Hindawi
Scientific Programming
Volume 2020, Article ID 8849011, 13 pages
https://doi.org/10.1155/2020/8849011

Research Article

Mushroom Toxicity Recognition Based on Multigrained Cascade Forest

Yingying Wang,[1,2,3] Jixiang Du,[1,2,3] Hongbo Zhang,[1,2,3] and Xiuhong Yang[1,2,3]

[1]*Fujian Key Laboratory of Big Data Intelligence and Security, Huaqiao University, Xiamen 361021, China*
[2]*Xiamen Key Laboratory of Computer Vision and Pattern Recognition, Huaqiao University, Xiamen 361021, China*
[3]*Department of Computer Science and Technology, Huaqiao University, Xiamen 361021, China*

Correspondence should be addressed to Jixiang Du; jxdu77@gmail.com

Received 24 April 2020; Revised 7 July 2020; Accepted 11 July 2020; Published 1 August 2020

Academic Editor: Chenxi Huang

Due to the tastiness of mushroom, this edible fungus often appears in people's daily meals. Nevertheless, there are still various mushroom species that have not been identified. Thus, the automatic identification of mushroom toxicity is of great value. A number of methods are commonly employed to recognize mushroom toxicity, such as folk experience, chemical testing, animal experiments, and fungal classification, all of which cannot produce quick, accurate results and have a complicated cycle. To solve these problems, in this paper, we proposed an automatic toxicity identification method based on visual features. The proposed method regards toxicity identification as a binary classification problem. First, intuitive and easily accessible appearance data, such as the cap shape and color of mushrooms, were taken as features. Second, the missing data in any of the features were handled in two ways. Finally, three pattern-recognition methods, including logistic regression, support vector machine, and multigrained cascade forest, were used to construct 3 different toxicity classifiers for mushrooms. Compared with the logistic regression and support vector machine classifiers, the multigrained cascade forest classifier had better performance with an accuracy of approximately 98%, enhancing the possibility of preventing food poisoning. These classifiers can recognize the toxicity of mushrooms—even that of some unknown species—according to their appearance features and important social and application value.

1. Introduction

Mushrooms are the fleshy fruiting bodies of certain fungus, some of which are edible, but a minority of them are toxic [1]. Every year, a large number of people die [2, 3] from eating poisonous mushrooms. It is useful to identify whether a mushroom is poisonous according to the appearance features of the mushroom. The automatic recognition of mushroom toxicity has important social and application value in effectively preventing food poisoning [4].

Current methods of recognizing poisonous mushrooms can be roughly divided into four categories: chemical determination, animal experimentation [5], fungal classification, and folk experience [6]. At present, the research of poisonous mushrooms based on these methods not only has been imperfect but also has left much to be desired [7].

The classification of poisonous mushrooms has evolved from the biological level to the molecular level [2]. Therefore, the application of chemical determination methods to detect poisonous mushrooms is becoming increasingly popular [8]. However, there are strict requirements for the experimental conditions, which are often limited to the laboratory. Due to cumbersome handling and the great number of unstable toxins, the method of toxic chemical detection cannot be used to distinguish edible mushrooms from poisonous ones [9]. This approach requires professional knowledge and is, therefore, not suitable for the average person.

Generally, mushrooms with intact cells, bright colors, and the lack of birds and insects interacting with them are likely to be poisonous, particularly if they are found in places where animals are foraging. To investigate the above situation empirically, the animal acute toxicity test is commonly

used to classify poisonous mushrooms [10]. Although the methods involved are simple, they carry some limitations, such as low efficiency, material and dosage concerns, and the varying sensitivities of different animals. Therefore, special institutions or facilities are needed to facilitate the application of these methods.

Fungal recognition includes three aspects: identification, classification, and phylogeny [11]. The development of fungal taxonomy has gone through two stages: traditional taxonomy and molecular biology. These methods have mainly been used to identify the mushrooms' species. The aim of these methods is subjective, however, because fungi contain many species and complex morphological features. These methods are limited to applications involving the artificial cultivation of fungi and are only suitable for professionals. Therefore, the identification of poisonous mushrooms is not straightforward.

For a long time, humans have recognized poisonous mushrooms by observing the shape, color, odor, and secretion features empirically [12]. This method is more intuitive, but is of low accuracy proven by the annual poisoning events. Thus, it is not a reliable method for identifying whether mushrooms are poisonous. However, this method relies on background knowledge acquired by humans. People get a lot of background knowledge and experiences so that the recognition accuracy rate is high. Otherwise, the accuracy rate is low. In this paper, automatic identification can break through the limitation to determine whether it is toxic. The machine learning methods not only do not require background knowledge but also can identify unknown species.

These mushroom toxicity recognition methods have some limitations, such as low accuracy, unqualified detection of unknown toxins, strict requirements for the experimental environments, sufficient professional knowledge, and complex experimental cycles. To solve these problems, an automatic model for mushroom recognition based on appearance features is constructed in this paper. According to the observed mushroom appearance data, a poisonous mushroom can be automatically and accurately identified by the proposed model.

With the advent of the data age, machine learning and deep learning have become the core of artificial intelligence [13]. In recent years, machine learning techniques have been used to identify the toxicity of mushrooms. Chaoqun [14] used machine learning models to identify poisonous mushrooms in an application. The android-based toadstool identification system can effectively classify toadstool in real time [15]. Zhifeng [16] proposed decision fusion based on the stacking algorithm to improve the accuracy of classification methods. The image database of mushroom, obtained from the Internet by Python Crawler, was constructed by Shuaichang et al. [17]. The model-based transfer learning and the Adam algorithm as the model optimization method were applied to construct the model structure of mushrooms' image recognition.

Deep neural networks require large-scale data volumes, making already complex models even more complicated. Machine learning has unique advantages for small-sample problems. For the identification of poisonous mushrooms, three different pattern recognition methods are discussed in this paper: logistic regression, support vector machine (SVM) [18], and multigrained cascade forest (gcForest) [19]. Mushroom toxicity recognition is regarded as a problem of binary classification. By observing the appearance features of mushrooms, these machine learning methods are used to determine whether a mushroom is toxic. gcForest has the following advantages: (1) feature-based learning and iterative classification through gcForest, which has the best performance; (2) no need for professional knowledge to use the system; (3) if the unknown mushroom varieties are toxic, they can be identified quickly; and (4) independence from the effects of the natural environment unlike other methods, expanding the scope of its use.

2. Methods

Driven by big data, deep neural networks (DNNs) show great potential [20]. DNNs have achieved remarkable success in various applications. However, deep neural networks have too many hyperparameters, and their learning performance critically depends on their careful tuning. At the same time, it has an impact on DNNs that is difficult to rein in [21].

In response to the above difficulties, Zhou and Feng proposed a multigrained cascade forest framework [19]. gcForest is a decision tree ensemble method. The gcForest method, which consists of multigrained scanning and a cascade forest, is explained in detail below.

As illustrated in Figure 1, after we had obtained the dataset, we checked the integrity of dataset. Firstly, we solved the problem of missing data by Process A and Process B. Secondly, the complete data was divided into test dataset and training dataset. Then, the training dataset was input to the constructed classifier model. After the model judgment, the result of classification depended on the category with high probability. Therefore, the experimental process is shown in Figure 1 in this paper.

In this paper, we use the cascade forest to discriminate the toxicity of mushrooms, which is regarded as a binary classification problem. The 22-dimensional features are easily obtained and used as data input. After running the gcForest model, the toxicity identification results of the corresponding samples can be obtained, as shown in Figure 1.

