Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/20/20231091

Spam classification based on different artificial intelligence methods

Xiaoke Wang

College of Engineering, The Ohio State University, 10 W Northwood Ave, Columbus, OH 43201, US

wang.14538@osu.edu

Abstract. The increased need for social communication has led to an increase in email users, and with it more spam is being spread. In this paper, by comparing and exploring the accuracy of some supervised machine learning methods and a deep learning method called Long short-term memory (LSTM) on the problem of spam classification, this paper aims to provide more solutions for the problem of spam filtering. This paper firstly conducts an in-depth understanding and analysis of the principles of different machine learning algorithm models, which is very helpful for the following research. Then the experimental comparisons after mastering the principles of different models are conducted. Regarding the process of the research, the data set was first pre-processed to facilitate the use of different algorithm models, and then the data set was put into different models for training. Finally, by comparing the accuracy and confusion matrix, it was concluded that LSTM was used in spam classification. problem is more advantageous.

Keywords: attention, LSTM, spam filtering.

1. Introduction

With the development of social communication technology and the expansion of information communication, the frequency of people's use of e-mail is gradually increasing. But with it came more spam, which also became a headache for many users. According to a statistic, from 2007 to 2019, the spam as a percentage of total email traffic in the world showed a significant decline after 2016 [1-5]. The reason is that the evolution of software that can help people detect those spam and filter them out. Nowadays, various machine learning methods are applied to the problem of spam classification, and due to different algorithms, different models also show different efficiencies. By comparing the efficiency and accuracy between different models, this paper aims to provides more possibilities and thoughts for spam classification and filtering.

Some of the more classic machine learning models today, such as Support Vector Machines (SVM), Naive Bayes (NB) and Random Forest (RF), perform quite well, but later there are many deep learning models. These new models can also be used for spam classification. At the same time, there are also advantages that ordinary models do not have. Those supervised machine learning models judge whether it is spam based on the word frequency in the email. The problem with this solution is that it's not a permanent solution for classification - that is, spammers now understand how your spam classifier works - they just look up certain words in the dataset and modify them words in spam to avoid filtering, then

DOI: 10.54254/2755-2721/20/20231091

these algorithms will fail. The neural network correlation model can solve this problem. It does not simply judge the category by the words appearing in the text but studies the relevance of the words [6-9].

In order to solve the problem of garbage classification, this paper first tries to use traditional machine learning models: SVM, RF, and NB, and uses a dataset with nearly 6,000 emails for systematic training, and finally compares and analyses the results. At the same time, this paper will also explore the deep learning algorithm Long short-term memory (LSTM), through the analysis of its principle and results, compare it with the traditional model mentioned above, and deeply analyze its advantages in spam classification and processing. After processing the same dataset, it can be seen that LSTM has the advantage in accuracy, followed by random forest and SVM, and NB has the worst performance.

2. Related work

Traditional methods such as SVM are widely used in spam classification problems, and they have two main advantages compared with modern methods such as the neural networks: their processing speed is faster, and with higher performance when dealing with smaller numbers of datasets [7].

SVM is one of the most classic models in solving spam, its performance and accuracy are quite impressive [1-3]. Bo and Zong-ben found in their study that since SVM classifiers only rely on support vectors, not all data in dataset can affect the classifier, like many neural network systems [4]. But one of the downsides of using SVM-based filters is the high rate of misclassification, especially when dealing with personalized emails [3]. At the same time, the performance of SVM is also very volatile when faced with different datasets, so its classification accuracy is somewhat affected by different datasets [5]. NB assumes that different attributes are opposed to each other. Although the assumption of mutual independence may be incorrect in some real-world problem domains, this assumption is commonly used in most Bayesian correlation calculations, and the results are still very reliable [8-10]. It is precisely because of this feature that its performance is pooled compared to other models [5]. In the study of Bassiouni et al., it was found that compared with NB, SVM and other algorithms, the performance of random forest is quite good: the accuracy reached to 95.45% [11]. In Qaroush, Ismail and Mahdi's research, they use five models: RF, C4.5 Decision Tree, NB, Bayesian Network, and SVM. After training and classifying a dataset containing over 50,000 data, Random Forest stands out [12].

