Research Article

Deep Belief Network-Based Multifeature Fusion Music Classification Algorithm and Simulation

Tianzhuo Gong

Music Collage, Harbin Normal University, Harbin 150000, China

Correspondence should be addressed to Tianzhuo Gong; gongtz@hrbnu.edu.cn

Received 1 October 2020; Revised 26 October 2020; Accepted 21 January 2021; Published 30 January 2021

Academic Editor: Wei Wang

Copyright © 2021 Tianzhuo Gong. 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, the multifeature fusion music classification algorithm and its simulation results are studied by deep confidence networks, the multifeature fusion music database is established and preprocessed, and then features are extracted. The simulation is conducted using multifeature fusion music data. The multifeature fusion music preprocessing includes endpoint detection, framing, windowing, and pre-emphasis. In this paper, we extracted the rhythm features, sound quality features, and spectral features, including energy, cross-zero rate, fundamental frequency, harmonic noise ratio, and 12 statistical features, including maximum value, mean value, and linear slope. A total of 384-dimensional statistical features was extracted and compared with the classification ability of different emotional features. The deficiencies of the traditional classification algorithm are first studied, and then by introducing confusion, constructing multilevel classifiers, and tuning each level of the classifier, better recognition rates than traditional primary classification are obtained. This paper introduces label information for supervised training to further improve the features of multifunctional fusion music. Experiments show that this information has excellent performance in multifunctional fusion music recognition. The experiments compare the multilevel classifier with primary classification, and the multilevel classification with the primary classification and the classification performance is improved, and the recognition rate of the multilevel classification algorithm is also improved over the multilevel classification algorithm, proving that the excellent performance with multiple levels of classification.

1. Introduction

Music occupies an important place in human civilization in another form [1, 2]. To this day, people use a variety of instruments, different rhythms, and arrangements to create a variety of music. How to efficiently classify and manage published music has become one of the frontiers of current related disciplines [3–5]. Researchers have proposed various solutions for the study of music retrieval and classification. Music sentiment analysis belongs to the category of music classification management and has become an important tool for music retrieval [6, 7], and the classification based on music sentiment can significantly improve the accuracy of music retrieval [8]. However, early studies only used single-modal data, and there are still some limitations in the accuracy of music sentiment classification [9–11]. In recent years, multimodal data have achieved some results in areas such as event detection, but less research has been conducted on music sentiment classification analysis. In music sentiment analysis, some scholars have used machine learning methods to investigate the conclusion that the lower-order features of music have a better effect on music sentiment classification [12, 13]. In recent years, with the development of big data on the Internet, all walks of life produce a large amount of data every day, the problem of information overload is becoming more and more serious, how to quickly find the information they want from a large amount of data, has become a challenge for people. To solve this problem, some classical personalized recommendation algorithms have been proposed, such as collaborative filtering and context-awareness. Personalized recommendations are a key approach to information retrieval and content discovery in today’s information-rich environments. By combining search and user history, valid information is extracted from them, allowing users to access relevant information efficiently and quickly when faced with large amounts of information. These recommendation algorithms are widely used in the industry. Currently, many major
websites have added a recommendation module to their homepage. However, such systems are still far from perfect and often recommend unsatisfactory results. This is partly because users’ tastes and music needs are highly dependent on multiple factors, but current recommendation systems do not explore multiple factors in depth. Current recommendation systems often focus on the user’s interaction with the item or on item-based descriptions and, by contrast, do not consider more implicit interaction information such as the inner, outer, and situational aspects of the listener [14].

Aceto et al. used a crawler approach to gather information from the Internet and then build a collaborative filtered user-item matrix to recommend music to users, which experimentally proved to be very effective [15]. In the rapidly growing music market, a recommendation system based on collaborative filtering represents a promising solution. However, the cofiltering algorithm has its limitations; when there are more new projects in the music library, its user project matrix will be sparse, which is the so-called cold-start problem of projects. For example, in a mobile environment, using explicit ratings to collect user preferences is often difficult, and user ratings have many missing values. To solve this problem, Möckl et al. proposed a collaborative filtering algorithm based on implicit user feedback, which completes the construction of the model by mining implicit ratings and scale profiles, and experiments on real mobile Internet data show that the proposed method provides better performance than existing collaborative filtering algorithms in mobile environment data [16]. Others have proposed content-based recommendation methods that mine the audio features of the music itself by eliminating the need to use the user’s historical data. Potential vectors are extracted based on the audio features, and then the potential factor training model is used to recommend songs. Ngiam et al. grouped songs and then constructed a song-based probabilistic model, and finally improved the performance of the model using user information and audio content features [17]. Shi et al. improved the performance of the logistic regression model on the ad-lib recommendation task [18]. Calvo-Zaragoza et al. used an optimized XGBoost model to recommend products for users on shopping sites, which achieved the highest ranking in evaluation metrics and faster time efficiency on a real dataset [19].

