

Retraction

Retracted: Analysis and Simulation of Multimedia English Auxiliary Handle Based on Decision Tree Algorithm

Complexity

Received 22 August 2023; Accepted 22 August 2023; Published 23 August 2023

Copyright © 2023 Complexity. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] K. Yan, "Analysis and Simulation of Multimedia English Auxiliary Handle Based on Decision Tree Algorithm," *Complexity*, vol. 2020, Article ID 9028303, 12 pages, 2020.

Research Article

Analysis and Simulation of Multimedia English Auxiliary Handle Based on Decision Tree Algorithm

Kaiwei Yan 

School of Foreign Languages, China West Normal University, Sichuan, Nanchong 637002, China

Correspondence should be addressed to Kaiwei Yan; tracey@cwnu.edu.cn

Received 14 July 2020; Revised 26 August 2020; Accepted 28 August 2020; Published 15 September 2020

Academic Editor: Zhihan Lv

Copyright © 2020 Kaiwei Yan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, through the improved decision tree algorithm, the handles in multimedia English assistance are parsed and simulated. In order to better perceive the sense of language in English composition and improve the rationality of intelligent evaluation, an N element based on association analysis is proposed. Sense value quantification calculates its support in the corpus by obtaining N-tuples of the composition. If the degree of support is lower than the threshold, the part where the language sense problem occurs is analyzed, and the type of language sense problem is judged for the students to provide assistance in modifying the composition. In addition, this paper also extracts word features, sentence features, and text structure features in the composition to fit the English handles analytical score. By testing the test set, the experiment shows that, by extracting the language sense features of the candidate's English composition, it can not only judge whether there is a problem with the language sense of the candidate, but also provide a basis for the overall evaluation of the composition.

1. Introduction

With the rapid development of information technology, people can use mobile phones, handheld computers, and other handheld mobile devices to obtain, process, and send information at any time or place, so that communication is everywhere, information is everywhere, and we also rely on handheld mobile devices. Carrying out educational activities and transmitting educational information with wireless networks has provided the possibility for human lifelong learning [1]. In this era of the popularity of handheld mobile devices, especially smart phones and tablets, almost every college student has one. In this environment, where learning can be done anytime, anywhere, people's learning habits and behaviors are quietly changing, and fragmented learning has also become one of our main learning methods. The learner's pronunciation problem is influenced by the phonological system mainly in that pronunciation-related organs or pronunciation actions are not standard, and there are large differences in phoneme discrimination. Learners often pronounce by blind imitation. They cannot fundamentally recognize how

to pronounce correctly, and pronunciation problems cannot be found in time, and if they are not found, there is no feedback to correct them. Therefore, learners sometimes do not even know whether their pronunciation is standard. In order to ensure that your pronunciation is correct, there are many people who are willing to pay high tuition fees and ask foreign teachers to correct their pronunciation. With the popularity of online language learning, an automatic pronunciation error detection and correction system has been spawned [2]. At present, there are only a few products on the market that has pronunciation problems, and most of the functions are relatively simple. Learners can only imitate the audio and video learning materials played by follow-up, and then the system plays the recording. There are only a few software that has feedback for detecting spoken pronunciation, but the flaw is that the feedback function is not enough to solve the root problem of the learner [3]. This function is only available after the learner has followed the pronunciation. It can be pointed out that the learner's pronunciation is not good enough, but the learner cannot understand where his pronunciation is wrong, and how to improve the

pronunciation, so that the learner cannot get the most valuable feedback to correct the information, and often this does not improve the learner's oral ability [4, 5].

The current research related to providing guidance for logging statement-level decisions is not that rich. In the empirical research on long practice, Ji et al. found that the revision history of logging statements by developers accounted for 72% of the changes at the logging statement level, so a simple logging statement level check was designed [6]. Their checker principle is based on the observation if the log code in a similar code block has an inconsistent logging statement level. Pan infers part of the execution path by mapping log messages to source code [7]. Cai et al. analyzed the logs to understand the correct dependencies between log messages from normal execution and used this information to identify anomalies in failed execution [8]. In addition, machine learning and data mining technologies also show great potential in tracking the scale and complexity of large-scale system monitoring and diagnostic challenges [9, 10]. Some studies learn statistical features to detect and diagnose abnormalities. Lai and Chen used classification techniques to group similar log sequences into a set of classes based on certain string distance metrics [11].

Subecz Z first extracted n graphs as the features of the system call sequence and then used support vector machines to classify the trajectories according to the similarity of the trajectories of known problems [12]. Ma et al. introduced the specific implementation of the text classification system in detail, which provided a basis for the subsequent text classification research [13]. McLarnon and O'Neill produced the L1-L2MAP tool, which included manually input phoneme data and then used this data to create a list of expected pronunciation errors [14]. Similarly, for learners whose mother tongue is Vietnamese, Professor Ha's research team studied the common phoneme substitution errors between Vietnamese and English [15]. Yan studied the distinguishing characteristics of flat tongue and tongue-twisted sounds. The research results show that there is a big difference between the peaks of the spectrum energy. Therefore, the energy concentration segment is selected as the distinguishing feature to discriminate between flat tongue and tongue phonemes [16]. Ma and Chow studied the pronunciation of consonants by Japanese learners while learning English. The study shows that, due to the aspirated variants of unvoiced stop sounds in Japanese, the use of non-aspirated/aspirated feature as a distinctive pronunciation feature can well distinguish the categories of consonant pronunciation [17]. An improvement of 17.35% reduction in error rate can be achieved. Finally, they also applied this technology to the CAPT system [18].

At present, although there are many related researches on the location of logging statements, they provide corresponding guidance for developers in the industry, but the relevant content in the recording content is relatively insufficient. In order to fill the gap, it provides developers with practical work. Logging statement-level recommendation functions. This paper implements a text classification technology based on machine learning and recommends levels through code block-related features. The paper conducts a comprehensive research on text classification technology

based on machine learning. The process and the text reprocessing, text representation, spatial dimensionality reduction, classification method, and classification performance evaluation were analyzed and discussed. After that, we focus on the random forest algorithm in the classification algorithm and propose improvements based on the analysis of its principles and characteristics [19]. The improvement mainly includes two aspects: on one hand, the paper explores and optimizes the handle simulation mechanism of the decision tree classification algorithm, performs weighted handle simulation based on the classification effect and prediction probability of the decision tree, and uses the weighted handle simulation to improve the traditional random forest classification algorithm. In the handle simulation mechanism, and text classification experiments to verify the improved effect. On the other hand, the concept of hyperparameters in the algorithm and commonly used hyperparameter adjustment algorithms are introduced. Based on the analysis of text classification scenarios and the random forest algorithm, an algorithm based on random handle simulation and grid handle simulation is proposed to carry out the random forest algorithm. Parameter optimization and, finally, design experiments prove the effectiveness of the algorithm.