The traditional classification algorithms mentioned above do have good results, but if the spam classification problem is brought back to reality again, the drawbacks of these algorithms will appear. The data brought into the different models of SVM, NB, and Random Forest are pre-processed vectors to represent different characters. That is to say, people usually train these models according to the frequency of different characters. These algorithms can easily fail if the writers who edit spam change their wording and writing. LSTM has shown great advantages in dealing with this problem. It can remember or forget data in a more efficient way, so performance should be better. This article will compare LSTM and other models for research.

3. Methods

In this research, the dataset first needs to be pre-processed as seen in the Fig 1. In this figure, data pre-processing is the first step: for the algorithm of supervised machine learning, considering that the data in the data set is content based, through pre-processing, all text data are converted into corresponding vectors, and the frequency of occurrence of different words is counted. Similarly, for the LSTM algorithm, it is necessary to perform tokenizer and other processing on the data before training the model. Finally, the processed data is brought into different algorithm models. After using different methods, results need to be compared and evaluated.

DOI: 10.54254/2755-2721/20/20231091

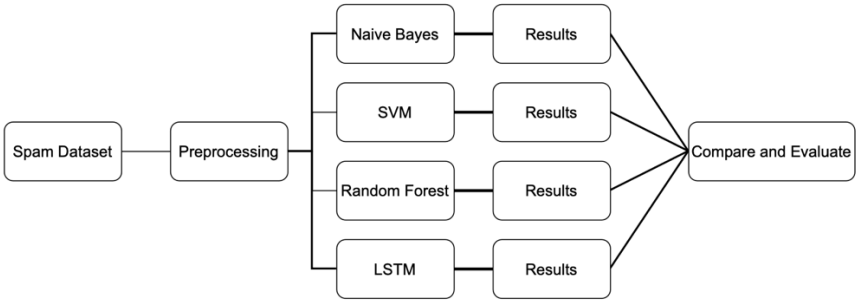


Figure 1. Research Flow Chart.

3.1. Naïve bayes classifier

The Naïve Bayes algorithm applies the Bayes Theorem, and it is one of the most classic and simplest classifiers for spam classification problem [12]. The default setting of the model is to consider all features to be independent of each other. That means all features contribute independently to the probability of categories. In the Naïve Bayes algorithm, the frequency of different words contained in emails is the feature and the category to be predicted is whether spam or not [1-2].

Given a vector $X = \{x_1, x_2, \dots, x_n\}$ which represents n features of an email. Given the category to be predicted $C = \{c_1, c_2, \dots, c_n\}$. For spam filtering, $C = \{spam, ham\}$. Using the Bayes Theorem, the conditional probability $P(C_k|X)$ can be calculated by:

$$P(C_k|X) = \frac{P(C_k)P(X|C_k)}{P(X)} \quad (1)$$

where $P(C_k)$ denotes the probability of different categories, $P(X|C_k)$ denotes the probability of X given category C_k. $P(X)$ means the probability of different features, which is independent of each other. Since $P(X)$ appears in every class, then it can be ignored. That means in the Naïve Bayes classifier, only $P(C_k)P(X|C_k)$ needs to be considered and it is denoted by $f(X)$ [13]. This function is called Bayes discriminant function. Although Naïve Bayes algorithm has the assumption that each input is independent with others, it is still quite efficient and useful for content-based classification [8]. Therefore, approximation is needed when features become multi-dimensional and interrelated [13]. The Bayes discriminant function mentioned above can be evolved into:

$$f^{NB}(X) = \Pi_{i=1}^{n} P(C = c_i)P(X = x_i|C = c_i) \quad (2)$$

3.2. Support vector machine classifier

SVM is supervised learning model that is very useful for classification. Given some labeled training data, SVM algorithm builds model to assign new data into different categories. The margin around the hyperplane is maximized and it is used to separate the different categories. Based on this principle, spam filtering could be considered as a simple application of SVM, that is, whether the new message is in the spam category or not [1]. In the SVM algorithms, one of the main elements to separate different classes of patterns is the kernel function. Using the kernel function, the SVM constructs a non-linear decision surface for the input, but linear for the feature [6]. Choosing the kernel function is of vital importance for the efficiency of SVM. There are four different types of kernel function in total: linear, polynomial, RBF, and sigmoid.

