

Research Article

Research on Sentiment Classification Algorithms on Online Review

Ruixia Yan , Zhijie Xia, Yanxi Xie , Xiaoli Wang, and Zukang Song

School of Management, Shanghai University of Engineering Science, Shanghai 201620, China

Correspondence should be addressed to Ruixia Yan; yanruixia8@163.com

Received 5 August 2020; Revised 30 August 2020; Accepted 31 August 2020; Published 8 September 2020

Academic Editor: Shuping He

Copyright © 2020 Ruixia Yan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The product online review text contains a large number of opinions and emotions. In order to identify the public's emotional and tendentious information, we present reinforcement learning models in which sentiment classification algorithms of product online review corpus are discussed in this paper. In order to explore the classification effect of different sentiment classification algorithms, we conducted a research on Naive Bayesian algorithm, support vector machine algorithm, and neural network algorithm and carried out some comparison using a concrete example. The evaluation indexes and the three algorithms are compared in different lengths of sentence and word vector dimensions. The results present that neural network algorithm is effective in the sentiment classification of product online review corpus.

1. Introduction

In the field of natural language processing, emotion analysis has always been a hot research field. With the development of the Internet, a large number of business reviews have emerged on various platforms, most of which are mixed with users' personal opinions on commodities. Therefore, the discriminative research on the emotional polarity of these review texts can help enterprises better understand the customer satisfaction of their products or services [1]. Based on the emotional polarity of comments, we could mine the advantages and disadvantages of products. Then, we could obtain suggestions for product promotion and improvement. The traditional method of discriminating emotional polarity of text is based on machine learning since the 1990s. Traditional machine learning methods are mainly divided into two steps. The first step is to construct the word vector feature manually to obtain the required text information. The second step requires the construction of a classifier to classify the emotional polarity of the text. Classical reinforcement learning methods can be basically used in text classification, such as support vector machine, random forest, Naive Bayesian, neural network, and other algorithms.

1.1. Construct Word Vector Features Manually. In this step, the traditional method of generating word vectors is Bag-of-word (BOW) [2], which converts each word into a one-hot vector based on the pre-established dictionary. One disadvantage of this method is that the text vector obtained has the characteristics of high dimension and lack of semantics. Therefore, methods such as TF-IDF and SVD model are used to reduce the dimension of word vector appearance. Karie and Venter calculate semantic similarity of the returned results by means of external engines and input words into searching engines, expanding the semantic information of word vectors [3]. In order to enable vectors to represent context information, models such as LDA and word embedding have also been presented [4]. The word embedding model is an important research result that introduces deep learning algorithm into the field of natural language.

1.2. Research on Text Classification Model Based on Naive Bayesian. Naive Bayesian model is one of the earliest classification algorithms used for text classification task. Its principle is very simple. Based on Bayes' theorem, it assumes that the word vectors of each word are independent from each other. Then, the prior probability of each word in the

corpus is calculated in the training set of the sample. The probability that the test set of the sample is summarized in each category is predicted [5]. Although Bayesian algorithm is simple, it depends on the prior probability of samples much more. Therefore, when the distribution of each sample category in the training set is different, the features of a small number of samples will be replaced by those of a large number of samples.

1.3. Research on Text Classification Based on Support Vector Machine Model. Support vector machine (SVM, for short) model, first proposed by Vapnik in 1995, is based on a combination of the VC dimension theory and risk structure minimization theory. The sample information needed by support vector machine is very limited, so it performs well in solving the problem of small sample and nonlinear text classification. Support vector machine can solve dual problems and use linear method to solve nonlinear problems. Support vector machine can solve the problem of linear inseparability of samples in low-dimensional space by introducing a kernel function. Support vector machine learning algorithm is proposed, and it combines the features of words, parts of speech, and named entities for the text classification task with named entity elements which could achieve good results in the text classification task [6].

