Hindawi
Scientific Programming
Volume 2022, Article ID 8971882, 11 pages
https://doi.org/10.1155/2022/8971882

Research Article

Assessing the Influence Level of Food Safety Public Opinion with Unbalanced Samples Using Ensemble Machine Learning

Bo Song,[1] **Kefan Shang**,[1] **Junliang He**,[2] **and Wei Yan**[1]

[1]*China Institute of FTZ Supply Chain, Shanghai Maritime University, Shanghai 201306, China*
[2]*Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 201306, China*

Correspondence should be addressed to Junliang He; jlhe@shmtu.edu.cn

Received 23 April 2021; Accepted 22 December 2021; Published 14 February 2022

Academic Editor: Xiaobo Qu

Assessing the public opinion on food safety events constitutes an important job of government regulators. To optimize the government's management of food safety affairs, a promising way is to use artificial intelligence to improve the efficiency of food safety public opinion assessment. In this paper, we model the assessment of public opinion influence as a text classification task. The whole model adopts the ensemble learning framework, and it integrates naive Bayes, support vector machine, extreme gradient boosting, convolutional neural network, long- and short-term memory network, FastText, and BERT classification methods into the framework to form an ensemble learner. The ensemble learner is able to classify textual public opinion into high, medium, and low influence levels by learning from the samples assessed by human experts. To overcome the problem of unbalanced samples, we propose a sample generation method consisting of synonym replacement and semantic filtering to increase the number of high-influence samples. Real public opinion data collected from the Food Safety Department of the Chinese government are used for experiment. Extensive comparison of the proposed method with baseline methods proves the effectiveness of the ensemble learner and the sample generation steps.

1. Introduction

Nowadays, people are used to expressing opinions on the Internet, which leads to an explosive growth in the amount of online public opinions. Because food safety is closely related to everyone's daily life, public opinions with this topic are very likely to develop into hot events in the society. For example, La Tourangelle is a walnut oil brand welcomed by the most discerning customers in China. In 2019, the news of this brand of oil containing plasticizer exceeding the standard triggered vast public opinion on the Web, as this oil was mainly used for feeding babies. It has been shown that the interaction of government agencies with public opinions through social media can help the government to respond to public events efficiently [1]. The government can use public opinion assessment to explore people's attitudes towards an event [2–4] and predict events that may lead to serious consequences [5]. Therefore, it is meaningful to assess the influence of food safety public opinion in the early stage of its formation.

The importance of food safety public opinion has been pointed out in various regulatory documents issued by the government [6, 7]. However, unlike many other management optimization fields which have been intensively studied [8, 9], currently there is not much research dedicated to food safety public opinion assessment. Instead of analyzing research on food safety public opinion assessment, we survey the literature of general public opinion assessment. Moreover, since we formalize the public opinion assessment problem as a text classification problem, the literature of text classification is also analyzed to show the character of our research.

2. Related Works

2.1. Public Opinion Assessment. For assessing the influence of general public opinion, researchers often construct an index system to carry out public opinion evaluation. For example, considering the influence of microblog messages

and the dynamic role of the target audience of online public opinion, a microblog public opinion indicator system is established based on the Information Source Index (ISI), Geographic Index (GI), Subject Index (SI), and Industry Index (II) [10]. Simple analysis methods such as principal component analysis and analytic hierarchy process [11] are also often used in the construction of public opinion index systems. However, the manually selected indexes in this kind of studies cannot fully measure the characteristics of public opinion influence. At the same time, the index selection has a strong dependence on the opinions of experts and thus has a strong subjectivity. Some scholars believe that user behaviors such as forwarding and commenting can be used as the basis for evaluating the future development of public opinion. Li and Li use cloud models and analytic hierarchy processes to analyze user behaviors in public opinion dissemination, and they use this method to accurately predict hot public opinions [12]. Considering the impact indicators that affect the amount of user forwarding, Zheng et al. build a prediction model of network public opinion forwarding behavior using BP neural networks [13]. Due to the randomness and ambiguity of user forwarding behavior, Liu et al. used cloud theory to optimize the activation function of RBF neural networks [14]. When the information publisher has some professional authority or high popularity, users may ignore the actual content of the information when forwarding it [15]. Therefore, only relying on the statistics of user behavior to assess the influence of public opinion will produce a certain deviation. Seeing the problems with using user behaviors and artificial indexes, scholars begin to resort to the textual content of public opinion to assess its influence.

2.2. Text Classification for Public Opinion Assessment. Text classification plays an important role in public opinion assessment. Some scholars use text classification to identify the sentiment of public opinion, as sentiment affects the behavior of people interacting with public opinion [16, 17]. Other scholars use text classification to directly classify public opinion into different influence categories [18]. Text classification is an intensively studied field in recent years. When text underwent feature extraction and turned into numerical features, various machine learning methods can be used to classify text. Al-Tabbakh et al. use support vector machine, k-nearest neighbors, naive Bayes, and decision trees to classify the same text collection, whose results show that k-nearest neighbors perform the best in the experiment [19]. In contrast, deep learning algorithms do not require text feature extraction [20]. Deep learning models complete text classification by autonomously acquiring the relationship between text and label [21]. Another way to enhance text classification performance is to use multiple classifiers, which is called ensemble learning. Ensemble learning can effectively improve the accuracy and generalization ability of machine learning by accommodating more model assumptions. Cotelo et al. use a stacking framework of ensemble learning to integrate the content feature and structure feature of text to do classification [22]. Song et al. propose an ensemble learner to assess the impact of food safety news, which improves the accuracy of impact prediction [18]. However, their work does not take into consideration the unbalanced sample distribution across different impact levels, so the result is not satisfactory for high-impact news.