Given a dataset of n points of a form: $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, where y_i is either 1 or -1 determines which category this point belongs to and x_i is a p-dimension vector. By using different kernel function, the maximized margin hyperplane which divides points having $y_i = 1$ from points having $y_i = -1$. For spam classification, y_i = 1 or -1 represents if this email is spam or ham. SVM has many advantages over other algorithms, one of which is that it is very fast, especially when the size of processing data is not particularly large [7]. Another advantage that it has to mention is that the SVM is

DOI: 10.54254/2755-2721/20/20231091

not affected by the imbalance in the number of different classes in the training set [4]. Based on these principles and advantages of SVM, using SVM is good for spam classification problems.

3.3. Random forest classifier

Random forest is a common machine learning algorithm that combines the outputs of many decision trees to arrive at a final outcome [9]. Its ease of use and flexibility drive its adoption as it can be used to handle classification problems. It is an extension of the bagging method as it exploits bagging and feature randomness to create forests of uncorrelated decision trees. The output of a random forest is the class chosen by most trees. First in a random forest, n random records are taken from the dataset with a total of k records, then many separate decision trees are built for different samples, and each decision tree will produce an output. Finally, output and regression are considered separately based on majority voting or averaging, respectively.

Given the following formula for Random Forest:

$$ni_j = w_j C_j - W_{left(j)} C_{left(j)} - W_{right(j)} C_{right(j)} \quad (3)$$

where ni_j denotes the node j's importance, w_j is the node j's weighted number, C_j is the impurity value of node j. The node j's left and right child are denoted by left(j) and right(j).

3.4. Long-short term memory classifier

The traditional neural network exposes shortcomings when dealing with the problem of data contextual correlation, and LSTM solves this problem very well.

LSTM networks are a special type of recurrent neural network (RNN). LSTMs have the ability to connect long-term dependencies between time steps in a sequential manner for questions such as spam classification, which are superior to traditional recurrent neural networks in terms of memory [10]. It was first proposed by Hochreiter & Schmidhuber in 1997 [13]. When dealing with long term information, RNN will be very laborious, and may even cause vanishing gradient problem. But LSTM can solve the problem of memorizing long-term information. The LSTM structure has LSTM units consisting of memory units that selectively store information for future use. Through these cells, the model in each cell can decide what information can be stored and what information should be discarded. The LSTM analyzes data in a time sequential manner. When dealing with the input information, it first takes the current input X_t and then grab the output from the previous hidden state h_{t-1} for each time period. That means when facing the problem of spam detection, LSTM can well connect the relationship between the words before and after and make a more accurate classification.

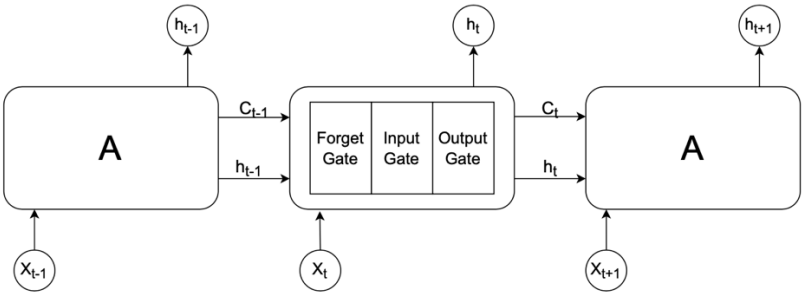


Figure 2. LSTM working unit examples.

In the Fig 2 above, each A represents copies of same neural network, so it can be considered as a working unit. *X* denotes the input and *h* denotes the output. Each working unit receives three different inputs: the current input information X_t, the hidden state, denotes by h_{t-1}, and the unit state from the previous working unit, denotes by C_{t-1} . To deal with all those input information, three different gates are used: input gate, forget gate, and output gate [14]. The forget gate is made by a sigmoid layer and decides what information to remember and what to forget:

DOI: 10.54254/2755-2721/20/20231091

$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f) \quad (4)$$

Here f_t is the forget gate output, W is neural network weight matrix and b is the neural network bias vector. The forget gate takes X_t and h_{t-1} and output a binary number (either 1 or 0) for each number in the unit state C_{t-1}, Here 1 represents keep this information in the current working unit and 0 represents forget this information.