1.4. Research on Text Classification Model Based on Deep Learning. In recent years, deep learning has been widely applied in the field of constructing classifiers to classify the emotional polarity of texts. Deep learning model can automatically extract features from the data [7, 8]. For example, Bengio et al. build a neural probabilistic model based on the idea of deep learning and use various deep neural networks to learn on a large-scale English corpus [9]. Deep learning can solve multiple tasks of natural language processing such as named entity recognition and syntactic analysis. In the industrial field, the controlled system usually has great nonlinearity [10–15]. Neural network models have been applied to the identification of nonlinear systems. Convolutional Neural Network (CNN, for short) and Recurrent Neural Network (RNN, for short) have been proved to be effective models for effective classification tasks in the nonlinear system. In terms of the emotional classification of the text, some models of cyclic neural network and convolutional neural network are used to classify the emotions of the short text, and excellent results were obtained [16–20]. However, due to the gradient explosion problem of RNN model, LSTM and GRU models based on the RNN model are more commonly used models [21–25]. Miyamoto et al.

applied the LSTM model to text prediction [26]. Duyu et al. applied the LSTM model to emotion classification and the LSTM model achieved good results [27]. The LSTM model has good long-distance feature extraction ability and can extract the relationship between two sequences that are far apart. For classification, important information is not uniformly distributed in the text in the LSTM model. In order to solve this problem, researchers have put forward the attention mechanism [28]. Different weights of each element in the text were presented in the mechanism in the LSTM model, and weights of each element in the text were iteratively updated through training in the LSTM model.

2. Reinforcement Learning of Text Sentiment Classification Algorithm

Reinforcement learning, one of the most active research areas in artificial intelligence, is a computational approach to learning whereby an agent tries to maximize the total amount of reward it receives when interacting with a complex, uncertain environment. Naive Bayesian algorithm, support vector machine, and neural network model are models of reinforcement learning for text sentiment classification.

2.1. Naive Bayesian Algorithm. Naive Bayesian is a machine learning algorithm based on probability theory. Its core theory is Bayes' Theorem. Suppose that, after word segmentation, a certain sentence corpus X is composed of words: x_1, x_2, \dots, x_n . X is presented as an n -dimensional vector by cut-off or adding 0. There are m categories in total, noted as y_1, y_2, \dots, y_m . The category with the highest probability is obtained through the Bayesian classifier. The highest probability is computed using

$$P(Y | x_1 x_2 x_3 \dots x_n) = \frac{P(x_1 x_2 x_3 \dots x_n | Y) P(Y)}{P(x_1 x_2 x_3 \dots x_n)}. \quad (1)$$

For each sentence of the fixed corpus X , $P(x_1 x_2 x_3 \dots x_n)$ is a certain value. So, formula (1) can be transformed into solving the maximum value of

$$P(Y | x_1 x_2 x_3 \dots x_n) = \max P(x_1 x_2 x_3 \dots x_n | Y) P(Y). \quad (2)$$

Naive Bayesian classifier is based on an assumption that each dimension of the word vector is independent of each other. It means that each feature of the data is independent when it applied to statistics. So, formula (2) can be converted into a solution maximum value of

$$P(Y | x_1 x_2 x_3 \dots x_n) = \max P(x_1 | Y) P(x_2 | Y) P(x_3 | Y) \dots P(x_n | Y) P(Y), \quad (3)$$

where $P(x_i | Y)$ is the prior probability, which represents the frequency of occurrence of word x_i in a certain category as

$$P(x_i | Y) = \frac{\text{count}(x_i | Y)}{\text{count}(Y)}, \quad (4)$$

where $\text{count}(x_i | Y)$ indicates the number of times the current word appears in the current category and $\text{count}(Y)$ indicates the total number of words in the current category.