2.3. Dealing with Unbalanced Samples in Text Classification. Using machine learning for public opinion classification must pay attention to the distribution of data samples, as high-influence samples take only a very small part of the whole. Studies have shown that when the imbalance of the data set reaches 4 : 1 or higher, the predictive ability of the model will be lost [23]. Methods of processing unbalanced samples can be divided into oversampling and undersampling. Oversampling tries to enrich the minority type of samples by generating more samples of this type, and undersampling uses a subset of the majority type to make the number of each type equal. The classic oversampling method SMOTE maps the original samples to a certain vector space and then uses the samples in the space that are close to each other to construct new samples [24]. SMOTE-IPF [25] optimizes the classic method considering noisy and borderline examples. ADASYN [26] considers the distribution of minority data and generates new samples corresponding to the actual distribution of minority samples. In addition to oversampling and undersampling, classification algorithms themselves can be adapted to fit unbalanced samples. Datta and Das propose an approximate Bayesian support vector machine based on boundary transition and asymmetric cost to minimize the classification error [27]. Ando proposes a nearest neighbor model based on class weighting to compensate for the sparsity of minority classes by adjusting the k radius [28]. Cheng et al. introduce cost-sensitive marginal mean, variance, and penalty to adjust the proportional distribution between different categories, so as to obtain a balanced detection rate [29]. Despite the usefulness of adapting algorithms to unbalanced data, the adapted algorithms are often only applicable to a specific model. Under the framework of ensemble learning, adapting different base models one by one to unbalanced samples will increase complexity and cost of the framework.

3. Ensemble Learning Framework

In this paper, we propose a food safety public opinion assessment model that considers unbalanced sample distribution using the ensemble learning framework. The structure of the model is shown in Figure 1. In order to make full use of the data samples labeled by domain experts, we retain all the available data and adopt a "replacement-filtering" oversampling method to replenish the minority samples. The first step of oversampling is to build a synonym dictionary based on the vectorized word representation acquired through word embedding operation. Then, we replace some words in a sample with similar words to generate more samples of the minority type. At the last step of oversampling, we train a Siamese LSTM network to filter

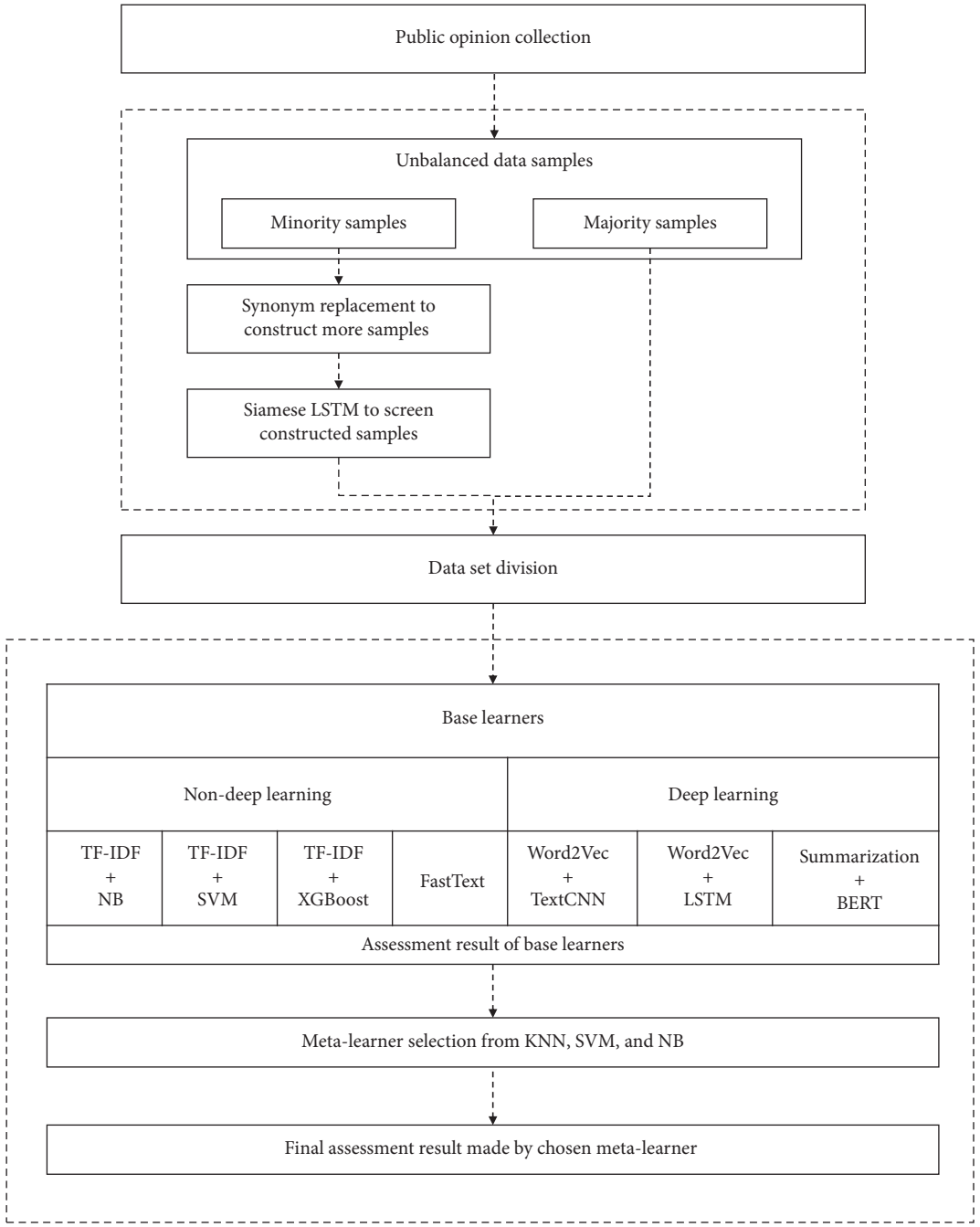

Figure 1: Ensemble model of food safety public opinion assessment with unbalanced samples.

out the newly constructed samples that are too dissimilar to real samples. To improve the accuracy and robustness of influence level classification, we use the stacking ensemble learning framework. The framework integrates naive Bayes (NB), support vector machine (SVM), extreme gradient boosting (XGBoost), convolutional neural network (CNN), long- and short-term memory network (LSTM), FastText, and BERT as base learners. Each base learner has its corresponding text preprocessing step: for NB, SVM, and XGBoost, each public opinion sample is turned into a vector of TF-IDF weights, together with the influence level label of this sample; for CNN and LSTM, each public opinion sample is turned into a matrix whose columns correspond to the embedding of words; for FastText, it takes the original text as input; for BERT, as it limits the length of input text for efficiency consideration, we apply automatic summarization to shorten oversized public opinion samples.