2. Improved Decision Tree Algorithm for Simulation Design of English Handle Resolution

2.1. Improved Decision Tree Algorithm. Decision tree algorithm is an instance-based inductive learning algorithm. It focuses on inferring the classification rules in a set of unordered and irregular examples, which are represented by decision trees. The purpose of constructing a decision tree is to get the relationship between attributes and categories and use this relationship to predict the category labels of samples of unknown categories. The algorithm uses a top-down recursive method, compares the attributes of the nodes within the decision tree, judges the branch of the node based on different attribute values, and draws conclusions from the leaf nodes of the decision tree. The main decision tree algorithms are ID3, C4.5 (C5.0), CART, PUBLIC, SLIQ, and SPRINT algorithms. The decision tree classification method has the advantages of processing both data-type and regular-type attributes, insensitivity to intermediate values, processing samples with missing attribute values, and easy to understand output results [20]. However, there are shortcomings that are prone to overfitting, and the results are biased toward features with more values, as shown in Figure 1. The decision tree algorithm is a meta-classifier of the random forest algorithm studied in this paper, which will be studied in more depth below.

There are various evaluation indicators according to different text classification application backgrounds. The metrics for classification performance evaluation include error rate, accuracy rate, recall rate, accuracy rate, F balance, microaverage and macroaverage, and ROC curve. The error rate is the proportion of samples with incorrect classification results to the total number of samples, and the accuracy rate is the proportion of samples with correct classification results to

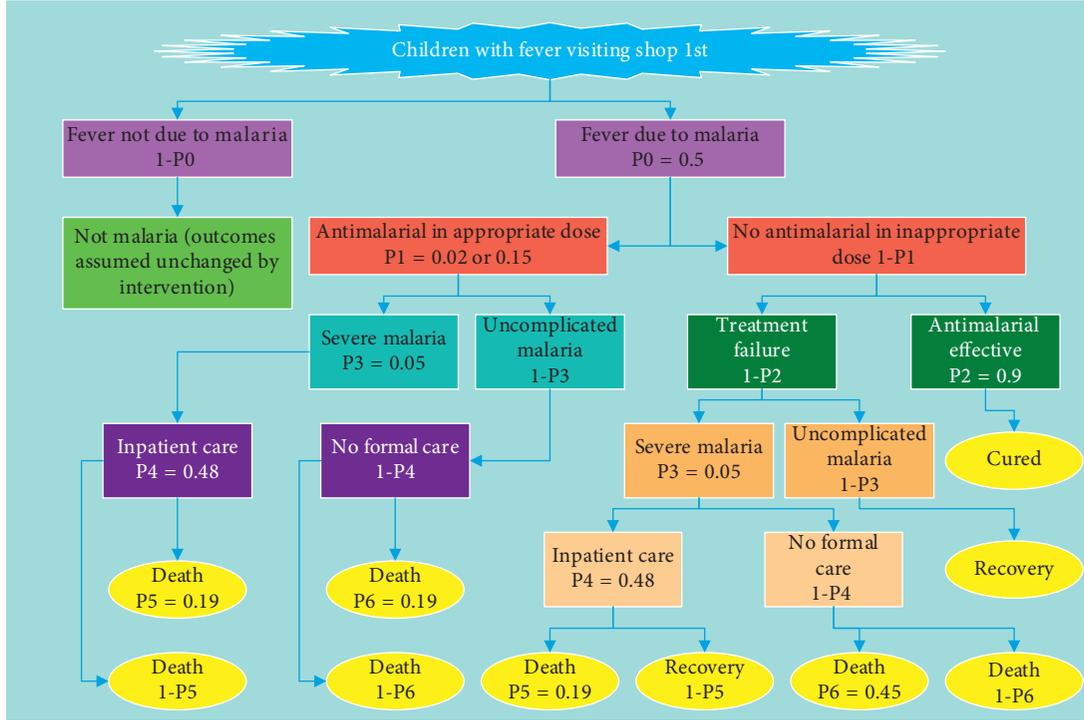


FIGURE 1: Improved decision tree algorithm model.

the total number of samples. They can be used for binary classification or multicategory. But this does not comprehensively reflect the performance of classification models, especially the problem of unbalanced classification [21]. The recall rate and accuracy rate can be calculated based on the confusion matrix established by the classification results. The confusion matrix is shown in Table 1. TP indicates that the text in category C is judged to belong to the text of category C, that is, the text with the correct classification result in category C. FP Text that does not belong to category C is judged to belong to category C; that is, text that is not classified in category C is incorrect, FN is the classification error in category C, and TN is the classification result in noncategory C. The recall rate is the ratio of the number of samples correctly judged to be in this category and the total number of samples belonging to this category, also called the recall (Recall), and the calculation method is formula (1).

$$\text{STFT}[A(t, w)] = \int_{-\infty}^{+\infty} a(\tau)g(\tau - t)e^{-jw\tau} d\tau. \quad (1)$$

Precision is the proportion of samples whose true category is C in the number of samples whose classification result is C, also known as the precision rate, and the calculation method is

$$R_m = \frac{E}{A \times F} * \beta. \quad (2)$$

F equilibrium is an evaluation index that takes into account both the recall rate and the accuracy rate, and its calculation method is shown in formula (3), where β is a value greater than 0, as well as the accuracy and recall weight adjustment parameters. $\beta < 1$ has a greater impact on the

accuracy rate, and $\beta > 1$ has a greater impact on the recall rate. When $\beta = 1$, the recall rate and accuracy rate have the same weight, and the F equilibrium at this time is the $F1$ indicator.

$$R_f = \frac{c}{A \times D} * \beta. \quad (3)$$

Microaverage and macroaverage are the evaluation indicators when evaluating the classifier on the entire data set. They average the results of all categories. They are different from the single-category classification evaluation indicators such as recall rate, accuracy rate, and F balance. The calculation methods of microaverage and macroaverage are formula (4) and formula (5), respectively. Micro-averaging first calculates the total number of instances with correct and incorrect classification results in all categories and then calculates the recall rate and accuracy rate of all categories. It also emphasizes the impact of the category with a larger number of samples on the overall result. Macroaverage first calculates the recall rate and accuracy rate of each category and then performs arithmetic average to get the recall rate and accuracy rate of all categories.

$$GJ_{ik} = \frac{GJ_{ik}^{jint}}{\sum_{i=1}^d GJ_{ik}^{jint}}, \quad (4)$$

$$D = \frac{(n-1)d_n}{\sum_{i=1}^{n-1} d_i}. \quad (5)$$

The full name of the ROC curve is the Receiver Operating Characteristic Curve, which is a curve in a two-dimensional plane. The horizontal coordinate of the plane is

TABLE 1: Classification result confusion matrix.

Type	Text for category C	Noncategory C text
Judge text that belongs to category C	14	22
Text that does not belong to category C	18	42

negative positive rate (False Positive Rate, FPR), which represents that the sample predicted by the classifier as positive is actually negative. The proportion of samples in all negative samples is True Positive Rate (TPR), which represents the proportion of actual positive samples to all positive samples in the samples predicted by the classifier as positive [22]. The ordinate is True Positive Rate (TPR), which represents the proportion of actual positive samples to all positive samples in the samples predicted by the classifier as positive. The curve is usually located on the coordinate lines (0, 0) and (1, 1), the area under the ROC curve is AUC, it is a number, and the effect of the classifier can be evaluated more intuitively. The larger the value, the better the classification effect.