2.2. Support Vector Machine. Support vector machine model is used to classify the data. T is the set of sample data points. s_i is the i sentence corpus. l_i is the corresponding label. Text sentiment classification defined the problem which is the optimization problem solved from

$$\min_{\alpha^{(*)} \in \mathbb{R}^{2l}} \left(\frac{1}{2} \sum_{i,j=1}^l (\alpha_i^* - \alpha_j)(\alpha_j^* - \alpha_i) K(x_i, x_j) + \varepsilon \sum_{i=1}^l (\alpha_i^* - \alpha_i) - \sum_{i=1}^l l_i (\alpha_i^* - \alpha_i) \right), \quad (6)$$

$\bar{\alpha}^* = (\bar{\alpha}_1, \bar{\alpha}_1^*, \dots, \bar{\alpha}_l, \bar{\alpha}_l^*)^T$ is the optimal solution. Formula (7) is the decision function:

$$l = \sum_{i=1}^l (\alpha_i^* - \alpha_i) K(s_i, s) + \bar{b}. \quad (7)$$

2.3. Neural Network Model. We use the Gated Recurrent Unit (GRU, for short) model as the basic model of the text sentiment classification model in this paper. The GRU model can learn long-term dependence information. GRU usually acts as a recurrent unit that leverages a reset gate and an update gate to control how much information flow from the history state and the current input, respectively. In the GRU model, the current unit state is obtained by calculating and summing the previous unit state. The GRU model can obtain historical information and current information, which is very helpful for extracting the above information in language processing.

The hidden layer of the GRU model could do most of the work. The GRU has two gates, a reset gate and an update gate. The GRU model can learn the long-term dependence of the text. The reset gate is calculated from

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]), \quad (8)$$

h_{t-1} is the state of the previous time step. x_t is the input of the current time step. W_r is the weight. The update gate is calculated from

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]). \quad (9)$$

GRU unit status is update from

$$\tilde{h}_t = \tan f\left(W_{\tilde{h}_t} \cdot [r_t * h_{t-1}, x_t]\right) \text{ and } h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t. \quad (10)$$

The unit output layer is calculated from

$$y_t = g(W_y \cdot h_t). \quad (11)$$

The input layer is noted as $\{x_1, x_2, x_3, \dots, x_{n-1}, x_n\}$. The output of the hidden layer is noted as $\{s_1, s_2, s_3, \dots, s_{m-1}, s_m\}$.

$$\begin{aligned} \min_{\omega, b} \quad & p \\ \text{s.t.} \quad & -p \leq (\omega \bullet s_i) + b - l_i \leq p, \quad i = 1, \dots, n. \end{aligned} \quad (5)$$

Because the data are linearly inseparable in the process of training, the kernel function and penalty factors of support vector machine are introduced to solve from

The calculation method for each layer of the GRU model is as follows:

$$t_i = W_{hx} x_i + W_{hh} h_{i-1} + b_h, \quad (12)$$

$$h_i = e(t_i), \quad (13)$$

$$s_i = W_{yh} h_i + b_y, \quad (14)$$

$$y_i = g(s_i). \quad (15)$$

Here, $\tan f(x) = e^x - e^{-x}/e^x + e^{-x}$ and $g(x) = \text{softmax}(x_i) = e^{x_i} / \sum_{k=1}^K e^{x_k}$.

In order to evaluate the generalization ability of the model on the test set accurately, we use the ten-fold training method to test the model performance. The basic steps of the ten-fold training method are as follows:

Step 1: first, we divide the test set data into ten parts.

Step 2: then, we put nine pieces of data into the classification for training. The remaining piece of data is used as the test set. We calculate the accuracy and recall rate of the test set on the classifier after training.

Step 3: we repeat Step 2. Here, we select one of the nine training sets that have been trained as the test set. Then, we convert the original test set to the training set for model training until ten sets of data are selected in turn.

Step 4: we calculate the average value of the evaluation parameters, which is the final result.