The gcForest classifier model can accurately judge whether a mushroom is poisonous in a timely manner; thus, it has strong practicality. The classifier also has the following features: (1) ease of trainability; (2) compatibility with various datasets; and (3) an adaptively adjustable hierarchy for the cascade structure depending on the desired the complexity of the model; such a small sample dataset can achieve good recognition performance.

2.1. Cascade Forest Structure. Feature learning by deep neural networks (DNNs) depends on the layer-by-layer processing of the original features [22]. Similar to DNNs,

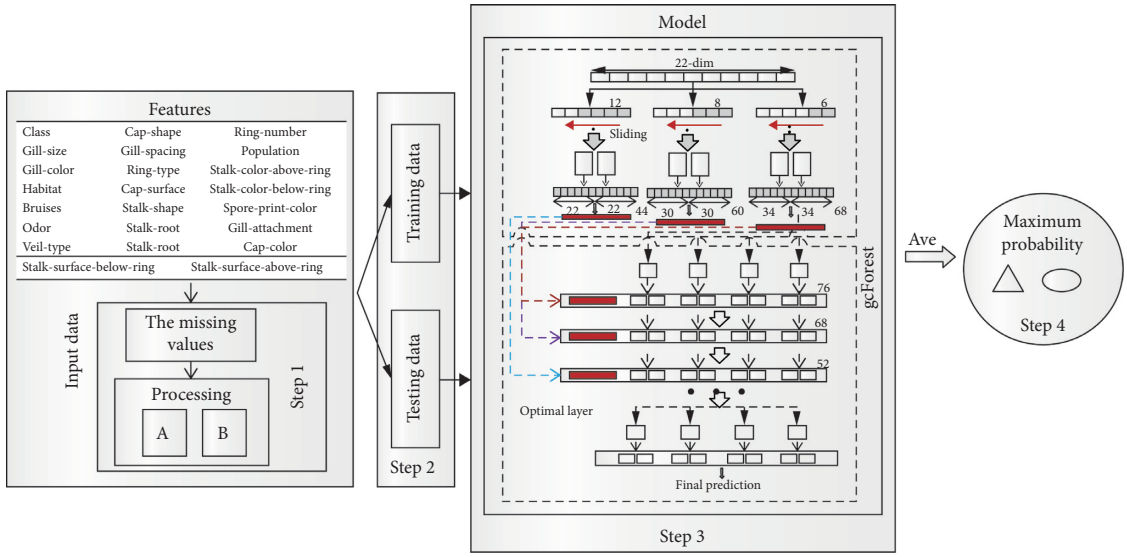

FIGURE 1: Flowchart of the classifier model. It consists of four main steps. (1) The missing values in the mushroom dataset are solved in two ways. (2) Seventy-five percent of the dataset is used as the training data. (3) The gcForest classifier is built on the mushroom dataset. (4) The maximum probability of whether the mushroom is toxic is determined according to the mushroom features.

gcForest uses a cascade forest structure where the information of each layer is processed in an upper layer and the result is delivered to the next layer.

As illustrated in Figure 2, suppose that the mushrooms can be divided into two categories: toxic and nontoxic. The leaf node of each forest will output a two-dimensional class as a vector, which is concatenated for the re-representation of the original input. Therefore, the next level of the cascade will receive $8 = 2 \times 4$ augmented features, and the vector dimension of the input feature will be $2 \times 4 + \text{length}(x)$. Namely, the feature dimension is equal to the number of enhanced features + the number of original (or transformed) features. Each forest will output two-dimensional class vectors, which are connected to the input features to produce the next original input. Additionally, each level contains several classifiers capable of ensemble learning [23].

For simplicity, we suppose that each layer of the cascade forest structure consists of two random forests and two completely random forests (CForests) [24]. Each forest is an aggregation of decision trees [25].

A completely random forest contains a number of completely random trees, generated by randomly selecting a feature for splitting at each node of the tree, and the tree is grown until each leaf node contains only the same class in Figure 3. Similarly, each random forest contains a number of trees, by randomly selecting $\sqrt{d}$ number of features as candidates (d is the number of input features) and choosing the one with the best Gini index (which refers to the index of optimal features when CART is used for classification problems) for splitting. The tree is grown tree until each leaf node contains only the same class of instances [26].

As illustrated in Figure 3, suppose that there are two classes, each forest will generate a two-dimensional class vector. Different symbols in the leaf nodes imply different classes. First, the red color highlights the paths along which the instance traverses to the leaf nodes, and each forest will generate the class distribution by counting the percentage of different classes of training examples at the leaf node where the concerned instance falls [19]. Then, the estimated class distribution forms a class vector. This vector is concatenated with the original feature vector as the input to the next level of the cascade. Finally, the cascade result vectors are averaged to two-dimensional vectors. The class of the maximum value is used to determine whether the mushroom is poisonous.

To reduce the occurrence of overfitting produced by each forest, k-fold cross-validation is used in this algorithm [27]. Each instance will be used as training data for $k - 1$ times to generate $k - 1$ vectors, which are finally averaged to produce the final class vector that represents the augmented features for the next level of the cascade. In other words, these augmented features are partially input in the new cascade layer, and the performance of the entire cascade on the validation set is evaluated. And if there is no significant performance improvement, the training process will automatically terminate. For that reason, the number of cascade layers can be adjusted automatically. Contrary to complex and fixed depth neural networks, gcForest can adaptively stop training to determine the number of cascade layers required.

2.2. Multigrained Scanning. The deep neural network has a powerful advantage in dealing with spatiotemporal and sequence features. Similarly, gcForest takes into account multigrained scanning prior to the cascade structure to enhance the feature learning ability of the overall structure. As shown in Figure 4, gcForest scans the original features using a multigrained sliding window. There are 22 raw features for the sample mushroom, and a window size of 8 features is used; this results in the production of 15 feature vectors. Assuming that there are two classes of mushroom

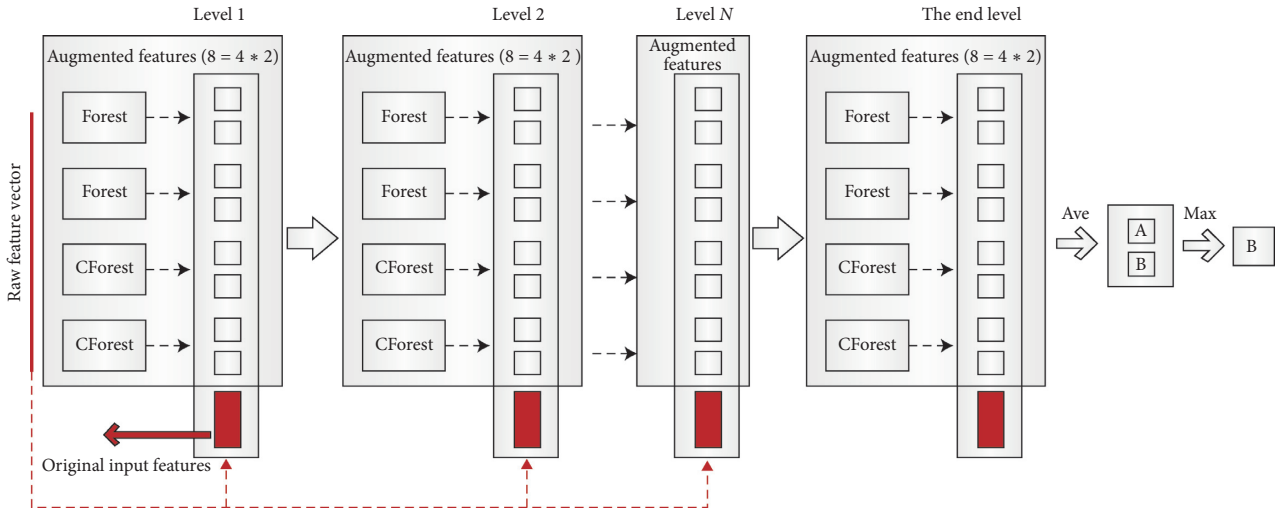

Figure 2: Illustration of the cascade forest structure [19].