Recommendation systems are mainly aimed at users who are not experienced enough in dealing with large amounts of information to choose the products or information they need. For example, there are so many different styles of music in a music library, which are updated and added every day; how to recommend the music that a user likes or might like according to his preferences is what a music recommendation system does. With hundreds of millions of registered users, tens of millions of regular visitors every day, hundreds of millions of products in the online system every day, and tens of thousands of products traded every minute on average, how to recommend products to users that they like or that meet their current needs is also a hotspot of research for recommendation systems in such complex data. While brainstorming, the team jumped on an idea: expanding the business to Europe could increase sales by $5 million to $100 million. Extrapolating back, the suggestion came from several assumptions: boldly envision opening new markets, rather than launching new products, expanding existing markets, and raising selling prices. The 5 to 100 million range is a rough calculation that will be refined over time.

Only audio information is used, but not fully combined with the user’s historical behavior information, so the recommended results are not personalized enough and cannot provide effective real-time feedback on user behavior. The tree model is easy to overfit, and modeling using the tree model can extract nonlinear features from the data, but when the sample is too small or the model complexity is too high, the generalization ability of the model will be poor. First, the nonlinear features of the data are explored in-depth to address the problem of weak model personalization. To avoid the overfitting problem of the tree model, I use an architecture that fuses the tree model and the logistic regression model. Then, some deficiencies of this fusion model were further improved. Finally, based on the above algorithm, a personalized music recommendation system based on the fusion model is implemented.

2. Multiple Features of Deep Confidence Networks for Music Classification and Simulation Design

2.1. Improved Deep Confidence Network Algorithm. Deep learning seeks to find the characteristics of network input data by simulating the way human brain neurons process information to achieve an optimal solution. The human brain can be viewed to some extent as a complex circuit structure, consisting of multiple layers of complex circuits stacked on top of each other. The signals generated in the human brain are passed through neurons into different parts of the circuit [20]. The different parts of the circuit are stimulated to trigger different responses, which in turn lead to the next step of judgment. To mimic this mechanism, neural networks use weighted connections to simulate the human brain. If the network structure is simple, the weights can be appropriately assigned based on the learned data and the network is better able to recognize and distinguish these responses. However, as the complexity of the network increases, the connections between the networks become denser, and it becomes correspondingly more difficult to distinguish between them based on weights alone. Also, when training the network, the corresponding model is constructed by reverse feedback. If the structure of the network is simple, the feedback can get a timely response, but if the network structure is complex, the feedback cannot get a timely response or even feedback may be diluted and disappeared.

A deep confidence network is a deep generative neural network composed of a stack of multilayer constrained Boltzmann machines [21–23]. The training of a deep belief network (DBN) is mainly divided into the following steps: firstly, pretrain the DBN to get the local optimal solution. After that, a classifier is added to the output layer to classify the
output features of the DBN. The SoftMax classifier is usually used for deep confidence network feature classification. Labels of data need to be added to the classifier for fine-tuning the network usually using a backpropagation algorithm. The training of the deep confidence network is divided into two main parts: pretraining of the entire deep confidence network and fine-tuning the network parameters. Throughout the training process, the weights, bias vectors, and probability distribution of each node are adjusted to achieve network convergence. After pretraining, the network is fine-tuned by back-propagating the errors across the network [24]. The fine-tuning of the DBN is accomplished by comparing the differences between the raw data and the output data of the network and propagating the differences from the lower layers of the network to the top layers of the network using the backpropagation algorithm. This layer-by-layer greedy training method does not explode the complexity of the algorithm during the pretraining and network fine-tuning process. Therefore, this method is highly reliable and efficient in the training process for many samples, as shown in Figure 1.

The probability value of each sample falling on each label is tested, and the label with the highest probability value is the recognition result. The deep neural network mainly includes a numerical model represented by a stack of denoised self-encoders and a probabilistic model represented by a deep confidence network. These two models are stacked by restricted Boltzmann machines and denoising self-encoders, respectively, and the different functions of the hidden nodes are the essential difference. The hidden nodes in the numerical model exist as actual computational units, while the hidden nodes in the probabilistic model are random variables of the algorithm rather than actual computational nodes. Numerous experiments have proven that discrete models are superior in training discrete data and numerical models are more efficient in training continuous data. Next, a classifier is added at the top of the network to fine-tune the weights of the network using the backpropagation algorithm. After that, the probability of each test sample falling on each label is tested, and the item with the highest probability is the identified result.