The stacking ensemble learning framework includes a meta-learner to synthesize the influence level rated by each of the base learners. We test k-nearest neighbors (KNN), SVM, and NB as three candidate meta-learners and select the best to use. To test the method proposed in this paper, we obtain food safety public opinion samples from the Risk Control Department of China Customs. Each sample has a

piece of text showing the original content of the public opinion, and an influence label ranging from high, medium, and low influence levels.

4. Processing Unbalanced Samples

In this study, we propose a "replacement-filtering" oversampling method to deal with the unbalanced data. The flowchart of the proposed oversampling method is shown in Figure 2. Details of the method are introduced in the following sections.

4.1. Minority Sample Generation Based on Synonym Replacement. In order to retain the information in the original data to the greatest extent, this paper uses the method of increasing minority samples to balance the sample set. To ensure that a newly added sample can achieve the purpose of equalizing the sample set, the new sample and the corresponding original sample should have similar characteristics. From the perspective of textual samples, the new sample and the corresponding original sample should be highly similar in terms of content and semantic meaning. Using synonym replacement to modify the original sample serves the goal of keeping text content similar. Synonym replacement is to replace each word in the original sample with a synonym of the word in the synonym dictionary and then obtain a new sample corresponding to the original sample. Although there is an existing platform that can do Chinese synonym replacement [30], the synonyms in this platform only include common terms and lack professional vocabulary such as law, medicine, and food safety. In this paper, we propose a synonym replacement method based on computed word vectors. The implementation steps are as follows:

4.1.1. Word Vector Computation. We use the Python package jieba to segment the original Chinese text. Then, we input the segmented text to the Word2Vec model realized in the Python package gensim to get word vectors. This step is also called word embedding. Word2Vec is a widely used word embedding model capable of capturing word meaning through self-supervised learning.

4.1.2. Top N Synonym Dictionary Construction. The cosine similarity of two words is calculated according to (1), where $\mathbf{x} = (x_1, \ldots x_i, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots y_i, \ldots, y_n)$ represent two word vectors.

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{n} (x_i \times y_i)}{\sqrt{\sum_{i=1}^{n} (x_i)^2} \times \sqrt{\sum_{i=1}^{n} (y_i)^2}}. \quad (1)$$

The closer the cosine similarity value is to 1, the higher the similarity between the two words. We calculate the cosine similarity of each word pair and choose the top N word most similar to a specific word to construct the synonym dictionary.

4.1.3. Generating Samples for a Minority Sample. We traverse each word in the minority sample and replace the word with its i^{th} similar word in the synonym dictionary, where i is a random number ranging from 1 to N. A new minority sample is generated after each word in the original minority sample has been replaced.

4.2. Sample Filtering Based on Siamese LSTM. Using the above method, we can generate any number of new samples for a given sample. However, this process only pays attention to the similarity of words but not the similarity of semantic meaning between the new samples and the original sample. To improve the quality of generated samples, we use Siamese neural networks to filter the new samples to ensure the semantic similarity between the new samples and the original sample.

The Siamese neural network [31] is composed of two identical neural networks with shared weights. It can be used to assess the similarity of two samples. Due to the excellent performance of the LSTM model in text understanding, this paper uses the Siamese LSTM model to complete sample filtering. The process is as follows:

4.2.1. Construction of a Siamese LSTM Model. Construct two LSTM models with identical structure and shared weights. This is done by training a LSTM text classifier using the original public opinion samples and duplicating the trained classifier.

4.2.2. Sample Filtering. For each newly generated sample, we input it with its corresponding original sample to the Siamese LSTM model. We can obtain the vector representation of the two samples at the LSTM layer prior to the softmax layer. Then, we compute the cosine similarity between the two vectors and see if the similarity value exceeds a predefined threshold. If so, we retain the generated sample, otherwise the generated sample is discarded.

5. Construction of an Ensemble Learner

To construct an ensemble learner includes three basic steps. The first is to select a group of base learners that are differently structured or differently trained. The second step is to divide the data set to properly train the base learners. The last step is to select a meta-learner to synthesize the results of base learners to get the final prediction result.

5.1. Base Learner Selection. To ensure the superiority and robustness of the final result, the selection of base learners follows the principle of accurate result and model diversity. The chosen base learners include those listed below.

5.1.1. Naive Bayes. Naive Bayes (NB) is a classic machine learning model based on the Bayes theorem and assumption of independent sample features. If the sample features meet the requirement of such assumption, a NB learner will have superior performance. In food safety public opinion

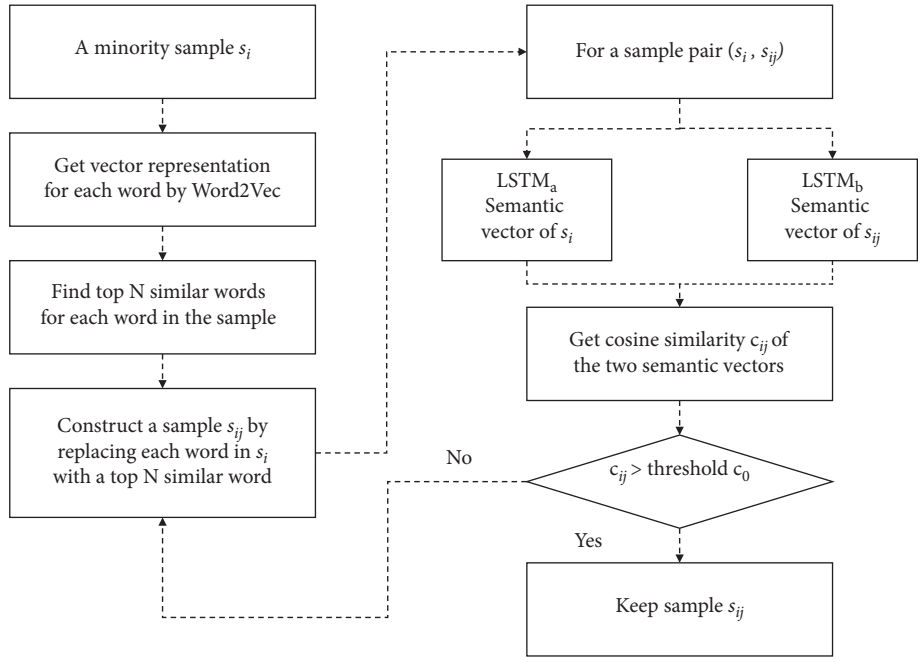

Figure 2: Flowchart of unbalanced sample processing.