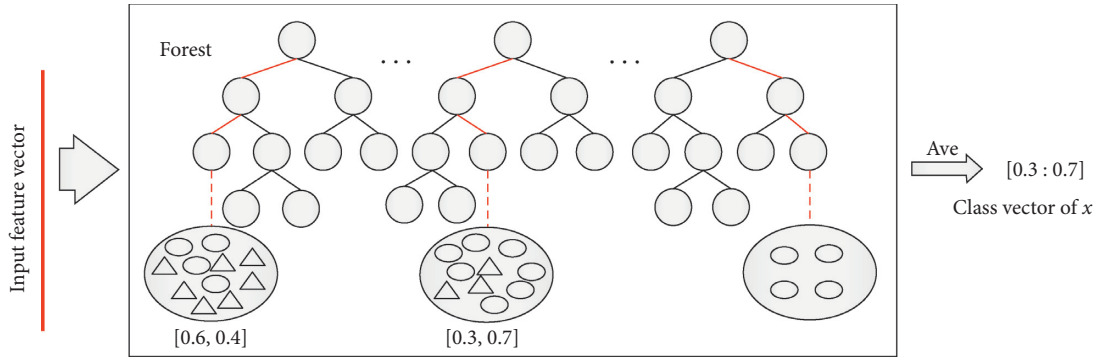

Figure 3: Illustration of class vector generation [19].

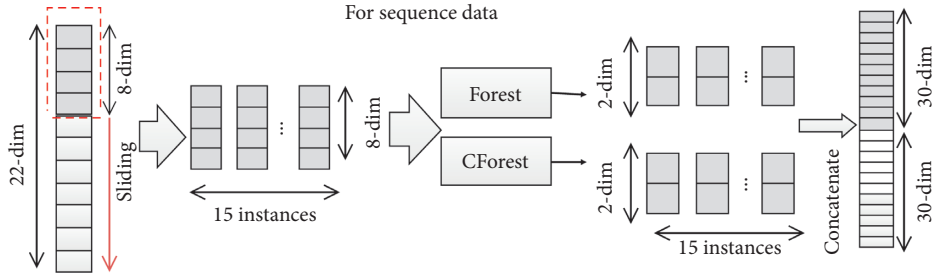

Figure 4: The procedure of sliding window scanning [19].

toxicity, the original 22-dimensional feature vector corresponds to 60 transformed dimensions.

Deep neural networks are effective in handling feature relationships. Inspired by this recognition, we enhance cascade forest with a procedure of multigrained scanning.

As illustrated in Figure 4, firstly, the complete appearance feature in the dataset is $i\,(i = 22)$ dimensions. A sliding sampling window with length $w\,(w = 8)$ is used to obtain $o\,(o = 15)$ subsample vectors with the size of v-dimensional feature. The process is similar to the sliding convolutional core $(o = \lfloor((i - w + 2{*}\ p)/s)\rfloor + 1 = 15)$, with the stride (s) is 1 and the padding (p) is 0. Secondly, each of the instances extracted is used to train a completely random tree forest and a random forest $(f = 1{*}2)$. Thirdly, each forest generates a length (c) probability vector, where c is the number of categories, here equal to two (corresponding to whether the mushroom is poisonous). Finally, the results of forests at each level are joined together to generate the output samples. A representation vector is generated in each forest, and these vectors can be concatenated together to obtain the final sample output in Figure 4. Therefore, there is output feature F-dimensions $(F = o * f * c = 60)$ [28].

Suppose the sliding windows with sizes of 8 and 12 features will generate 60-dimensional and 44-dimensional feature vector for each original training example, respectively. The transformed feature vectors, which contain augmented features by the previous grade, can be used to train the 2nd grade and 3rd grade of cascade forests, respectively. The procedure, in every three levels, will iterate until the boost of accuracy rate is less than the threshold, as illustrated in Figure 5. The repetition process of the training is completed.

Feature vectors of dataset will enter into the cascade forest structure in batches and connect with the upper output data to increase the disturbance of mushroom samples. The input feature vector of the cascaded forest structure will be connected with the output data of the first layer to form the input data of the second layer. Therefore, we have obtained the transforming features from the process. The final model is actually a cascade of cascade forests [19]. Each level in the cascade consists of multiple grades (cascade forests), and each corresponds to a grain of scanning, as shown in Figure 5.

The last layer classifies the upper-layer input data. Counting the percentage of different classes of training examples at the leaf node where the concerned instance falls, we then compute the average value in all forests to generate the maximum value of the class distribution to determine whether the mushroom is poisonous in Figure 5.

3. Results

3.1. Mushroom Dataset. The mushroom dataset provided by the University of California, Irvine was used to classify the toxicity of poisonous mushrooms [29]. The input features of the mushroom include class, cap shape, cap surface, cap color, bruises, odor, gill appendages, gill spacing, gill size, gill color, stalk shape, stalk root, stem-surface-above-ring, stem-surface-below-ring, stem-color-above-ring, stem-color-below-ring, veil type, veil color, ring number, ring type, spore print color, population, and habitat, for a total of 23 features (see Table 1). These features, which can be observed directly, are classified with the feature calculation. There are 8,124 recordings of mushroom data, which can be divided into two nearly balanced classes: poisonous (48%) and nonpoisonous (52%). We have created a new table (see Table 1) based on data attributes.

Each recording of mushroom has 22 features and one class label; however, part of the recording in the mushrooms' dataset is missed in the stalk-root feature. There are two ways to solve the missing data problem in Table 2:

(1) Using a KNN to complete the missing data, which is called Process A in this paper. The value of parameter K is set to 12.

(2) Treating missing attribute values as special values [30]. A special value ("m") is used to fill in the missing values of the stalk-root, which is called Process B.

The first row shows an example of complete data. The second row shows that the original data of the stalk-root is missing. In Process A, the KNN algorithm predicts the value of stalk-root ("c"). In Process B, a special value ("m") is used as the new label for all of the missing data. And the range of the stalk-root $S\_R \in \{b, c, u, e, z, r, ?\}$ is changed to $S\_R \in \{b, c, u, e, z, r, m\}$.

The two processes were specifically verified and compared in the experiment to determine the feature's effect in identifying mushroom toxicity.

The missing values of the original data are processed to obtain a complete list of character data. In the experiment, the error in the numerical data is small relative to the result, and the data can be converted to the target type data by LabelEncoder [31].

When the support vector machine classifier is used, a toxicity category value of −1 or +1 is generally assigned. Therefore, the range of the feature data toxicity y is changed to $y \in \{-1, +1\}$, which is advantageous for obtaining the hyperplane of toxicity.

Three models were run in this paper: logistic regression, SVM, and gcForest.

3.2. Evaluation Standard. To evaluate the classification models, we adopt two measures.

The index for evaluating the performance of the classifier is the accuracy of classification generally. For the test dataset, it is the ratio of the correctly labeled samples to the whole pool of samples in (1).