To determine what information to store in the new unit state C_t, the input gate is needed:

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i) \quad (5)$$

$$\tilde{C}_t = tanh(W_c \cdot [h_{t-1}, X_t] + b_c) \quad (6)$$

The sigmoid layer i_t decides what value will be used in the next unit state, and $\tilde{C}_t$ creates candidate values by using tanh layer that uses to update unit state. Then the unit state C_t can be updated by using the following formula:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (7)$$

The final step is to decide what values to output, first the network uses a sigmoid layer to change different values into 1 or 0, so that the working unit can decide what values to output (1 means output this value, 0 means not). Then, the C_t from the previous step is pushed into the tanh layer and multiply values from the sigmoid layer. Thus, only the portion that the network wants to output will be output. The formulas for this part are shown below in formula 6 and 7.

$$o_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o) \quad (8)$$

$$h_t = tanh(C_t) * o_t \quad (9)$$

3.5. Attention - LSTM

Although the LSTM has shown advantages in dealing with text classification problems, there are still some problems. LSTM defaults that the impact of different features on the model is the same, but in fact the impact of different features on the LSTM network is different, and the emergence of attention solves this problem. Instead of assigning weights equally to all features, attention can assign weights to different features according to the correlation between the input value and output value of the algorithm, just like when the brain receives a lot of information at the same time and decides which information is more important.

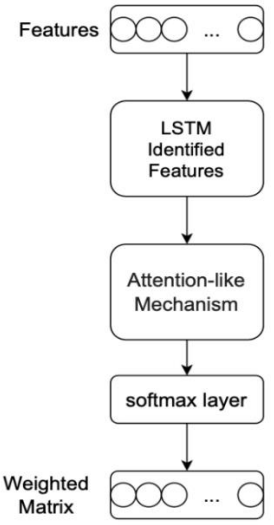


Figure 3. Attention-LSTM Feature Processing

DOI: 10.54254/2755-2721/20/20231091

First, as shown in Figure 3, the input feature is encoded by the function of LSTM, and then it is input into the attention mechanism to obtain the correlation value between the input feature and the output of the LSTM algorithm. In order to assign weights to different features, we need to use a softmax layer to normalize the obtained correlation values. Finally, the weight matrix can be obtained. Different features are weighted according to the obtained weight matrix, and then input into the LSTM model.

4. Results & Discussion:

4.1. Dataset:

The dataset that is used in this research is from the UCI website, 5572 instances included [14]. In this dataset, the text content of the e-mail is the only feature. All those two datasets have one target attributes, that is e-mail categories: 1 stand for spam and 0 stand for ham, the percentage distribution is shown in Figure 4 below, among the 5572 data, nearly five thousand data are not spam, and the remaining less than 1000 data belong to spam.

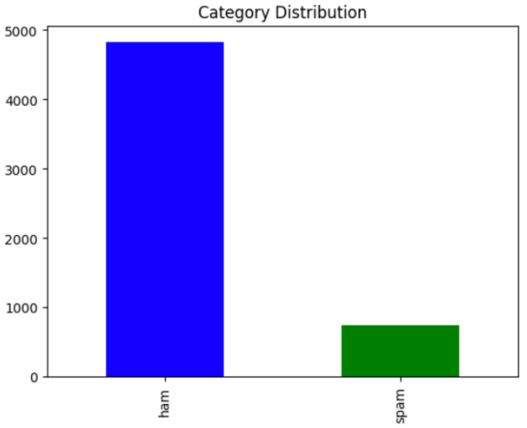


Figure 4. Dataset Category Distribution.