assessment, a NB learner uses TF-IDF weighted words in the public opinion as features, and we use the sklearn package to carry out the training and predicting with the NB learner.

5.1.2. SVM. SVM learns to classify by solving a optimization problem. It maximizes the distance between a cutting hyperplane and the support vectors in the sample space. Due to its good performance, SVM has been used as a benchmark in many classification tasks. When there are more than two classes, a one-versus-rest method is usually adopted: by treating one class of the total n classes as a class, and the other $n-1$ classes as another class, totally n SVM classifiers will be constructed for an n-class classification problem.

5.1.3. XGBoost. XGBoost itself is an ensemble learner integrating multiple CART (classification and regression tree) models based on the boosting mechanism. The training process of XGBoost is to create a series of CARTs and let each tree learn to fit the prediction error of a previous tree. Leveraging the different assumptions in the constructed trees, XGBoost can improve the generalization ability of a learned model.

5.1.4. FastText. FastText is a simple three-layer neural network deliberately trained for accomplishing natural language processing tasks. FastText can achieve text classification precision comparable to that of deep neural networks but is many orders of magnitude faster in training time. At the input layer of FastText, n-grams in the text undergo a bucket hashing process and become embedding vectors. Since FastText generate word vectors by itself, we do not apply Word2Vec to the FastText classifier.

5.1.5. CNN. CNN is a deep neural network architecture originally proposed for image classification. Yoon Kim proposed a variant of CNN, namely TextCNN, for text classification [32]. In this paper, we use the word vectors generated by Word2Vec to replace the random word embedding used in TextCNN, so as to incorporate more prior knowledge in the classification model.

5.1.6. LSTM. LSTM adds an input gate, a forgetting gate, an output gate, and a memory unit to a RNN neuron, making the modified model capable of memorizing important information and forgetting unimportant information in a time series [33]. LSTM is very useful for modeling text as the word sequences in text represents time series signals. The training of LSTM requires vector representation of each word in the text, which in this paper is acquired using Word2Vec.

5.1.7. BERT. BERT [34] and its variations are among the state-of-the-art techniques for natural language processing. BERT is based on Transformer [35], an encoder-decoder architecture built on multihead self-attention mechanism. BERT is structured as a multilayer bidirectional Transformer encoder and is deliberately pretrained with two types of tasks: masked language model and next sentence prediction. Using BERT to classify is called fine-tuning, which is to learn only the weight matrix of the softmax layer. BERT consumes significantly more resources to compute as the length of input text grows. To fit the capacity of our computing resources, we set 128 Chinese characters as the max length of input text for BERT and use TextRank [36] to summarize the food safety public opinion, so as to keep as much information in the original text as possible.

5.2. Data Processing for the Ensemble Model. To train the 7 base learners and the meta-learner, the sample data set should be properly divided and fed to the model following the process depicted in Figure 3.

The overall training set is divided into 7 sets of equal size, shown as $a_1, \ldots, a_7$ in Figure 3. For each base learner, it is trained 7 times. In the first time, a_1 is used as the inner test set and the rest sets are used as the training set; in the second time, a_2 is used as the inner test set and the rest sets are used as the training set, and so on. The predicted class labels for inner test set a_i by base learner j is denoted as b_{ij}. By merging $b_{i1}, \ldots, b_{i7}$ along the same test samples, we get B_i, and $B_1, \ldots, B_7$ comprise the training set for the meta-learner. Since each base learner has 7 differently trained versions, and when testing the ensemble learner the meta-learner needs a determined class label from each base learner, we test the seven versions of a base learner one by one using the overall test set and choose the most frequently appearing class label for each test sample to get T_i, the test result of base learner i. Finally, by merging $T_1, \ldots, T_7$, we get T, the test set for the meta-learner.

5.3. Meta-Learner Selection. A meta-learner uses the output of all the base learners as input and makes final decision on the class label of a sample. Since the output of base learners for a given sample comprises a digital vector, and the results of different base learners do not affect each other, the meta-learner predicting with this vector needs not to be complicated. In this paper, we choose KNN, SVM, and NB as the candidate meta-learners. They have good performance in the ensemble learning framework and relatively short training time. We will test the performance of the three models and select the best model to use.

6. Experiment

6.1. Data Collection. The experiment data in this article come from the Food Safety Department of China Customs. Each public opinion has been manually rated by the customs officers. The original data collection includes 21,145 samples of food safety public opinion. After deleting invalid information, the total number of data sample is 21,065, including 10,247 low-influence samples, 10,314 medium-influence samples, and 504 high-influence samples.

6.2. Model Settings. The settings of each compositional machine learning model are shown in Table 1. Among the models, TextCNN, LSTM, and BERT need to set the reading length of the text. As samples with the length of less than 1000 characters account for 98% of the total samples, the reading length of TextCNN and LSTM is set to 1000 characters, and the excessive text is cut off. Due to the reason explained in Section 5.1, the reading length of the BERT model is set to 128 characters, and we apply automatic summarization to compensate for the information loss.