The multifeature fusion music signal is sampled. X communication sources to be identified are obtained, each radiation source has N sample signals, and then the kth signal of the i-th radiation source can be represented as n [25].

\[ x^i_k(n) = \sum_{n=0}^{T} x^i_k(n)e^{-(2\pi n/T)}, \quad (n = 0, 1, 2, \ldots, T), \]  

(1)

T is the data length of the sample signal as follows:

\[ x^i_k(w) = \sum_{n=0}^{T} x^i_k(n)e^{-iwn}. \]  

(2)

Calculate the frequency spectrum of the sampled signal as follows:

\[ A^i_k(w) = A^i_k(w_1)A^i_k(w_2)A^i_k(-w_1 - w_2). \]  

(3)

Calculate the rectangular integral spectrum SIB of the sampled signal as follows:

\[ X^i_k(w) = \sum_{j} A^j_k(w_1, w_2). \]  

(4)

The higher the number of paths L selected for integration when calculating the integral spectrum of a signal’s perimeter, the more bispectrality features will be obtained accordingly. As the number of dimensions increases, more redundant information will be entrapped in the feature vector, which may lead to a decrease in the recognition rate of multifeature fusion music; therefore, the selection of a reasonable number of integral path strips has an important impact on the recognition results. Each piece of information in the data set does not contain semantic errors or contradictory data. Each piece of data in the data set accurately represents real-world entities. The data set contains enough data to answer a variety of queries and support a variety of calculations.

The preprocessed rectangular integral spectrum is input into the deep confidence network, and the network parameters are adjusted through extensive training. The specific network parameter adjustment can be divided into the following two parts. The network weight parameters of each hidden layer in the deep confidence network are adjusted by unsupervised learning, and the adjusted hidden layer state is used as the input of the next layer. Then, the expected term is approximated, as follows:

\[ \frac{\partial \ln R(v)}{\partial c_{i_j}} = p(h_i = 1|v^i_{10})v^i_{10} - p(h_i = 1), \]  

(5)

\[ \frac{\partial \ln R(v)}{\partial w_{i,j}} = v^i_{10} - v^i_{1}, \]  

(6)

\[ \frac{\partial \ln R(v)}{\partial w_{i,j}} = p(h_i = 1|v^i_{10})v^i_{1} - p(h_i = 1). \]  

(7)

Each layer of the network is fully trained from the bottom up to obtain the final weight parameters, hidden layer deviations, and visible layer deviations. Secondly, the whole network parameters are adjusted by a supervised backpropagation algorithm. A SoftMax regression classifier is used for target recognition, and the recognition results are output. Two main models of multilayer networks are introduced, which are the numerical model-based denoising self-encoder and the probabilistic model-based constrained Boltzmann machine. These two models form stacked denoising self-encoders and deep confidence networks through multilayer stacking, respectively. Both models can be used for the multifeature fusion music individual identification in this paper, and the use of deep confidence networks is more reasonable. This chapter then introduces the SoftMax classifier for the classification of multifeature fusion music features [26]. Finally, a deep confidence network-based model for multifeature fusion music individual recognition is developed by the above theory.

An autoencoder is a network that learns in an unsupervised state. The network model is a simple neural network that maps input data to output. An autoencoder maps input vector x to output layer y through the intermediate layer z of
the network. The process of representing the input layer with an intermediate layer is called the encoding process. The process of mapping from the intermediate layer to the output layer is called the decoding process. The decoding process is the process of adding noise to the self-encoder. The denoising autoencoder adds noise to the autoencoder, breaks the original input data, and then restores the corrupted data to the original input data so that the restored data are as close as possible to the original data. If the network is still able to reconstruct the original data after adding noise, it proves that the network has better robustness.

2.2. Multifeature Fusion Music Classification Algorithm and Simulation Design. To enhance the user experience and satisfy their quest for a high-quality entertainment lifestyle, many music platforms have developed their music recommendation systems. Music recommendation systems are part of a broader category of recommendation systems that filter information to predict user preferences for items. Music recommendation systems can mine listeners’ music preferences and generate user portraits based on listener search and other interactions. When listeners are bored with songs, they have heard many times and want to explore new songs, the music recommendation system digs deeper into listener needs, preferences, and intentions to expand the user’s musical boundaries. For users, the music recommendation system meets their real needs and enhances the music experience. For the platform, the music recommendation system increases the user’s dwell time and activity, introduces greater traffic, and provides a stronger impetus for the platform’s user growth.

The prototype music recommendation system designed in this paper is mainly divided into the client, server, and database. The client is mainly a mobile terminal, which is responsible for interacting with users, obtaining user information, music information, and contextual information, as well as displaying the different services of the platform to users. The client is the terminal closest to the user and is also the entrance for the user to access services, and whether the interface design is beautiful affects the user’s experience. The service side gets the information entered by the client, retrieves the corresponding information from the database, and then performs offline model training or real-time recommendation. The service side is the most critical part, requiring the service state to be stable, and a few seconds of downtime can affect the experience of tens of millions of users. In a real-time prediction system, because the recommendation system needs to use a model to predict user preferences in real-time, and the candidate set of data may be large, so this process can be time-consuming, but making users wait for a long time can seriously affect the user experience, so rapid response to user service requests is a necessary factor to meet the real-time recommendation system. The database stores user data, song data, and interaction behavior, and is responsible for providing data support for large-scale computing. The recommendation engine system is divided into an offline model system and a real-time prediction system. The offline model system performs offline calculations and model training after obtaining data and periodically updates the model iteratively. When the real-time prediction system receives a client request, it uses the existing model to predict the user’s song preferences in the candidate set and then pushes the user’s favorite songs to the user terminal in real-time to complete the closed-loop recommendation. The system design architecture is shown in Figure 2.

The client is the platform entrance to interact with the user; the user’s browsing, listening, liking, favorite, and other behaviors are generated on the client; when the user enters the personalized recommendation module, the client
requests recommendation services from the server; the server’s real-time prediction system on the candidate music set will match the user and the candidate song one-to-one, extracting the corresponding user and music features from the database, using the higher-order multi-information dimensionality reduction. The model makes preference prediction, generates a list of candidate recommendations, and then, after fine sorting, selects the top-N final recommendation list and pushes it to the client. The offline model system on the server-side periodically extracts data from the database to update the model training and then selects the optimal model for the real-time prediction system based on the performance evaluation.

The recommendation engine module is the core module in the prototype system, which mainly includes the offline system and the real-time system. In the offline system, there are steps such as data cleaning, feature computation, and model training. The data recorded by the logging system is raw, and the offline system extracts these data and filters and cleans it according to business rules or policies to obtain denoised data. To facilitate model training, this paper carries out a series of feature engineering to generate more complex and effective cross-feature data, which is imported into the database table after completion. The model is periodically trained and optimized based on online business metrics such as click-through rate and retention rate. The offline system does not require high responsiveness but requires large-scale data processing capabilities. The platform generates increased data every day, and the dimensions of the data are getting higher and higher, so to improve the efficiency of the use of data, many companies have built their big data offline platform. Real-time systems are a key part of the recommendation process to reach users. After the server receives the user’s recommendation request, the system generates the user’s candidate music set through certain rules such as collaborative filtering. The rules are used to finely sort the list, and then, the top-N songs with the highest user preference are pushed to the client terminal as the final top recommendation list. The real-time system consists of multiple sorting processes and requires a big data platform with low time consumption to reduce the user’s waiting time.

2.3. Classification and Simulation Performance Indicator Analysis. This paper uses the public dataset of a music website to study the cyclic neural network consumption model, and a statistical approach is used to describe the dataset to facilitate an intuitive understanding of the dataset. The public dataset is used to organize the structure of the data, store the valid fields and information, extract the fields we need and their data, and then divide the data into training test sets. The next step is to experimentally find out the reasonable length of the long sequences, as the memory length of a recurrent neural network is limited. The last 10% of the obtained long sequences are then used as short sequences to match the popular music with the long sequences to obtain the popular sequences. We keep items that have been consumed at least 10 times in the dataset, i.e., we keep the playback records of those artists that the user has listened to more than 10 times. We then sort the data for each user in chronological order, as our cyclic neural network consumption behaviour prediction model is particularly effective for the analysis of temporal data. Also, we limit the maximum sequence length to 1000 because we do not find any qualitative differences in the realization of the effect.
beyond this length, as shown in Figure 3, where the errors of the standard RNN and GRU-based consumer behavior prediction models are minimal at a maximum sequence length of 1000.

