Feature Selection-based Integrated Learning Model in Violation Comment Detection

Shuo Lv
Department of Information Engineering,
Wuhan Business University
Wuhan, China

Qing Shen*
Department of Information Engineering,
Wuhan Business University
Wuhan, China
shenqing0611@126.com

2024 IEEE 7th Eurasian Conference on Educational Innovation (ECEI) | 979-8-3503-0720-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/ECEI60433.2024.10510851

***Abstract*—With the rapid development of Internet technology and the increasing popularity of social network media, people often encounter insults, abuse, slander, and other illegal comments on the Internet, which is called "network violence". These Social phenomena are continuously emerging, leading to vicious incidents. To maintain a healthy network environment, illegal speech must be identified and monitored. It is essential to use technology governance to regulate the network environment and curb cyber violence incidents. Based on the data set of Chinese violation comments, we proposed an ensemble learning model based on feature selection using TF-IDF to extract text features. The feature importance method of random forest was used for feature selection. Then, the grid search method was used to optimize the parameters of multiple classification models. After identifying several models with strong classification performance, they were integrated to obtain the ensemble learning model with the best performance. Compared with other traditional machine learning models, the proposed model showed superior detection ability and identified illegal comments.**

Keywords—violation comment recognition, importance of random forest features, integrated learning, hyperparameter optimization

I. INTRODUCTION

The Internet has the characteristics of openness, virtual nature, concealment, divergence, permeability, and randomness [1]. Therefore, people can use the characteristics of the Internet to express various remarks and opinions freely. However, several netizens lack a sense of social responsibility when expressing their opinions, use bad expressions to vent their emotions, and even escalate into personal attacks. This leads to the emergence of network violence and seriously affects the mental state of the parties involved in the incident. Accordingly, the managers of such illegal information must monitor them to protect the rights and interests of the victims and maintain the order and stability of society.

In order to identify these illegal comments, scholars have carried out related research using the knowledge of text emotion analysis. Huang and Zhang [2] conducted the emotion analysis of commodity reviews based on word2vec to establish an emotion dictionary through semantic similarity calculations. Zhang and Song [3] proposed a method to classify texts in bursts using TF-IDF. Cao and Bai [4] proposed a text classification method based on the combination of domain emotion dictionary and word feature fusion. Lv et al. [5] transformed malicious comment classification into dichotomy using deep recurrent neural networks and improved model accuracy with stack generalization integrated.

We proposed an integrated learning model based on feature extraction to reduce the amount of operation of model learning and improve the accuracy of the model. We combined the prediction of multiple basic models to improve the prediction performance. Integrated learning can reduce the variance of the model and improve the generalization ability, thus achieving better results in solving complex problems.

II. DATA ACQUISITION AND PREPROCESSING

A. Data Collection

The data were derived from the Chinese offensive language detection dataset on GitHub [6]. It contains 37,480 sentences covering topics such as race, gender, and region. 18,041 sentences were labeled as normal comments (labeled 0), and 19,439 sentences were labeled as offensive comments (labeled 1). Some of the data are shown in Table I.

TABLE I. TEXT DATA SET TABLE

Topic	Label	Text
Region	1	Henan people steal manhole covers, saying that there are rotten things
Race	0	The range of the yellow race is larger than that of Asia...
Gender	0	It's the same as the word "down-to-earth" that praises men.

[a.] Part of text data set table

B. Data Preprocessing

There are spaces between each word of English, while there is no obvious boundary between Chinese sentences. In order to extract the characteristic information of the text, the Chinese sentences were classified. In this study, Chinese text was segmented using the Jieba word library in the Python programming language. Jieba segmentation is based on a statistical dictionary. It constructs a prefix dictionary and utilizes this dictionary to segment input sentences into all possible segmentations. Subsequently, it constructs a directed acyclic graph based on the segmentation positions. Through dynamic programming algorithms, it calculates the maximum probability path to obtain the final segmentation form [7].

In addition to the text, each comment contains meaningless symbols, such as numbers, emojis, web tags, and other non-Chinese characters. These symbols hinder providing useful information for text classification and lower classification efficiency. Therefore, it is necessary to eliminate these useless characters during the text classification process. Moreover, it is needed to filter out frequently occurring words in the text that lack practical

meaning. These words are commonly referred to as stop words and include personal pronouns, prepositions, tone auxiliary words, and other terms. Such words do not contribute effectively to the analysis but consume considerable resources and time. The stop word list from the Harbin Institute of Technology was used in this study. This list encompasses a vast array of commonly used Chinese stop words, including terms from specific domains, making it applicable to most text analysis tasks.