6.3. Evaluation Index. To conduct a comprehensive evaluation of the model results, four evaluation indexes are used: accuracy, precision, recall, and F1-score. The calculation of these indicators is listed below, where TP, TN, FP, and FN stand for the number of true positive, true negative, false positive, and false negative predictions of sample influence level.

$$\begin{aligned} \text{accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \\ \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ \text{F1} &= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \end{aligned} \tag{2}$$

Accuracy reflects the overall performance of a model. Precision and recall reflect the ability of the model to correctly predict class labels regarding all classified samples and samples of a certain type respectively. The F1 value is the harmonic average of precision and recall.

6.4. Minority Sample Generation and Filtering. We use Word2Vec to train a word vector table of $88{,}296 \times 50$ from the original 21,145 public opinion samples. After removing useless words such as punctuations and numbers, we get a table of $62{,}981 \times 50$. Each row in the table corresponds to a Chinese word, and its similarity with another word is calculated through cosine similarity. For a real minority sample, we traverse each word in it and replace the word with its i^{th} similar word in the word vector table. By ranging i from 1 to 20, we obtain 10,080 new samples.

Word replacement only ensures the word-level similarity between a generated sample and the original sample, but in fact we need the semantic meaning between the two samples to be similar. To achieve this goal, we train Siamese LSTM networks to filter out the generated samples whose semantic similarity with the original sample is low. Adopting a semantic similarity threshold of 0.8, we retain 5544 high-influence samples constructed from word replacement. The final set of high-influence samples has a size of 6048.

6.5. Meta-Learner Selection Based on Performance. Three types of meta-learner have been tested using the balanced samples. Their performances are shown in Table 2. From Table 2 we can see that NB has the best accuracy, which is 0.8530. So we choose NB as the meta-learner in our ensemble learning model.

7. Result and Analysis

To show the effectiveness of the proposed ensemble learning framework and sample balancing method, we present results of influence assessment in three scenarios. The first is the result of base learners and ensemble model with original and

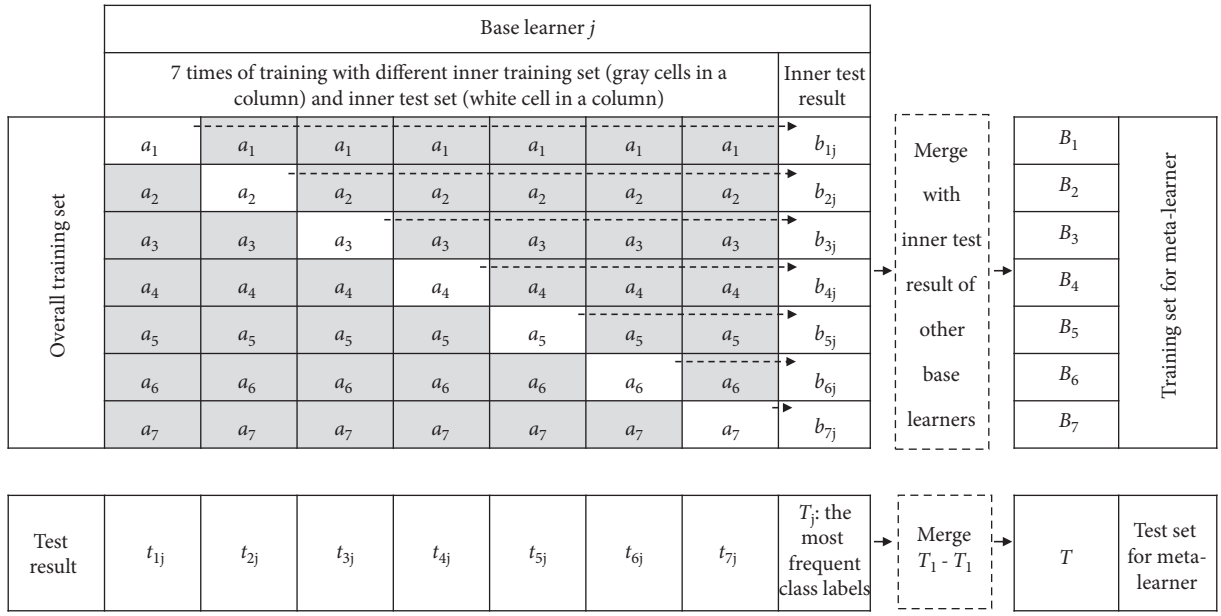

Figure 3: Construction of training and test data sets.

Table 1: Model settings.

Model	Settings
NB	Uniform prior probability of classes; other parameters follow the default setting of sklearn MultinomialNB model.
SVM	Parameters follow the default setting of sklearn LinearSVC model.
XGBoost	Early stopping rounds = 10; eval_metric = "logloss"; other parameters follow the default setting of the Python package XGBoost.
FastText	Minimal number of word occurrences = 2; other parameters follow the default setting of the Python package FastText.
TextCNN	Keras-based implementation of a TextCNN [11]-like CNN, with a dropout layer after the embedding layer (dropout rate = 0.2); the 1D convolutional layer has 250 filters (kernel length = 3); a 3-max pooling layer follows and is followed by a flatten layer, a 50-unit dense layer, and a 3-unit softmax layer; the activation function of the convolutional layer and the dense layer is ReLU; input length = 1000, batch size = 256, epochs = 5.
LSTM	Keras-based implementation of LSTM; the embedding layer is connected to a LSTM layer with 200 neurons, where a 0.2 dropout rate of the input and recurrent state is applied; following the LSTM layer is a dropout layer (dropout rate = 0.2), a 64-unit dense layer (ReLU activation function) and a 3-unit softmax layer; input length = 1000, batch size = 128, epochs = 5, Adam optimizer, learning rate = 0.01.
BERT	Chinese pretrained model, $L = 12$, $H = 768$, $A = 12$; batch size = 32, epochs = 5, learning rate = $2e - 5$; input length = 128.
KNN	Parameters follow the default setting of the sklearn neighbors model.

Table 2: Comparison of meta-learners.

Meta-learner	Evaluation index			
	Accuracy	Precision	Recall	F1
KNN	0.8357	0.8364	0.8357	0.8360
SVM	0.8500	0.8506	0.8500	0.8502
NB	0.8530	0.8541	0.8530	0.8534

balanced samples as input (Table 3). The second is the result of some base learners under 3 ways of sample balancing: none, SMOTE, and replacement-filtering (Table 4). The third is the performance of influence assessment for each sample class (Figure 4).

