An Improved Convolutional Neural Network-Based Spam Recognition Model

Jinyuan Liu

School of Automation, Chongqing University of Posts and Telecommunications, Chongqing 400000, China
2021213399@stu.cqupt.edu.cn

Abstract. Spam is one of the significant threats to cyber security by not only sending unwanted messages but also by potentially carrying viruses. Conventional spam detection methods, such as keyword matching and rule-based filtering, are less effective since spammers could advance their method to bypass these simple detection approaches. Machine learning algorithms can quickly and effectively identify subtle relationships among email texts, thus providing a more promising defense against spams. In this work, a Convolutional Neural Network (CNN)-based approach to spam recognition, leveraging the power of deep learning to process and analyze email content. This method is designed to address the shortcomings of traditional methods by employing a deep learning architecture that can generalize well to new, unseen data. Through detailed experimental analysis, the paper demonstrates that the proposed model not only achieves high performance in detecting spam but also significantly reduces the incidence of false positives, which is crucial for maintaining user trust and ensuring that normal emails will not be wrongly classified as spam.

Keywords: Convolutional Neural Networks, Spam Recognition, Cyber Security.

1 Introduction

With the wide use of e-mail, spam has become a problem that users can't ignore in their daily mailbox. Spam is not only annoying, but also may bring serious security risks. First of all, spam mails will occupy users' valuable time and energy, affect the normal use of email experience, and even lead to the burial of important emails. Secondly, spam often hides fraudulent information, malicious links or virus attachments, which brings potential risks to users' personal information and computer systems [1,2].

In light of the prevailing spam issue, it holds paramount significance to delve into spam recognition technology. An effective spam recognition system can help users filter out a large number of waste pieces and improve work efficiency and information security. At the same time, the study of spam recognition technology can also

Y. Wang (ed.), *Proceedings of the 2024 2nd International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2024)*, Advances in Computer Science Research 115,
https://doi.org/10.2991/978-94-6463-540-9_66

promote the application of artificial intelligence in the field of information security and promote the development of intelligent information management.

In the past, spam recognition methods are: rule-based, Bayesian classifier-based, support vector machine-based, decision tree-based, etc. These traditional spam recognition methods have achieved some results in the past research and practice, but with the continuous evolution and change of spam, they may face problems such as poor adaptability and low accuracy. Therefore, deep learning methods based on, with better feature learning and classification capabilities have gradually become a hot research topic. Deep learning models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) can automatically learn intricate feature representations from data, aiding in precise discrimination between spam and legitimate emails [3,4]. In this paper, by constructing a CNN-based deep learning model and training and optimizing it, the accuracy and robustness of the spam recognition system can be improved to create a cleaner and safer email environment for users.

2 Related Work

Nowadays there are many machine learning-based methods for spam recognition, for example, El-Alfy discovered that employing various methods to complete the feature set can mitigate complexity. They utilized Support Vector Machines and Plain Bayes for classification, amalgamating factors like URLs, potential spam vocabulary, sentiments, symbols, special strings, and subject matter [5]. This method utilizes a variety of feature set filtering and can improve classification accuracy, but there are problems of feature selection and model tuning.

The method of Jialin considers symbolic terms, background summaries and subject values and represents unwanted emails based on the possibility of hidden system analysis [6]. The paper applies the k-mean algorithm to solve the rare problem. This method integrates multiple factors for classification and can effectively deal with the rare problem, but there may be a problem of local optimal solution due to the improper selection of the parameters and initial values.

Mohammed proposes a new technique to improve the accuracy of spam filtering by reweighing the service burden and improving text processing [7]. The approach operates based on message size and severity, with a particular focus on the misleading character workbook's output. It also introduces a reweighing function to recalibrate the weights. In this way, spam emails that try to deceive users through misleading character workbooks can be identified and filtered more efficiently, and the accuracy of message importance assessment can be improved. However, this approach may have some potential disadvantages. For example, re-burdening the service may increase the computational and processing load on the system, thus affecting performance. Also, improved word processing may require complex algorithms and models, which may increase the cost of implementation and maintenance.

Such traditional machine learning-based spam recognition methods improve classification accuracy and enhance the model's ability to recognize spam by utilizing

multi-feature ensemble filtering, integrating multiple factors, and improving text processing [8, 9]. However, challenges such as feature selection problems, parameter selection problems, and computational and processing burdens arise.