Assuming that the model of classification is $Y = \widehat{f}(X)$, given by

$$\text{accuracy} = r_{\text{test}} = \frac{1}{N}\sum_{i=1}^{N} I\left(y_i = \widehat{f}\left(x_i\right)\right), \tag{1}$$

where N is the test sample size and I is the indicator function.

The second is a standard evaluation performance of the binary classification used, namely, recall and precision [32]. In addition, there are F1-score and ROC curve.

The mushroom class has two possible predicted classes: edible (e) and poisonous (p). And there are four situations in confusion matrix (see Table 3): true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

The precision and recall are given by the following relation:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{2}$$

Precision is defined as the number of true positives TP over the number of true positives plus the number of false positives FP [32].

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{3}$$

Recall is defined as the number of true positives TP over the number of true positives plus the number of false negatives FN [32].

These quantities are also related to the F1-score, which is defined as the harmonic mean of precision and recall [32].

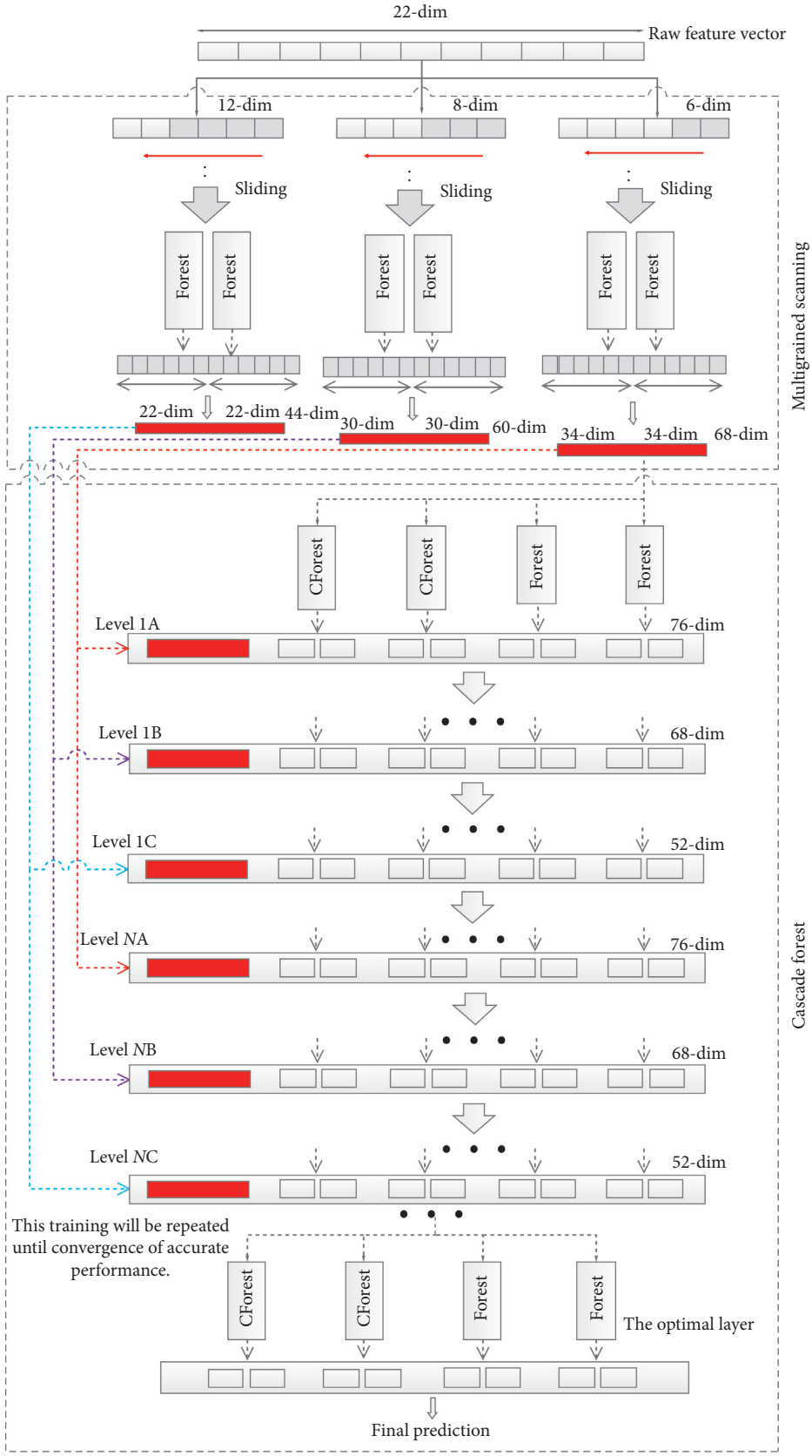

Figure 5: The overall procedure of gcForest [19].

Table 1: Attribute information.

Attribute	Range of value(s)
Class	{edible (*e*), poisonous (*p*)}
Cap-shape	{bell (*b*), conical (*c*), convex (*x*), flat (*f*), knobbed (*k*), sunken (*s*)}
Cap-surface	{fibrous (*f*), grooves (*g*), scaly (*y*), smooth (*s*)}
Cap-color	{brown (*n*), buff (*b*), cinnamon (*c*), gray (*g*), green (*r*), pink (*p*), purple (*u*), red (*e*), white (*w*), yellow (*y*)}
Bruises	{bruises (*t*), no (*f*)}
Odor	{almond (*a*), anise (*l*), creosote (*c*), fishy (*y*), foul (*f*), musty (*m*), none (*n*), pungent (*p*), spicy (*s*)}
Gill-attachment	{attached (*a*), descending (*d*), free (*f*), notched (*n*)}
Gill-spacing	{close (*c*), crowded (*w*), distant (*d*)}
Gill-size	{broad (*b*), narrow (*n*)}
Gill-color	{black (*k*), brown (*n*), buff (*b*), chocolate (*h*), gray (*g*), green (*r*), orange (*o*), pink (*p*), purple (*u*), red (*e*), white (*w*), yellow (*y*)}
Stalk-shape	{enlarging (*e*), tapering (*t*)}
Stalk-root	{bulbous (*b*), club (*c*), cup (*u*), equal (*e*), rhizomorphs (*z*), rooted (*r*), missing (?)}
Stalk-surface-above-ring	{fibrous (*f*), scaly (*y*), silky (*k*), smooth (*s*)}
Stalk-surface-below-ring	{fibrous (*f*), scaly (*y*), silky (*k*), smooth (*s*)}
Stalk-color-above-ring	{brown (*n*), buff (*b*), cinnamon (*c*), gray (*g*), orange (*o*), pink (*p*), red (*e*), white (*w*), yellow (*y*)}
Stalk-color-below-ring	{brown (*n*), buff (*b*), cinnamon (*c*), gray (*g*), orange (*o*), pink (*p*), red (*e*), white (*w*), yellow (*y*)}
Veil-type	{partial (*p*), universal (*u*)}
Veil-color	{brown (*n*), orange (*o*), white (*w*), yellow (*y*)}
Ring-number	{none (*n*), one (*o*), two (*t*)}
Ring-type	{cobwebby (*c*), evanescent (*e*), flaring (*f*), large (*l*), none (*n*), pendant (*p*), sheathing (*s*), zone (*z*)}
Spore-print-color	{black (*k*), brown (*n*), buff (*b*), chocolate (*h*), green (*r*), orange (*o*), purple (*u*), white (*w*), yellow (*y*)}
Population	{abundant (*a*), clustered (*c*), numerous (*n*), scattered (*s*), several (*v*), solitary (*y*)}
Habitat	{grasses (*g*), leaves (*l*), meadows (*m*), paths (*p*), urban (*u*), waste (*w*), woods (*d*)}

Table 2: Two ways to generate the missing data.