4.2. Data preprocessing:

In order to use this dataset, preprocessing is needed. All content except word will be removed, such as single letters, URL links. At the same time, the shortened words in the text will be converted into separate words, and all words will be converted to lowercase. Stop words are also removed to improve the analytics. The left side of the Figure 5 below shows the 10 most frequently occurring words before processing, and the right side shows the results after processing. There have been significant changes before and after processing which is great for increasing the accuracy of the models.

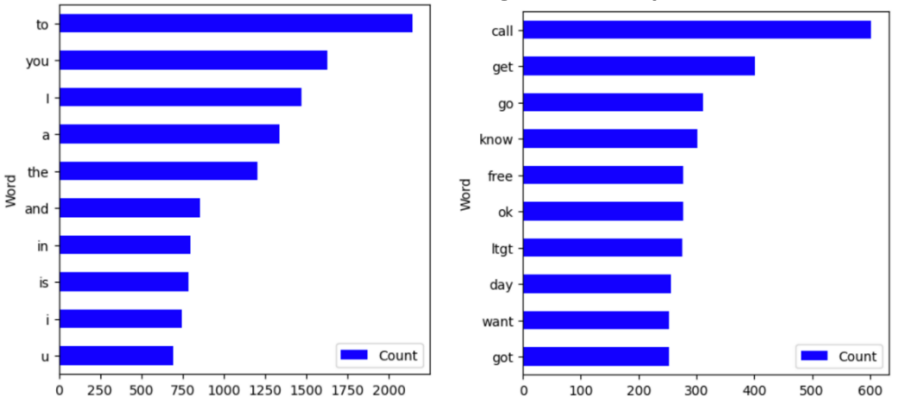


Figure 5. Top 10 Words in Email Before/After Processing.

DOI: 10.54254/2755-2721/20/20231091

Then, in order to bring the text data into the model, according to the frequency of occurrence of different words, CountVectorizer () is used to convert the text data into a vector.

4.3. Compare & Evaluate:

Next, the processed data is brought into different algorithm models, and it can be seen that the accuracy of different algorithms are different, and the accuracy of LSTM is the highest.