The decision tree algorithm is the base classifier of the random forest. It is an inductive learning algorithm. It learns the classification rules in the form of a tree structure with a large number of samples without order and rules and uses this rule to predict unknown samples. The decision tree is composed of nodes and directed edges, and nodes have intermediate nodes and leaf nodes. Each intermediate node has 4 parameters. One is the decision function, which is the value of a feature. When the feature is less than or equal to this value, the decision path is to the left, and when the feature is greater than this value. The decision path is to the right. The second is the impurity value, which reflects the prediction ability of the current node. The third is the number of covered samples, which refer to the number of samples participating in the decision of this node. The more the covered samples, the more stable the decision functions.

The learning process of decision tree mainly includes feature selection, decision tree generation, and decision tree pruning. Feature selection is to select features from the features of the training data as the segmentation criteria of the current node, where different criteria for feature selection produce different decision tree algorithms. Decision tree generation is a recursive process that generates child nodes recursively from top to bottom according to the selected feature evaluation criteria and stops generating after reaching certain constraints. Pruning is because decision trees are prone to overfitting. To reduce overfitting of decision trees, methods such as prepruning and postpruning can be used to reduce the structure and size of the decision tree. The core algorithm for decision tree growth is the branching criterion, including how to select the best grouping variable from many variables, and how to select the best segmentation point of multiple values of the grouping variable. According to the different methods of decision tree node splitting, it can be divided into two categories; one is ID3, C4.5, and other decision trees based on information entropy splitting, and the other is the CART decision tree based on the Gini index classification.

The full name of CART is Classification and Regression Trees, which can do both classification and regression. It is a classification tree when the result to be predicted is a discrete value, and a regression tree when the result to be predicted is a continuous value. CART is a binary tree, and non-leaf nodes have two branches because it recursively divides the samples on the current node into two subsets during the node splitting process. CART uses the Gini coefficient when the nodes are split and uses the Gini coefficient minimization principle. The calculation method of the Gini coefficient is

$$k_{nn}(x) = \sum_{i=1}^N w_i s_i(x) = W^T S(x), \quad (6)$$

$$M^* = \arg \min \left\{ \sup_{x \in \Omega_x} |b(x) - W^T S(x)| \right\}. \quad (7)$$

Constraints for the decision tree to stop growing, that is, to no longer perform node splitting, include the node's impurity purity reaching the threshold, the node's sample number reaching the threshold, the attributes to be split reduced to a certain value, and the depth of the decision tree growing to a certain value.

2.2. Design of Multimedia English Auxiliary Handle Analysis Simulation System. The main function of this system is to provide a self-learning method, especially for those who have left school but want to take the exam. It is difficult to get the teacher's timely examination. Students may not approve test questions, and teachers may not have time to approve a large number of test papers. This system is a learning system for simulated exams developed to solve the actual needs of this society. The overall design of this system is designed based on the independence of the modules. Module independence is in the design and development process, the function of each module is independent, and it has less interaction with other modules. The independence of modules can make effective modular software easy to develop, and independent modules are more convenient for testing and maintenance. The modules of this system are loosely coupled with each other, and the internal elements of the modules are closely combined.

The system contains a module of question bank management and test paper management. The question bank management module provides teachers with the functions of adding, modifying, and deleting test questions. The test paper management module adds or deletes the test questions in the question bank to the test paper. When adding test questions, it is only the question bank management module that has an impact. Since the test paper just added does not

add to the test paper, it does not have any impact on the test paper management module. When adding test questions in the test paper, only rely on the foreign key to generate the association relationship, so as to form a test paper. This shows that the independence between the two modules is high, so the coupling relationship is loose, as shown in Figure 2.

For the test bank management module, its function is to maintain the test bank. You can add, delete, modify, and check the test questions. You can only operate the large and small question databases in the database, and you cannot and are not allowed to operate the content in other databases. That is to say, the only thing the question bank management module does is to maintain the test question bank. Things that are not within its responsibilities can only be requested by other modules to complete. This is called high cohesion.

In addition to the basic requirements of the system, there are the most important functional business requirements for pronunciation correction. Automatic pronunciation correction system mainly includes two functional modules: a user information management module and a pronunciation correction module. The user management module includes student users, teacher users, and system administrators. Pronunciation correction module includes pronunciation data collection module and pronunciation data error detection module, pronunciation data correction module, and historical data display module. Figure 3 is a functional block diagram of the automatic pronunciation error correction system.

The server is the core of the automatic pronunciation correction system. All functions are implemented through the server Java code. The server uses the framework of spring + spring MVC + Mybatis. Among them, spring is like a container to automatically create and manage the instances in the project. By configuring parameters in the configuration file to call Java entity classes to create objects, its core idea is control inversion (IOC) and dependency injection (DI); using the Spring framework to develop projects can enable developers to focus on the development of business functions and improve development efficiency without paying attention to the creation and management of instance objects. Spring MVC based on Java is a popular lightweight Web MVC pattern framework, which simplifies our development through a request-response driving model. MyBatis is a database interaction layer framework used by many companies. MyBatis does not need JDBC code to operate the database and result set. Simply configure the mapping relationship in XML to complete the mapping between database fields and Java entity class attributes.

2.3. Analysis and Simulation Evaluation Analysis. The running efficiency of the algorithm can be evaluated by the time and space resources required by the computer when running the algorithm. The time resource required by the computer when running the algorithm is also called the time complexity of the algorithm, which mainly

depends on the following factors: the time required for the input of the algorithm data, the time required for the algorithm to be compiled into an executable program, and the time required for the computer to execute each instruction, and the number of times to execute the algorithm statement repeatedly. Since the first three factors mainly depend on the performance of the device, it is customary to use the number of times the algorithm statement is repeatedly executed as the time complexity of the algorithm. When comparing, generally do not care about its precise measurement; only care about the order of magnitude. In the actual environment, there is a more convenient way to measure the time complexity of the algorithm, for example, by comparing the time difference before and after the execution of the algorithm on the same machine.

The storage space resource occupied on the computer memory when running the algorithm is also called the space complexity of the algorithm. The space complexity of the algorithm includes static storage space, storage space required for the input and output data of the algorithm, and variable storage space. The static storage space is a fixed part, and its size does not have a great relationship with the needs of users, including the space occupied by the algorithm code and the space occupied by variable constant characters in the algorithm. The storage space required for the input and output data of the algorithm depends on the problem to be solved. The input and output data are passed through the calling function, and the storage space they need is not different due to the difference of the algorithm. Variable storage space refers to some auxiliary space temporarily generated when running the algorithm. This part has no great relationship with the algorithm itself but is related to the auxiliary algorithm storage structure called by the operating system to execute the algorithm. These spaces are larger than those required by the algorithm itself and are issues that require objective consideration. Because of the rapid development of hardware technology, computers have more and more storage space, and the limitation of storage capacity has little effect on the algorithm. When analyzing the running efficiency of the algorithm, we cannot consider the space complexity of the algorithm more and consider the time complexity of the algorithm more, as shown in Figure 4.

The value range of Brier Score is 0 to 1. The lower the score, the better the performance of the model. Brier Score represents the error between the predicted probability and the actual observation value. That is, the probability of the predicted logging statement level is the same as the actual logging statement level. Probability prediction can assign a very high probability correct logging statement level to instance data or just assign the correct category with a probability that is only slightly higher than the error level probability to instance data. Brier Score is conducive to identifying the ability of the classification model to accurately predict the corresponding category in the former case and can also explain the performance of the classifier model in predicting the category compared to random guessing.

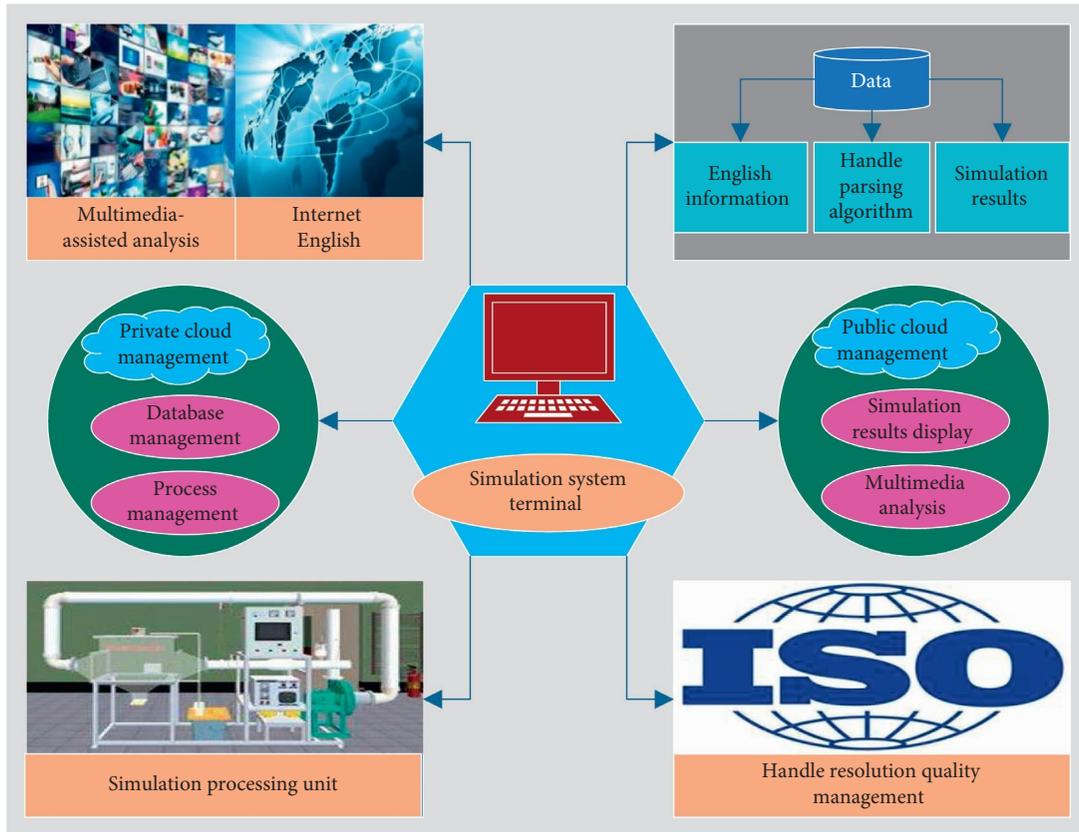


FIGURE 2: Multimedia English auxiliary handle parsing simulation system design.

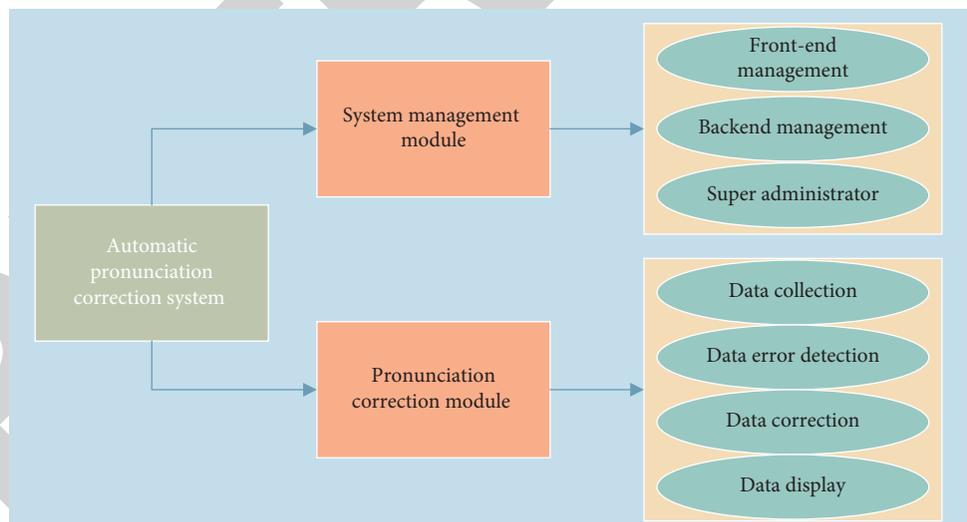


FIGURE 3: Functional block diagram of automatic pronunciation correction system.

3. Result Analysis

3.1. *Evaluation of Experimental Results.* The main work of this paper is to use machine learning text classification technology to implement a logging sentence level recommendation method. As an important source of experience for machine learning, this article selects the top 100 ranked GitHub and uses the logging statements in the Java

language project to build a classification model. The AUC and Brier Score performance results of the classification models constructed by the three classic classification algorithms are shown in Figure 5.

It can be seen from Figure 5 that the AUC of the three classification algorithms is up to 0.815, and only 0.798 when the learning effect is poor. Although the experimental dataset in this paper (25 GitHub projects) is

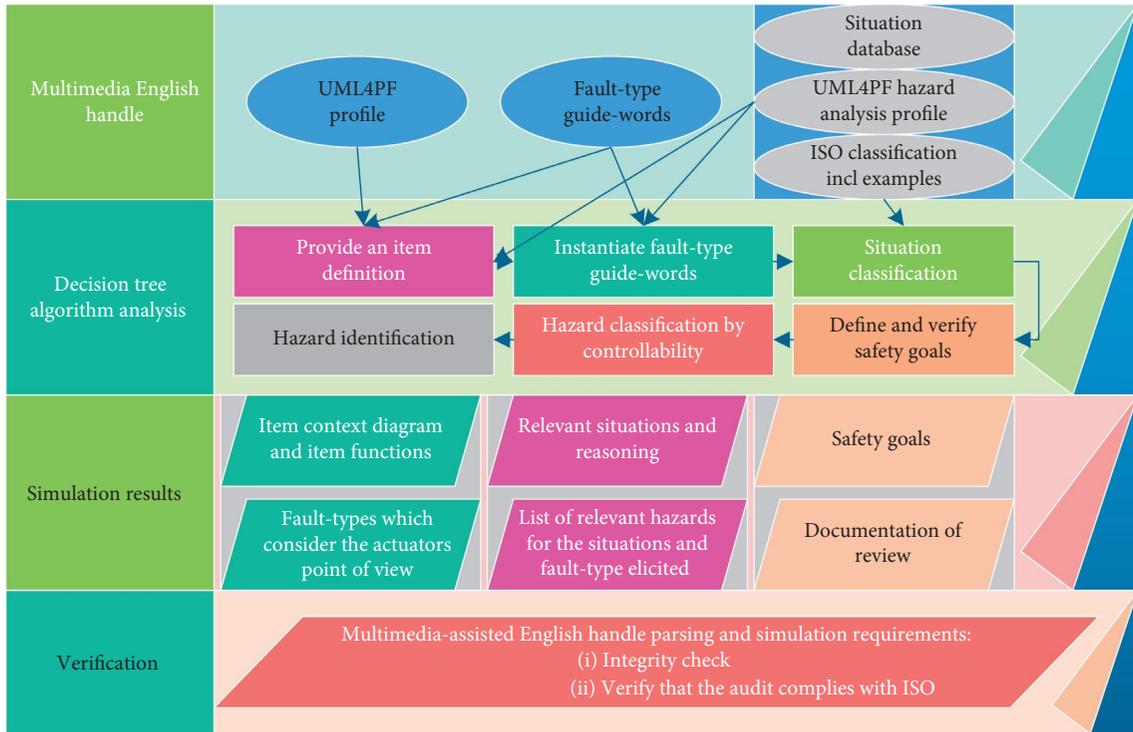


FIGURE 4: Analysis and simulation evaluation design.

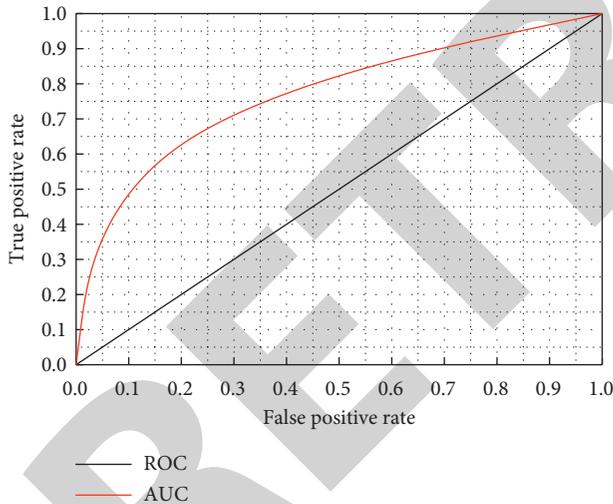


FIGURE 5: ROC and AUC after decision tree classification.

different from the approximate study in the field of logging statement-level recommendations (4 open source projects), it is different from the AUC score in the approximate study (0.75 to 0.81). In contrast, the experimental results of this article are generally equivalent to them, with a slight advantage. Moreover, considering that the increase in the number of items may lead to an increase in the difference in the characteristics of logging statements, the AUC score of 0.798 to 0.815 is sufficient to illustrate that the text classification model constructed in this paper can well overcome the uneven distribution of cross-project machine learning predictions. And

regarding other challenges, it has a better performance in distinguishing different levels of logging statements.

On the other evaluation standard, Brier Score, three classification algorithms selected in this paper can reach 0.440 at the best time and 0.462 at the worst time. Compared with the Brier Score of 0.44 to 0.66 in the approximate study, it has better performance. It shows that, as a probabilistic prediction algorithm, the machine learning model constructed in this paper has a high accuracy rate, which provides a reliable guarantee for the correct recommendation of logging statement level.

It can be seen from the two evaluation indicators that the decision tree is the best among the three selected algorithms. In the ROC curve, one point that needs attention is (0, 1), which represents $FPR=0$ and $TPR=1$, which means that $FN=0$ and $FP=0$, and the classifier in this state can use all the samples. The datum is classified into the correct category, which means that it is an ideal classifier. Therefore, the closer the value of AUC to 1, the stronger the classification performance of this classifier.

This article conducted multiple samplings and then split the sample data in the same way according to the ratio of 9:1. 90% of the data was used for training, and the remaining data was used as test data to verify the performance of the model. Considering the influence of the number of data on the performance of the classification model, this paper selects from four to four representative sampling data of the experimental results for display, as shown in Figure 6, and the number of logging statements for each sample data has been Callout.

It can be seen from Figure 6 that when the number of items decreases and the differences in the factors that affect

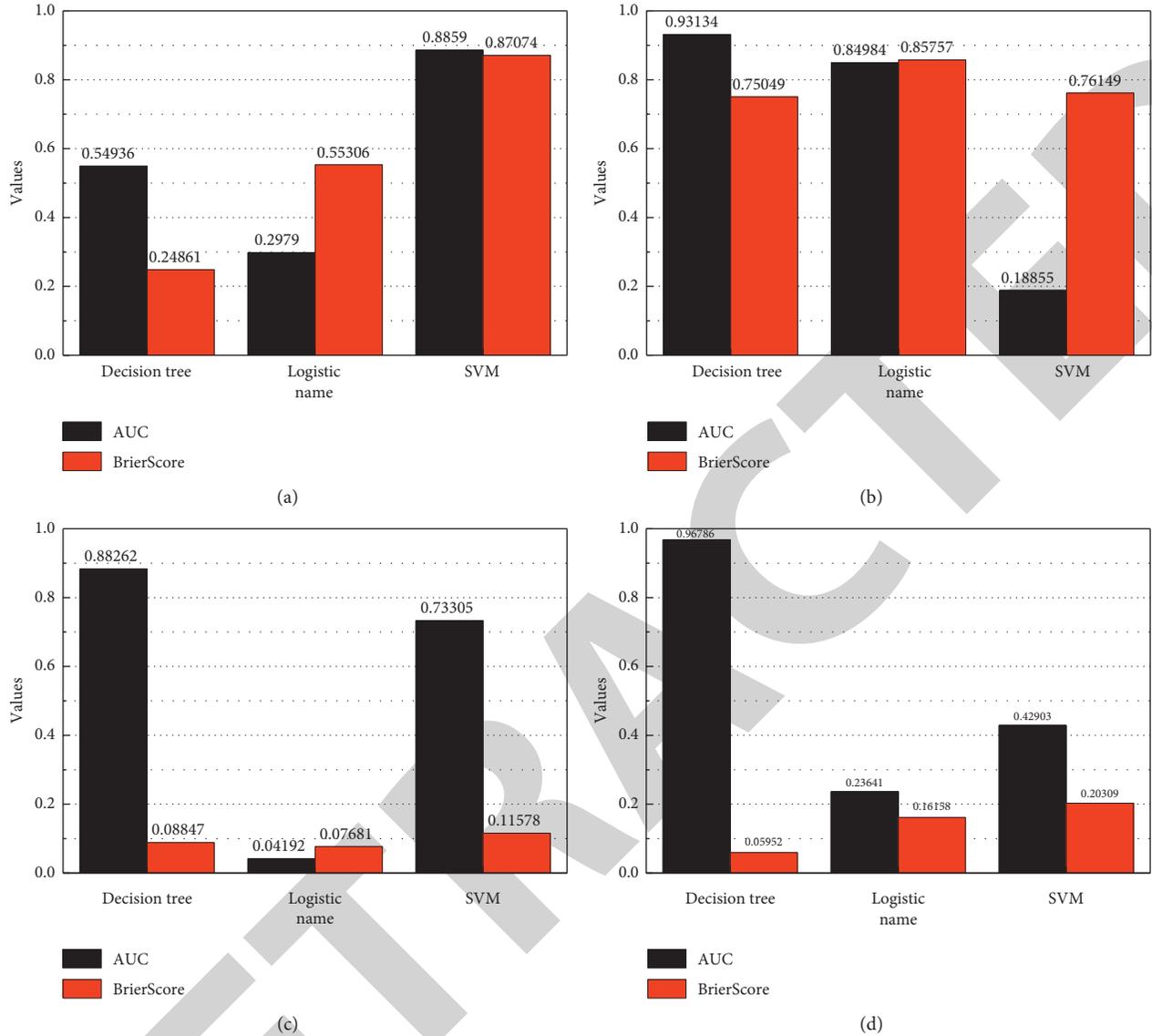


FIGURE 6: Sample experimental results.

the logging statement level become less, the method proposed in this article performs excellent in both AUC and Brier Score indicators. Compared with the test results on multiple projects, the AUC of each sample experiment has a certain degree of improvement, and it can reach 0.836 when the performance is the best. Such a high value means the distance between its ROC curve and the upper left corner. In the process of classifying the log statements, it is likely to be classified into the correct category, and the prediction results of the model are very reliable. The Brier Score indicator has also made no small progress, from the lowest 0.440 on the multi-project data set to the lowest 0.373, successfully below 0.4, and closer to 0, indicating that there is feature learning on approximate regular data; the method proposed in this paper can accurately predict the level of logging statements, and the error in probability prediction is small.

3.2. Optimal Decision Tree Algorithm Handles Analysis Mechanism Result Analysis. Aiming at the traditional random forest classification algorithm and the random forest classification algorithm optimized in this paper to optimize the handle simulation mechanism in text classification, a comparative experiment is conducted. The experimental data use all the data in the 20 news by date data sets, use the training set in the data set to train the classifier, and use its test set to evaluate the classification effect of the algorithm, and the selection time of the evaluation indexes of the performance of the two classification algorithms and the accuracy rate of the algorithm prediction. The time is the time used by the algorithm to train the data set to obtain the classifier and use the classifier to predict the test set for the category result. The prediction accuracy of the algorithm is the ratio of the number of samples that the algorithm

predicts the test samples to obtain the correct classification result to the total number of test samples. Since the number of decision trees in the random forest will have an impact on the performance of the algorithm, when conducting a comparative experiment on text classification using the traditional random forest classification algorithm and the random forest classification algorithm with the optimized handle simulation mechanism in this paper, we should ensure that the number of decision trees is excluded. Under the same conditions except for others, choose different numbers of decision trees for text classification. In the experiment, choose the number of decision trees as 10, 30, 50, 100, 200, 300, 400, and other hyperparameters are the default values for comparison experiments. In addition, in order to eliminate the effect of randomness on the experimental results, comparison experiments under the same number of decision trees are conducted 10 times, and then the average value of the 10 results is taken as the final result.

According to the experimental design, the text classification experiment is carried out, and it is concluded that, under different numbers of decision trees, the traditional random forest classification algorithm and the random forest classification algorithm with optimized handle simulation mechanism in this paper are used. The time for text classification is shown in Figure 7.

It can be seen from Figure 7 that the number of decision trees in the random forest classification algorithm has a certain impact on the time of text classification under the two algorithms. As the number of decision trees in the random forest classification algorithm increases, the two algorithms are used for text classification. The time spent is increasing accordingly. Figure 7 lists the time to use two algorithms for text classification under different numbers of decision trees. Although the random forest classification algorithm that optimizes the handle simulation mechanism in this paper is more complicated than the traditional random forest algorithm used in handle simulation, according to the prediction training accuracy of the out-of-bag data, each decision tree is given a certain weight. When the handle simulation is performed, it is not just outputting the category of the sample, but the probability that the sample belongs to each category. There is no significant increase.

The traditional random forest classification algorithm and the random forest classification algorithm with optimized handle simulation mechanism in this paper are used for text classification experiments. The prediction accuracy of the two algorithms is shown in Figure 8.

It can be seen that the number of decision trees in the algorithm has a greater impact on the accuracy of the algorithm. Generally speaking, the larger the number of decision trees, the higher the accuracy of the algorithm prediction. In the case of different decision tree numbers, the prediction accuracy of the random forest classification algorithm using the optimized handle simulation mechanism in this paper is improved to a certain extent compared with the traditional random forest algorithm. Combining the time comparison used in the text classification experiment using the two algorithms in Figure 8, the random forest

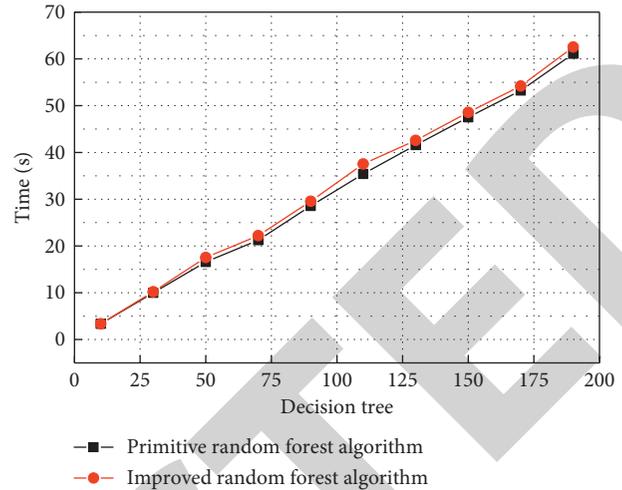


FIGURE 7: Algorithm time comparison.

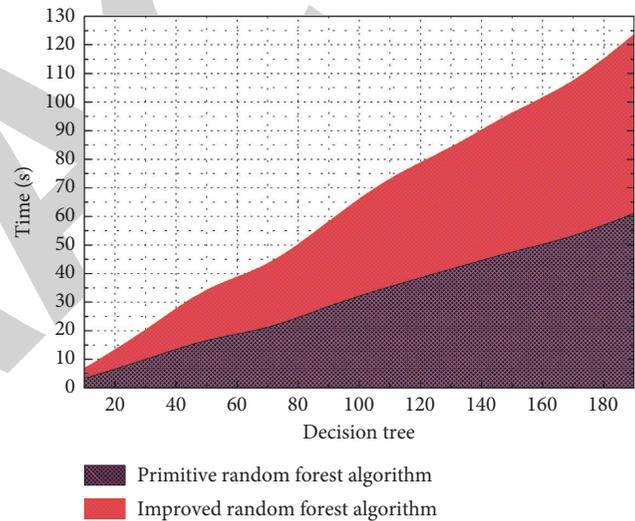


FIGURE 8: Comparison of algorithm prediction accuracy.

classification algorithm that optimizes the handle simulation mechanism in this paper improves the prediction accuracy of text classification on the basis of almost no increase in time resources, compared with the traditional Random forest algorithm that has higher performance. In addition, the prediction accuracy of the two algorithms has not reached a high level, which is related to the use of default values for the hyperparameters of the algorithm except for the number of decision trees, which also illustrates the importance of optimizing the hyperparameter values of the algorithm.

3.3. Handle Simulation Result Analyses. The design comparison experiment proves that combining the optimized handle simulation mechanism and the proposed hyperparameter optimization algorithm has improved the effectiveness of the traditional random forest algorithm. Because the values of the number of decision trees and the number of feature attribute subsets in the random forest algorithm have

a great influence on the performance of the random forest algorithm, these two hyperparameters are selected in the experiment. First, the proposed hyperparameter optimization algorithm is used to find the hyperparameter optimization value of the random forest algorithm with the optimized handle simulation mechanism, then two text classifications are performed, and the first time the hyperparameter optimization value of the random forest algorithm with the optimized handle simulation mechanism is used to perform text classification. For the second time, the traditional random forest algorithm hyperparameters use default values for text classification.

Using a random handle simulation algorithm to perform hyperparameter handle simulation, the experimental results of random handle simulation are shown in Figure 9. Figure 9 is a scatter plot of three-dimensional coordinates, where the x and y coordinates represent the number of decision trees and the number of features in the feature attribute subset, respectively, and the z coordinate is the evaluation index of the experimental algorithm, that is, the average score of the scores on the test set. Each point corresponds to the score on the test set under the condition that the hyperparameter value is taken.

It can be seen from Figure 9 that, in the case where the number of decision trees and the number of features in the feature attribute subset are different value combinations, the score on the test set is significantly different. When the number of decision trees is small and the number of features in the subset of feature attributes is large, the score on the test set is significantly lower. If only one of the number of decision trees and the number of features in the subset of feature attributes is considered, basically the greater the number of decision trees, the higher the score on the test set; the smaller the number of features in the subset of feature attributes, the higher the score on the test set high. According to the results, the best five sets of experimental hyperparameter values are shown in Figure 10.

Results of random handle simulation are analyzed to determine the number of grid handle simulation. By analyzing the five sets of hyperparameter value combinations in Figure 10, it can be seen that the first set of experimental results is significantly better than the other four sets. According to the algorithm proposed above, the hyperparameter values of the five sets of results are similar in performance. The gap is large, and a grid handle simulation is performed around the optimal value. That is, only one grid handle simulation is needed near $k = 279$ and $m = 3$. Carry on the grid handle simulation to get the final hyperparameter value. Set the hyperparameter range of the grid handles simulation algorithm. The range of k is $250 \leq k \leq 340$, the step size is 10, the range of m is $2 \leq m \leq 6$, and the step size is 1. The handle simulation result of the grid handles simulation algorithm, as shown in Figure 11. Figure 11 is a graph under three-dimensional coordinates, where the x coordinate and the y coordinate represent the number of decision trees and the number of features in the feature attribute subset, and the z coordinate is the evaluation index of the experimental algorithm, that is, the average score of the score on the test set. The points on each grid represent the

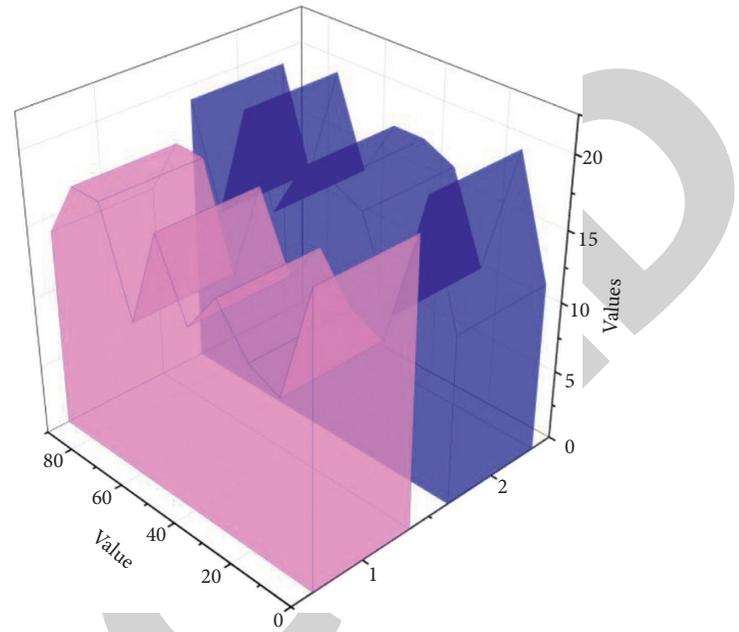


FIGURE 9: Random handle simulation results.

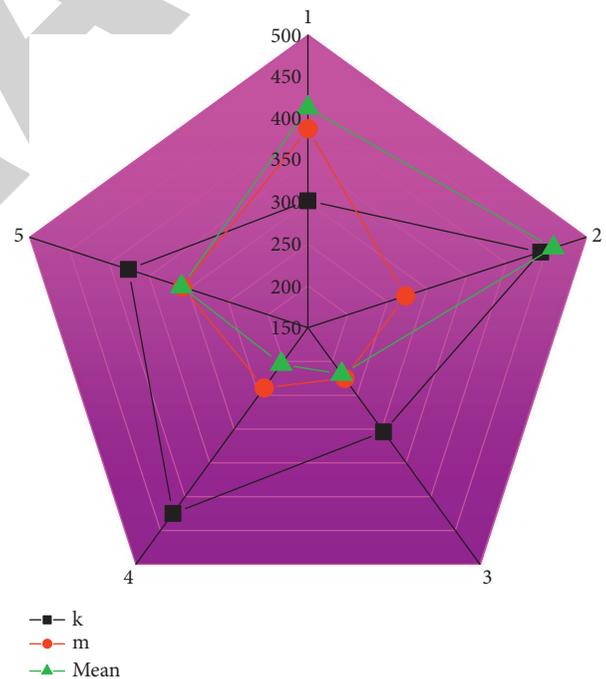


FIGURE 10: The best five sets of experimental parameter values and experimental results.

score of the algorithm on the test set when the number of decision trees and the number of features in the subset of feature attributes are the corresponding horizontal and vertical coordinates. Connect each point in a line to form a grid-like graph. Different colors are used to distinguish the score of different test sets. The transition from dark blue to red corresponds to the high to low score on the test set. According to the results of the grid handle simulation, the

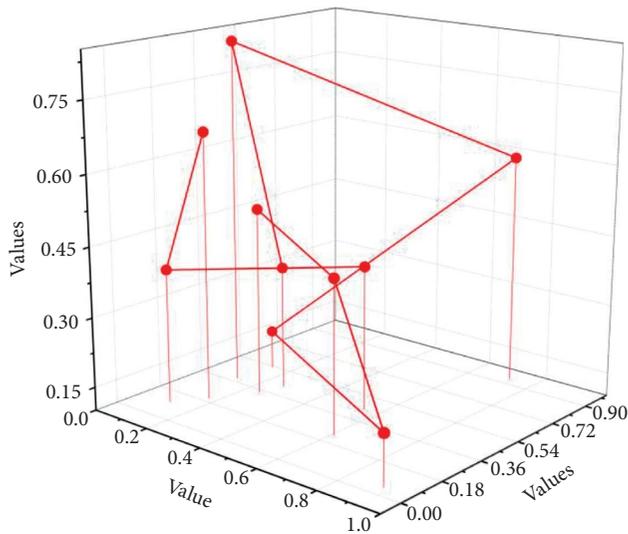


FIGURE 11: Grid handle simulation results.

optimal value of the hyperparameters of the algorithm is obtained, $k = 270$, $m = 2$, and the mean value of the score is the highest at 0.8362.

After text-processing the data set and expressing it as a data set suitable for training the classifier, the random forest algorithm that performs the traditional random forest algorithm and the optimization method of the handle simulation mechanism based on weighted handle simulation proposed in the paper are used for text classification, experimental algorithm comparison experiment. Using time and algorithm prediction accuracy as indicators to evaluate the performance of the two algorithms, the experimental results can be concluded by analyzing the experimental results. The random forest algorithm proposed in this paper based on the optimization method of the handle simulation mechanism based on weighted handle simulation has more advantages than the traditional random forest classification algorithm.

4. Conclusion

In this paper, through the improved decision tree algorithm, multimedia English-assisted handle parsing and simulation are studied. After analyzing the research status, the vacancies in the current price segment pronunciation error detection research field are summarized, and a classification error detection model method based on machine learning algorithms is proposed. To solve the pronunciation problem caused by the learner's pronunciation is not standard, so as to make more intuitive suggestions for the learner's pronunciation. A random forest model that optimizes the handle simulation mechanism is proposed. Handle simulation weight of each decision tree is obtained by the classification correct rate of the decision tree. The probability that the output sample belongs to each class when deciding on the sample result is determined. The output of the decision tree is used. The class probabilities and the weight of the decision tree are weighted to simulate the handles to

obtain the handle simulation of samples belonging to each class, and the final classification result is obtained. Design text classification experiments to compare the traditional random forest algorithm and the random forest algorithm of the optimized handle simulation mechanism proposed in this paper. The analysis of the experimental results proves that the random forest algorithm of the optimized handle simulation mechanism improves when the acceptable time complexity increases. The handle simulation ability of the algorithm is improved, and the purpose of improving the classification performance is achieved.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] Y. Huo, "Analysis of intelligent evaluation algorithm based on English diagnostic system," *Cluster Computing*, vol. 22, no. 6, pp. 13821–13826, 2019.
- [2] Y. Choi and H. Lee, "Data properties and the performance of sentiment classification for electronic commerce applications," *Information Systems Frontiers*, vol. 19, no. 5, pp. 993–1012, 2017.
- [3] A. Kumar and A. Jaiswal, "Swarm intelligence based optimal feature selection for enhanced predictive sentiment accuracy on twitter," *Multimedia Tools and Applications*, vol. 78, no. 20, pp. 29529–29553, 2019.
- [4] B. Münzer, K. Schoeffmann, and L. Böszörményi, "Content-based processing and analysis of endoscopic images and videos: a survey," *Multimedia Tools and Applications*, vol. 77, no. 1, pp. 1323–1362, 2018.
- [5] Z. Li and Y. Zhang, "Hierarchical evaluation algorithm of logistics carrying capacity based on transfer learning in multimedia environment," *Multimedia Tools and Applications*, vol. 78, no. 4, pp. 4481–4501, 2019.
- [6] H. Ji, Y. Yun, S. Lee, K. Kim, and H. Lim, "An adaptable UI/UX considering user's cognitive and behavior information in distributed environment," *Cluster Computing*, vol. 21, no. 1, pp. 1045–1058, 2018.
- [7] L. Pan, "A big data-based data mining tool for physical education and technical and tactical analysis," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 14, no. 22, pp. 220–231, 2019.
- [8] B. Cai, L. Huang, and M. Xie, "Bayesian networks in fault diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2227–2240, 2017.
- [9] L. Deng, D. Li, X. Yao, D. Cox, and H. Wang, "Mobile network intrusion detection for IoT system based on transfer learning algorithm," *Cluster Computing*, vol. 22, no. 4, pp. 9889–9904, 2019.
- [10] Z. Xu, "Dynamic monitoring and management system for land resource based on parallel network algorithm and remote sensing," *Journal of Intelligent & Fuzzy Systems*, vol. 37, no. 1, pp. 249–262, 2019.

- [11] Z. Lai and Q. Chen, "Reconstructing granular particles from X-ray computed tomography using the TWS machine learning tool and the level set method," *Acta Geotechnica*, vol. 14, no. 1, pp. 1–18, 2019.
- [12] Z. Subecz, "Event detection and classification in natural texts," *GRADUS*, vol. 6, no. 1, pp. 16–21, 2019.
- [13] Z. Ma, Y. Lai, W. B. Kleijn, Y.-Z. Song, L. Wang, and J. Guo, "Variational Bayesian learning for Dirichlet process mixture of inverted Dirichlet distributions in non-Gaussian image feature modeling," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 2, pp. 449–463, 2018.
- [14] M. J. W. McLarnon and T. A. O'Neill, "Extensions of auxiliary variable approaches for the investigation of mediation, moderation, and conditional effects in mixture models," *Organizational Research Methods*, vol. 21, no. 4, pp. 955–982, 2018.
- [15] M. A. Hariri-Ardebili and F. Pourkamali-Anaraki, "Support vector machine based reliability analysis of concrete dams," *Soil Dynamics and Earthquake Engineering*, vol. 104, pp. 276–295, 2018.
- [16] R. Yan, "Researches on hybrid algorithm for moving target detection and tracking in sports video," *Cluster Computing*, vol. 22, no. 2, pp. 3543–3552, 2019.
- [17] J. Ma and T. W. S. Chow, "Topic-based algorithm for multilabel learning with missing labels," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 7, pp. 2138–2152, 2018.
- [18] Y. Wu and Q. Ji, "Facial landmark detection: a literature survey," *International Journal of Computer Vision*, vol. 127, no. 2, pp. 115–142, 2019.
- [19] M. B. Mustafa, M. A. M. Yusoof, Z. M. Don, and M. Malekzadeh, "Speech emotion recognition research: an analysis of research focus," *International Journal of Speech Technology*, vol. 21, no. 1, pp. 137–156, 2018.
- [20] Y. Li, L. Zhang, Y. Xu, Y. Yao, R. Y. K. Lau, and Y. Wu, "Enhancing binary classification by modeling uncertain boundary in three-way decisions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 7, pp. 1438–1451, 2017.
- [21] E. Gundogdu, H. Ozkan, and A. A. Alatan, "Extending correlation filter-based visual tracking by tree-structured ensemble and spatial windowing," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5270–5283, 2017.
- [22] L. Bilke, B. Flemisch, T. Kalbacher, O. Kolditz, R. Helmig, and T. Nagel, "Development of open-source porous media simulators: principles and experiences," *Transport in Porous Media*, vol. 130, no. 1, pp. 337–361, 2019.