Class	Stalk-root		
	The original data	Process A	Process B
Edible (*e*)	Bulbous (*b*)	Bulbous (*b*)	Bulbous (*b*)
Poisonous (*p*)	?	Club (*c*)	Missing (*m*)

$$F_1 - \text{score} = \frac{2}{(1/\text{precision}) + (1/\text{recall})} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \tag{4}$$

The ROC curve describes the change process of classifier performance changing with the change of thresholds settings, where the x-axis represents false positive rate and the y-axis represents true positive rate [33].

$$\text{FPR} = \frac{\text{FP}}{(\text{TP} + \text{TN})}, \quad \text{TPR} = \frac{\text{TP}}{(\text{TP} + \text{FN})}. \tag{5}$$

Ideally, we want the fraction of correct positive class predictions to be 1 (top of the plot) and the score of incorrect negative class prediction to be 0 (left of the plot) [33].

3.3. Experimental Analysis Using Logistic Regression. Assume that $x_0, x_1, \ldots, x_{22}$ is the feature set for mushrooms containing features such as cap shape, cap surface, cap color, swelling, and odor. According to the features, we calculate the probability of this mushroom as toxic, which is called a score. This score is used as the input of the sigmoid function:

$$s = \sum_{i=0}^{21} \omega_i x_i + \alpha. \tag{6}$$

At the same time, a sigmoid function is used to convert the scores into values within the interval of [0,1].

In the logistic regression function, the maximum likelihood function is applied to obtain the parameters of the model. Then, a logistic regression model is constructed, which is turned into an optimization problem using the log-likelihood function [34]. During the period of learning the logistic regression model, the gradient descent method or other improved scores is usually used [35].

Analysis of the cleaned features may result in excessively long training time or memory overflow, due to an excessively large feature matrix. Consequently, rescreening is the most direct and effective method to screen effective features with a random logistic regression model [36]. There are differences between the data obtained from Processes A and B, and so different effective features will be obtained.

As shown in Table 4, the effective feature stalk-color-below-ring is different across the different processes. The only variable is stalk-surface-above-ring. There may be a

Table 3: Confusion matrix.

		Predicted class	
		P	*N*
Actual class	*P*	TP	FN
	N	TP	TN

Table 4: Effective features under different treatments.

Features	Effective features by Process A	Effective features by Process B
1	Cap-surface	Cap-surface
2	Bruises	Bruises
3	Odor	Odor
4	Gill-spacing	Gill-spacing
5	Gill-size	Gill-size
6	Gill-color	Gill-color
7	Stalk-shape	Stalk-shape
8	Stalk-root	Stalk-root
9	Stalk-surface-above-ring	Stalk-surface-above-ring
10	Stalk-surface-below-ring	Stalk-surface-below-ring
11	Stalk-color-above-ring	Stalk-color-above-ring
12	Veil-color	Stalk-color-below-ring
13	Population	Veil-color
14	Habitat	Population
15		Habitat

correlation functional relationship between stalk-surface-above-ring and stalk-color-below-ring. Although the values of stalk-surface-above-ring are different processes, this feature is one of the reasons that affect accuracy.

First, the dataset is divided into 4 even parts; 3 of the parts are selected as the training set, while the remaining 3 are used as the testing set. Second, we establish a model for discriminant poisonous mushrooms on the basis of the training dataset and then judge whether the testing data indicate poisonousness. Then, statistics on the accuracy of prediction are calculated. Finally, the statistics for the effective feature analysis are compared with the statistics for the analysis using all the input features, and the results are described in Table 5.

It is easy to observe that, following Process A, the accuracy obtained with effective features and with all features differs by approximately 0.01% with maximum and minimum accuracy values of 0.955 and 0.940, respectively. Following Process B, the effects of the effective feature set and of the overall feature set on the accuracy vary greatly, with a minimum accuracy of 0.940 and a maximum accuracy of 0.955 (see Table 5). The effective feature set is an important factor in judging accuracy, but an incorrectly judged feature could reduce the accuracy of estimating mushroom toxicity.

3.4. Experimental Analysis Using SVM. The labels for the toxicity category are set to $\{-1, +1\}$ indicating whether a mushroom is toxic. The other features are converted to $0, 1, 2..., n$ using LabelEncoder to represent the samples. The dataset is divided into a training set with 75% of the data and a testing set with the remaining 25% by random selection.

Support vector machine (SVM) is essentially a binary classification model [37]. The basic idea is to find the optimal classification line (surface) from the feature space. The optimal demarcation line maximizes the distance in the binary classification of data.

The minimum accuracy under Process A is approximately 0.02% higher than that under Process B (see Table 6) on the test dataset. Further experiments with KNN constraints on the original data will produce partially correct data. Therefore, increasing the correct proportion of mushroom features can improve the accuracy of mushroom toxicity classification.

3.5. Experimental Analysis on gcForest. To reduce the contrast error between the three experiments, 25% of the dataset is used as the testing data. In this paper, the gcForest model requires two stages: multigrained scanning and the cascade forest [19]. Multigrained scanning generates the features, and the cascade forests use multiple forest cascades to derive prediction probability results [38]. The criterion for selecting each parameter is that the accuracy should be the lowest; this process is then iterated. Due to the inconsistent magnitude of the parameters, the weight assignment will cause analysis errors. The influencing factors of the parameters on the experiment will not be commented upon in detail (e.g., the number of trees in a random forest, the size of the sliding window, the sliding step, the number of cascading random forest, and the number of trees in a single cascade random forest). The maximum fluctuation in the experimental results is less than 8%.

The maximum average denotes the average of the sum of the maximum values of each parameter in the interval range. In the experimental results, when the original data are further tested under the KNN constraint, the processing will produce erroneous data with a higher error than the new category (Process B). Incorrectly judged features will reduce the accuracy of discriminating mushroom toxicity. Thus, the effect of predicting incorrect data in the experiment is greater than that from filling in special values. Table 7 shows that the average accuracy of the multigrained cascade forest classifier fluctuates between a maximum of 0.9835 and a minimum of 0.9260 on the test dataset.

3.6. Analysis of the Results. Three classifiers were built on a mushroom dataset to determine whether the mushrooms are poisonous according to their features.

In the logistic regression experiment, the accuracy of each feature in determining mushroom toxicity is calculated following steps such as dimension reduction and computing accuracy. At the same time, the results from the SVM and gcForest models are analyzed separately.

Table 5: Logistic regression results by different treatments.

Model results	Accuracy of results by Process A		Accuracy of results by Process B	
	Effective features	Overall features	Effective features	Overall features
Maximum value	0.9547021	0.953594	0.953964	0.954585
Minimum value	0.9401772	0.940423	0.945593	0.940431
Average value	0.9507167	0.950822	0.951318	0.950717

Table 6: SVM results with different treatments.

Model results	Accuracy of the results by Process A	Accuracy of the results by Process B
Maximum value	0.963873	0.962088
Minimum value	0.949419	0.937708
Average value	0.960022	0.959802

Table 7: gcForest results with different treatments.