CNN-based spam recognition methods have advantages in automatic feature learning, generalizability, scalability, and the effect of adapting to process text data, and the model performance can be improved by adjusting the network structure, introducing the attention mechanism, data enhancement, and hyper-parameter tuning.

3 Method

3.1 Overall Design

In spam classification, every email can be regarded as an amalgamation of distinct words, with each word serving as a feature. The probability of occurrence of each word in a spam email and the probability of occurrence of each word in a normal email are calculated by tallying the frequency of each word's occurrence in both spam and legitimate emails within the training dataset. For classification, the email text could be represented as a collection of words. Next, a CNN model is constructed, which performs well in text classification tasks because it is able to capture both local features and global information efficiently. The model consists of an embedding layer for converting the text data into a dense vector representation, convolutional and pooling layers for extracting features from the text, and a fully connected layer for the final classification.

3.2 Model Implementation

CNN-Based Modeling. Convolutional neural nets are used for spam classification, where an embedding layer is used to map the text vocabulary to a dense vector representation, a convolutional layer is utilized to extract localized features from the text, followed by a maximum pooling layer for dimensionality reduction and preservation of the most significant features. Subsequently, a fully-connected layer is utilized for classification, producing a probability value that signifies the text's likelihood of being spam. CNNs have been very successful in the field of image processing but also perform well in tasks such as text classification. CNNs have had great success in image processing, but also perform well in tasks such as text classification, as their convolutional layers automatically extract local features from the text, and progressively extract more advanced features through multiple layers of convolutional layers and pooling.

The CNN model used in this paper has six layers, first the text data is processed through an embedding layer to map each word into a 100-dimensional vector for subsequent processing. Next, it is processed through a one-dimensional convolutional layer, where local features are extracted using 128 filters of size 3 and nonlinearly transformed by the Rectified Linear Unit (ReLU) activation function. Multi-scale convolution and residual concatenation are introduced after each convolutional layer to capture features at different scales and deepen the learning capability of the model.

Subsequently, a maximum pooling layer is applied to reduce the dimensionality of the convolutional layer outputs while preserving crucial features, employing a pooling window size of 2. The Flatten layer spreads the outputs of the convolutional and pooling layers into a one-dimensional vector for connecting to the fully connected layer. A self-attention layer is introduced before the fully connected layer to increase the model's attention to the important information in the text sequence. Finally, the fully connected layer has only one unit that outputs the binary classification results using the Sigmoid activation function, which realizes the processing from text data to binary classification results.

Attention Mechanisms. Attention mechanisms are extensively employed in natural language processing to enable models to concentrate on pertinent segments when handling sequential data. Incorporating an attention mechanism within a CNN empowers the model to adeptly grasp crucial information within a text sequence, thereby enabling it to automatically discern the significant segments of the text and allocate more attention to those segments. The self-attention mechanism introduced after the convolutional layer allows the model to dynamically weight the important information in the input sequence globally, thus improving the model's expressiveness and generalization ability.

In the self-attention mechanism, the level of attention of the model during processing is determined by calculating the weights of each position in the input sequence. These weights are usually obtained by calculating the similarity between different locations in the input sequence and then converting these similarities into attention weights. Eventually, these weights are used to weight the feature representations in the input sequence to obtain more important information.

By integrating the attention mechanism, convolutional neural networks can effectively capture pivotal information within text sequences. This enhancement bolsters the model's performance, augments its generalization capability, and facilitates superior adaptation to diverse natural language processing tasks.

Segmentation and Removal of Stop Words. The data preprocessing phase performs operations on the dataset such as disambiguation and deactivation to convert the text into a feature vector representation that can be used to train the model. Segmentation allows continuous text sequences to be sliced into the smallest meaningful units (words or symbols), while deactivation reduces noise and improves feature quality.

Feature Extraction Using Term Frequency-Inverse Document Frequency (TF-IDF). Methods such as TF-IDF are used to convert preprocessed text features into vector representations that can be used for model training [10]. This is a commonly used text feature extraction method that takes into account the trade-off between word frequency and inverse document frequency, and is able to transform text data into a numerical form that can be processed and understood by computers.