3. Training and Evaluation Parameters of Reinforcement Learning

We select accuracy and F value as evaluation parameters in this paper. First, we introduce the confusion matrix in information retrieval, which is shown in Table 1. Among them, TP is a pair of feature views that are correctly classified as positive emotions. FP is misclassified as negative emotions. About feature opinion pairs, FN is a feature opinion pair that

TABLE 1: Confusion matrix.

| | | Positive emotion | |
|-----------------------------|--|---|--|
| | | True (the number of texts belonging to this type of sentiment) TP True positive FN False negative | False (the number of texts that do not fall into this category) FP False positive TN True negative |
| Negative or invalid emotion | Positive (the number of real positive cases in the data) | | |
| | Negative (the number of real negative cases in the data) | | |

is misclassified as positive emotion. TN is a feature opinion pair that is classified as negative emotion correctly.

The calculation formulas for accuracy and F value are as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (16)$$

$$F = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \quad (17)$$

In the training process of the model, we use the grid search technique to find the optimal parameters of Naive Bayesian classifier support and the support vector machine classifier in this paper. The training parameters of each classifier are shown in Tables 2 and 3.

4. Model Test of Reinforcement Learning

In order to evaluate the results of the classification model in different word vector dimensions and different sentence lengths, we train models in different dimensions of word vectors and different sentence lengths. The training set contains 7000 online comments of a certain brand of tablet computers (the data is product online review corpus in 2017-2018 which is crawled by us from an online mall. It can be downloaded from <https://pan.baidu.com/s/16AYTrzjWZDKXJ0iPqLZUvw> (Extraction code: wr01)). The training set and the test set are divided according to the ratio of 7 : 3. From the statistical information of the corpus, it can be seen that the number of words in most of the sentence is below 50 words. If the number of words in the sentence is too low, it would lose the meaning of training. Therefore, we select sentences in which the number of words is in an interval of [15, 50] and the word vector dimension to an interval of [100, 250].

The training result is shown in Tables 4–6. Table 4 is the classification results in different sentence lengths and different dimensions by the Bayesian classifier.

Table 5 is the classification results in different sentence lengths and different dimensions by support vector machine classifier.

Table 6 is the classification results in different sentence lengths and different dimensions by the GRU classifier.

From Tables 4–6, we find that the RNN model algorithm has a better classification effect and the RNN model algorithm effect is significantly better than the other two algorithms. From Table 4, we find the accuracy is stable at about 64% from the Naive Bayesian model. However, the F value continues to rise with the increase of sentence length and the highest value is still

TABLE 2: Support vector machine training parameters.

| Parameters | Parameter values |
|--|------------------|
| C: penalty coefficient | 1 |
| Kernel: the sum function type used in the algorithm | RBF |
| Probability: whether to use probability AVB Estimation | True |
| tol: residual convergence condition | 0.001 |

TABLE 3: Recurrent neural network training parameters.

| Parameters | Parameter values |
|-------------------|------------------|
| Input layer | 128 |
| Decision function | Relu |
| Output layer | 1 |

TABLE 4: Bayesian classifier classification results.

| Sentence length | Word vector dimension | Accuracy (%) | F value |
|-----------------|-----------------------|--------------|-----------|
| 15 | 100 | 61.43 | 0.580 |
| 15 | 150 | 60.90 | 0.570 |
| 15 | 200 | 60.86 | 0.572 |
| 15 | 250 | 61.27 | 0.577 |
| 20 | 100 | 61.87 | 0.613 |
| 20 | 150 | 61.97 | 0.616 |
| 20 | 200 | 62.66 | 0.625 |
| 20 | 250 | 61.99 | 0.613 |
| 25 | 100 | 60.74 | 0.629 |
| 25 | 150 | 61.43 | 0.631 |
| 25 | 200 | 61.83 | 0.633 |
| 25 | 250 | 61.71 | 0.631 |
| 30 | 100 | 61.24 | 0.649 |
| 30 | 150 | 61.81 | 0.649 |
| 30 | 200 | 61.57 | 0.645 |
| 30 | 250 | 62.60 | 0.652 |
| 35 | 100 | 61.21 | 0.652 |
| 35 | 150 | 62.99 | 0.664 |
| 35 | 200 | 61.47 | 0.650 |
| 35 | 250 | 61.84 | 0.655 |
| 40 | 100 | 62.53 | 0.677 |
| 40 | 150 | 61.44 | 0.660 |
| 40 | 200 | 63.03 | 0.675 |
| 40 | 250 | 61.86 | 0.668 |
| 45 | 100 | 60.24 | 0.641 |
| 45 | 150 | 62.14 | 0.674 |
| 45 | 200 | 61.44 | 0.668 |
| 45 | 250 | 61.49 | 0.662 |