Model results	Accuracy of the results by Process A	Accuracy of the results by Process B	Average value
Average of the maximum	0.981905	0.983506	0.982706
Average of the minimum	0.926004	0.931912	0.928958
Average value	0.953955	0.957709	0.955832

Based on the mushroom dataset, SVM has a slightly higher effect with Process A than that with Process B. Predicting erroneous data in the logistic regression and gcForest models will produce more errors than will adding a new class (Process B) from a certain feature; the data requirements are stricter, and incorrectly judged features will reduce the accuracy of determining mushroom toxicity. Table 8 shows that, among the three classifiers, the average accuracy of gcForest is 0.9835 and fluctuates less than 8%. The implementation of gcForest obviously improves the accuracy of the classification, but this improvement is not stable. Thus, it is necessary to improve the experiment to further improve the effect of the gcForest classification.

In addition, we used four indexes (precision, recall, F1-score, and ROC curve) to compare the performance of the three classification algorithms in this paper (see Table 9 and Figure 6). AUC stands for the area under the ROC curve.

From Table 9 and Figure 6, it is verified that Process B is better than Process A, and more features are beneficial to classification. In other words, the effective feature set is an important factor in judging accuracy, but an incorrectly judged feature could reduce the accuracy of estimating mushroom toxicity.

According to Table 10 and Figures 7 and 8 can be described as follows. It is verified that the result of partially correct data by KNN constraint is better than that by Process B.

The gcForest classifier obtains the higher precision rate and the lower recall rate than other classifiers on the balanced dataset. In addition, we intuitively get the information that ACU and ROC have reached the highest value on the gcForest classifier from Figures 7 and 8 and Table 10. The results further prove the outstanding classifier in the three classifiers and the applicability of gcForest classifier on this dataset.

4. Discussion

In this paper, we proposed an automatic mushroom toxicity identification method. The gcForest method proposed by Zhou has a recognition accuracy of more than 98% [19]. Based on the high accuracy requirements, we analyzed three pattern classification models. Logistic regression yielded classification results by analyzing the effective features necessary to identify toxicity. The accuracy of the SVM method is better than that of logistic regression. Compared to SVM and logistic regression, gcForest achieved better results in terms of identification accuracy. Therefore, gcForest is a good method for automatically identifying whether a mushroom is poisonous.

A number of common mushroom toxicity recognition methods are currently in use. These methods use different contributions for determining toxicity, but they have a number of limitations, such as low accuracy, unsatisfactory detection of unknown toxins, the need for a strict experimental environment, and sufficient professional knowledge and complex experimental testing techniques. To circumvent the limitations of these methods and apply them to small-sample data analysis, we used machine learning. In contrast to deep neural networks, which require great effort in hyperparameter tuning, gcForest is much easier to train and can be applied to different kinds of data in different domains. The gcForest algorithm has the following advantages: (1) it has a simple structure; (2) it can be applied to datasets of different sizes; (3) the testing techniques and handling are simple; and (4) for our experiments, mushroom toxicity is recognized quickly. Feature-based learning and iterative classifiers in the gcForest method have the best performance among the three methods of machine learning used proposed in this paper. This automatic identification

Table 8: Accuracy of the three classifiers with different treatments.

Model results	Accuracy of the results by Process A				Accuracy of the results by Process B			
	Logistic regression		SVM	gcForest	Logistic regression		SVM	gcForest
	Effective features	Overall features		Average value	Effective features	Overall features		Average value
Maximum value	0.9547	0.9536	0.9638	0.9819	0.9540	0.9560	0.9621	0.9835
Minimum value	0.9402	0.9404	0.9494	0.9260	0.9456	0.9404	0.9377	0.9319
Average value	0.9507	0.9508	0.9600	0.9540	0.9513	0.9507	0.9598	0.9577

Table 9: Indexes of precision, recall, F1-score, and AUC.

Model results	The dataset by Process A		The dataset by Process B	
	Effective features	Overall features	Effective features	Overall features
Precision	0.9623	0.9933	0.9644	0.9932
Recall	0.9596	0.9095	0.9595	0.9163
F1-score	0.9596	0.9534	0.9596	0.9573
AUC	0.9832	0.9943	0.9811	0.9964

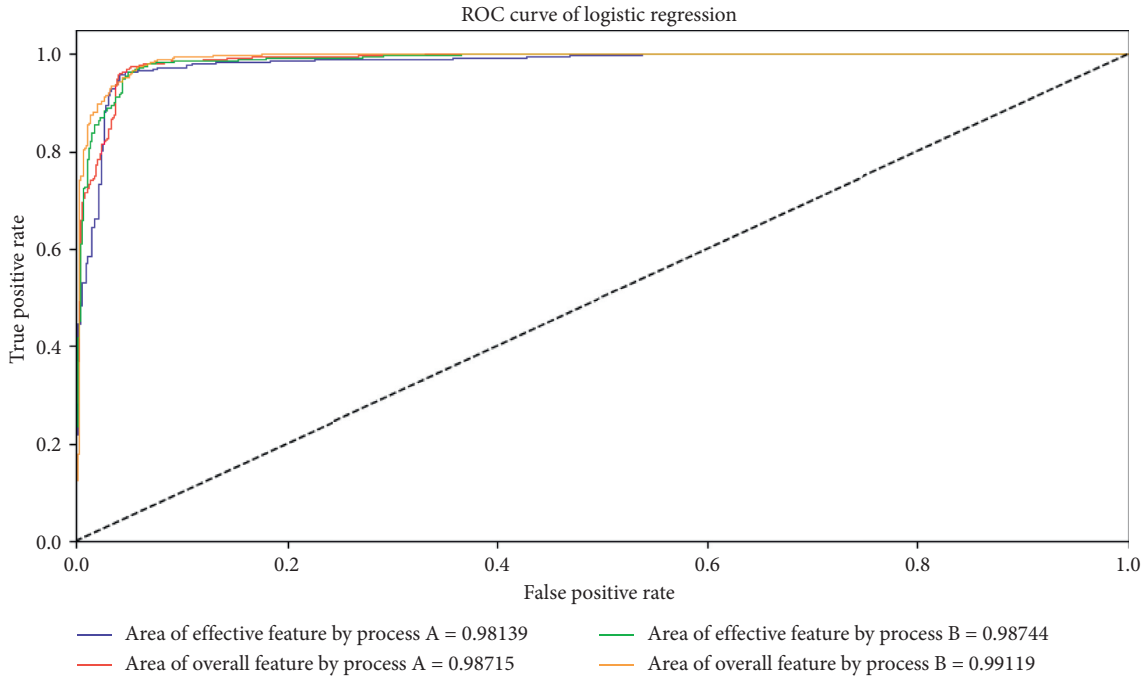

Figure 6: ROC curve of logistic regression.

Table 10: The second evaluation standard on the classifiers of SVM and gcForest.