Training Details. The binary crossentropy is one of the commonly used loss functions when performing binary classification tasks. This loss function is suitable for binary classification problems in the form of model output probability, it calculates the loss by comparing the probability distribution of the model output with the distribution of the actual labels, its formula is as follows:

$$Loss = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log\big(p(y_i)\big) + (1 - y_i) \cdot \log\,(1 - p(y_i)) \tag{1}$$

where y_i is the binary label 0 or 1, and $p(y_i)$ is the probability that the output belongs toy_i probability of the label.

Adaptive Moment Estimation (Adam) is an adaptive learning rate optimization algorithm that amalgamates the benefits of Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square prop (RMSProp). By computing the adaptive learning rate for each parameter, the Adam algorithm dynamically adjusts the step size of parameter updates, leading to accelerated convergence and enhanced performance during training.

4 Experiments and Results

4.1 Dataset

This experiment uses a labeled dataset obtained from the web, which contains 12500 Chinese emails, in which normal emails are labeled as “ham” and spam emails are labeled as “spam” [11]. In order to validate the generalization performance of the model, this paper randomly divides the dataset according to the ratio of 7:2:1, in which 70% is used for training, 20% is used for validation, and the remaining 10% serves as the test set. At the end of the experiment, an unlabeled dataset of another 20,000 mixed normal and spam emails was used to further validate the model. This dataset is derived from the Chinese dataset (trec06c) in the TREC 2006 Spam Track Public Corpora public spam corpus provided by the International Conference on Text Retrieval, and is composed of a mixture of normal and spam e-mails after removing labels from them and randomly extracting normal e-mails and spam e-mails. And click on the prediction after entering the emails into the created software window to test whether its model building is successful or not.

4.2 Experimental Steps

First, the labelled model is trained using the training and validation sets. Performance is then evaluated using the validation set. Subsequently, the test set, which is not involved in training, is utilized to validate the accuracy of the model. By comparing the results with those on the training and validation sets, the generalization performance of the model can be evaluated more comprehensively. Finally, the accuracy is compared with other training models to evaluate the performance of the model on the spam recognition task. In the experiments, to calculate the accuracy of the classifier, it is done by comparing the predicted results with the true labels to get

an array of Boolean type with True or False elements indicating whether the prediction is correct or not for each sample. Then this array is averaged to get the accuracy of the classifier.

After the model training and testing is completed, another set of unlabeled email dataset is entered into the software window, and if it is determined to be spam, the output is "This is a spam email, please delete it." If it is recognized as spam, it will output "This is a spam mail, please delete it." and if it is a normal mail, it will output "This is a normal mail, please send it." In this way, spam is recognized.

4.3 Result Analysis

After testing, the model training took 3735s, and its accuracy is 99.88% on the validation set and 99.67% on the test set, and it can output the correct results in the software window, which indicates that the CNN-based machine learning model has high accuracy for spam recognition.

In order to better compare the model performance, other machine learning models were also selected for testing in this experiment, resulting in the accuracy rates under the same dataset as shown in Table 1.

Table 1. Results Comparison.

Method	Accuracy
Plain Bayesian Model	98%
Logistic Regression Model	98%
Support Vector Machine with TF-IDF	93%
Support Vector Machine with bag of-words	98%
CNN-Based Model	99%

Convolutional neural network-based text categorization methods have the advantage of automatically learning features and processing complex text, but they are prone to overfitting on small data, require more resources to process long text, and may not perform as well as other models under specific text structures.

Support Vector Machine based on bag-of-words model has advantages and limitations in text categorization. It is suitable for handling short text tasks and has good generalization ability under high dimensional feature space. However, it ignores the order and semantic information between words, is slow to train, and is not flexible enough to handle synonyms.

Both support vector machines based on bag-of-words model and support vector machines based on TF-IDF are common in text categorization. The bag-of-words model is suitable for processing short text tasks and has good generalization ability under high-dimensional feature space. However, it ignores the order and semantic information between words, is slow in processing large-scale data, and is not flexible enough for synonyms. In contrast, TF-IDF improves the model's consideration of word importance and is especially suitable for tasks that require attention to word

weights. However, it also ignores semantic relationships and is computationally expensive when dealing with large-scale data or long texts.

Logistic regression-based text categorization is suitable for simple binary classification tasks with simple and easy to interpret models. However, the performance may be poor when the data has a complex nonlinear structure. In addition, when dealing with high-dimensional features, logistic regression has a weak generalization ability and is prone to overfitting. Logistic regression may also not perform flexibly enough for texts containing complex semantic and contextual information.

