Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/20/20231103

Exploiting ensembled neural network model for social platform rumor detection

Bowen Huang[1, †], Ruoheng Feng[2, 4, †] and Jiahao Yuan[3, †]

[1]School of Mathematics and Statistics, Northeastern University, Shenyang, Liaoning, 110819, China
[2]School of International Economics and Management, Beijing Technology and Business University, Beijing, 100048, China
[3]School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, 200433, China

[4]7uanm@st.btbu.edu.cn
[†]These authors contributed equally

Abstract. With the spread of the internet and social media, it has become difficult to detect rumors from the vast amount of event information. In order to improve the accuracy of rumor detection, deep learning neural network models are often used in rumor detection tasks. First, this paper reproduces the rumor detection experiments of four single neural network models: Long Short-term Memory Networks (LSTM), Text Convolutional Neural Networks (TextCNN), Text Recurrent Neural Network with Attention Mechanism (TextRNN_Att), and Transformer. On this basis, a model based on pre-trained feature extractor and ensemble learning is proposed, and a weighted average ensemble algorithm is adopted. The results show that the rumor-detecting ensemble learning model is better than the single model in all indicators. Then, aiming at the problem that the weighted average ensemble method cannot determine the optimal ensemble parameters, this paper proposes to improve the adaptive ensemble model. Multilayer Perceptron (MLP) is selected as the metamodel, and the weight parameters are automatically trained finetuning on the predicted output of the base model by weighted summation and MLP neural network is used, which improves the traditional integrated weighted average model and realizes the function of automatic weight adjustment. Finally, the Fast Gradient Sign Method (FGSM) algorithm is used to train the model adversarily. The results show that the ensemble model after adversarial training obtains stronger generalization, robustness and attack resistance under the premise of ensuring that the classification performance is not reduced.

Keywords: rumor detection, text classification, ensemble learning, adversarial training.

1. Introduction

Social networking platforms have now become important channels for accessing hot social practices, public issues, and economic development trends. However, these platforms lack effective regulation of the vast amount of information they provide, allowing misleading and false information to spread rapidly through social networks, thus accelerating the dissemination of rumors [1]. Online rumors,

DOI: 10.54254/2755-2721/20/20231103

characterized by their wide audience and ease of dissemination, can easily fuel emotions such as anxiety and panic among the masses, triggering public sentiment and various forms of collective events, thereby posing a serious threat to social stability [2].

Currently, rumors have become a global hot topic that continues to draw people's attention. The search popularity of relevant keywords on Google over the past five years is depicted in Figure 1.

Figure 1. Trend of search popularity for keyworks related to “Rumors”.

Deep learning and artificial intelligence have entered the public's consciousness since 2013. Deep learning is primarily based on neural networks, which are machine learning models designed to simulate the workings of the human brain's neurons [3]. In this context, this paper aims to improve the performance of a single neural network model using an ensemble approach by adjusting parameters for optimization. Additionally, data augmentation techniques are employed to expand the dataset, thereby enhancing robustness and improving the accuracy of rumor detection technology.

Rumor detection can be categorized into two types: text and image. This paper specifically focuses on text-based rumor detection. Within this type, rumor detection is considered a binary text classification problem. Research on rumor detection methods, both domestically and internationally, primarily encompasses techniques based on conventional machine learning and deep learning methodologies [4].

Compared to traditional methods, deep learning approaches have the advantage of learning effective features from data, thereby improving the accuracy and efficiency of feature selection. They also overcome various limitations associated with manual annotation. Convolutional neural networks (CNN), recurrent neural networks (RNN), and long short-term memory networks (LSTM) have been used for text sequence representation and text classification [2]. Deep neural networks have shown significant improvements in Chinese text classification compared to traditional machine learning methods [5].

This paper proposes a rumor detection method based on ensemble learning and textual content. Firstly, the rumor detection model of a single neural network is improved by selecting TextCNN, TextRNN_Att, LSTM, and Transformer as base models. A Multi-Layer Perceptron (MLP) is used as the meta-algorithm for weighted averaging ensemble. The proposed improved ensemble model not only seeks the optimal weighted parameters but also aims to achieve the best testing accuracy, F1 score, and resistance to perturbed texts. Additionally, the authors employ the Fast Gradient Sign Method (FGSM) for adversarial training to further enhance the security, generalization, and robustness of the MLP model in the ensemble learning framework. Finally, this paper compares and analyzes the rumor detection accuracy of various ensemble models and individual models, exploring the characteristics of different networks and their roles in the rumor detection process.

