the convolutional neural network and make the convolutional neural network effectively converge. Meanwhile, the calculation amount is smaller than the saturated nonlinear activation function, and the convergence speed is faster than the others.

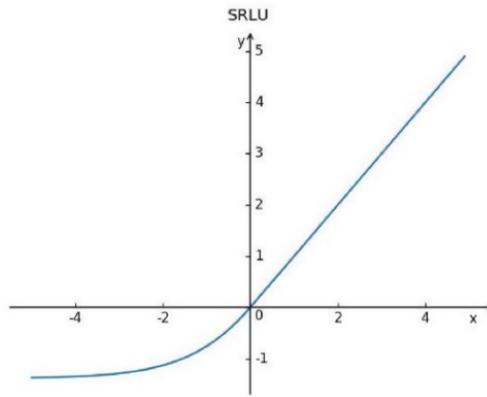


Figure 2: SRLU activation function

3.5 Dropout layout

The function of the dropout layer is preventing over-fitting and improve the training effect of convolutional neural networks. Let r_i obeys the Bernoulli distribution, then the final output of the model is as shown in Eqs. (9) and (10):

$$r_i \sim \text{Bernoulli}(p) \quad (9)$$

$$y_i = r_i \times s_i \quad (10)$$

where s_i is the i th dimension vector of the upper layer output and y_i is the i th dimension vector of the dropout layer output.

3.6 Output layout

Putting together the sentence vectors belonging to the same comment, we can reinvent the output of the sentence representation to get the input of the document representation.

The document indicates that the top of the model with a softmax layer added, which can

comprehensively consider all the extracted local features to complete the task of identifying fraudulent spam opinion. Softmax regression is an algorithm that divides the target component into multiple classes.

The document representation model first converts the document vector into a real-valued vector \mathbf{v}_i whose length is the classification number N . The softmax layer then converts the real-value vector to the conditional probability P_i , which uses the average of several output values from the previous layer, namely considers the eigenvalues represented by the document to identify deceptive spam opinion. The calculation of the softmax function is as shown in Eq. (11):

$$P_i = \frac{\exp(v_i)}{\sum_{j=1}^N \exp(v_j)} \quad (11)$$

3.7 Model training

We use a small batch gradient descent method to train convolutional neural networks. The advantage of the small batch gradient descent method is both considering the advantages of the batch gradient descent method and the stochastic gradient descent. In each iteration, a part of the training sample should be considered to accelerate the convergence rate and avoid the optimal local solutions as much as possible [Ruder (2016)].

Assuming that the number of training samples is 1000, we apply the small batch gradient descent method and set 10 training samples for each iteration, and then the whole data set can be divided into 100 mini-batch. Let the loss function of the batch descent method be as shown in Eq. (12):

$$\frac{\partial J(\theta)}{\partial \theta_j} = \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad (12)$$

The iterative formula for the batch gradient drop is as shown in Eq. (13):

$$\begin{aligned} \theta'_j &= \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j} \\ &= \theta_j - \alpha \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} \end{aligned} \quad (13)$$

where α is the learning rate. Then the iteration of the small batch gradient drop at each iteration is as shown in Eq. (14):

$$\theta'_j = \theta_j - \alpha \frac{1}{10} \sum_{k=i}^{i+9} (h_{\theta}(x^{(k)}) - y^{(k)}) x_j^{(k)} \quad (14)$$

Besides, how to solve the over-fitting phenomenon in model training is significant for obtaining accurate results. We use L2 regularization to constrain the parameters of the convolutional neural network and add dropout layer after generating sentence

representation and document representation [Srivastava, Hinton, Krizhevsky et al. (2014)]. In the training of many deep learning models, the dropout layer is a commonly used method to avoid overfitting.

4 Experiment

4.1 Datasets

We use the spam opinion standard dataset from Ott et al. [Ott, Choi, Cardie et al. (2011)] first published a false positive standard dataset in 2011, which includes 400 true praises and 400 false positives. Later, in 2013, a false negative evaluation standard dataset was published, which included 400 true bad reviews and 400 false bad reviews [Ott, Cardie and Hancock (2013)]. In 2014, Li et al. [Li, Ott, Cardie et al. (2014)] used the same method to collect 1,636 false comments and 1,200 real comments in three different fields. Also, we use a dataset named YelpZip [Rayana and Akoglu (2015)], which contains 608,598 reviews to testify our model.

4.2 Results and analysis

In order to verify the accuracy of different methods, we use the following four indicators, namely Accuracy, Precision, Recall and F1 (F1-score). In this paper, the 10-fold cross-validation method is adopted [Zhang, Jin, Sun et al. (2018); Zhang, Wang, Lu et al. (2019); Zhang, Wu, Feng et al. (2019); Zhang, Lu, Li et al. (2019); Yu, Long and Cai (2017)], that is, the original data set is equally divided into ten parts, and 1/10 of them are taken as the test set, and 1/10 of remaining data is used as verification set. The last remaining data is used as the train set. For each experiment, ten results obtained were averaged as the results.