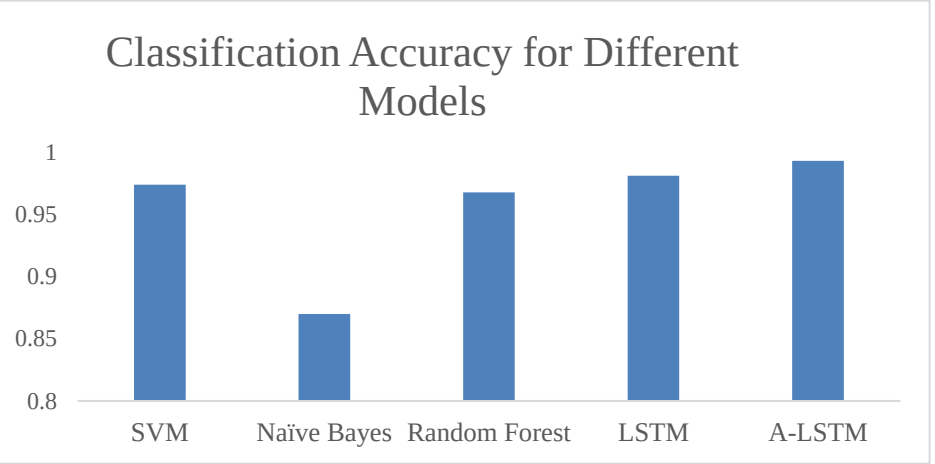
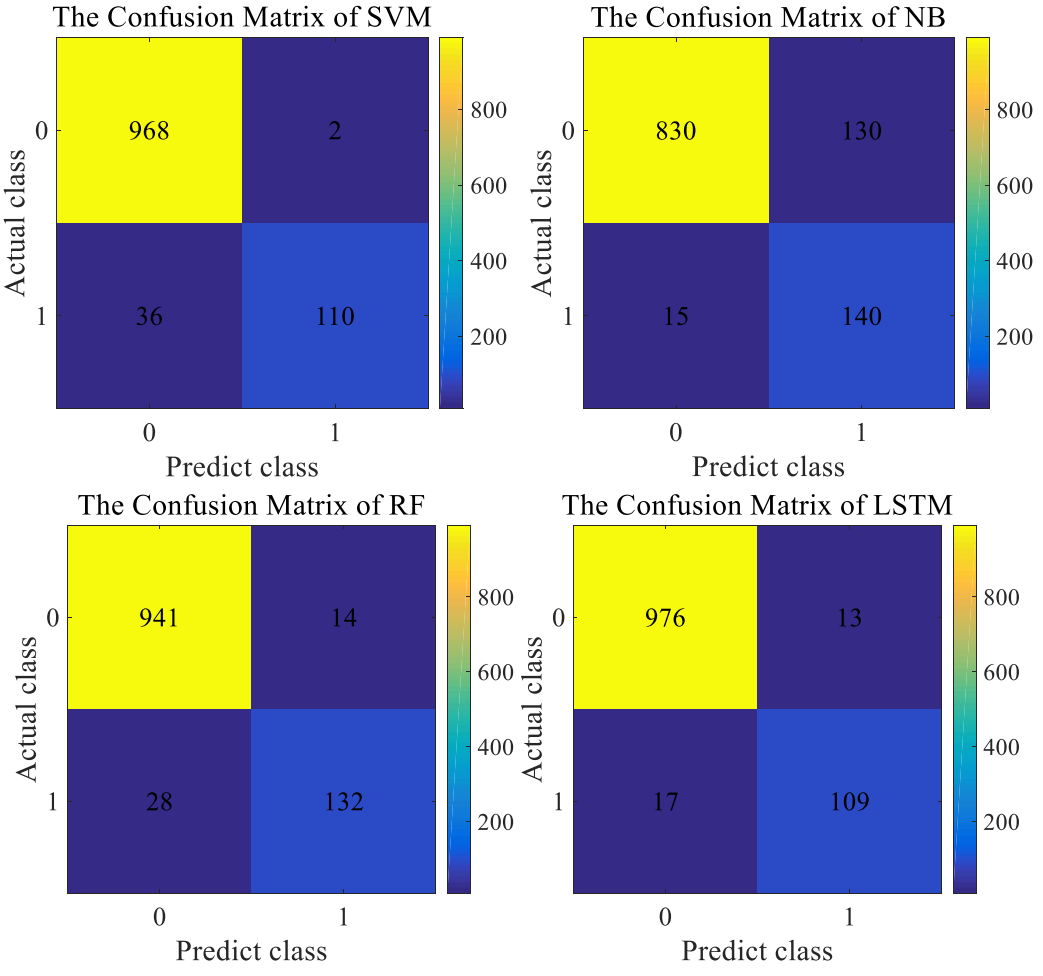


Figure 6. Accuracy comparisons among different models.



DOI: 10.54254/2755-2721/20/20231091

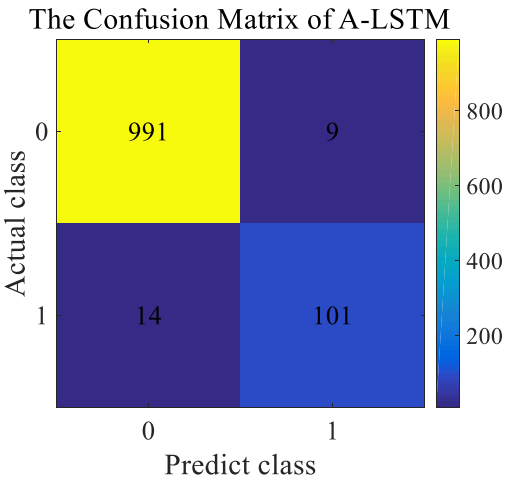


Figure 7. Confusion matrix for SVM, NB, RF, LSTM, and A-LSTM.