Model results	The dataset by Process A		The dataset by Process B	
	SVM	gcForest	SVM	gcForest
Precision	0.9623	0.9846	0.9644	0.9816
Recall	0.9596	0.9189	0.9595	0.9292
F1-score	0.9596	0.95397	0.9596	0.9575
AUC	0.9832	0.9913	0.9811	0.9938

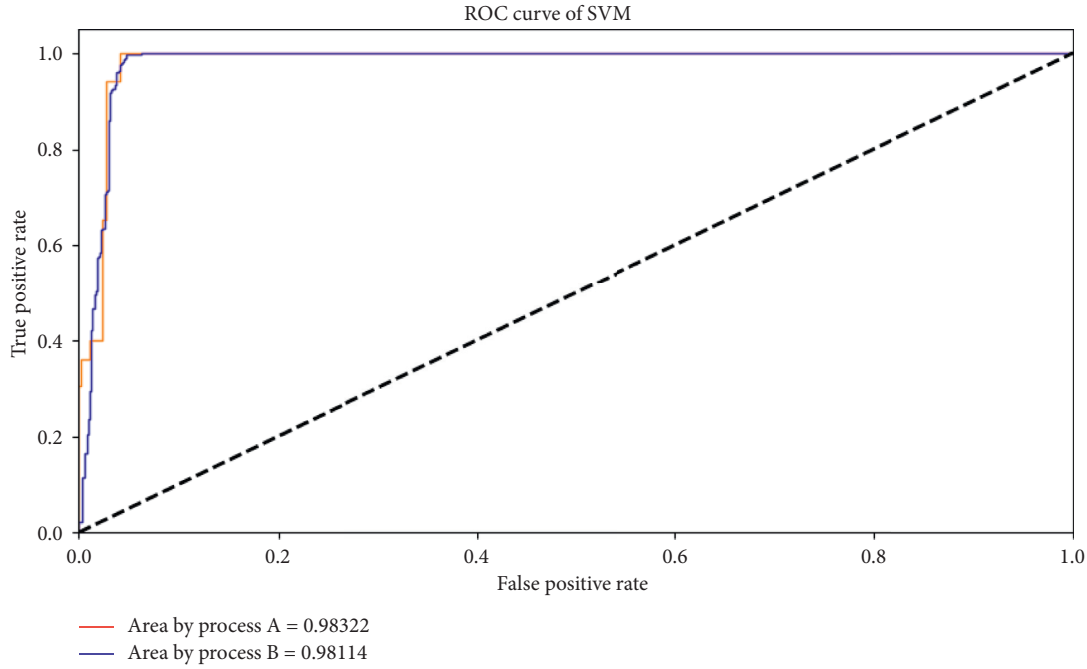

Figure 7: ROC curve of SVM.

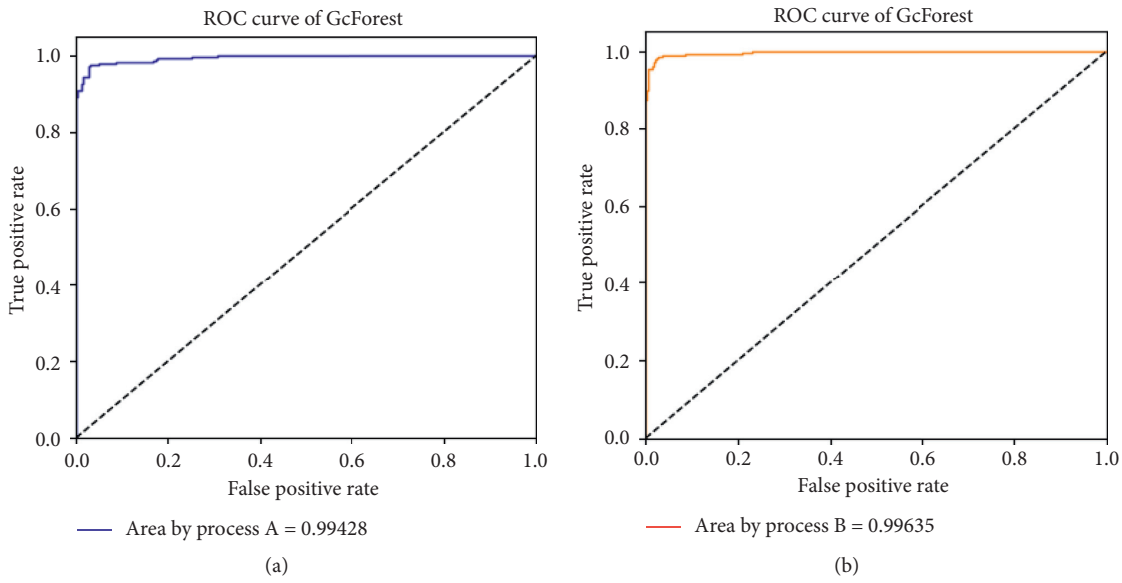

Figure 8: ROC curve of gcForest.

method is suitable for nonprofessional identification and for unknown mushroom varieties.

Among the three machine learning methods, gcForest yielded the best accuracy. However, the stability of the classifier, as shown in Table 8, needs to be improved. A reason for this error may come from the absence of features, the feature labeled stem-surface-above-ring, or it may come from the algorithm itself. Therefore, improving stability is a top priority when trying to improve the accuracy of the classifier. Since gcForest can be used for different types of datasets and recognition based on image features is more convenient than with the other classifiers [39], this method of identifying whether a mushroom is toxic can be extended to image recognition. Nonetheless, there is currently no dataset of mushroom images.

In this paper, we studied whether mushrooms were toxic, by comparing, analyzing, and summarizing four classic and traditional identification methods. According to their shortcomings, we adopted automatic identification methods to conduct the analysis based on machine learning. Based on the mushroom dataset, three pattern recognition analyses were performed. In contrast, gcForest has higher accuracy, but its stability needs to be improved. The used method identifies whether the unspecified mushroom

species is toxic in a timely manner. Because this automated identification method is not affected by the natural environment, it has important social and application value in effectively preventing food poisoning. Meanwhile, people also need to improve their safety awareness of mushrooms.

5. Conclusions

In this paper, multigrained cascade forest was used to determine whether a mushroom was poisonous based on its appearance features. LabelEncoder was used to encode the processed data to form numerical data for the mushroom dataset. According to the analysis of the dataset features, the accuracy of gcForest in data classification was approximately 98%. The maximum fluctuation of its accuracy was less than 8%, however, so the stability of the classifier needs to be improved. The gcForest structure can be used not only for large data but also for small-sized samples, and the adaptive selection of cascade layers can achieve the same accuracy as fixed patterns of deep neural networks in other datasets.

At present, research on toxins in poisonous mushrooms is still underway [40], and cases of mushroom poisoning still occur. Therefore, it is necessary to establish an automatic model for the appearance feature recognition of mushrooms' toxicity. Compared with other mushroom identification methods, the method proposed in this paper uses short cycles, is highly efficient, has low requirements in terms of the natural environment, and results in the timely identification of the toxicity of unknown species. Consequently, this method has important social and application value.

Data Availability

The mushroom data used to support the findings of this study have been deposited in the UCI (https://archive.ics.uci.edu/ml/datasets/mushroom) repository.

Conflicts of Interest

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

Authors' Contributions

All authors have made significant contributions to this research. JXD conceived the research ideas. YYW designed the experiments, conducted the data analysis, and provided the writing of this paper. YYW performed the majority of the data processing, and the University of California Irvine [29] provided rapeseed mushrooms data. HBZ provided important insights and suggestions into this research from the perspective of a computer scientist. All authors read and approved the final manuscript.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (No. 2019YFC1604700), the National Natural Science Foundation of China (Nos. 61673186 and 61871196), the Natural Science Foundation of Fujian Province of China (No. 2019J01082), the Promotion Program for Young and Middle-Aged Teacher in Science and Technology Research of Huaqiao University (Nos. ZQN-YX601 and 18014083020), and the Subsidized Project for Postgraduates' Innovative Fund in Scientific Research of Huaqiao University.

References

[1] J. H. Tegzes and B. Puschner, ""Toxic mushrooms," the veterinary clinics of north America," *Veterinary Clinics of North America: Small Animal Practice*, vol. 32, no. 2, pp. 397–407, 2002.

[2] C. Lei, W. Tangkanakul, and L. Lu, "Mushroom poisoning surveillance analysis," *OSIR Journal*, vol. 1, no. 1, pp. 8–11, 2006.