We set sentence representation and document representation convolution kernel width as (1, 2, 3) and (1, 2, 3), the number of convolution kernel channels is 100, the dropout is 0.5, and the learning rate is 0.001. The experimental results compared with other algorithms are as shown in Tab. 1.

Table 1: Contrast experimental results

Methods	Accuracy	Precision	Recall	F1-score
Logistic Regression	55.63	55.50	56.00	55.75
CNN	78.49	80.97	76.42	78.63
Kim Y. [Kim (2014)]	82.45	85.49	80.32	82.82
Ren Y. [Ren and Ji (2017)]	88.39	90.64	85.47	87.98
MFCNN	91.06	92.25	88.74	90.46

As shown in Tab. 1, the automatically optimized logistic regression method only gets F1-score of 55.75%. CNN gives better results by capturing the relationship between words. Furthermore, we also compare our method with others, which gives results lower than ours.

The experimental results obtained by applying the MFCNN-SRLU model is the best, and our best development result is 91.78%, followed by the experimental result 89.80% using only the MFCNN model, which can indicate the effectiveness of our proposed method.

The result of the experiments on other datasets is shown in Tab. 2. Dataset published by Li consists of three different categories, with the Hotel data set containing the most data, followed by the Doctor data set, and the smallest data is the Restaurant data set.

Table 2: Experiment results of other datasets

Dataset	Accuracy	Precision	Recall	F1-score
Hotel [Li, Ott, Cardie et al. (2014)]	83.58	84.69	88.46	86.53
Restaurant [Li, Ott, Cardie et al. (2014)]	68.82	54.71	65.75	59.72
Doctor [Li, Ott, Cardie et al. (2014)]	79.18	74.51	97.18	84.35
All [Li, Ott, Cardie et al. (2014)]	82.44	78.52	98.64	87.44
YelpZip [Rayana, Akoglu (2015)]	84.31	85.74	87.27	86.50

The experiment based on the dataset of Li mainly verifies the influence of data volume on experimental results, ignoring the influence of parameter setting on experimental results. As can be seen from Tab. 2, the best experimental results were obtained on the Hotel dataset, followed by the Doctor dataset, and the worst experimental result was the Restaurant dataset. The result of such an experiment is mainly due to the different data amounts of the data set. Obviously, the larger the amount of data, the more accurate the experimental results. However, the experimental results obtained in experiments with all data sets are not the best. The Dataset category also has an impact on the experimental results. Multi-category spam recognition is less accurate than single-category spam identification. The experiment based on YelpZip shows the practical value of our method.

5 Conclusion

In this paper, we construct a word vector model by looking up the trained word vector model provided by Google News. Moreover, a new document representation model MFCNN based on the convolutional neural network with a fixed-length obtained input and a new activation function is proposed. Experimental results based on standard evaluation show that compared with other existing methods, this model has better performance in resolving the problem of opinion spam detection.

Funding Statement: This work is supported by The National Key Research and Development Program of China (2018YFB1800202, 2016YFB1000302, SQ2019ZD090149, 2018YFB0204301).

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

References

- Akoglu, L.; Chandy, R.; Faloutsos, C.** (2013): Opinion fraud detection in online reviews by network effects. *Proceedings of the 7th AAAI International Conference on Weblogs and Social media*.
- Asghar, M. Z.; Ullah, A.; Ahmad, S.; Khan, A.** (2019): Opinion spam detection framework using hybrid classification scheme. *Soft Computing*, pp. 1-24.
- Chen, C.; Zhao, H.; Yang, Y.** (2015): Deceptive opinion spam detection using deep level linguistic features. *Proceedings of the 4th CCF Conference on Natural Language Processing and Chinese Computing*, pp. 465-474.
- Chen, H.; Yu, J.; Liu, F.; Cai, Z.; Xia, J.** (2019): Archipelago: a medical distributed storage system for interconnected health. *IEEE Internet Computing*.
- Duan, J.; Ai, Y.** (2015): LDA topic model for microblog recommendation. *International Conference on Asian Language Processing*, pp. 185-188.
- Jain, G.; Sharma, M.; Agarwal, B.** (2019): Spam detection in social media using convolutional and long short term memory neural network. *Annals of Mathematics and Artificial Intelligence*, vol. 85, no.1, pp. 21-44.
- Jansen, J.** (2010): Online product research, Pew internet & American life project. <http://pewinternet.org/Reports/2010/Online-ProductResearch.aspx>.
- Jindal, N.; Liu, B.** (2007): Review spam detection. *Proceedings of the 16th International Conference on World Wide Web*, pp. 1189-1190.
- Jindal, N.; Liu, B.** (2008): Opinion spam and analysis. *Proceedings of the International Conference on Web Search and Data Mining*, pp. 219-230.
- Jindal, N.; Liu, B.; Lim, E. P.** (2010): Finding unusual review patterns using unexpected rules. *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pp. 1549-1552.
- Kim, Y.** (2014): Convolutional neural networks for sentence classification. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1746-1751.
- Le, Q.; Mikolov, T.** (2014): Distributed representations of sentences and documents. *Proceedings of the 31st International Conference on International Conference on Machine Learning*, pp. 1188-1196.
- Li, F. H.; Huang, M.; Yang, Y.; Zhu, X.** (2011): Learning to identify review spam. *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 2488-2493.
- Li, J.; Ott, M.; Cardie, C.; Hovy, E.** (2014): Towards a general rule for identifying deceptive opinion spam. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 1566-1576.
- Liu, F.; Guo, Y.; Cai, Z.; Xiao, N.; Zhao, Z.** (2019): Edge-enabled disaster rescue: a case study of searching for missing people. *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 6, pp. 1-21.
- Liu, F.; Tang, G.; Li, Y.; Cai, Z.; Zhang, X. et al.** (2019): A survey on edge computing systems and tools. *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1537-1562.