It can be seen from Table 3 that before sample balancing, only FastText and LSTM achieve accuracy more than 0.8. After using replacement-filtering measure to process unbalanced samples, all base learners achieve accuracy more than 0.8, and the ensemble model proposed by this paper has the highest score of 0.8530. After processing unbalanced samples, the best performance of a single learner is 0.8494 of accuracy achieved by LSTM. BERT represents a more advanced model than LSTM in processing text, but the performance of BERT predicting public opinion influence in this paper is only better than the NB model. This is due to the limitation of input text length caused by hardware constraints. From the above result we can see that both the ensemble learning framework and the replacement-filtering oversampling measure improve the performance of public opinion assessment. While the ensemble learning framework achieves a 0.42% improvement regarding the best single learner, the oversampling measure achieves an improvement of 5.8%. Moreover, in real application of artificial intelligence, people usually consider an accuracy above 0.85 as the baseline, so in this sense, only the result of ensemble learning model with balanced samples meets the requirement of real application.

To verify the advantage of the proposed oversampling method versus traditional oversampling methods, we

Table 3: Influence assessment results.

Model		Evaluation index			
		Accuracy	Precision	Recall	F1
NB	Original sample	0.7587	0.7615	0.7587	0.7592
	Balanced sample	0.8126	0.8127	0.8126	0.8124
SVM	Original sample	0.7968	0.7990	0.7968	0.7972
	Balanced sample	0.8435	0.8439	0.8435	0.8434
XGBoost	Original sample	0.7902	0.7918	0.7902	0.7905
	Balanced sample	0.8491	0.8490	0.8491	0.8490
FastText	Original sample	0.8027	0.8037	0.8027	0.8030
	Balanced sample	0.8470	0.8468	0.8469	0.8469
TextCNN	Original sample	0.7870	0.7907	0.7870	0.7879
	Balanced sample	0.8392	0.8417	0.8392	0.8393
LSTM	Original sample	0.8018	0.8064	0.8018	0.8024
	Balanced sample	0.8494	0.8513	0.8494	0.8494
BERT + summarization	Original sample	0.7919	0.7934	0.7919	0.7921
	Balanced sample	0.8300	0.8298	0.8300	0.8299
Ensemble model (ours)	Original sample	0.8062	0.8071	0.8062	0.8052
	Balanced sample	0.8530	0.8541	0.8530	0.8534

Table 4: Comparison of oversampling methods.

Model		Evaluation index			
		Accuracy	Precision	Recall	F1
NB	Original sample	0.7312	0.7652	0.7312	0.7321
	SMOTE	0.7714	0.7703	0.7714	0.7697
	Replace-filter	0.8052	0.8167	0.8052	0.8030
SVM	Original sample	0.7532	0.7563	0.7532	0.7535
	SMOTE	0.8235	0.8257	0.8235	0.8242
	Replace-filter	0.8064	0.8070	0.8064	0.8063
XGBoost	Original sample	0.7986	0.7992	0.7986	0.7984
	SMOTE	0.8384	0.8389	0.8384	0.8385
	Replace-filter	0.8485	0.8488	0.8485	0.8486
TextCNN	Original sample	0.7782	0.7882	0.7782	0.7806
	SMOTE	0.8153	0.8171	0.8153	0.8157
	Replace-filter	0.8375	0.8375	0.8375	0.8372
LSTM	Original sample	0.7845	0.7852	0.7845	0.7848
	SMOTE	0.8054	0.8080	0.8054	0.8054
	Replace-filter	0.8332	0.8339	0.8332	0.8331

conduct experiments with three types of samples, saying the original samples without oversampling, the balanced samples using SMOTE oversampling method, and the balanced samples using the proposed replacement-filtering oversampling method. Since the SMOTE method generates new samples by locating points between two original sample points in the hyperspace, only learners using vectorized representation of input text are suitable for testing with this oversampling method. From Table 4 we can see that both oversampling methods can improve the accuracy of influence level prediction. From the perspective of improved performance, the SMOTE method has a better improvement effect on non-deep learning models, and the replacement-filtering method proposed in this paper has good improvement effect on both non-deep learning models and deep learning models.

An important capability of a public opinion assessment model is to recognize high-influence samples, as these samples are more likely to trigger public events. Figure 4 shows the class level results of influence assessment. It can be seen that the abilities of different models to distinguish between low- and medium-influence-level samples are close. But for high-influence-level samples, the performances of different models vary greatly. For single models, SVM and LSTM are better than NB and CNN in recognizing high-influence public opinion with unbalanced samples. For the ensemble model, it has high precision and low recall when recognizing high-influence public opinion with unbalanced samples, but when the samples are balanced, it has the highest precision, recall, and overall performance when recognizing high-influence public opinion.