DOI: 10.54254/2755-2721/20/20231103

2. Method

2.1. Dataset

The dataset used in this study is sourced from the publicly available CED Chinese Rumor Dataset [6]. This dataset comprises a total of 3,387 Chinese rumor data collected from Sina Weibo, a popular microblogging platform. Among these instances, 1,538 are labeled as rumors, while 1,849 are labeled as non-rumors. The dataset also includes information such as retweets and comments related to the original Weibo posts.

Before training the ensemble model based on text content, the text content is extracted from the entire target dataset and labeled as rumor or non-rumor. The dataset is partitioned into two sets, with a 9:1 ratio between the training set and the test set, respectively. The single network model and the ensemble learning model are then trained in a hierarchical manner using the training set, while the test set is used to evaluate the accuracy, F1 score, and recall of the models. The emphasis of this study lies in the generalization ability of the models, and the F1 score is used as the criterion for model selection and improvement [7].

2.2. Neural network

Using the architecture of neural network models designed for text classification, the models can be categorized into LSTM, TextCNN, TextRNN, TextRCNN, Transformer, and ensemble learning-based hybrid network models [8]. The properties and focuses of each neural network model are presented in Table 1.

Table 1. Properties of various neural network models.

Neural Network Model	Properties and Focus
LSTM	Capturing long-range dependencies.
TextCNN	Utilizes convolutional operations for efficient extraction of text features, maps text sequences to fixed-length vectors for classification
TextRNN	Utilizes convolutional operations for efficient extraction of text features, maps text sequences to fixed-length vectors for classification
TextRCNN	Utilizes convolutional operations for efficient extraction of text features, maps text sequences to fixed-length vectors for classification
Transformer	A self-attention mechanism-based encoder-decoder architecture, appropriate for text classification tasks downstream.

TextRNN_ATT, an extension of TextRNN, incorporates an attention mechanism. The following provides a brief explanation of the attention mechanism. The introduction of the attention mechanism improves the performance and interpretability of text classification models. Without the attention mechanism, each word in the text sequence contributes equally to the classification task. However, in practical applications, there are often irrelevant words that do not contribute to the classification task. Therefore, the attention mechanism is introduced to effectively measure the contribution of each word to the classification task.

2.3. Ensemble learning

Ensemble learning refers to a machine learning approach that aggregates multiple base classifiers for making collective decisions. It involves using simple classification algorithms to obtain diverse base classifiers, and then combining them in a certain way to form a strong classifier [9].

In this paper, the Bagging (training multiple classifiers and averaging) series algorithms are adopted. Subsampling is performed on the training set to create sub-training sets for each base model. The predictions of all base models are then combined to generate the final prediction result.

The working mechanism of Bagging is as follows:

DOI: 10.54254/2755-2721/20/20231103

(1) To generate training sets from the original sample set, the Bootstrapping method is used to randomly select n samples in each round. Some samples may be chosen multiple times in the training set, while others may not be selected at all. This process is repeated k times, producing k sets of independent training data.

(2) Models are created for each training set, using various classification or regression methods depending on the problem at hand. This yields k models.

(3) In a classification problem, the predictions of the k models are combined through voting to obtain the final result. For a regression problem, the mean of the predictions from the k models is calculated, with all models having equal importance.

2.4. Adversarial training

Adversarial training technique is to add a perturbation to the original input sample, and after obtaining the adversarial sample, use it for training to improve the robustness of the model. An improved version was proposed in the 2020 ICLR conference paper [10]. FGSM method uses the idea of projection gradient descent to generate adversarial samples, and proposes a range-limiting initial method that enables FGSM to be compatible with Mini-batch homogeneous computing. This paper employs the FGSM method, and the workflow description is shown in Table 2.

Table 2. FGSM-Ensemble Model Algorithm.

Input: The original dataset(D1)
Output:FGSM-Ensemble Model *H*
1:step1: Word vectorization embedding
2:step2: Train the base classifier
3:step3: Ensemble learning training:
4: Initialize base classifier weighted parameters wi(i=1,2,3...T)
5: **for** i=1 to T **do**
6: Base classifier prediction result weighted sum
7: Adversarial sample generation is sent to MLP secondary training
8: Train the adaptive ensemble model H with attack_D2
9: **end for**

Given a network determined by the parameter θ in adversarial training techniques, f_θ, a dataset of (x_i, y_i), a loss function of ι, and an attack model Δ, the learning problem of adversarial training techniques can be defined as the following robust optimization problem:

$$\min_{\theta} \sum_i \max_{\delta \in \Delta} \ell(f_\theta(x_i + \delta), y_i) \tag{1}$$

The optimization goal is $\max - \min$, δ is a parameter of the attack model. Its optimization means to find a message with the maximum interference to add to the input sample, and the resulting interference sample has as little impact on the training network f_θ as possible. It can also be understood that the addition of interference information does not change the polarity of the original sample, and at the same time learns part of the noise information, which also improves the anti-interference of the model, that is, robustness. In this way, in adversarial training, there is one more parameter learning than the normal training network, δ and its parameters can be solved according to the following formula:

$$\delta^* = \epsilon \cdot sign\left(\nabla_x \ell(f(x), y)\right) \tag{2}$$

DOI: 10.54254/2755-2721/20/20231103

3. Result

3.1. Experimental setting and preprocessing

The research experiment in this chapter uses the preprocessed and labeled rumor content dataset as the training object, and the proposed ensemble learning adopts pytorch framework version 1.12.1 and is implemented using Python 3.7. This experiment is paired with GPU A5000 under the Windows operating system. In the experiment, the loss function adopts "CrossEntropyLoss", the optimizer adopts "Adamw", the learning rate scheduler adopts "CosineAnnealingWarmRestarts", the ratio of training set and test set is 9:1. Four single neural network models train 150, 200, 30, 350 epochs, respectively, Ensemble learning trains a total of 20 epochs.

Next, feature engineering is carried out. Considering the dimensionality disaster of Onehot and the shortcomings of TF-IDF in ignoring semantics, this paper selects Word2Vec (considering semantic association) to train the feature extraction model, and uses the LSTM model for training and testing in advance. The pre-training extraction feature extractor, Sogou, is compared Chinese pre-trained word vector, which is also brought into LSTM as an embedding layer. The test accuracy obtained by experiments is shown in Table 3 below, and it can be seen that the rumor detection pre-training feature extractor brings into the model better, contains more semantic information, and does not need to start training from 0, and speculates that wor2vec is brought in LSTM is limited by the size of this data set, so subsequent experiments are fine-tuned using Sogou News Chinese pre-trained word vector, which has a dimension of 300 dimensions.

Table 3. Results of a pre-trained feature extractor experiments.

Index	Word2vec-LSTM	Sougou-LSTM
Precision	0.7968	0.8135
Recall	0.7992	0.7873
F1 Score	0.7980	0.8002

3.2. Result comparison of single models

This paper has successively reproduced the rumor detection experiments of LSTM, TextCNN, TextRNN_Att, Transformer, and combined theory and practice to feel their respective advantages. In order to finally select and improve the hybrid model based on ensemble learning, the rumor detection experimental results of the single network model are shown in Figure 2 below.

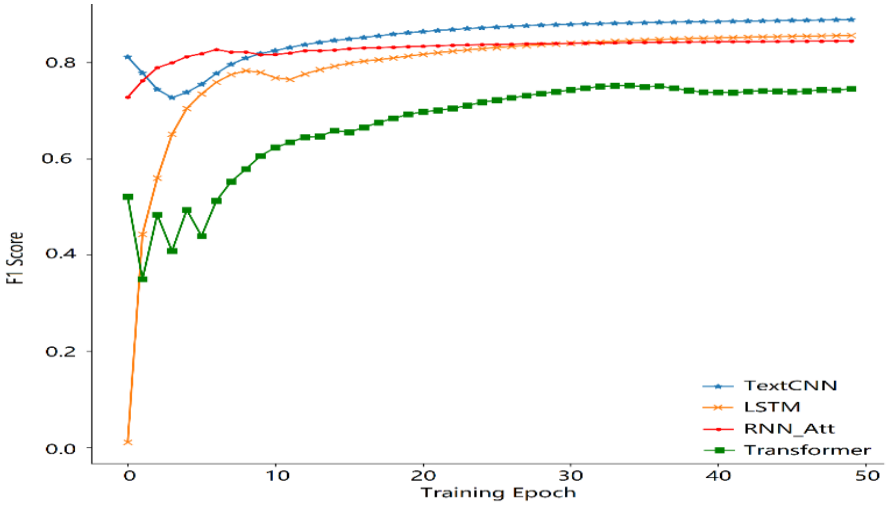

Figure 2. Rumor detection experiment results for single network models.

DOI: 10.54254/2755-2721/20/20231103

From the data in the figure, TextCNN performs better, has the advantage of being fast, can capture local text features, and reduces the risk of overfitting. LSTM is mediocre but solves the problem of vanishing gradients and gradient explosions to some extent, while being able to handle long-term dependencies. TextRNN_Att is average, but through attention mechanisms, a model can adaptively focus on important parts of the text. Transformers perform poorly, capturing global text features, but require long training time and a lot of computing resources.

In this paper, four types of models, TextCNN, Transformer, LSTM and TextRNN_Att, have been chosen to significantly enhance the precision of rumor detection. Their weights are 0.85, 0.02 0.11 and 0.02 respectively. The experimental results are shown in Table 4.

Table 4. Comparison of network model results.

Model Name	Precision	Recall	F1
Transformer	0.741	0.740	0.740
TextRNN_Att	0.814	0.818	0.814
LSTM	0.826	0.823	0.824
TextCNN	0.874	0.876	0.875
Ensemble Model	0.921	0.916	0.918

By weighting the predictions of multiple models, it can improve the overall performance of the model and reduce the risk of overfitting. In this problem, TextCNN has a larger weight, while Transformer and TextRNN_Att have a smaller weight, and LSTM has a moderate weight, indicating that TextCNN performs better on this task.

In the ensemble model, TextCNN can be one of the more important components responsible for capturing local information. This paper experiments that the advantages of TextCNN can be compensated for by integrating the advantages of other models. The Transformer model can better handle long text and capture global information more fully, so in the ensemble model, it can work with the TextCNN model to improve the overall classification performance. LSTM and TextRNN_Att models are better able to handle longer text, so they can work in tandem with TextCNN models to improve overall classification performance.

3.3. Results of ensemble learning

Table 5 and Figure 3 display the performance of the adaptive integration model and numerous heterogeneous deep learning base models on various indices. The data gap between the adaptive integration model and the integrated learning model in Precision, Recall, and F1 Score is very minimal, reaching more than 90%, when compared with the indicators of the integrated learning model in Table 3. The adaptive integration model, on the other hand, has very potent adaptive and self-learning characteristics and does not require manual adjustment of the weight ratio. By not relying on human expertise in the area, time is saved.

By examining the index data of the four single models in this module, it could be observed that TextCNN excels in text feature extraction but has limited ability to handle long texts. TextRNN_Att and LSTM, on the other hand, are better suited for accurate and stable sequential data processing. Transformer works well for many different NLP tasks, although it performs worse when trained on tiny data sets.

Table 5. Results of ensemble model.

	Ensemble Model	TextCNN	TextRNN_Att	LSTM	Transformer
Precision	0.9090	0.8760	0.8375	0.8262	0.8067

DOI: 10.54254/2755-2721/20/20231103

Table 5. (continued).

Recall	0.9034	0.8745	0.8373	0.8258	0.8079
F1 Score	0.9058	0.8752	0.8374	0.8260	0.8072

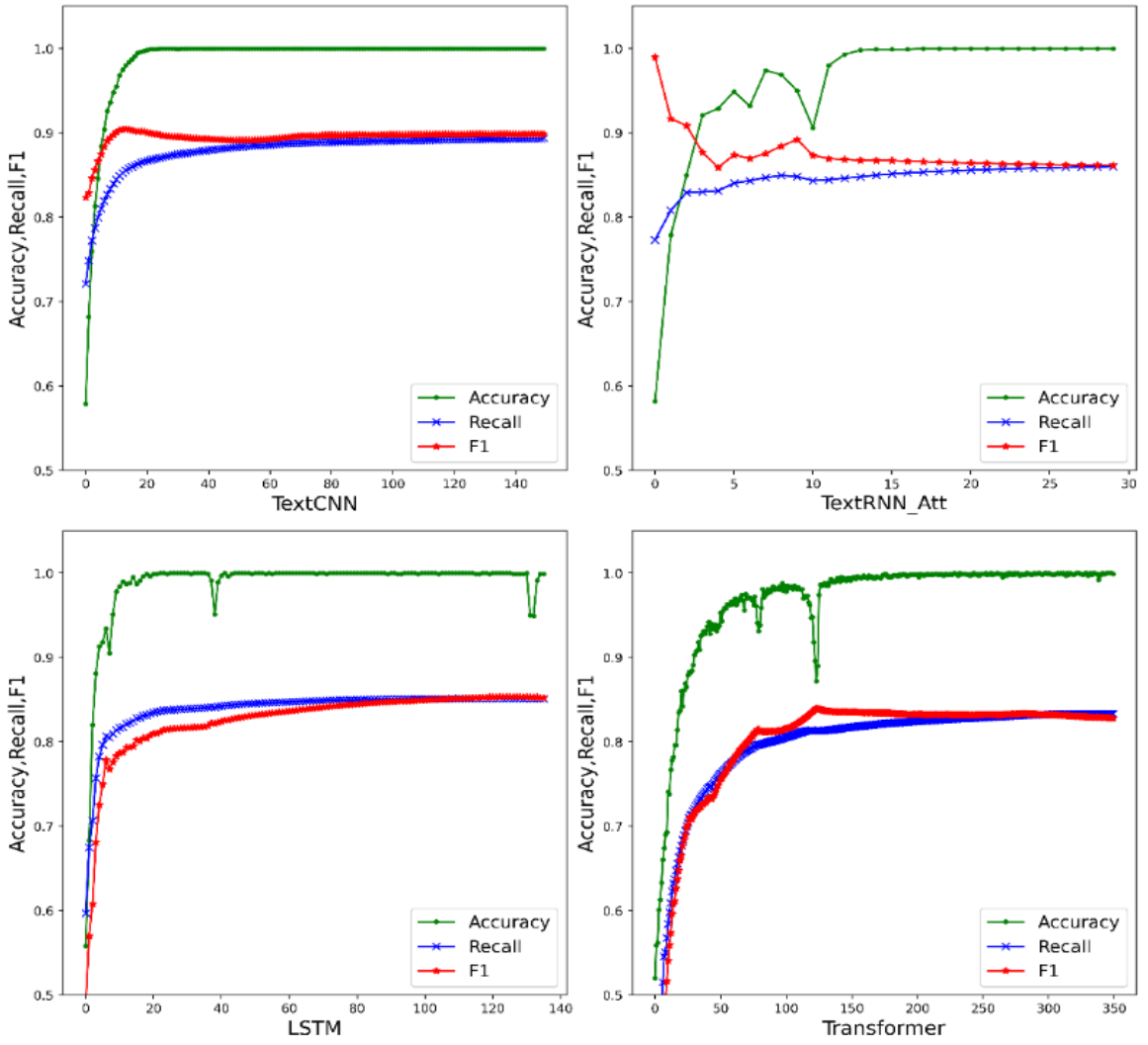

Figure 3. Training process of base models.

Figure 4 depicts the adaptive integration model's training procedure. The adaptive model has a higher training efficiency than the single model. High accuracy was attained and had a convergence trend after three training sessions. Precision, Recall, and F1 Score were each 90.90%, 90.34%, and 90.58% after 15 training rounds. Each index has significantly improved compared to the four standalone models. This is since the integration model's capabilities could be expanded by integrating a single model that specializes in many fields.

DOI: 10.54254/2755-2721/20/20231103

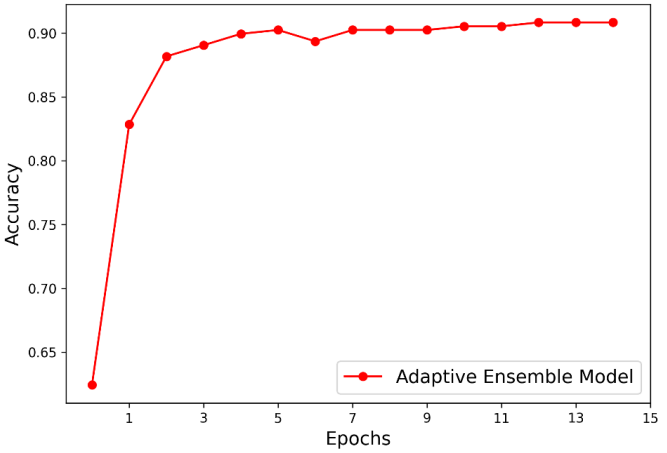

Figure 4. Training process of ensemble model.

3.4. Result of FGSM-ensemble model

This study performs FGSM countermeasure sample training test on the original data set for generalization performance test, and the text disturbance amount epsilon is set to 0.1. The test is run on the original test set after countermeasure training. Table 6 depicts a comparison of the outcomes of numerous indicators. The integrated model of confrontation training has test accuracy rates, recall rates, and F1 scores as high as 91.16%, 90.68%, and 90.89%, demonstrating that the FGSM-Ensemble model described in this study has some anti-text disturbance ability.