The results shown in Figure 6 reflect that in addition to LSTM, A-LSTM, SVM and Random Forest perform relatively well compared to Naive Bayes. In Figure 7, the type II error for Naïve Bayes model is significantly larger than SVM and Random Forest, so the accuracy for NB is lower than others. As for why the performance of random forest is so good, a natural advantage of this model is that it can assign a clear value to each small decision tree and each output, and finally make the judgment with the highest accuracy by voting or averaging [9] This decreases vagueness in decision-making. The feature of this model makes its accuracy is better compared to previous techniques when filtering spam. Regarding LSTMs, as expected, this model outperformed the three previously mentioned models. LSTM is a special classification in RNN, and its significance is to solve the long-term dependence problem between information that RNN cannot solve. Therefore, sometimes remembering some information for a long time or forgetting some information immediately are the characteristics of LSTM, not something that the developers need to worry about. These features make it effortless when dealing with problems like spam classification. They have a special architecture that enables it to forget information that is unnecessary. This model learns which information to keep for the future use and which to forget and throw away, so the final accuracy is about 98.1%. The attention mechanism is added on the basis of LSTM, which changes the consistency of the weights of different features, so that the LSTM model pays more attention to those more important features, so as to make more accurate judgments. Therefore, the accuracy of A-LSTM is closer to one, reached 99.3%.

5. Conclusion

With the increasing demand for communication in society and the increasingly prominent spam problem, this paper compares the traditional machine learning algorithms with the deep learning algorithm LSTM to explore the classification of the neural network model in deep learning in processing content-based data. The advantages in the question provide better ideas and inspiration for the spam detecting problem. In this paper, for the classification of spam, this research first tried several classic machine learning methods such as SVM, NB, and RF. Although both SVM and RF have good performance, these methods are limited to supervised machine learning. algorithm, so in the second half of this paper, LSTM is chosen. The results are obvious, LSTM shows an absolute advantage in accuracy. The accuracy of LSTM is about 98.1%, which is obviously higher than that of classic machine learning methods (SVM, NB, and RF). After adding the attention mechanism to A-LSTM, its performance is even better, so it can be shown that for the answers of spam classification, the deep learning method is obviously more advantageous.

DOI: 10.54254/2755-2721/20/20231091

References

[1] Yu B, Xu Z, A comparative study for content-based dynamic spam classification using four machine learning algorithms. 2008 Knowledge-Based Systems 21.4: 355-362

[2] Feng L, Wang Y, Zuo W, Quick online spam classification method based on active and incremental learning. 2016 Journal of Intelligent & Fuzzy Systems 30.1: 17-27.

[3] Abayomi A, Olusola, et al. A review of soft techniques for SMS spam classification: Methods, approaches and applications. 2019 Engineering Applications of Artificial Intelligence 86: 197-212.

[4] Drucker, Wu D, Vladimir N. Support vector machines for spam categorization. 1999 IEEE Transactions on Neural networks 10.5: 1048-1054.

[5] Shams, Rushdi, Robert E. Mercer. Supervised classification of spam emails with natural language stylometry.2016 Neural Computing and Applications 27.8: 2315-2331.

[6] Almeida, Tiago, Renato, and Akebo Y, Machine learning methods for spamdexing detection. 2016 International Journal of Information Security Science 2.3: 86-107.

[7] Rodrigues, Anisha P., et al. Real-time twitter spam detection and sentiment analysis using machine learning and deep learning techniques. 2022 Computational Intelligence and Neuroscience.219-232.

[8] Yang F, An implementation of naive bayes classifier. 2018 International conference on computational science and computational intelligence. 57-68.

[9] Breiman, Leo. Random forests.2001 Machine learning 45.1: 5-32.

[10] Jain, Gauri, Manisha Sharma, and Basant Agarwal. Spam detection in social media using convolutional and long short-term memory neural network. 2019 Annals of Mathematics and Artificial Intelligence 85.1: 21-44.

[11] Bassiouni M., Ali M. Ham and Spam E-Mails Classification Using Machine Learning Techniques, 2018 Journal of Applied Security Research, 13:3, 315-331.

[12] Kim, Chanju, and Kyu-Baek Hwang. "Naive Bayes classifier learning with feature selection for spam detection in social bookmarking." Proc. Europ. Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD). 2008.

[13] Rish, Irina. "An empirical study of the naive Bayes classifier." IJCAI 2001 workshop on empirical methods in artificial intelligence. Vol. 3. No. 22. 2001.

[14] https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection