Hindawi
Mobile Information Systems
Volume 2022, Article ID 6457286, 10 pages
https://doi.org/10.1155/2022/6457286

Research Article

An English Pronunciation Error Detection System Based on Improved Random Forest

Haiyang Cao and Chengmei Dong

Department of International Business, Qingdao Huanghai University, Qingdao, Shandong 266427, China

Correspondence should be addressed to Haiyang Cao; adelle413@126.com

Received 28 January 2022; Revised 20 February 2022; Accepted 2 March 2022; Published 19 April 2022

Academic Editor: Chia-Huei Wu

The existing English pronunciation error detection methods are more oriented to the detection of wrong pronunciation, and lack of targeted improvement suggestions for pronunciation errors. With the aim of solving this problem, the paper proposes an English pronunciation error detection system based on improved random forest. Firstly, a speech corpus is constructed along with the evaluation of the acoustic features. Then an improved random forest detection algorithm is designed. The algorithm inputs rare mispronunciation data into a GAN neural network to generate new class samples and improve the uneven distribution of mispronunciation data in the sample set. The distribution rules of the pronunciation data are extracted layer by layer by stacking deep SDAEs, and the coefficient penalties and reconstruction errors of each coding layer are combined to identify the features associated with the wrong pronunciation in the high-dimensional data. Furthermore, a forest decision tree is constructed using the reduced-dimensional feature-based data to improve the pronunciation detection accuracy. Finally, the extracted 39 Mel Frequency Cepstral Coefficient (MFCC) acoustic features are used as the input of the improved random forest classifier to construct a classification error detection model. The experimental results indicate that the designed system achieves a high accuracy of English pronunciation detection.

1. Introduction

In the context of the comprehensive development of economic globalization and the continuous promotion of China's opening-up process, communication between countries around the world is intensifying. More and more people are eager to learn another language (Second Language, L2) in addition to their native language [1, 2], which in turn gives them an advantage in their study, life, and work. As it is well known, the purpose of language is to communicate, while mastering a language necessarily requires learning its spoken pronunciation. Currently, L2 oral language teaching [3] is teacher-centered and still uses the traditional one-to-many teaching model like other subjects, ignoring the development of students' independent learning ability. In addition, this face-to-face teaching mode is limited by many external objective conditions, such as time and place. More importantly, L2 teachers are scarce in China, especially in the central and western regions where educational resources are insufficient. Students and teachers do not have one-on-one opportunities, so it is difficult to find out the problems in students' pronunciation. Students do not get any advice from their teachers to correct their pronunciation, which in turn does not promote the learning of L2 oral language [4].

In recent decades, the birth and development of Computer Assisted Language Learnging (CALL) [5] has been promoted along with the development of science and technology. This digitally assisted teaching technology makes language learning more convenient for learners, who can use fragmented time and any location for language learning. Some scholars have divided the development of CALL into stages [6]. In the early stage, CALL systems could only provide some text learning of words and grammar. With the development of Internet technology, the current stage of CALL system can meet the interactive L2 learning scenarios of users, for instance, remembering words by looking at pictures and watching videos, and practicing pronunciation by simulating dialogues in scenes. CALL systems that focus on L2 pronunciation learning are called Computer Assisted Pronunciation Training (CAPT) [7, 8].

When people learn to pronounce L2, they are often influenced by the pronunciation habits of their native language, which in turn leads to mispronunciation. Learners whose native language is Mandarin often pronounce the English vowel/i:/with incorrect duration because of the influence of the rhyme phoneme/i/in Mandarin. Therefore, the CAPT system, which aims to improve L2 learners' pronunciation, is designed with two types of basic modules, one of which is the pronunciation assessment of L2 learners' pronunciation, i.e., the rating of L2 learners' pronunciation quality according to the system's pre-agreed scoring rules. The APED system has received increasing attention from researchers in the field of phonetics [9]. The APED system can detect the mispronunciation of L2 learners, i.e., Mispronunciation Detetion (MD). Currently, researchers are working on further implementation of the APED system for Mispronunciation Detetion and Diagnose (MD&D) [10], which not only identifies mispronunciations, but also diagnoses the causes of mispronunciations and provides feedback to L2 learners to promote their pronunciation. Learners, in order to promote their speaking skills.

With the continuous improvement in the performance of automatic speech recognition algorithms and the positive achievements of related applications, new ideas for the research of APED algorithms are put forward. The literature [11] used Gaussian mixture model (GMM) and linear discriminant analysis (LDA) for voice recognition. The literature [12] employed SVM with multilayer perceptron classifier for phoneme classification of Bengali. Automatic speech recognition (ASR) based pronunciation error detection and ASR have similarity in the construction of acoustic models. ASR techniques can be essentially divided into two categories [13], which are Hidden Markov Model (HMM) based acoustic model for speech recognition and end-to-end acoustic model-based language recognition, in view of the achievements of deep neural networks on language recognition. The literature [14] proposed a cross-language HMM-DNN speech recognition model to improve the automatic language recognition performance. The literature [9] proposed an end-to-end pronunciation error detection algorithm based on connectionist temporal classification (CTC) with attention blending. Due to the lack of a corpus specifically annotated by linguists with L2 pronunciation error information, the pronunciation error detection algorithm in the paper achieves error detection for L2 pronunciation but cannot provide diagnostic information.

The current problems faced by pronunciation error detection are, firstly, the small coverage of pronunciation error types and the great limitation of error detection types, then, the importance of corrective feedback is ignored, and most of the studies at this stage are only biased to how to conduct pronunciation error detection, and the detected errors can only indicate the problems of learners' pronunciation. The detected errors can only indicate the problems of learners' pronunciation, but cannot suggest the targeted improvement of pronunciation errors. The effect on learners' improvement of pronunciation level is minimal.

In order to solve the above problems, this paper proposes an improved random forest English pronunciation error detection system. The system adopts the Mel Frequency Cepstral Coefficient (MFCC) and the improved random forest (RF) algorithm in the English pronunciation detection module to classify and detect pronunciation errors caused by the nonstandard position of pronunciation-related organs, movements, and duration of pronunciation, to clarify the pronunciation problems of learners and make it possible to provide feedback to correct different types of errors.

Section 2 of the paper is a theoretical study of the related algorithms. Section 3 contains a concrete implementation of an English pronunciation error detection system with improved random forest. Section 4 contains the experimental part. Section 5 is the conclusion.

2. Related Theories

2.1. Generative Adversarial Networks. GAN is a novel generative model proposed by Goodfellow et al. in 2014 [15], which belongs to one of the deep learning algorithms. GAN is composed of a discriminative model D and a generative model A, which simulates the probability distribution of some data in a specific way, so that it is the same as the probability statistical distribution of some target data or as similar as possible. At the beginning of the algorithm, some data are generated based on the input of noise k to A. D determines from the real data and the data generated by A which are the fake data generated by A. The whole process is based on the input of noise k to A. The whole process is equivalent to a game between A and D. The purpose of A is to make the data it generates not easily recognized by D, and the purpose of D is to determine the source of the data as accurately as possible, iterating and optimizing the process to finally reach a steady state, when D can generate fake data close to the real data distribution, not just a reproduction of the real data, and achieve the data expansion by the fake data generated by A. The role of data expansion is achieved by the falsified data generated by A. The core idea can be expressed mathematically as follows.

$$\min_{\mathrm{A}} \max_{\mathrm{D}} \mathrm{Q}(\mathrm{D}, \mathrm{A}) = \mathrm{E}_{\mathrm{i} \sim \mathrm{U}_{\mathrm{w}}}[\ln (\mathrm{D}(\mathrm{i}))] + \mathrm{E}_{\mathrm{k} \sim \mathrm{U}_{\mathrm{k}}}[\ln (1 - \mathrm{D}(\mathrm{A}(\mathrm{k})))] \tag{1}$$

where $Q(D, A)$ is the loss function? U_w is the true data distribution. U_k is the generated data distribution. $A(k)$ denotes the falsified sample generated by A through the input noise k . $D(i)$ denotes the probability that D determines i to be the true data. $D(i)$ and $A(k)$ alternately maximize and minimize the loss function, and finally find the generative model with the approximate optimal solution.

2.2. Stacked Denoising Autoencoders (SDAE). For the current data with high dimensionality and complexity, this paper uses self-encoding networks for feature extraction of data. The learning goal of AE is to make the output data vector $\boldsymbol{i}'$ equal to the input data vector $\boldsymbol{i}$ with the maximum similar reconstruction.

The basic process of DAE is to add noise to the original data $\boldsymbol{i}$, convert $\boldsymbol{i}$ to $\boldsymbol{i}'$ using the random mapping function, and then use the encoding function f to obtain the encoded features from $\boldsymbol{i}'$ containing the noise data.

$$\mathrm{j} = \mathrm{f}(\mathbf{Wi} + \mathrm{u}) \tag{2}$$

Where $\boldsymbol{W}$ is the weight matrix. $\boldsymbol{u}$ is the deviation vector. Then the decoded data is obtained by the decoding function a.

$$\tilde{\mathbf{i}} = \mathrm{a}\left(\mathbf{W}^{\mathrm{T}}\mathbf{j} + \mathbf{v}\right) \tag{3}$$

Here f and a are the activation functions, which are set as sigmoid functions. $\boldsymbol{j}$ represents the hidden layer vector. $\boldsymbol{v}$ represents the inverse of the deviation vector. The parameters of the DAE are adjusted by optimizing the reconstruction error $J(\boldsymbol{i},\boldsymbol{i}^{\sim})$, and the optimal parameters using the gradient descent method are

$$\theta, \theta' = \arg \min_{\theta,\theta'} \mathrm{J}(\mathrm{i}, \mathrm{a}_{\theta'}(\mathrm{f}_{\theta}(\mathrm{i}))) \tag{4}$$

The hidden layer parameter $W_{(1)}$ is saved as the input to the next layer of DAEs for layer-by-layer extraction of anomalous features. When multiple noise reduction self-encoders are cascaded up and down to form a stack structure, the coding vectors of each DAE are then combined to form an N-layer neural network. The DAE process is repeatedly iterated layer by layer until the model reaches the final output layer, and the deep structured stacked noise-reducing self-encoder SDAE is obtained.

The training process of SDAE [16]: In the first stage, the unsupervised layer-by-layer training parameters, each implicit layer is the feature extracted layer by layer for each DAE pre-training process. In the second stage, the parameters of the whole stack structure are adjusted to obtain the optimal solution of the model.

2.3. Random Forest. Random Forest, which can be thought of as a collection of multiple decision trees, is an integrated learning algorithm proposed by Breiman in 2001 [17]. The random forest uses Bagging algorithm to randomly sample the original data set and obtain each subset of data with the same number but different from each other. First, N decision trees correspond to N subsets of data, and the best division among m attributes of M feature variables is selected by the principle of exponential minimum. The CART algorithm is used for attribute classification of nodes. For the data set D, the smaller the Gini index $Gini(D) = 1 - \sum_{z=1}^{z}(|C_z/D|)^2$, the smaller the probability that the selected samples in the data set are misclassified. The number of selected attributes is a random feature variable, and many decision trees are trained sequentially, and then all the decision trees are formed into a forest.

The classification result of random forest is generated after considering all the decision tree results in a combined vote, and the final classification of the sample is the category with the highest ratio of votes to the total number of votes received. When classifying data, the random forest algorithm has good noise immunity, can handle missing values and outliers, and has better robustness due to the introduction of partial randomness. However, the random forest method alone cannot determine the minimum correlation of data samples and the balanced distribution of a few classes of data in the face of high and unbalanced data sets, resulting in low classification accuracy. Therefore, the imbalance expansion and feature dimensionality reduction are needed before training the random forest classifier.

3. The Model in This Paper

3.1. General System Architecture. With the development of the international environment, the number of learners of spoken English in China has increased greatly, and it is crucial to provide learners with a scientific and systematic way to correct errors in speech. In this paper, the construction of the error detection model is pronounced in the following steps.

Step 1: Pre-processing. The preprocessing part includes forced text-to-speech alignment and phoneme separation of the speech data.

(1) The data obtained from the speech corpus are whole-sentence audio data files. The audio files are force-aligned to the reference text using Hidden Markov Model Toolkit (HTK), which aligns speech to sentence, word, and phoneme level

(2) Get the phoneme level alignment time information by forcing alignment. Cut and separate the phonemes according to their alignment time information to obtain phoneme data

Step 2: Acoustic feature extraction. The phoneme data obtained in the first step are extracted by MFCC acoustic features. In this paper, a total of 13-dimensional MFCC plus 13-dimensional first-order differential and 13-dimensional second-order differential coefficients are extracted to form a total of 39-dimensional MFCC coefficients.

Step 3: Data set pre-processing. This part mainly includes dividing the acquired feature data set into training data set and test data set, and normalizing them, respectively. The normalization operation has no effect on the original distribution of the data. In this paper, a linear function transformation normalization method is chosen.

Step 4: The 39-dimensional MFCC feature vector of the training dataset is used as input to train the improved random forest model. The default initial model parameters, generally the default parameters of random forest can get good classification accuracy, but this paper still choose cross-validation to tune the parameters.

Step 5: Test the pronunciation error detection classification model built by the algorithm using the test set of pronunciation error detection feature data. The accuracy of pronunciation error classification detection is obtained. The evaluation metrics are used to further determine the optimal model. The evaluation metrics used are: accuracy, recall rate and false alarm rate.

3.2. Corpus Construction. The data for the corpus constructed in this paper was obtained from the website http://accent.gmu.edu (hereafter collectively referred to as the English accent website). The website mainly showcases the English accent archives they have constructed. The English accent website collects spoken English from speakers of various linguistic backgrounds. The phonetic data collection and analysis was done by Steven H, a linguist, with the help of many research institutions (George Mason University, etc.) and linguistic researchers, who wanted the collection to include as many native languages as possible from all backgrounds of the world. The English accent website was created to unify the large number of speech sounds from various language backgrounds for use in language teaching and English accent research. To date, the English accent website has collected 2937 English accent audio data, which were obtained by reading the same English passage to 2937 respondents from 386 countries and regions around the world, such as Guangdong, China, Denmark, Bangladesh, France, and others. The website also gives information about each respondent's age, gender, years of English learning, and English learning style, combined with the English pronunciation theory and the analysis of the information of L2 pronunciation errors marked by many linguists in the English accent website. In this paper, L2 English pronunciation errors are counted.

Consonant pronunciation is more complex than vowel pronunciation, and the three main factors that determine the characteristics of consonant pronunciation are position, manner and vocal fold vibration. According to the information of the English accent website, the linguistic experts have found that the errors of the consonants are: the clear consonants have been cleared, appearing at the beginning of words like "p", "t", "k", etc., and the process of airflow explosion is missing when pronouncing. Comprehensive pronunciation theory and English accent website, consonant error types are shown in Table 1.

English accent sites are well suited for training English pronunciation error correction models. Since it is used for training models, a very large number of data samples need to be collected. If the audio is downloaded manually, the workload and time cost are huge, and the downloaded audio needs to have a matching tag file. It is impractical to create tag files manually based on the annotation information given on the web page, and the tag files are often faulty, which affects the performance of the model. Therefore, in this paper, crawler technology is employed to obtain audio and web page information and organize tag files in batch. In addition, website labeling error text labels are not easy to obtain, and the linux virtual servers (LVS) technique in the deep web crawler system is used to obtain the label text by combining the domain knowledge of error pronunciation. This method greatly improves the efficiency compared to manual methods.

This article uses python to simulate a user clicking on a web page and getting the data returned by the server. Using the get method of Python's requests library generates a requests object and a response object, and the requests object encapsulates the client-side data and sends it to the server. Other parameters such as handle and cookie can use the method's defaults. Requests are the return value of the get method, meaning that the request sent by the get method gets a response from the server, and the server passes the returned resources encapsulated in the response object to the get method. The get method in Python's requests library is the core tool for this article to get the English accent website.

3.3. MFCC Acoustic Feature Extraction. The purpose of acoustic feature extraction is to reduce the dimensionality of the raw data and to obtain more characterizing features. In the field of speech signals, acoustic feature extraction is the basis of subsequent research and an important factor in ensuring the accuracy of the results. MFCC was developed based on the use of human auditory models. It is by taking advantage of this feature that MFCC is more stable than linear-based prediction of cepstral coefficients and maintains a good performance even when the signal-to-noise ratio is reduced. In this paper, the MFCC extraction process is studied as follows.

Step1: Pre-emphasis framing. First, a high-pass filter is used to pre-emphasize the signal, considering that audio signals are quasi-smooth signals. That is, when N samples are taken as observation units, they remain relatively smooth and static for a given time of about 15-20 ms, and the time length information of such N samples is the framing of the audio signal. However, the frame size should not be too small or too large, otherwise the spectrum distributed over different time windows on the time axis cannot be obtained. In order to make a smooth transition between frames, instead of a large variation, an overlapping period is allowed to accompany the frames, which is generally taken as one third of the frame length. In this paper, the sampling frequency of the speech signal is 16khz and the frame size is 512 samples, which corresponds to a frame duration of $512/16000*1000 = 32$ ms. The duration of the overlapping area is 11 ms.

Step2: Add windows and multiply each frame with a Hamming window. Adding windows helps to minimize the discontinuity of the frame signal. Even though the signal is smoother and more stable, using the Hamming window to smooth the signal can attenuate the partials and reduce the spectral leakage in the subsequent fast Fourier transform step (FFT). The window function used in this paper is.

$$\mathrm{M}(\mathrm{t}, \mathrm{g}) = (1 - \mathrm{g}) - \mathrm{g} \times \cos \left[\frac{2\pi t}{\mathrm{T} - 1}\right], 0 \leq \mathrm{t} \leq \mathrm{T} - 1 \qquad (5)$$

T is the number of frames. Different coefficients g will have different Hamming windows. In the text, g is taken as 0.46.

Step3: Fast Fourier Transform. In the time domain, the transform of an audio signal generally does not show the characteristics carried by the signal. The opposite is true for the energy spectrum in the frequency domain, which can be used to show different audio characteristics by the level of energy in the frequency domain. In addition to multiplying each frame by a Hamming window, a Fast Fourier Transform

Table 1: Types of consonant errors.

Error type	Specific description
Non_aspiration	Clear consonants such as “p”, “k”, and “p” appear at the beginning of words and lack the process of airflow when pronouncing them.
Final_ devoicing	The vocal folds do not vibrate when the turbid consonants are pronounced, resulting in incorrect clearing of the turbid consonants
Interdental fricative to_stop	Friction sounds are pronounced as bursting sounds
Interdental_fricative to labial fricative	Interdental friction develops into interlipolar friction
Interdental fricative to alveolar fricative	Interdental friction develops into gingival friction
Stop to fricative	Blast sounds are pronounced incorrectly as fricatives
R_to_trill	The curly tongue “r” has a vibrato
W_to_labial_fricative	The “w” creates a friction between the lips
H_to_velar-fricative	“h” produces soft palate friction

is performed on each frame, aiming to convert the time domain signal to the frequency domain signal.

Step4: Mel scale conversion. The power spectrum of each frame is multiplied by the Mel filter bank, which has m delta bandpass filters. In this paper, m is taken as 20. The first filter is narrower, and as the frequency increases, the filter size becomes wider. The use of triangular bandpass filters not only smoothes the spectrum by removing harmonics, but also reduces the complexity of the operation. The conversion of Mel scale frequency to normal frequency is shown in formula (6).

$$\mathrm{Mel}(f) = 2595 \times \lg\left(1 + \frac{f}{700}\right) \tag{6}$$

Step5: Calculate the log spectrum energy. The logarithmic energy of the Mel spectrum obtained by the Mel filter set above is taken to get the logarithmic spectrum. The log spectrum improves the noise immunity of the extracted features and the stability of the spectrum error.

Step6: Discrete cosine transform (DCT). The discrete cosine transform is applied to the above logarithmic spectral energy to find the 13th order MFCC coefficient.

Step7: Find the dynamic differential parameters. Since the speech signal is continuous in the time domain, the MFCC is extracted from the frame to reflect the static feature information contained in this frame. By differencing the 13-dimensional static MFCC to increase the dimensionality of the feature information in the front and back frames, the dynamic and static features are combined to make the features better reflect the speech continuity. In this paper, the 13-dimensional first-order difference of the 13-dimensional MFCC and the 13-dimensional second-order difference of the 13-dimensional MFCC are used as the input feature vectors of the model.

3.4. Data Pre-Processing. The collected speech data were separated into phonemes, extracted acoustic features and processed into a feature dataset, and then divided into 7 groups based on manual annotation by phonetic experts. These represent seven types of pronunciation, namely correct, raising, lowing, fronting, backing, lengthing and shorting, the last six of which represent different types of errors. During the study, it was found that most of the learners' pronunciation errors were concentrated in a few common error types, and the sample size of the common error types was large, while some error types were only present in a small number of learners' pronunciations. The sample sizes of not all pronunciation error types remained balanced, but were normally distributed.

Since the sample data of fronting, backing and lengthing error types were small, three types of error types, namely, raising, lowing, and shorting, were selected for the experiment. Each type of error sample contains 32000 training samples. In addition, 32000 samples with standard pronunciation were selected, and all samples were divided into training and test sets, with three-fourths of the training data set consisting of 24000 samples per set and the remaining one-fourth of the test set consisting of 8000 samples per set. The model was regrouped in the cross-validation session and repeated four times to cross-validate the model.

The data set is normalized separately using the linear function conversion normalization method. The purpose of normalization is to restrict the automatic pronunciation error detection feature data to a certain range and to reduce the variability of the data by reducing the dispersion of the automatic pronunciation error detection feature data, so that the fluctuation of the data is limited to a certain range. The normalization operation has no effect on the original distribution of the data. The linear function conversion normalization is shown in Formula (7).

$$\mathrm{h} = \frac{\mathrm{g} - \mathrm{g}_{\min}}{\mathrm{g}_{\max} - \mathrm{g}_{\min}} \tag{7}$$

where g is the MFCC feature value before normalization. h is the MFCC feature value after normalization. $\mathrm{g}_{\min}$ is the minimum value among the MFCC features. $\mathrm{g}_{\max}$ is the maximum value among the MFCC features.

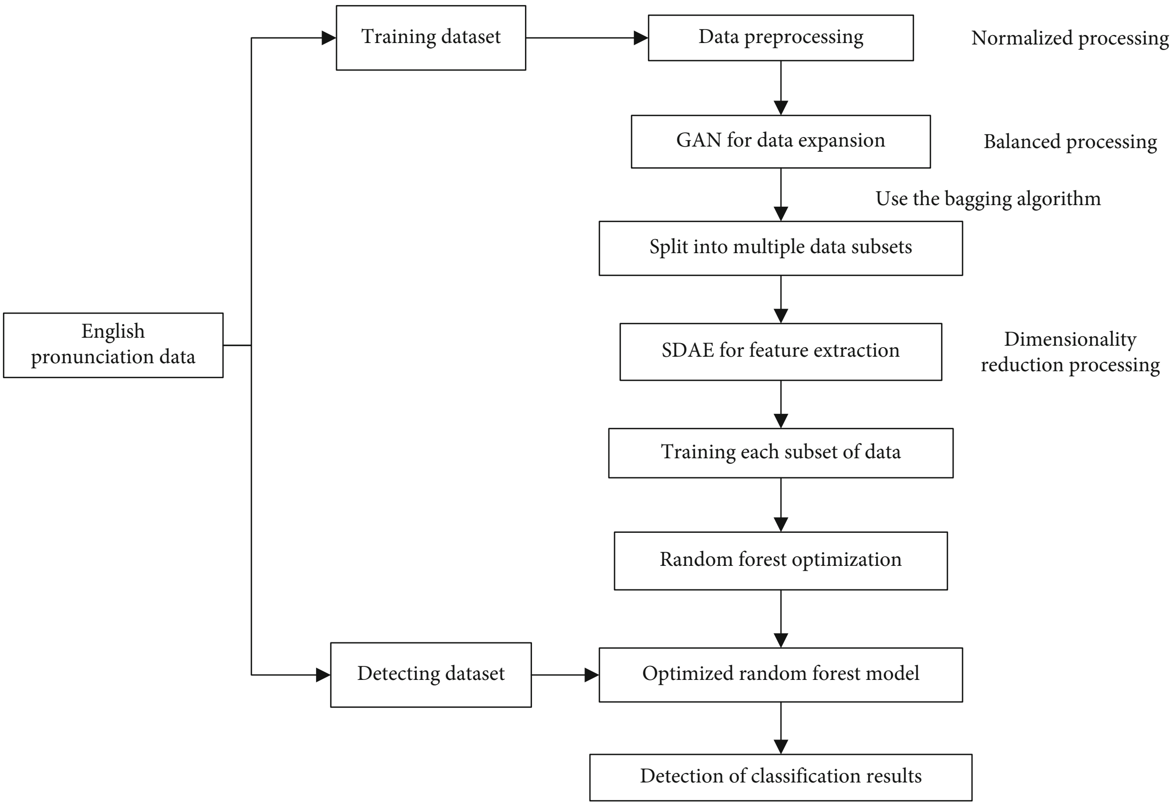

English pronunciation data
Training dataset
Data preprocessing
Normalized processing
GAN for data expansion
Balanced processing
Use the bagging algorithm
Split into multiple data subsets
SDAE for feature extraction
Dimensionality reduction processing
Training each subset of data
Random forest optimization
Detecting dataset
Optimized random forest model
Detection of classification results

Figure 1: Framework of English pronunciation detection model based on GAN-SDAE-RF.

3.5. Improving Random Forest Algorithms. GAN is a generative model that uses an adversarial approach to learn the distribution of real samples. The model can generate new samples with high quality without pre-modeling. In the process of pronunciation detection, the data set used for detection is unbalanced due to the small amount of anomalous data in certain classes, thus, this paper uses GAN to generate a few classes of training samples to reduce the impact of unbalanced training samples on detection accuracy.

SDAE is a deep learning method that contains input layers, N implicit layers and output layers. SDAE takes DAE as the basic unit and stacks them sequentially layer by layer to form a deep network structure, which has deep feature extraction ability and enhances the generalization ability of the model by using the noise reduction property of DAE. SDAE can downscale the high-dimensional data to the maximum extent possible to get the most characteristic data and get the reconstructed the original data can be more easily learned by random forest.

Considering the characteristics of data with high dimensionality and imbalance, we combine the advantages of GAN and SDAE to improve the classification accuracy of the random forest algorithm by addressing the shortcomings of the traditional random forest algorithm to the pronounced data. After using GAN to generate minority samples, we combine the generated minority samples and the original dataset to form a new dataset, which is then sampled by the Bagging algorithm to produce multiple subsets with a balanced sample distribution. Each subset is then feature-dimensioned using SDAE. Each reduced-dimensional data sample corresponds to each decision tree and is trained. In the detection phase, the GAN-SDAE-RF-based English pronunciation detection model is constructed by combining the classification results of each decision tree for voting and finally pooling all the decision trees to form a forest and derive the classification results. The overall framework is shown in Figure 1.

3.5.1. Minority Class Training Data Expansion. Data generation for fewer pronunciation types in the training data using GAN generation adversarial networks is performed by intra-category expansion in the following steps.

Step 1 Firstly, multiple real data sets including few wrong pronunciation types are separated separately.

Step 2 The 128 data is converted into a 12×12 matrix vector according to the GAN model input format, with the remaining 16dimension0 complemented.

Step 3 Given a 144-dimensional noise s with values in the range [-1, 1] of the generated model, the generated fake data is mixed with the separated real data and the discriminator is trained.

Step 4 The discriminative model is trained according to the set number of iterations until the discriminative result is optimal. At this point, the parameters of the discriminant model are fixed and the discriminant results are fed back to the generation model.

Step 5 The training iterations of the generative model are performed according to the set number of iterations until the worst discriminative result is obtained. At this point, the parameters of the generative model are fixed and the process is iterated until the GAN model is balanced.

Step 6 The generated minority class data are supplemented with the original data as expanded samples, and the expanded samples are reorganized into 144-dimensional features, and the front-dimensional128 data are taken as the expanded samples to obtain a balanced training dataset.

3.5.2. SDAE Training Process. The expanded data is subjected to feature extraction using SDAE. The training process is.

Step 1 Construct the first DAE. set each rule $\theta_y : i_1, i_2, \cdots, i_t$ is the hidden layer neuron of the object network, and $i_1, i_2, \cdots, i_t$ is the set of input layer neurons.

Step 2 Determine the connection weights $M\theta_y$ between θ_y and $i_1, i_2, \cdots, i_t$. When the input neuron corresponds to the activation element in the rule, then $M=1$, otherwise $M=-1$. The remaining weights that have little relationship with θ_y are set to smaller random values. The neuron bias is set to a random value.

Step 3 Train the network using the back propagation algorithm and update the connection weights.

Step 4 Repeat steps 1 ~ 3 for each DAE until all DAEs are trained.

3.5.3. Random Forest Generation Process. The GAN model, SDAE feature extraction and random forest algorithm are constructed to parallelize the design. As shown in Figure 2, the whole parallelized design idea is as follows.

Step 1 The dataset captured on the network is first numerically and normalized, and then the GAN model is expanded with a minority class sample from the minority class sample.

Step 2 The few class samples generated by the GAN model with the original data samples are integrated to obtain a new and balanced dataset, which is randomly sampled by the Bagging algorithm to produce several equally distributed subsets of data.

Step 3 Each data subset is subjected to feature extraction by SDAE to obtain the reconstructed new data subset.

Step 4 Each data subset is trained with the corresponding decision tree model according to the decision tree generation method.

Step 5 All the decision trees are aggregated to form an improved forest.

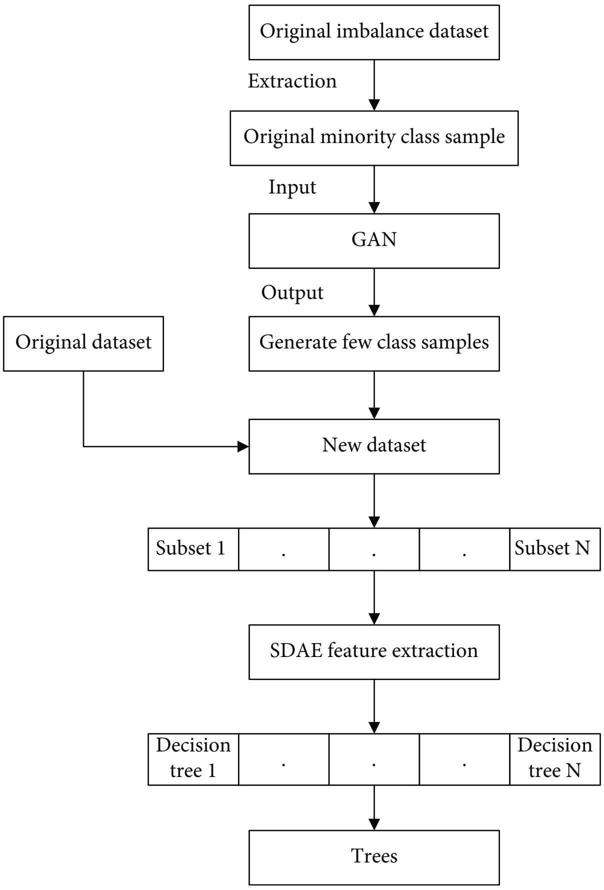


Figure 2: New Random Forest Generation.

Table 2: Configuration of experimental environment.

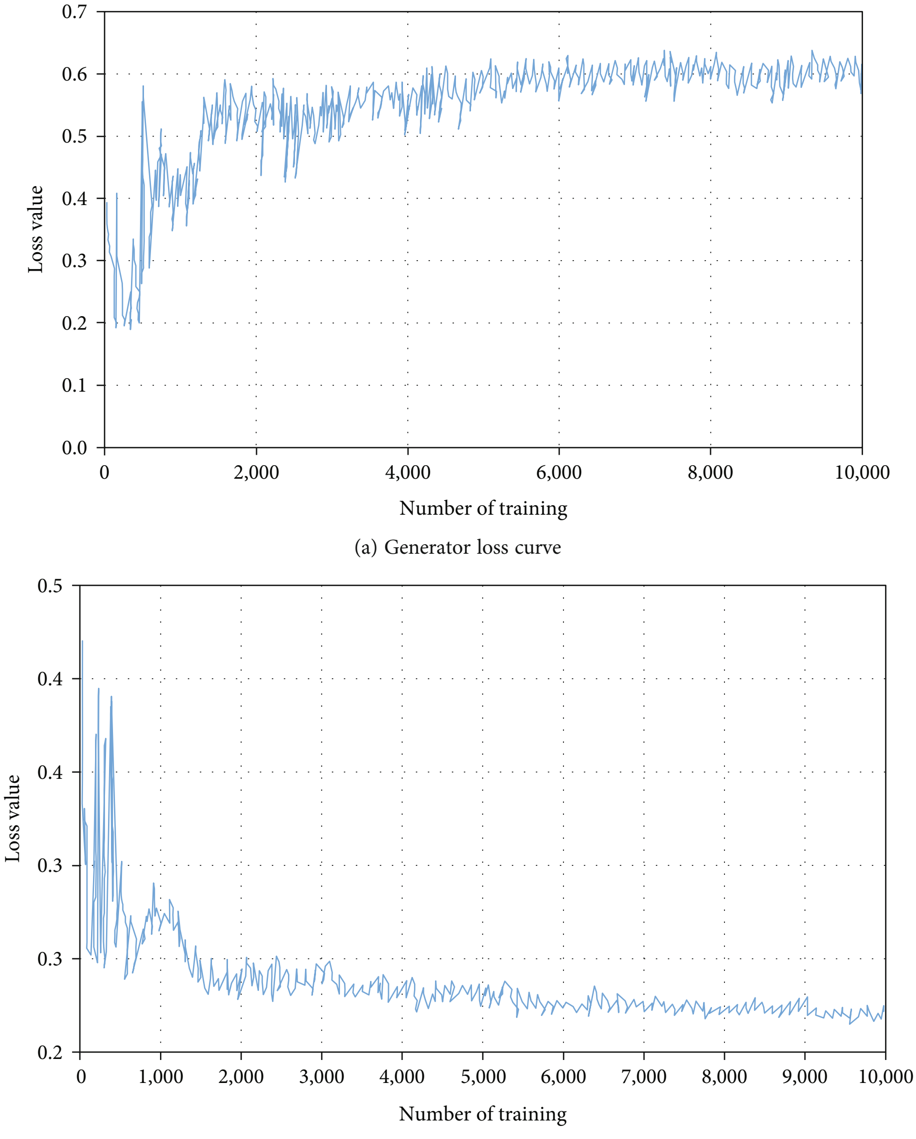
Experimental environment	Environment configuration
Deep learning framework	Keras+ Tensorflow
The operating system	Windows 10
Graphics card	NVIDIA GTX960
CPU	Intel i5-6300HQ
Memory	12G

4. Experimental Data Analysis

In order to validate the deep learning-based intrusion detection model in this paper. The experimental environment configuration of this paper is listed as Table 2.