Text classification based on plain Bayesian model is a common probability-based classification method. Its main advantages lie in its computational simplicity and speed, which makes it particularly suitable for dealing with small-scale datasets and able to achieve good classification results on a relatively small amount of data. Indeed, the plain Bayesian model's assumption of feature independence can be limiting, especially when handling high-dimensional and complex datasets. In such cases, dependencies between features may exist, affecting classification accuracy. This limitation underscores the importance of exploring more sophisticated probabilistic models or incorporating techniques like feature engineering to address the challenges posed by intricate datasets. Therefore, in practical applications, it is necessary to choose an appropriate classification method based on specific data characteristics and task requirements to achieve better classification results.

Finally, the experiment proves that the CNN based machine learning model has better accuracy as well as application in the text classification task for spam recognition.

5 Conclusion

In this paper, the machine learning based spam recognition method is studied and explored, which shows that it has important research significance, but there are still many problems such as feature selection, parameter selection, and accuracy to be improved. In this paper, a CNN-based machine learning method is used to build a model for spam prediction, and its model structure includes six layers: embedding layer, convolutional layer, maximum pooling layer, fully connected layer, and self-attention layer, and through the introduction of the attention mechanism, the introduction of word splitting and the removal of deactivated words in the preprocessing stage, the use of TF-IDF feature extraction, the use of binary crossentropy loss function, adam optimizer, etc., combined with the advantages of CNN in extracting local features to improve the model for better performance, accuracy and generalization.

After building the model, the Chinese email dataset downloaded from the Internet is used to train and test its performance. To verify the superiority of the model, this paper uses other machine learning methods such as TF-IDF-based support vector machine, logistic regression model, and plain Bayesian model for training and prediction under the unified dataset, and the experiments prove that CNN-based

model has a better performance and accuracy for spam email recognition with better performance and accuracy in this study.

Future research on spam classification may increasingly emphasize the utilization of deep learning techniques due to technological advancements. Pre-training models possess robust feature learning capabilities, enabling them to effectively capture intricate relationships and semantic information in text, consequently enhancing the accuracy of spam classification.

In addition, personalized filtering is an important development direction. Future spam classification systems may pay more attention to the personalized needs of users, adjusting according to their historical behavior and feedback, providing filtering strategies that are more in line with their habits and preferences, and improving the user experience.

References

1. Chen, Z., Tao, R., Wu, X., Wei, Z., & Luo, X.: Active learning for spam email classification. In Proceedings of International Conference on Algorithms, Computing and Artificial Intelligence, 457-461 (2019).
2. Mujtaba, G., Shuib, L., Raj, R. G., Majeed, N., & Al-Garadi, M. A.: Email classification research trends: review and open issues. IEEE Access, **5**, 9044-9064 (2017).
3. Yoon, K.: Convolutional Neural Networks for Sentence Classification. arXiv preprint arXiv:1408.5882 (2014).
4. John-Africa, E., & Emmah, V. T.: Performance Evaluation of LSTM and RNN Models in the Detection of Email Spam Messages. European Journal of Information Technologies and Computer Science, **2**(6), 24-30 (2022).
5. El-Alfy, E. S. M., & AlHasan, A. A.: Spam filtering framework for multimodal mobile communication based on dendritic cell algorithm. Future Generation Computer Systems, **64**, 98-107 (2016).
6. Ma, J., Zhang, Y., Liu, J., Yu, K., & Wang, X.: Intelligent SMS spam filtering using topic model. In 2016 International Conference on Intelligent Networking and Collaborative Systems, 380-383 (2016).
7. Mohammed, T. A., & Mohammed, A. B.: Security architectures for sensitive data in cloud computing. In Proceedings of the 6th International Conference on Engineering & MIS, 1-6 (2020).
8. Crawford, M., Khoshgoftaar, T. M., Prusa, J. D., Richter, A. N., & Al Najada, H.: Survey of review spam detection using machine learning techniques. Journal of Big Data, **2**, 1-24 (2015).
9. Teja Nallamothu, P., & Shais Khan, M.: Machine Learning for SPAM Detection. Asian Journal of Advances in Research, **6**(1), 167-179 (2023).
10. George, P., and Vinod, P.: Machine Learning Approaches for Spam Email Filtering. in Proceedings of International Conference on Security of Information and Networks, 271-274 (2015).
11. Spam Email Detection. URL: https://cloud.tencent.com/developer/news/275120. Last Accessed: 2024/05/11