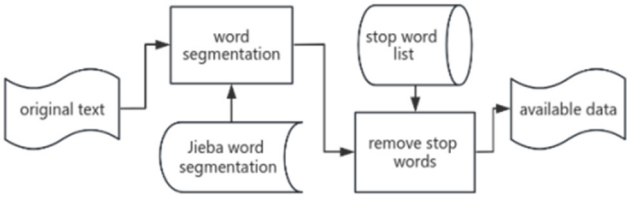

Fig. 1. Flow chart of text data preprocessing.

III. Feature Extraction and Selection

A. Term Frequency (TF)-IDF (Inverse Document Frequency) Feature Extraction

After preprocessing the text, the TF-IDF feature extraction method was employed to identify and model text keywords. TF represents the frequency of occurrence of a feature word in the text set relative to the total occurrence of all feature words in the text set and indicates whether the feature word adequately represents the main information of the text. IDF reflects the importance of feature words that appear in a small number of texts compared to those appearing in a large number of texts. IDF diminishes the significance of feature words that are prevalent across numerous texts. The feature extraction function for a feature word '*t*' was defined as follows.

$$\mathrm{W}(t) = TF(t) \times IDF(t) = (t_i / c) \times \log(N / n + 1) \tag{1}$$

where *ti* is the number of times the feature character *t* appears in the sentence; *c* is the total number of words in the sentence, and *N* is the total number of texts in the entire data set. (1) was used to calculate the weight of the TF-IDF of each feature word in the suspended word list of the text and to convert them into a vector representation and conduct vector operation and similarity calculation.

B. Random Forest (RF) Feature

The RF machine learning algorithm applies to classification and regression tasks and techniques used to assess the extent of features' contribution to model prediction [8]. Its principle is to quantify the contribution of each feature to the impurity reduction in the node partition, obtaining a feature importance score, usually calculated using the Gini index.

The variable importance score is denoted as VIM, and the Gini index, expressed as GI to present the presence of *J* features *X1*, *X2*, *X3*, ..., *Xj*, *I* decision trees, *C* categories, and *Pqc* representing the proportion of category *c* in node *q*. The Gini index of the i-th tree node, denoted as *q*, is calculated as follows.

$$\mathrm{GI}_q^{(i)} = \sum_{c=1}^{|C|} \sum_{c' \neq c} \mathrm{P}_{qc}^{(i)} \mathrm{P}_{qc'}^{(i)} = 1 - \sum_{c=1}^{|C|} (\mathrm{P}_{qc}^{(i)})^2 \tag{2}$$

The importance of feature *Xj*in the *i* tree node *q*, that is, the change of Gini index before and after the branch of point *q*, is represented as

$$\mathrm{VIM}_{jq}^{(Gini)(i)} = \mathrm{GI}_q^{(i)} - \mathrm{GI}_l^{(i)} - \mathrm{GI}_r^{(i)} \tag{3}$$

where *GIl (i)* and *GIr (i)* respectively represent the Gini indices of the two new nodes after branching. If the feature *Xj* appears as a joint Q in the decision tree *i*, then the importance of *Xj* in the i-th tree is expressed as follows.

$$\mathrm{VIM}_j^{(Gini)(i)} = \sum_{q \in Q} \mathrm{VIM}_{jq}^{(Gini)(i)} \tag{4}$$

Suppose the RF has an *I* tree, and then,

$$\mathrm{VIM}_j^{(Gini)} = \sum_{i=1}^{\mathrm{I}} \mathrm{VIM}_j^{(Gini)(i)} \tag{5}$$

All importance scores were normalized to obtain feature importance assessment scores (6).

$$\mathrm{VIM}_j^{(Gini)} = \frac{\mathrm{VIM}_j^{(Gini)}}{\sum_{j'=1}^{J} \mathrm{VIM}_{j'}^{(Gini)}} \tag{6}$$

IV. Classification Method of Illegal Text

A. Hyperparameter Optimization

Grid search is a traditional method of machine learning hyperparameter optimization. The basic process is to go through all possible combinations of hyperparameters in a given prior hyperparameter search space, and then find the optimal hyperparameter setting according to the evaluation index [9]. We evaluated the parameter performance by cross-validation in this study. Cross-validation is used to validate the performance of the classifier. This method randomly divides the dataset into roughly equal k subsets and then performs k experiments using one of the subsets as the test set and the remaining k-1 subset as the training set, and the subset sampled each time is not repeated [10].