Table 6. Result comparison of adversarial training.

	FGSM-Ensemble Model	Adaptive Ensemble Model
Precision	0.9116	0.9090
Recall	0.9068	0.9034
F1 Score	0.9089	0.9058

4. Conclusion

In this study of rumor detection based on integrated neural network models, this paper first trains for four independent single models, TextCNN, TextRNN_Att, LSTM, and Transformer.Then, using weighted averaging, these individual models are combined to create a more potent integrated model. In this thesis, the integrated model is retrained to identify the best integration parameters. The importance of each model in the integration is specified by a weight vector in this research, and the parameters are automatically updated by a back-propagation process. The model's performance can be optimized, and the ideal set of parameters can be discovered by changing the values in the weight vector.

In this paper, a weight vector is set to specify the importance of each model in the integration, and the parameters are updated by a back-propagation algorithm to automatically adjust the parameters. By adjusting the values in the weight vector, the performance of the model can be optimized and the best combination of parameters can be found. This approach allows the integrated model to learn the advantages of different models adaptively, thus improving the overall performance. Finally, in this paper, four single models are weighted and integrated according to the ratio of 0.5562, 0.1477, 0.1470, and 0.1491, and the models have Precision, Recall, and F1 Score of 90.90%, 90.34%, and 90.58%, respectively, in the test set. The integrated model significantly outperformed the four single models in

DOI: 10.54254/2755-2721/20/20231103

the test set in all metrics, and thus it can be determined that the integrated model has higher classification performance.

Subsequently, to enhance the model's resilience, the FGSM algorithm is used for adversarial training in this paper. A minor perturbation is applied to the original data to create an adversarial sample. The best model is created by retraining the model with fresh samples and then evaluating it on the original dataset. The precision, recall, and F1 scores of the integrated model after adversarial training in the test set are 91.16%, 90.68%, and 90.89%, respectively, using four single models created with weights of 0.5307, 0.1718, 0.1411, and 0.1564. In comparison to the integrated model prior to adversarial training, the integrated model after adversarial training achieves greater generalization, robustness, and attack resistance.

The research aspects of this thesis can be further explored and more deeply thought about, and some shortcomings can be improved and looked forward to. Firstly, for the research object, since the forms and types of rumors are very diverse, a larger and representative dataset is needed to solve the data intensification problem. Secondly, in terms of text pre-processing, the current rumor detection models are lacking in interpretability and generalization ability. Thirdly, for the core technology, visualization can begin with the addition of an attention mechanism to observe more intuitively the applicability of each model when combined. Based on this it is possible to find better methods for integrating learning model combinations, such as using dynamic classifier integration, to improve the early detection and generalization performance of rumors.

References

[1] Shunzhi, X. (2022). Overview of rumor detection based on Neural Network, Changjiang Information & Communications, 35(01), 53-56.

[2] Nuo, X, Wei, Z, Keyuan, S, et, al. (2022). Health Rumor Detection based on Pre-Trained Language Model, Journal of Systems Science and Mathematical Sciences, 42(10), 2582-2589.

[3] Al-Sarem, M., Boulila, W., Al-Harby, M., Qadir, J., & Alsaeedi, A. (2019). Deep learning-based rumor detection on microblogging platforms: a systematic review. IEEE access, 7, 152788-152812.

[4] Cao, J., Guo, J., Li, X., Jin, Z., Guo, H., & Li, J. (2018). Automatic rumor detection on microblogs: A survey. arXiv preprint arXiv:1807.03505.

[5] Choi, D., Oh, H., Chun, S., Kwon, T., & Han, J. (2022). Preventing rumor spread with deep learning. Expert Systems with Applications, 197, 116688.

[6] Song, C., Yang, C., Chen, H., Tu, C., Liu, Z., & Sun, M. (2019). CED: credible early detection of social media rumors. IEEE Transactions on Knowledge and Data Engineering, 33(8), 3035-3047.

[7] Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. International journal of data mining & knowledge management process, 5(2), 1.

[8] Luan, Y., & Lin, S. (2019). Research on text classification based on CNN and LSTM. In 2019 IEEE international conference on artificial intelligence and computer applications (ICAICA), 352-355.

[9] Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2020). A survey on ensemble learning. Frontiers of Computer Science, 14, 241-258.

[10] Wong, E., Rice, L., & Kolter, J. Z. (2020). Fast is better than free: Revisiting adversarial training. arXiv preprint arXiv:2001.03994.