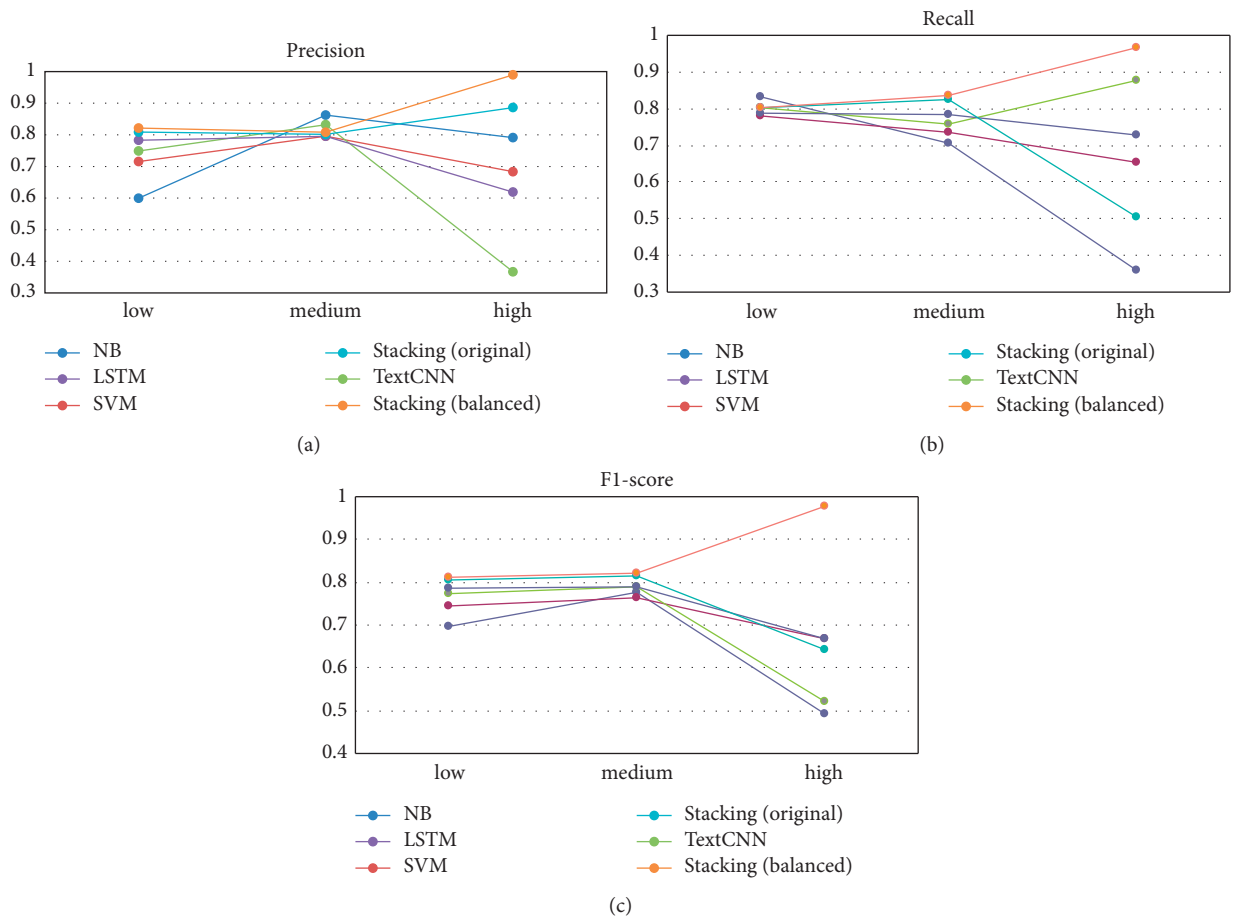

Figure 4: Comparison of class level results.

8. Conclusion

In this paper we study the problem of assessing the influence level of food safety public opinion. An ensemble machine learning model is proposed to classify food safety public opinions into three influence levels: high, medium, and low. Given that the number of high-influence public opinion samples is much smaller than that of low-influence samples, an oversampling method is proposed to balance the sample number and improve the assessment accuracy. The oversampling method includes using synonym replacement to generate pseudo-high-influence samples and using Siamese LSTM neural network to filter out low-quality pseudo-samples. Experiments with real data collected from the Food Safety Department of China Customs show that the ensemble machine learning model outperforms single machine learning model including NB, SVM, XGBoost, FastText, TextCNN, LSTM, and BERT in terms of assessment accuracy. The oversampling operation is also tested to be beneficial, as after sample balancing, the accuracy of recognizing high-influence samples reaches more than 0.9 and the F1-score raises from below 0.7 to above 0.9. The result of the study shows that the proposed method can be used in real life to optimize the trade-off between accuracy and efficiency of food safety public opinion assessment.

As pointed out by Zheng [37], it is important to do reproducible research by making the data and model definite. Regarding the oversampling model and ensemble learning model proposed in this paper, the result can be made reproducible if the random seeds used in these models were set definite. However, we have not studied how to tactically eliminate the randomness of the model to achieve beneficial effects such as avoiding the selection of outliers during sampling [38]. This kind of study will be done in the future research.

Data Availability

Data are available upon request.

Conflicts of Interest

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

Acknowledgments

This work was sponsored by the National Natural Science Foundation of China (71601113 and 72072112), Shanghai Science and Technology Committee Project (21010501800), and Shanghai Rising-Star Program (19QA1404200).

References

[1] D. Wu and Y. Cui, "Disaster early warning and damage assessment analysis using social media data and geo-location information," *Decision Support Systems*, vol. 111, pp. 48–59, 2018.

[2] E. D'Andrea, P. Ducange, and F. Marcelloni, "Monitoring negative opinion about vaccines from tweets analysis," in *Proceedings of the 2017 Third International Conference on Research in Computational Intelligence and Communication Networks*, Kolkata, India, November 2017.

[3] E. D'Andrea, P. Ducange, A. Bechini, A. Renda, and F. Marcelloni, "Monitoring the public opinion about the vaccination topic from tweets analysis," *Expert Systems with Applications*, vol. 116, pp. 209–226, 2019.

[4] B. O'Connor, "From tweets to polls: linking text sentiment to public opinion time series," in *Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010*, ICWSM, Washington, DC, USA, May 2010.

[5] J. Chen, H. Zhou, H. Hu et al., "Research on agricultural monitoring system based on convolutional neural network," *Future Generation Computer Systems*, vol. 88, pp. 271–278, 2018.

[6] European Food Safety Authority, "EFSA Image Qualitative Research Report," 2020, http://www.efsa.europa.eu/sites/default/files/event/2010/mb100318-ax4.pdf.

[7] US Food and Drug Administration, "DFA's Strategic Plan for Risk Communication," 2020, http://www.fda.gov/Food/default.htm.

[8] L. Zhen, Z. Liang, D. Zhuge, L. H. Lee, and E. P. Chew, "Daily berth planning in a tidal port with channel flow control," *Transportation Research Part B: Methodological*, vol. 106, pp. 193–217, 2017.

[9] J. He, Y. Wang, C. Tan, and H. Yu, "Modeling berth allocation and quay crane assignment considering QC driver cost and operating efficiency," *Advanced Engineering Informatics*, vol. 47, Article ID 101252, 2021.