Each long-term behavior sequence of the user consists of a chronological log of 1000 items of music played in a chronological order. The short sequence consists of the last 10% of the above long-term behavior sequence since the last 10% better reflects the user’s behavior habits and behavior patterns in the short term currently than the first 10%. To ensure that the length of the popular sequence is not too small, we rank the popular music according to its popularity and match it with the long-term sequence until the length of the popular sequence reaches 600 because the number 1000 is too time-consuming and laborious to collect data for such a match, but 600 is not too long. The data records are not too small to affect the prediction error of the model.

There are two commonly used methods for multifeature fusion: simple fusion and decision fusion. Simple fusion refers to the fusion of different features into a feature vector, while decision fusion refers to the creation of a single feature-based classifier on each feature, and then adopting a certain strategy to fuse the different single features. Since the human brain is a typical chaotic system, the signal is a nonstationary signal of change, and by simply fusing different features into a feature vector, the different feature indicators will affect each other, resulting in partial loss of the nonlinear features of the signal. Multifeature decision fusion is to establish a base classifier on a single feature and then select the classification result according to a certain strategy; this method can effectively use the information of different features and maintain the stability of the classification of different features, and the accuracy of recognition is expected to be improved compared with simple fusion. The single-feature characterization patterns based on approximate entropy features and marginal spectral features, respectively, have been established with some results, but different features are parsimoniously reconstructed from their perspectives on geometric signals, and these single-feature classifiers may show different advantages on different experimenters, so the feature fusion method will improve the classification performance of signals in single-person feature extraction and recognition. Considering that simple feature fusion can lead to the interaction between different features, resulting in the loss of nonlinear structure of the signal, this paper designs a decision fusion method based on signal approximate entropy and marginal spectral features to establish single feature sub-classifiers, respectively, and then select the classification results according to a certain strategy, which can maintain the advantages of different features and is expected to improve the classification accuracy of basic geometry signals.

When designing a deep confidence network, many of the parameters of the deep confidence network rely on the designer’s own experience in selecting the parameters based on previous models of this type. A portion of the network parameters in this paper was obtained based on experimental results, while the remaining key parameters were set based on previous network models that were specific to this experiment. The learning rate is divided into the pretraining learning rate and the learning rate during fine-tuning, which is often referred to as the learning rate for unsupervised learning. The learning rate when fine-tuned is the supervised learning rate, and these two learning rates are DBNs of different training stages, which usually take different values. However, they represent the same meaning, i.e., the step size of each gradient iteration. The momentum is the weight of the last update that is preserved when the weights are updated so that when training the network using stochastic gradient descent, the direction of each update of the network gradient can form an angle with the direction of the last gradient update to avoid the system falling into a locally optimal solution. According to experience, the value of momentum is 0.5. The difference in the network structure mainly refers to the difference in the number of hidden layers and the number of hidden layer neurons in the deep confidence network, and there are differences in feature extraction and characterization of the same data with a different number of hidden layers.

3. Analysis of Results

3.1. Analysis of Multifeature Fusion Music Classification and Simulation Results. The simulations are used to analyze the differences in the recognition rate of individual features of the radiation source when the number of data is different in the deep confidence network-based multifeature fusion music mutual modulation interference algorithm. The simulation mainly compares the extraction of different intermodulation component features to reflect individual differences under different numbers of training samples. The simulation model is a DBN network composed of three RBMs, and the number of neurons in the visible layer and three hidden layers of the deep confidence network are 1024, 512, 256, and 128, respectively. To improve the accuracy of the experiment, the experiment was repeated 10 times and the average recognition rate was taken as the result.

From the five simulated signals generated by the simulation, $10^3$ samples were randomly extracted, and $10^3$, $2 \times 10^3$, $3 \times 10^3$, and $4 \times 10^3$ labelled samples were randomly selected from each radiation source signal as the labelled sample set, and then, $3 \times 10^4$ samples were taken from the remaining samples as the unlabelled sample set, followed by $10^3$ samples were selected from each radiation source signal as the experiment was repeated 10 times and the average recognition rate was calculated.

Figure 4 shows the average recognition rate of each method with a different number of training samples. Under different "small sample" conditions, the deep confidence network has the highest average recognition rate of individual features, which means that the proposed method can accurately characterize individual features, followed by the SIB method, and the recognition of R features is the poorest. In the case of sufficient samples, the recognition rate of deep belief network method exceeds 80%, which means that they can accurately characterize the characteristics of different signal sources in small samples. The recognition performance of each algorithm is significantly improved in the case.
of sufficient samples, and the proposed algorithm can still reach the optimal recognition performance, which shows that the deep confidence network can meet the requirements of individual recognition of communication radiation sources in both large and small samples.