TABLE 5: Support vector machine classifier classification results.

| Sentence length | Word vector dimension | Accuracy (%) | <i>F</i> value |
|-----------------|-----------------------|--------------|----------------|
| 15 | 100 | 75.54 | 0.738 |
| 15 | 150 | 75.00 | 0.732 |
| 15 | 200 | 74.47 | 0.729 |
| 15 | 250 | 73.11 | 0.720 |
| 20 | 100 | 76.59 | 0.753 |
| 20 | 150 | 76.63 | 0.754 |
| 20 | 200 | 76.27 | 0.750 |
| 20 | 250 | 76.33 | 0.751 |
| 25 | 100 | 77.74 | 0.769 |
| 25 | 150 | 77.51 | 0.766 |
| 25 | 200 | 77.03 | 0.761 |
| 25 | 250 | 77.03 | 0.761 |
| 30 | 100 | 77.99 | 0.772 |
| 30 | 150 | 77.67 | 0.769 |
| 30 | 200 | 77.67 | 0.768 |
| 30 | 250 | 77.64 | 0.767 |
| 35 | 100 | 78.11 | 0.773 |
| 35 | 150 | 78.79 | 0.779 |
| 35 | 200 | 78.10 | 0.773 |
| 35 | 250 | 78.94 | 0.782 |
| 40 | 100 | 78.13 | 0.773 |
| 40 | 150 | 78.47 | 0.779 |
| 40 | 200 | 79.19 | 0.785 |
| 40 | 250 | 78.73 | 0.781 |
| 45 | 100 | 76.87 | 0.750 |
| 45 | 150 | 78.06 | 0.770 |
| 45 | 200 | 78.41 | 0.777 |
| 45 | 250 | 79.19 | 0.785 |

TABLE 6: GRU classification results.

| Sentence length | Word vector dimension | Accuracy (%) | <i>F</i> value |
|-----------------|-----------------------|--------------|----------------|
| 15 | 100 | 85.02 | 0.863 |
| 15 | 150 | 87.44 | 0.864 |
| 15 | 200 | 83.77 | 0.862 |
| 15 | 250 | 87.78 | 0.866 |
| 20 | 100 | 89.31 | 0.881 |
| 20 | 150 | 87.92 | 0.887 |
| 20 | 200 | 89.03 | 0.889 |
| 20 | 250 | 87.97 | 0.886 |
| 25 | 100 | 85.64 | 0.866 |
| 25 | 150 | 90.71 | 0.888 |
| 25 | 200 | 89.47 | 0.885 |
| 25 | 250 | 92.60 | 0.903 |
| 30 | 100 | 88.26 | 0.894 |
| 30 | 150 | 87.72 | 0.871 |
| 30 | 200 | 88.13 | 0.901 |
| 30 | 250 | 90.60 | 0.894 |
| 35 | 100 | 90.70 | 0.905 |
| 35 | 150 | 87.47 | 0.904 |
| 35 | 200 | 90.16 | 0.901 |
| 35 | 250 | 87.23 | 0.899 |
| 40 | 100 | 87.60 | 0.897 |
| 40 | 150 | 91.33 | 0.919 |
| 40 | 200 | 93.34 | 0.922 |
| 40 | 250 | 73.62 | 0.809 |
| 45 | 100 | 89.07 | 0.912 |
| 45 | 150 | 91.16 | 0.900 |
| 45 | 200 | 92.01 | 0.912 |
| 45 | 250 | 89.25 | 0.908 |