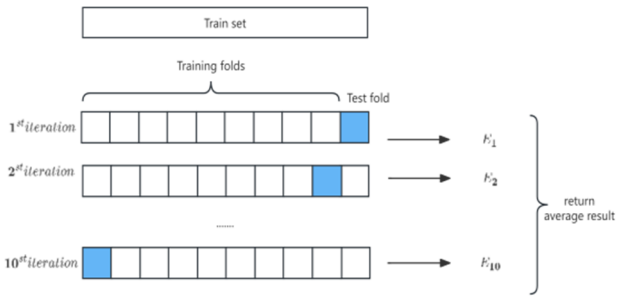

Fig. 2. K fold cross-validation.

B. Integrated Learning Method

In the integrated learning method, a set of classifiers is constructed, and the weighted votes of each classifier are used to predict new data. Compared to independent models, integrated learning methods have the advantages of eliminating bias, reducing prediction variance, and being less prone to overfitting. Currently, various integrated learning methods have been proposed and implemented, such as

simple averaging, weighted averaging, majority voting, weighted voting, and integrated stacking [11].

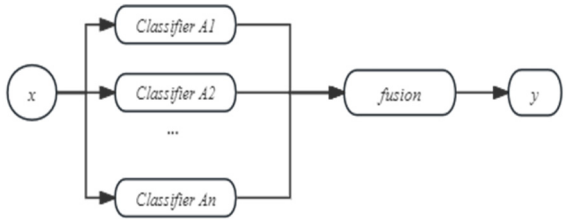

Fig. 3. Flow chart of integrated learning.

We employed the weighted voting method to integrate classification models for classification algorithms. In this method, weights are assigned to votes from different classifiers during the voting process and calculates the sum of weighted probabilities from all models to obtain the final prediction result. By utilizing the probability information of model predictions, the models' estimation of uncertainty is better expressed, and the contributions of different models in the integrated are balanced, thereby enhancing the detection performance.

V. Results and Discussions

The following methods and models were implemented through the Python third-party library sklearn in this study.

A. Experiment

First, the TF-IDF feature extraction was performed on the preprocessed dataset's 'train' table. The obtained features were split into a training set and a testing set with a ratio of 7:3 for learning. The number of features affects the model's accuracy. To find the optimal number of features, the feature importance scores are obtained using the RF feature importance method, ranking the importance of each feature word from highest to lowest. The top *n* feature words are then selected. Finally, the RF model is used to calculate the Area Under Curve (AUC) value after running, to find the optimal number of features:

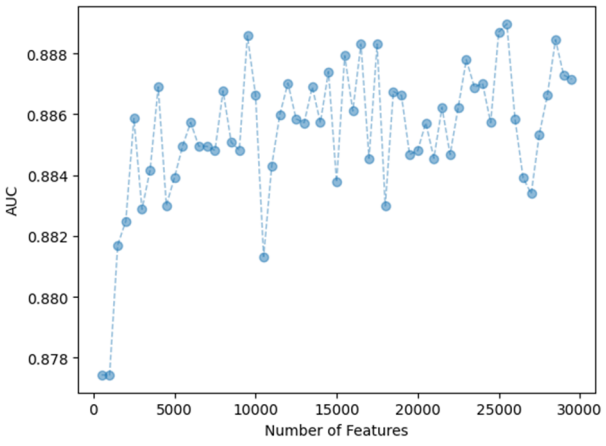

Fig. 4. AUC values for different number of features.

When the number of features was 25500 and 9500, AUC values were 0.8889 and 0.8886, while the feature number 25500 was more complex than 9500 and was more time-consuming to process. Therefore, we used the feature word number *n* of 9500 to be the best feature selection.

B. Evaluation Index

In machine learning and data mining, evaluation metrics are used to evaluate the performance of classification models. The commonly used evaluation index is the accuracy (A) and is calculated as

$$A = \frac{TP + TN}{TP + TN + FP + PN} \tag{7}$$

where True Positives (TP) represent the number of true positive instances, True Negatives (TN) represent the number of true negative instances, False Positives (FP) represent the number of false positive instances, and False Negatives (FN) represent the number of false negative instances. However, accuracy can only reflect the overall performance of the model, and in situations of imbalanced datasets, accuracy may be misleading. Therefore, we introduced three metrics, precision (P), recall (R), and F1-score (F1) to analyze the model's performance ((8)-(10)).

$$P = \frac{TP}{TP + FP} \tag{8}$$

$$R = \frac{TP}{TP + FN} \tag{9}$$

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{10}$$

C. Comparative Analysis

We trained eight classification models, namely 'KNN', 'Decision Tree', 'Naive Bayes', 'Logistic Regression', 'RF', 'AdaBoost', 'XGBoost' and 'SVM.' The GridSearchCV() method was introduced to provide an automated hyperparameter search approach. This method was used to systematically explore the parameter space to find the best model configuration. A parameter grid was set up, and each parameter combination was evaluated on the validation set using a 5-fold cross-validation technique to assess performance. Finally, the optimal parameters for each model regarding this problem were obtained as shown in Table II.