The simulation mainly compares the recognition rate of the individual recognition model of the deep confidence network when the phase noise of the carrier of the communication radiation source is used as the basis of recognition. There is no difference in the frequency deviation of the carrier in the simulation, but there is a difference in its phase noise. A rectangular integral bispectrality transform is performed on the sampled signal to obtain the preprocessed data. In each set of simulations, $2 \times 10^4$ randomly extracted from 5 radiation source signals were used as training samples, and then, 100 samples from each radiation source signal were selected as test sets. The experiments were repeated five times. The recognition rates of the five experiments are expressed as recognition rate 1, recognition rate 2, recognition rate 3, recognition rate 4, and recognition rate 5. The specific simulation results are shown in Figure 5.

From Figure 5, the average recognition rate of the signals is 55.43% when there is only a slight difference in phase noise. It has the smallest phase difference in the middle and the smallest phase difference variance between each signal, so it is more difficult to identify than other signals. 5 groups of experiments have roughly the same recognition rate, indicating that the stability of this method is high. However, the simulation results show that the average recognition rate of the signal is 55.43%. In general, it is more difficult to recognize the carrier frequency difference of the communication radiation source, which is not used as the fine features of the communication radiation source.

Among them, the learning rate and the depth of the tree parameters are more important and are related to the model’s generalization ability. In this paper, we observe the effect of the learning rate and the maximum depth of the tree on the AUC of the decision tree. The results are shown in Figure 6.

From Figure 6, we can see that the logistic regression model has improved after constructing the feature project, with the accuracy ACC reaching 0.6938 and AUC reaching 0.7268. However, the performance of the logistic regression and a single model is still not as significant as the two fusion models. This shows that the manual feature extraction is not only inefficient but also the nonlinear features are not rich enough. The boosted tree model can automatically extract
nonlinear cross features, and from the results, these features improve the learning of the logistic regression model greatly and significantly reduce the model bias. In terms of the fusion model, the multi-information downscaling fusion algorithm is also slightly improved over the low-order fusion algorithm, with an ACC of 0.7624 and an AUC of 0.8087.

This paper looks at the performance of the high-order boosting tree XGBoost and the low-order boosting tree GBDT on the test set, as shown in Figure 7.

From Figure 7, XGBoost has a lower error on the test set than GBDT for lower-order boost trees. This means that the XGBoost feature extraction layer will be more efficient than GBDT, which to some extent avoids the overfitting problem of GBDT and allows to get a more realistic leaf drop and features of the sample in the dataset reconstruction stage. Among all the models, the ACC and AUC of this paper’s multi-information downscaling fusion algorithm are the best, which proves that the overall performance of the fusion algorithm framework in this paper is improved by using high-order gradient boosting tree XGBoost and multi-information fusion, data-sparse level downscaling processing, and overcoming some of the limitations of the previous low-order fusion algorithm.

3.2. Analysis of Metrics Results. To determine the values of the coefficient $k$ and the number of subsequence data points’ $m$ in the approximate entropy characteristic algorithm and to analyze the effect of $k$ and $m$ on the approximate entropy of different signals, this section firstly conducts a simulation experiment to verify the analysis. After eliminating the interference of irrelevant factors, Figure 8 shows the experimental waveforms of the three simulated signals.

According to the definition of approximate entropy, approximate entropy is a nonlinear dynamics parameter that quantifies the regularity and unpredictability of EEG signals, which uses a non-negative number to represent the complexity of a time series, reflecting the possibility of new information occurring in the time series. As shown in Figure 8 of the waveform of the simulated signal, the random noise has the strongest irregularity among the three signals, the chirp signal has the second strongest irregularity, and the sine signal is the most regular, so the approximate entropy value of the random noise should be the largest, the approximate entropy value of the chirp signal is the second smallest, and the approximate entropy value of the sine signal is the smallest.

Since the similarity tolerance coefficient $k$ and the number of subseries data points’ $m$ are important parameters in the approximation entropy calculation method, they need to be determined before specific experiments. To study the effect of similarity tolerance coefficient $k$ and the number of subsequence data points’ $m$ on the approximate entropy of different signals and to determine the values of $k$ and $m$ applicable in this paper, a comparative analysis of the approximate entropy of three different simulated signals is carried out. Figure 9 shows the variation of approximate entropy with similar threshold coefficients $k$ for the three signals under $m = 2$ and $m = 3$, respectively. Figure 9 shows the variation of the approximate entropy with the coefficient $k$ of the similar threshold under the conditions of $m = 2$ and $m = 3$.
The approximate entropy of random noise is higher than that of chirped signal and sine signal when the number of data points \( m = 2 \) and \( m = 3 \) in the subsequence and the approximate entropy of sine signal is the smallest among the three. The gap between the signal approximate entropy is maximized as much as possible, and the EEG signals treated in this section are complex chaotic signals, so the \( k \) value can be chosen as 0.2 or 0.3 is more appropriate.