[10] L. Zhao, X. Zhang, M. He, D. Zhang, W. Liu, and C. Liu, "Research on public opinion index system of Chinese microblog," in *Proceedings of the 2014 IEEE 5th International Conference on Software Engineering and Service Science*, pp. 385–388, IEEE, Beijing, China, June 2014.

[11] H. Xing, J. Huidong, and Z. Yu, "Risk assessment of earthquake network public opinion based on global search BP neural network," *PLoS One*, vol. 14, 2019.

[12] Y. Li and S. Li, "The propagation behavior prediction of Tibetan network public opinion based on cloud model," in *Proceedings of the: 2nd International Conference on Information Science and Control Engineering, ICISCE 2015*, pp. 992–995, Shanghai, China, April 2015.

[13] C. Zheng, Y. Song, and Y. Ma, "Public opinion prediction model of food safety events network based on bp neural network," *IOP Conference Series: Materials Science and Engineering*, vol. 719, no. 1, Article ID 012078, 2020.

[14] Y. Liu, J. Zhao, and Y. Xiao, "C-RBFNN a user retweet behavior prediction method for hotspot topics based on improved RBF neural network," *Neurocomputing*, vol. 275, pp. 733–746, 2018.

[15] L. Zhang, H. Li, C. Zhao, and X. Lei, "Social network information propagation model based on individual behavior. Wireless Communication over ZigBee for automotive inclination measurement," *China Communications*, vol. 14, no. 7, pp. 78–92, 2017.

[16] M. Zhang, R. Zheng, J. Chen et al., "Emotional Component analysis and forecast public opinion on micro-blog posts based on maximum entropy model," *Cluster Computing*, vol. 22, 2018.

[17] L. Servi and S. B. Elson, "A mathematical approach to gauging influence by identifying shifts in the emotions of social media users," *IEEE Transactions on Computational Social Systems*, vol. 1, no. 4, pp. 180–190, 2015.

[18] B. Song, K. Shang, J. He, W. Yan, and T. Zhang, "Impact assessment of food safety news using stacking ensemble learning," *Advances in Transdisciplinary Engineering*, vol. 12, pp. 353–362, 2020.

[19] S. M. Al-Tabbakh, H. M. Mohammed, and H. El-Zahed, "Text mining techniques for intelligent grievances handling system: WECARE project improvements in EgyptAir," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 2, pp. 603–614, 2019.

[20] Y. Bengio and Y. Lecun, "Scaling learning algorithms towards AI," in *Proceedings of the Large-Scale Kernel Machines*, vol. 34, no. 5, pp. 1–41, 2007.

[21] B. Zhong, X. Xing, P. Love, X. Wang, and H. Luo, "Convolutional neural network: deep learning-based classification of building quality problems," *Advanced Engineering Informatics*, vol. 40, pp. 46–57, 2019.

[22] J. M. Cotelo, F. L. Cruz, F. Enríquez, and J. A. Troyano, "Tweet categorization by combining content and structural knowledge," *Information Fusion*, vol. 31, pp. 54–64, 2016.

[23] L. Victoria, F. Alberto, G. Salvador, P. Vasile, and H. Francisco, "An insight into classification with unbalanced data: empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113–141, 2013.

[24] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[25] A. S. José, L. Julián, J. Stefanowski, and H. Francisco, "SMOTE–IPF: addressing the noisy and borderline examples problem in unbalanced classification by a re-sampling method with filtering," *Information Sciences*, vol. 291, 2015.

[26] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: adaptive synthetic sampling approach for unbalanced learning," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, IEEE, Hong Kong, June 2008.

[27] S. Datta and S. Das, "Near-Bayesian support vector machines for imbalanced data classification with equal or unequal misclassification costs," *Neural Networks*, vol. 70, pp. 39–52, 2015.

[28] S. Ando, "Classifying imbalanced data in distance-based feature space," *Knowledge and Information Systems*, vol. 46, no. 3, pp. 707–730, 2016.

[29] F. Cheng, J. Zhang, C. Wen, Z. Liu, and Z. Li, "Large cost-sensitive margin distribution machine for unbalanced data classification," *Neurocomputing*, vol. 224, pp. 45–57, 2016.

[30] W. Che, Z. Li, and T. Liu, "LTP: a Chinese language technology platform// COLING 2010," in *Proceedings of the 23rd International Conference on Computational Linguistics, Demonstrations*, pp. 23–27, Beijing, China, August 2010.

[31] S. Chopra, R. Hadsell, and Y. Lecun, "Learning a similarity metric discriminatively, with application to face verification," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, San Diego, CA, USA, June 2005.

[32] Y. Kim, "Convolutional neural networks for sentence classification," 2014, https://www.researchgate.net/publication/265052545_Convolutional_Neural_Networks_for_Sentence_Classification.

[33] J. Chung, C. Gulcehre, K. H. Cho, and B. Yoshua, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, https://www.semanticscholar.org/paper/Empirical-Evaluation-of-Gated-Recurrent-Neural-on-Chung.

[34] J. Devlin, M. W. Chang, K. Lee, and T. Kristina, "BERT: pre-training of deep bidirectional transformers for language understanding," 2018, https://www.semanticscholar.org/paper/BERT%3A-Pre-training-of-Deep-Bidirectional-for-Devlin-Chang.

[35] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention Is All You Need," 2017, https://papers.nips.cc/paper/7181-attention-is-all.

[36] R. Mihalcea and P. Tarau, "Textrank: bringing order into texts," in *Proceedings of the EMNLP*, pp. 404–411, Barcelona, Spain, July 2004.

[37] Z. Zheng, "Reasons, challenges, and some tools for doing reproducible transportation research," *Communications in Transportation Research*, vol. 1, Article ID 100004, 2021.

[38] J. Zhang, X. Qu, and S. Wang, "Reproducible generation of experimental data sample for calibrating traffic flow fundamental diagram," *Transportation Research Part A: Policy and Practice*, vol. 111, pp. 41–52, 2018.