[3] J. White, S. A. Weinstein, L. De Haro et al., "Mushroom poisoning: a proposed new clinical classification," *Toxicon*, vol. 157, pp. 53–65, 2019.

[4] J. H. Diaz, "Evolving global epidemiology, syndromic classification, general management, and prevention of unknown mushroom poisonings," *Critical Care Medicine*, vol. 33, no. 2, pp. 419–426, 2005.

[5] T. Fukuwatari, E. Sugimoto, K. Yokoyama, and K. Shibata, "Establishment of animal model for elucidating the mechanism of intoxication by the poisonous mushroom Clitocybe acromelalga," *Journal of the Food Hygienic Society of Japan (Shokuhin Eiseigaku Zasshi)*, vol. 42, no. 3, pp. 185–189, 2001.

[6] P. Wexler, B. D. Anderson, and S. C. Gad, *Encyclopedia of toxicology*, Vol. 1, Academic Press, Cambridge, MA, USA, 2005.

[7] M. Lu, "Present status and future prospects of the mushroom industry in China," *Acta Edulis Fungi*, vol. 13, no. 1, pp. 1–5, 2006.

[8] A. Salman, E. Shufan, and I. Lapidot, "Application of multivariate analysis and vibrational spectroscopy in classification of biological systems," in *Proceedings of the International Conference of Computational Methods in Science and Engineering*, Athens, Greece, March 2015.

[9] J. Brzezicha-Cirocka, M. Grembecka, I. Grochowska, J. Falandysz, and P. Szefer, "Elemental composition of selected species of mushrooms based on a chemometric evaluation," *Ecotoxicology and Environmental Safety*, vol. 173, pp. 353–365, 2109.

[10] J. Zhao, M. Cao, J. Zhang, Q. Sun, Q. Chen, and Z.-R. Yang, "Pathological effects of the mushroom toxin α-amanitin on BALB/c mice," *Peptides*, vol. 27, no. 12, pp. 3047–3052, 2006.

[11] J. Guarro, J. Gené, and A. M. Stchigel, "Developments in fungal taxonomy," *Clinical Microbiology Reviews*, vol. 12, no. 3, pp. 454–500, 1999.

[12] K. Tanaka, S. Miyasaka, and T. Inoue, "Histopathological effects of illudin S, a toxic substance of poisonous mushroom, in rat," *Human & Experimental Toxicology*, vol. 15, no. 4, pp. 289–293, 1996.

[13] W. A. Reynolds and F. H. Lowe, "Mushrooms and a toxic reaction to alcohol," *New England Journal of Medicine*, vol. 272, no. 12, pp. 630-631, 1965.

[14] Z. Chaoqun, *Recognition and Research of Poisonous Mushroom Based on Machine Learning*, Taigu: Shanxi Agricultural University, Jinzhong, China, 2019.

[15] P. FengLi, *Research and Design of Virus-Based Mushroom Identification System Based on Android*, Hohhot: Inner Mongolia University of Technology, Hohhot, China, 2019.
[16] Y. Zhifeng, *Application of Multi-Classifier Fusion Based on Stacking Algorithm in Identification of Poisonous Mushrooms*, Taigu: Shanxi Agricultural University, Jinzhong, China, 2019.
[17] F. Shuaichang, Y. Xiaomei, and L. Jian, "Toadstool image recognition based on deep residual network and transfer learning," *Journal of Transduction Technology*, vol. 33, no. 1, pp. 74–83, 2020.
[18] A. Kaur, K. Verma, A. P. Bhondekar, and K. Shashvat, "Implementation of bagged SVM ensemble model for classification of epileptic states using EEG," *Current Pharmaceutical Biotechnology*, vol. 20, no. 9, pp. 755–765, 2019.
[19] Z. H. Zhou and J. Feng, "Deep forest: towards an alternative to deep neural networks," in *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 3553–3559, Melbourne, Australia, August 2017.
[20] R. Giryes, G. Sapiro, and A. M. Bronstein, "Deep neural networks with random Gaussian weights: a universal classification strategy?" *IEEE Transactions on Signal Processing*, vol. 64, no. 13, pp. 3444–3457, 2016.
[21] J. Wang, Y. Yang, and J. Mao, "Cnn-rnn: a unified framework for multi-label image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2285–2294, Las Vegas, NV, USA, June 2016.
[22] P. Swietojanski and A. Ghoshal, "Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR," in *Proceedings of the 2012 IEEE Spoken Language Technology Workshop*, IEEE, Miami, FL, USA, pp. 246–251, December 2012.
[23] M. P. Perrone and L. N. Cooper, "When networks disagree: ensemble methods for hybrid neural networks," in *Proceedings of the Brown University Providence Rhode Islad Institute For Brain And Neural Systems*, pp. 342–358, Providence, RI, USA, August 1992.
[24] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[25] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.
[26] L. V. Utkin and M. A. Ryabinin, "A deep forest for transductive transfer learning by using a consensus measure," in *Proceedings of theConference on Artificial Intelligence and Natural Language*, pp. 194–208, Springer, Cham, Switzerland, November 2017.
[27] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the Fourteenth International. Joint Conference on Artificial Intelligence*, pp. 1137–1145, Montreal, Canada, August 1995.
[28] G. Hu, H. Li, Y. Xia, and L. Luo, "A deep Boltzmann machine and multi-grained scanning forest ensemble collaborative method and its application to industrial fault diagnosis," *Computers in Industry*, vol. 100, pp. 287–296, 2018.
[29] J. Schlimmer, *UCI Machine Learning Repository: Mushroom Data Set*, University of California, School of Information and Computer Science, Irvine, CA, USA, 1987.
[30] P. Balasubramaniam and R. Uthayakumar, "Mathematical modelling and scientific computation," in *Proceedings of the International Conference on Mathematical Moodelling and Scientific Computing*, Springer Science & Business Media, Gandhigram, Tamil Nadu, India, March 2012.
[31] A. Gogna, A. Majumdar, and R. Ward, "Semi-supervised stacked label consistent autoencoder for reconstruction and analysis of biomedical signals," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 9, pp. 2196–2205, 2016.
[32] E. Alpaydin, *Introduction to Machine Learning*, MIT press, Cambridge, MA, USA, 2020.
[33] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, "Machine Learning," *Neural and Statistical Classification*, vol. 13, no. 1994, pp. 1–298, 1994.
[34] K. Khamaru and R. Mazumder, "Computation of the maximum likelihood estimator in low-rank factor analysis," *Mathematical Programming*, vol. 176, no. 1-2, pp. 279–310, 2019.
[35] D. W. Hosmer and L. Stanley, *Applied logistic regression*, Vol. 398, John Wiley & Sons, Hoboken, NJ, USA, 2013.
[36] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting with discussion and a rejoinder by the authors," *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.
[37] K. Peng, V. Leung, and L. Zheng, "Intrusion detection system based on decision tree over big data in fog environment," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 4680867, 10 pages, 2018.
[38] L. V. Utkin and M. A. Ryabinin, "A Siamese deep forest," *Knowledge-Based Systems*, vol. 139, pp. 13–22, 2018.
[39] B. Li, Z.-T. Fan, X.-L. Zhang, and D.-S. Huang, "Robust dimensionality reduction via feature space to feature space distance metric learning," *Neural Networks*, vol. 112, no. 4, pp. 1–14, 2019.
[40] R. Baselt, "Encyclopedia of toxicology," *Journal of Analytical Toxicology*, vol. 38, no. 7, p. 464, 2014.