Cross-sectionally, for the choice of kernel functions, polynomial kernels always have the lowest emotion recognition rate, while for rhythmic features, RBF kernels have a higher recognition rate than linear kernels, and for MFCC and harmonic noise ratio, linear kernels have a higher recognition rate than RBF kernels. By reviewing the data, the linear kernel is more favourable to handle the linear case, while the SVM based on the RBF kernel can better handle the case after the learning of nonlinear features; in this paper, the strong nonlinear fitting ability of the RBF kernel is needed, and the RBF kernel function is chosen for the next speech emotion experiment in this paper. The recognition rate of emotions on different features and kernel functions is shown in Figure 10.

Among the three major types of emotional features, the strongest classification ability is the MFCC feature, that is, the spectral feature, with the highest recognition rate of 54.58\%, followed by the rhythm feature, with the highest recognition rate of 45\%, and the lowest recognition rate of the harmonic noise ratio, with the highest 28.3\%, which is due to the singularity of the harmonic noise ratio voice emotional features; compared to the MFCC and rhythm features, the classification ability is also weak. By fusing the three types of features, the affective speech recognition rate is greatly improved, reaching 62.08\% for the linear kernel and 59.17\% for the RBF kernel.

4. Conclusion

In this paper, a design of a deep confidence network-based multifeature fusion algorithm for music sentiment classification is proposed. The multifeature fusion is based on fundamental tone frequency and band energy distribution; meanwhile, the traditional DBN model is improved to obtain the music sentiment-oriented classification model. From the test results, the model has a high classification accuracy, but there are still some shortcomings. The main reason is that the variety of audio signal features is small, which limits the variety resolution of musical emotions. In the future, other audio signal features and combinations of them will be added to the study of music emotion classification. To address the limitations of single morphological data in music sentiment classification, a multifeature fusion music classification algorithm based on deep confidence networks is proposed in the paper. Firstly, the feature vectors of the music signals are extracted from multiple angles to form multifeature data and fused. At the same time, the
traditional deep confidence network is improved for music emotion classification by adding fine-tuning nodes to enhance the model’s tunability. The training set obtained from the fusion is trained in the improved deep confidence network, and the optimal performance of the model is achieved by adjusting the weights between the visible and hidden layer units in the RBM. The test results show that the highest music sentiment classification result is 82.23%, which can be a good aid for music retrieval.