TABLE II. Best Parameters of Individual Classifiers

Model	Optimal parameters
KNN	{'n_neighbors': 7}
Decision Tree	{'max_depth': None}
Naive Bayes	{alpha: 1.0}
Logistic Regression	{'C': 10.0}
RF	{'n_estimators': 200}
AdaBoost	{'n_estimators': 200}
XGBoost	{'n_estimators': 200}
SVM	{'C': 1.0, 'kernel': 'rbf'}

[b.] Best parameters of the individual classifiers

For the evaluation metrics of the model, we used precision, recall, and F1-score serves as an evaluation index. The result showed that the XGBoost model had the highest accuracy of the problem, while the KNN model was the lowest. Five models, 'XGBoost', 'Logistic Regression', 'SVM', 'AdaBoost', and 'RF', showed better results for offending text

recognition. The results of each model evaluation index are shown in Table III.

TABLE III. PREDICTION RESULTS OF INDIVIDUAL CLASSIFIERS

Model	Precision(%)	Recall(%)	F1(%)
KNN	74.04	34.77	47.31
Decision Tree	84.79	85.02	84.91
Naive Bayes	80.93	90.93	85.64
Logistic Regression	90.02	88.29	89.15
RF	87.33	89.26	88.29
AdaBoost	88.66	83.94	86.23
XGBoost	90.99	88.24	89.59
SVM	89.98	89.50	89.74

[c.] The prediction results of the individual classifiers

Three of the five models with better classification performance were selected for integrated learning through permutation and combination, resulting in 10 ensemble learning models in the end, numbered from 1 to 10. Weighted voting was used for classification selection among the three groups of classifier combination models. The classification effects are illustrated in Fig. 5.

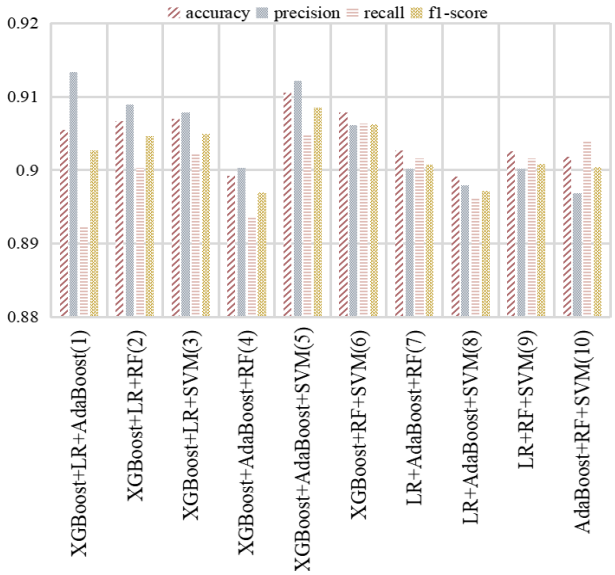

Fig. 5. Combined prediction performance plots of three groups of classifiers.

Model (1) showed the highest prediction accuracy of 91.33%, while its recall was the lowest among the 10 models, 89.24%. Model (8)'s classification effect was the worst, and its four evaluation indexes were less than 90%. The best overall evaluation index was obtained in Model (5), namely the 'XGBoost + ADaBoost + SVM' integrated learning model, with accuracy and precision of 91.05 and 91.22%. Considering that the more combined models showed better classification effects. Four models were combined, and the result is shown in Fig. 6.