only about 67% from the Naive Bayesian model. We believe that the Bayesian model is a prior probability model and is more dependent on the big data. In this paper, the sample set is small or medium and its prior probability distribution is not accurate. So, the result is not well as expected from the Naive Bayesian model. In contrast, the result from the support vector machine model is much better. With the continuous increase of sentence length and word vector dimensions, the accuracy and *F* value from the support vector machine model are maintained at about 78%. It can be seen that when the sentence length is increased to 40, the accuracy has also decreased slightly, and similar accuracy rates are maintained before and after this length from the support vector machine. It can be considered that the model has converged at this length from the support vector machine. The result from the neural network model is the best. The accuracy is around 90% from the neural network model. Due to the powerful feature extraction capabilities, the neural network model is better than the other two models. It can be seen from Table 6 that the GRU model has the best classification effect on a sentence of which the number of words is 40 words and 200 dimensions of word vector.

5. Conclusions

At present, more general scenarios for reinforcement learning and adaptive optimization present a major challenge in complex dynamic systems. The judgment of text sentiment tendency is a hot direction in the field of natural language. We study the sentiment classification algorithm of online reviews. Due to the remarkable effect of machine learning, we select three kinds of machine learning methods: Naive Bayesian, support vector machine, and neural network for comparative research. In order to evaluate the performance of the algorithm on different sentence lengths and word vector dimensions, we train these three models in different dimensions. Finally, using an experiment on an online, we find that the neural network algorithm is effective in classification.

Data Availability

Data used to support the findings of this study are available from the corresponding author upon request or can be downloaded from <https://pan.baidu.com/s/16AYTrzjWZDKXJ0iPqLZUvw>, Extraction code: wr01.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was funded by the National Philosophy and Social Science General Foundation of China (no. 19BGL234) and Ministry of Education of Humanities and Social Science Foundation of China (no. 17YJCZH199). The authors gratefully acknowledge the National Office for Philosophy and Social Sciences of China and Ministry of Education of China for financial support.