Data Availability

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

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

References

[1] S. Chatterjee, S. Sarkar, S. Hore et al., “Structural failure classification for reinforced concrete buildings using trained neural network based multi-objective genetic algorithm,” Structural Engineering and Mechanics, vol. 63, no. 4, pp. 429–438, 2017.
[2] D. J. Miller, Z. Xiang, and G. Kesidis, “Adversarial learning targeting deep neural network classification: a comprehensive review of defenses against attacks,” Proceedings of the IEEE, vol. 108, no. 3, pp. 402–433, 2020.
[3] Y.-H. Lai, Y. Tsao, X. Lu et al., “Deep learning-based noise reduction approach to improve speech intelligibility for cochlear implant recipients,” Ear and Hearing, vol. 39, no. 4, pp. 795–809, 2018.
[4] Z. Gojcic, C. Zhou, and A. Wieser, “F2S3: robustified determination of 3D displacement vector fields using deep learning,” Journal of Applied Geodesy, vol. 14, no. 2, pp. 177–189, 2020.
[5] K. A. Anderson, “Skill networks and measures of complex human capital,” Proceedings of the National Academy of Sciences, vol. 114, no. 48, pp. 12720–12724, 2017.
[6] A. Maier, C. Syben, T. Lasser, and C. Riess, “A gentle introduction to deep learning in medical image processing,” Zeitschrift für Medizinische Physik, vol. 29, no. 2, pp. 86–101, 2019.
[7] T. T. Q. Bui, N. T. Thang, and T. H. Le, “A robust PCA-SURE thresholding deep neural network approach for mental task brain computer interface,” Journal of Informatics and Mathematical Sciences, vol. 11, no. 3–4, pp. 383–406, 2019.
[8] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, “Convergence of edge computing and deep learning: a comprehensive survey,” IEEE Communications Surveys & Tutorials, vol. 22, no. 2, pp. 869–904, 2020.
[9] R. Wason, V. Jain, G. S. Narula, and A. Balyan, “Deep understanding of 3-D multimedia information retrieval on social media: implications and challenges,” Iran Journal of Computer Science, vol. 2, no. 2, pp. 101–111, 2019.
[10] R. Haeb-Umbach, S. Watanabe, T. Nakatani et al., “Speech processing for digital home assistants: combining signal processing with deep-learning techniques,” IEEE Signal Processing Magazine, vol. 36, no. 6, pp. 111–124, 2019.
[11] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pesca¨o, “Toward effective mobile encrypted traffic classification through deep learning,” Neurocomputing, vol. 409, pp. 306–315, 2020.
[12] A. Mesaros, T. Heitlola, E. Benetos et al., “Detection and classification of acoustic scenes and events: outcome of the DCASE 2016 challenge,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, no. 2, pp. 379–393, 2018.
[13] D. M. Dimiduk, E. A. Holm, and S. R. Niezgoda, “Perspectives on the impact of machine learning, deep learning, and artificial intelligence on materials, processes, and structures engineering,” Integrating Materials and Manufacturing Innovation, vol. 7, no. 3, pp. 157–172, 2018.
[14] R. Vishwakarma and A. K. Jain, “A survey of DDoS attacking techniques and defence mechanisms in the IoT network,” Telecommunication Systems, vol. 73, no. 1, pp. 3–25, 2020.
[15] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pesca¨o, “Mi-METIC: mobile encrypted traffic classification using multimodal deep learning,” Computer Networks, vol. 165, p. 106944, 2019.
[16] L. Möökl, A. R. Roy, and W. E. Moerner, “Deep learning in single-molecule microscopy: fundamentals, caveats, and recent developments,” Biomedical Optics Express, vol. 11, no. 3, pp. 1633–1661, 2020.
[17] J. Ngiam, A. Khosla, M. Kim, J. Nam, and A. Y. Ng, “Multimodal deep learning,” in Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Washington, DC, USA, June 2011.
[18] L. Shi, J. K. Nielsen, J. R. Jensen, M. A. Little, and M. G. Christensen, “Robust bayesian pitch tracking based on the harmonic model,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 27, no. 11, pp. 1737–1751, 2019.
[19] J. Calvo-Zaragoza, H. Hijic Jr., and A. Pacha, “Understanding optical music recognition,” ACM Computing Surveys, vol. 53, no. 4, pp. 1–35, 2020.
[20] N. F. Fauzi, A. S. M. Jaya, M. I. Jarrah et al., “Thin film roughness optimization in the TiN coatings using genetic algorithms,” Journal of Theoretical and Applied Information Technology, vol. 95, no. 24, pp. 6690–6698, 2017.
[21] X. Guo and N. Ansari, “Localization by fusing a group of fingerprints via multiple antennas in indoor environment,” IEEE Transactions on Vehicular Technology, vol. 66, no. 11, pp. 9904–9915, 2017.
[22] M.-C. Chen, S.-Q. Lu, and Q.-L. Liu, “Global regularity for a 2D model of electro-kinetic fluid in a bounded domain,” Acta Mathematicae Applicatae Sinica, English Series, vol. 34, no. 2, pp. 398–403, 2018.
[23] M. Chen, S. Lu, and Q. Liu, “Uniform regularity for a Keller-Segel-Navier-Stokes system,” Applied Mathematics Letters, vol. 107, p. 106476, 2020.
[24] M. Naderan, E. Namjoo, and S. Mohammadi, “Trust classification in social networks using combined machine learning algorithms and fuzzy logic,” Iranian Journal of Electrical and Electronic Engineering, vol. 15, no. 3, pp. 294–309, 2019.
[25] L. Liu, O. De Vel, Q.-L. Han, J. Zhang, and Y. Xiang, “Detecting and preventing cyber insider threats: a survey,” IEEE Communications Surveys & Tutorials, vol. 20, no. 2, pp. 1397–1417, 2018.
[26] M. Azad-Manjiri, A. Amiri, and A. Saleh Sedghpour, “MLSLTSVM: a new structural least square twin support vector machine for multi-label learning,” Pattern Analysis and Applications, vol. 23, no. 1, pp. 295–308, 2020.