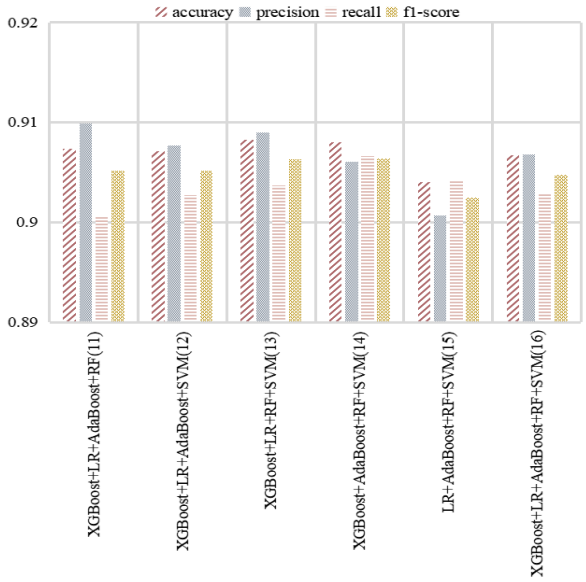

Fig. 6. Combined prediction performance plot of four and five classifiers.

The integrated models with more than three sets of classifiers outperformed three-set classifier integrated models. However, none of the models performed better than Model (5) in terms of classification effectiveness. Model (11) exhibited the highest precision at 90.99%, while its recall was the lowest among the six models at 90.06%.

In summary, the accuracy of the 16 ensemble learning models in identifying illegal comments was higher than 90%. This indicated that the recognition ability of integrated learning models did not necessarily increase with the number of model combinations. The integrated learning model that performed the best in identifying inappropriate comments was the "XGBoost+AdaBoost+SVM" model. Compared to the XGBoost model, which was the best-performing single model in classification, all evaluation metrics improved with a noticeable increase in recall and F1-score by 2.24 and 1.26%.

VI. CONCLUSIONS

For the Chinese review, we proposed an integrated learning model based on feature selection that extracted text features by TF-IDF, selected features with RF feature importance method to save resource cost, and found the best parameters for a single classifier by the grid search and 5-fold cross-validation. Integrated learning was conducted to improve performance. The final result indicated that the " XGBoost + ADaBoost + SVM " integrated learning model performed the best in the identification effect, while each evaluation index was higher than a single classifier. Due to the rapid iteration of Internet comments and the frequent change of network terms, if the training set of the model is not updated, it decreases accuracy. In future studies, it is needed to introduce neural network algorithms combined with this model for violation comment classification to further improve the identification accuracy.

ACKNOWLEDGMENT

The research was supported by the Ministry of Education industry-university cooperative education project (220602695021021) and the Student Innovation and Entrepreneurship Program of Wuhan Business University (202211654009).

REFERENCES

[1] Zhao Yanyan, Qin Bing, Liu Ting. Text sentiment analysis [J]. Journal of Software, 2010,21 (08): 1834-1848.

[2] Huang Ren, Zhang Wei. The emotional tendency study of Internet commodity reviews based on word2vec [J]. Computer Science, 2016,43 (S1): 387-389.

[3] Zhang Xinyue, Song Shaocheng. Study on text classification based on SVM algorithm in emergencies [J]. Information Technology and Informatization, 2022 (08): 13-16.

[4] Cao Tao, Bai Shucheng Chen. Emotional recognition model of public health events based on BERT-CW [J]. Science and Technology Innovation, 2023 (10): 72-76.

[5] Lv Pin, Yu Wenbing, Wang Xin, etc. Heterogeneous classifier stacking generalization and its application in the detection of malicious comments [J]. Journal of Electronics, 2019,47 (10): 2228-2234.

[6] Data sources: https: / / github.com/thu-coai/COLDataset

[7] Xu Borong. Wordcloud design and optimization using Jiba and Wordcloud libraries [J]. Fujian Computer, 2019,35(06):25-28.DOI:10.16707/j.cnki.fjpc.2019.06.006.

[8] Qiu Yang, Li Sheng, Jin Liang, etc. Bridge anomaly monitoring data identification method based on the mixture of statistical features and RF importance ranking [J]. Journal of Sensing Technology, 2022,35 (06): 756-762.

[9] Li Haixia, Song Danlei, Kong Jianing, etc. Evaluation of hyperparameter optimization techniques for traditional machine learning models [J / OL]. Computer Science: 1-24 [2024-02-26].http://kns.cnki.net/kcms/detail/50.1075.TP.20231201.0853.002.html.

[10] Zhou Yongxin, Yu Guo. Vector mosquito classification under deep learning [J]. Computer system application, 2023,32(05):234-243.DOI:10.15888/j.cnki.csa.009072.

[11] Wang Jun, Si Changfu, Wang Kaipeng, etc. Intrusion detection method [J / OL] based on feature selection of the PSO-GA algorithm with integrated learning and improvement. Journal of Jilin University (Engineering edition): 1-9 [2024-02-27].https://doi.org/10.13229/j.cnki.jdxbgxb.20230751.