- Luo, M.; Wang, K.; Cai, Z.; Liu, A.; Li, Y. et al.** (2019): Using imbalanced triangle synthetic data for machine learning anomaly detection. *Computers, Materials & Continua*, vol. 58, no. 1, pp. 15-26.
- Mikolov, T.; Chen, K.; Corrado, G.; Dean, J.** (2013): Efficient estimation of word representations in vector space. *International Conference on Learning Representations Workshop*.
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; Dean, J.** (2013): Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, pp. 3111-3119.
- Mukherjee, A.; Liu, B.; Glance, N.** (2012): Spotting fake reviewer groups in consumer reviews. *Proceedings of the 21st International Conference on World Wide Web*, pp. 191-200.
- Ott, M.; Cardie, C.; Hancock, J. T.** (2013): Negative deceptive opinion spam. *Proceedings of the Conference of the North American chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 497-501.
- Ott, M.; Choi, Y.; Cardie, C.; Hancock, J. T.** (2011): Finding deceptive opinion spam by any stretch of the imagination. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human language technologies*, vol. 1, pp. 309-319.
- Rayana, S.; Akoglu, L.** (2015): Collective opinion spam detection: bridging review networks and metadata. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 985-994.
- Ren, Y.; Ji, D.** (2017): Neural networks for deceptive opinion spam detection: an empirical study. *Information Sciences*, vol. 385, pp. 213-224.
- Ruder, S.** (2016): An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.
- Sedighi, Z.; Ebrahimpoor-Komleh, H.; Bagheri, A.; Kosseim, L.** (2019): Opinion spam detection with attention-based neural networks. *Proceedings of the 32nd International Flairs Conference*.
- Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D. et al.** (2013): Recursive deep models for semantic compositionality over a sentiment treebank. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1631-1642.
- Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.** (2014): Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958.
- Trottier, L.; Gigu, P.; Chaib-draa, B.** (2017): Parametric exponential linear unit for deep convolutional neural networks. *Proceedings of the 16th IEEE International Conference on Machine Learning and Applications*, pp. 207-214.
- Yin, W.; Schütze, H.** (2015): Convolutional neural network for paraphrase identification. *Proceedings of the Conference of the North American chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 901-911.

Yu, G.; Cai, Z.; Wang, S.; Chen, H.; Liu, F. et al. (2019): Unsupervised online anomaly detection with parameter adaptation for KPI abrupt changes. *IEEE Transactions on Network and Service Management*.

Yu, Y.; Long, J.; Cai, Z. (2017): Network intrusion detection through stacking dilated convolutional autoencoders. *Security and Communication Networks*.

Zhang, J.; Jin, X.; Sun, J.; Wang, J.; Sangaiah, A. K. (2018): Spatial and semantic convolutional features for robust visual object tracking. *Multimedia Tools and Applications*, pp. 1-21.

Zhang, J.; Lu, C.; Li, X.; Kim, H. J.; Wang, J. (2019): A full convolutional network based on DenseNet for remote sensing scene classification. *Mathematical Biosciences and Engineering*, vol. 16, pp. 3345-3367.

Zhang, J.; Wang, W.; Lu, C.; Wang, J.; Sangaiah, A. K. (2019): Lightweight deep network for traffic sign classification. *Annals of Telecommunications*, pp. 1-11.

Zhang, J.; Wu, Y.; Feng, W.; Wang, J. (2019): Spatially attentive visual tracking using multi-model adaptive response fusion. *IEEE Access*, vol. 7, pp. 83873-83887.