References

- [1] Z. Wang, X. Zhang, X. Yang, S. Wang, and W. Xia, "PortraitAI: a deep learning-based approach for generating user portrait for online dating website," in *Proceedings of the International Conference on Education, Economics and Information Management (ICEEIM 2019)*, p. 2020, December 2019, Wuhan, China.
- [2] M. Andrew, D. Raymond, P. Pham, D. Huang, A. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Florence, Italy, June 2011.
- [3] K. Nickson and H. . s. Venter, "Measuring semantic similarity between digital forensics terminologies using web search engines," in *Proceedings of the Information Security for South Africa IEEE*, pp. 1–9, Johannesburg, South Africa, July 2012.
- [4] J. Fei, L. Yiqun, L. Huanbo, and S. Ma, "Microblog sentiment analysis with emoticon space model," *Journal of Computer Science and Technology*, vol. 30, no. 5, pp. 1120–1129, 2014.
- [5] K. Yoon, "Convolutional neural networks for sentence classification," *Eprint Arxiv*, vol. 61, no. 6, pp. 2–12, 2014.
- [6] S. Tong and K. Daphne, "Support vector machine active learning with applications to text classification," *The Journal of Machine Learning Research*, vol. 2, no. 1, pp. 999–1006, 2002.
- [7] J. Dai and J. Chen, "Feature selection via normative fuzzy information weight with application into tumor classification," *Applied Soft Computing*, vol. 92, pp. 106299–106314, 2020.
- [8] J. Dai, H. Hu, W.-Z. Wu, Y. Qian, and D. Huang, "Maximal-discernibility-pair-based approach to attribute reduction in fuzzy rough sets," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 4, pp. 2174–2187, 2018.
- [9] Y. Bengio, H. Schwenk, S. Jean-Sébastien, F. Morin, and J.-L. Gauvain, "Neural probabilistic language models," *Innovations in Machine Learning*, vol. 194, pp. 137–186, 2006.
- [10] P. Cheng, J. Wang, S. He, X. Luan, and F. Liu, "Observer-based asynchronous fault detection for conic-type nonlinear jumping systems and its application to separately excited DC motor," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, pp. 1–12, 2019.
- [11] P. Cheng and S. He, "Observer-based finite-time asynchronous control for a class of hidden Markov jumping systems with conic-type nonlinearities," *IET Control Theory & Applications*, vol. 14, 2019.
- [12] P. Cheng, S. He, J. Cheng, X. Luan, and F. Liu, "Asynchronous output feedback control for a class of conic-type nonlinear hidden markov jump systems within a finite-time interval," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 99, pp. 1–8, 2020.
- [13] J. Liu, T. Yin, Y. Dong, H. Reza Karimi, and J. Cao, "Event-based secure leader-following consensus control for multi-agent systems with multiple cyber-attacks," *IEEE Transactions on Cybernetics*, vol. 99, 2020.
- [14] J. Liu, W. Suo, L. Zha, E. Tian, and X. Xie, "Security distributed state estimation for nonlinear networked systems against DoS attacks," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 3, pp. 1156–1180, 2020.
- [15] J. Liu, Y. Wang, J. Cao, Y. Dong, and X. Xie, "Secure adaptive-event-triggered filter design with input constraint and hybrid cyber-attack," *IEEE Transactions on Cybernetics*, 2020.
- [16] Y. Shuhan, W. Xintao, and X. Yang, "Incorporating pre-training in long short-term memory networks for tweets classification," in *Proceedings of the IEEE International Conference on Data Mining IEEE*, pp. 12–19, Pisa, Italy, December 2017.
- [17] J. P. A. Vieira and R. S. Moura, "An analysis of convolutional neural networks for sentence classification," in *Proceedings of the Computer Conference IEEE*, pp. 1–5, Cordoba, Argentina, December 2017.
- [18] Y. Zhao, B. Qin, and T. Liu, "Encoding syntactic representations with a neural network for sentiment collocation extraction," *Science China (Information Sciences)*, vol. 60, no. 011, pp. 3–14, 2017.
- [19] Y. Zhang and J. Meng, N. Wang and M. Pratama, "Sentiment classification using comprehensive attention recurrent models," in *Proceedings of the International Joint Conference on Neural Networks IEEE*, pp. 45–54, Vancouver, BC, Canada, July 2016.
- [20] Q.-H. Vo, N. Huy-Tien, B. Le, and M.-Le Nguyen, "Multi-channel LSTM-CNN model for Vietnamese sentiment analysis," in *Proceedings of the 2017 9th International Conference on Knowledge and Systems Engineering (KSE) IEEE*, pp. 12–19, Hue, Vietnam, October 2017.
- [21] L.-x. Luo, "Network text sentiment analysis method combining LDA text representation and GRU-CNN," *Personal and Ubiquitous Computing*, vol. 23, no. 3-4, pp. 405–412, 2019.
- [22] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.
- [23] T. H. Shiou, M. Changsung, J. Paul, and N. F. Samatova, "A hybrid CNN-RNN alignment model for phrase-aware sentence classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain, April 2017.
- [24] H. Liu, B. Lang, M. Liu, and H. Yan, "CNN and RNN based payload classification methods for attack detection," *Knowledge-Based Systems*, vol. 163, pp. 332–341, 2019.
- [25] R. Rajalakshmi, H. Tiwari, J. Patel, and R. Rameshkannan, "Bidirectional GRU-based attention model for kid-specific URL classification," *Deep Learning Techniques and Optimization Strategies in Big Data Analytics*, IGI Global, Hershey, PA, USA, 2020.
- [26] M. Yasumasa and K. Cho, "Gated word-character recurrent language model," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1992–1997, Austin, TX, USA, November 2016.
- [27] T. Duyu, Q. Bing, and L. Ting, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1422–1432, Lisbon, Portugal, September 2015.
- [28] Y. Wang, M. Huang, Xiaoyan Zhu, and Li Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 606–615, Austin, TX, USA, November 2016.