4.1. Parameter Setting. The initial parameters of the GAN include batch-size set to 50, epoch set to 100. learning rate set to 0.0002. Relu function is selected as the activation function of the model. Adam optimizer is utilized. The final generator loss and discriminator loss variation curves are shown in Figure 3. From the figure, it can be seen that the generator and discriminator losses start to converge when the training times reach about 5500.

The SDAE is a deep learning model whose initial parameters are weights obtained by minimizing the reconstruction errors of the original and reconstructed data through layer-by-layer greedy training. The cross-entropy of the initial parameters is fine-tuned by the BP algorithm to ensure the minimum reconstruction error to obtain the optimal result. The number of nodes in the input layer is consistent with the eigenvalues of the numerically processed data, which is set to 122. The highest accuracy is obtained by comparing different SDAE network structures, which is 122-100 for DAE1, 100-60 for DAE2, 60-30 for DAE3, and 30-5 for DAE4. The batch-size is set to 64 while epoch is set to



(a) Generator loss curve

(b) Discriminator loss curve

Figure 3: Loss curves for generators and discriminator.

3000. After several experiments, the mean square error is chosen as the criterion of reconstruction error, and the MSE stabilizes after 5 training sessions. The number of training times for each layer of DAE and SDAE model is set to 10. The influence of noise ratio on the accuracy is analyzed by experiment. The accuracy of the noise ratio is highest in the interval of [0.2, 0.6], and the mean value is selected by combining the results of multiple experiments, so this setting is 0.4.

The categorical_crossentropy is selected as the loss function, which is specifically used for the multi-categorization problem, and the bach_size is set to 64 and epoch to 10. The model is trained by back propagation using the Adam optimization algorithm.

Through simulation experiments, the overall random forest model performance is synthesized, as shown in Figure 4. The final forest size chosen by the algorithm is 550 trees, with the deepest decision tree 12 and weights of 1、3、1、5、3.5, respectively.

4.2. Analysis of Experimental Results. For decision tree generation algorithms, there are different feature selection measures, such as ID3, C4.5 and CART, etc. In the academic research on decision tree algorithms, there is no definite statement on how to select a decision tree algorithm, therefore, this paper selects the optimal decision tree algorithm via experimental method. The ID3, C4.5 and CART decision tree algorithms are compared in test set validation for classification error detection accuracy and algorithm performance (training time) in cross-validation. The results are showcased in Table 3.

C4.5 Decision tree algorithm has the highest classification error detection accuracy. For the phoneme elongation class of pronunciation errors, there is little difference in the

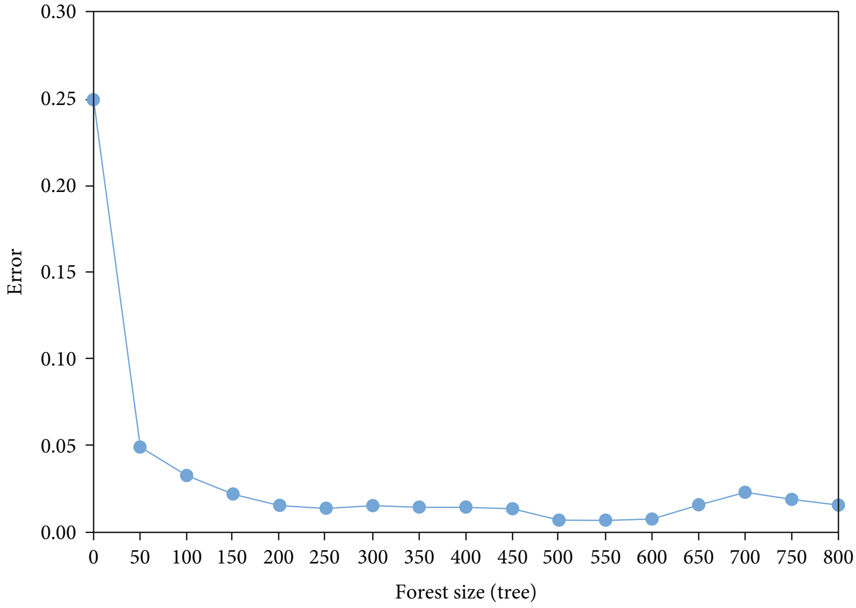


Figure 4: Forest size and algorithm performance impact graph.

Table 3: Comparison of the accuracy and performance of the three algorithms.

Category	Performance	Riseing	Lowing	Shorting
ID3	Accuracy (%)	75.6	78.6	82.1
	Test set error (%)	5.6	6.5	4.5
C4.5	Accuracy (%)	86.7	81.6	82.6
	Test set error (%)	5.1	4.3	3.9
CART	Accuracy (%)	85.1	79.6	82.5
	Test set error (%)	7.9	7.52	6.95

classification error detection accuracy between the CART and ID3 algorithms. For the test set errors, the best performance was achieved by the C4.5 decision tree algorithm. There was no significant difference in the classification accuracy for the three types of pronunciation errors, which was stable between 81% and 87% in each test.

Experimental verification of the error detection performance of the system in English consonants and English vowels is shown in Table 4.

In this paper, the error detection and correction performance of the English pronunciation system for English vowels is significantly higher than that for consonants. The error detection rate of the system for vowels is 84.11%, which is 7.74% higher than that of the system for consonants. The reasons for the higher error detection and correction performance of the system for vowels than for consonants are: Firstly, the pronunciation characteristics of vowels are relatively homogeneous compared with those of consonants, which mainly depend on the position of the tongue in the oral cavity. Besides, the vocal folds, some organs of the oral cavity and the tongue are all involved in the articulation of consonants. Secondly, vowel error types are more comprehensively labeled.

Table 4: Comparison of performance on vowels and consonants.

Type	Precision	Recall	F-value	Error detection accuracy	Error correction accuracy
Consonants	82.17%	82.78%	76.37%	76.37%	61.15%
Vowels	89.72%	86.48%	84.11%	84.11%	75.87%

5. Conclusion

An English pronunciation detection system based on improved random forest is proposed in the paper. Firstly, the process of forced alignment and acoustic feature extraction in data acquisition is explained step by step. Then the corpus applyed in this paper is constructed and acoustic feature extraction of MFCC is performed. Additionally, the random forest algorithm based on GAN-SDAE-RF is designed, and the algorithm selection of English pronunciation error detection system based on improved random forest is described in detail. Finally, the results of model validation from the test set are analyzed to compare the effect of pronunciation error detection in this paper and the comparison between various pronunciation error types. The accuracy and performance of three decision tree algorithms to construct a random forest are also compared and analyzed. The feasibility of the English pronunciation detection system based on the improved random forest is verified from various perspectives. The next step is to further study the English pronunciation error detection system and improve the performance of the system for real-time applications.

Data Availability

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

Conflicts of Interest

The authors declare no competing interests.

Acknowledgments

This study is sponsored by Qingdao Huanghai University.

References

[1] Y. Teimouri, J. Goetze, and L. Plonsky, "Second language anxiety and ACHIEVEMENT," *Studies in Second Language Acquisition*, vol. 41, no. 2, pp. 363–387, 2019.

[2] Y. Suzuki, T. Nakata, and R. Dekeyser, "The desirable difficulty framework as a theoretical foundation for optimizing and researching second language practice," *The Modern Language Journal*, vol. 103, no. 3, pp. 713–720, 2019.

[3] E. Kartchava, E. Gatbonton, A. Ammar, and P. Trofimovich, "Oral corrective feedback: pre-service English as a second language teachers' beliefs and practices," *Language Teaching Research*, vol. 24, no. 2, pp. 220–249, 2020.

[4] K. Saito and L. Plonsky, "Effects of second language pronunciation teaching revisited: a proposed measurement framework and meta-analysis," *Language Learning*, vol. 69, no. 3, pp. 652–708, 2019.

[5] C. Troussas, K. Chrysafiadi, and M. Virvou, "An intelligent adaptive fuzzy-based inference system for computer-assisted language learning," *Expert Systems with Applications*, vol. 127, pp. 85–96, 2019.

[6] J. Buendgens-Kosten, "The monolingual problem of computer-assisted language learning," *ReCALL*, vol. 32, no. 3, pp. 307–322, 2020.

[7] P. M. Rogerson-Revell, "Computer-assisted pronunciation training (CAPT): current issues and future directions," *RELC Journal*, vol. 52, no. 1, pp. 189–205, 2021.

[8] C. Tejedor-García, D. Escudero-Mancebo, E. Cámara-Arenas, C. González-Ferreras, and V. Cardeñoso-Payo, "Assessing pronunciation improvement in students of English using a controlled computer-assisted pronunciation tool," *IEEE Transactions on Learning Technologies*, vol. 13, no. 2, pp. 269–282, 2020.

[9] L. Zhang, Z. Zhao, C. Ma et al., "End-to-end automatic pronunciation error detection based on improved hybrid ctc/attention architecture," *Sensors*, vol. 20, no. 7, p. 1809, 2020.

[10] W. K. Leung, X. Liu, and H. Meng, "CNN-RNN-CTC based end-to-end mispronunciation detection and diagnosis," in *2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8132–8136, Brighton, UK, 2019.

[11] R. Thiruvengatanadhan, "GMM and LDA based speech recognition using sonogram," *NOVYI MIR Research Journal*, vol. 6, no. 1, pp. 24–29, 2021.

[12] M. G. Hussain, M. Rahman, B. Sultana, A. Khatun, and S. Al Hasan, "Classification of Bangla Alphabets Phoneme based on Audio Features using MLPC & SVM," in *International Conference on Automation, Control and Mechatronics for Industry*, pp. 1–5, Rajshahi, Bangladesh, 2021.

[13] D. Palaz, M. Magimai-Doss, and R. Collobert, "End-to-end acoustic modeling using convolutional neural networks for HMM-based automatic speech recognition," *Speech Communication*, vol. 108, pp. 15–32, 2019.

[14] T. Tanaka, R. Masumura, T. Moriya, T. Oba, and Y. Aono, *A Joint End-to-End and DNN-HMM Hybrid Automatic Speech Recognition System with Transferring Sharable Knowledge [C]//INTERSPEECH*, ISCA, Graz, Austria, 2019.

[15] Y. Cheng, Z. Gan, Y. Li, J. Liu, and J. Gao, *Sequential Attention GAN for Interactive Image Editing [C]//Proceedings of the 28th ACM International Conference on Multimedia*, ACM, Westminster, USA, 2020.

[16] J. Yu, "A selective deep stacked denoising autoencoders ensemble with negative correlation learning for gearbox fault diagnosis," *Computers in Industry*, vol. 108, pp. 62–72, 2019.

[17] Y. Chen, W. Zheng, W. Li, and Y. Huang, "Large group activity security risk assessment and risk early warning based on random forest algorithm," *Pattern Recognition Letters*, vol. 144, pp. 1–5, 2021.