Hindawi
Complexity
Volume 2021, Article ID 4300059, 12 pages
https://doi.org/10.1155/2021/4300059

Research Article

Matching Subsequence Music Retrieval in a Software Integration Environment

Zhencong Li,[1] **Qin Yao**,[1] **and Wanzhi Ma**[2]

[1]*School of Music and Dance, Ningxia Normal University, Guyuan, Ningxia 756000, China*
[2]*Department of Educational and Culture Contents Development, Woosuk University, Jeonju 55338, Republic of Korea*

Correspondence should be addressed to Wanzhi Ma; 997443418@stu.woosuk.ac.kr

Received 22 April 2021; Revised 7 May 2021; Accepted 12 May 2021; Published 24 May 2021

Academic Editor: Zhihan Lv

This paper firstly introduces the basic knowledge of music, proposes the detailed design of a music retrieval system based on the knowledge of music, and analyzes the feature extraction algorithm and matching algorithm by using the features of music. Feature extraction of audio data is the important research of this paper. In this paper, the main melody features, MFCC features, GFCC features, and rhythm features, are extracted from audio data and a feature fusion algorithm is proposed to achieve the fusion of GFCC features and rhythm features to form new features under the processing of principal component analysis (PCA) dimensionality reduction. After learning the main melody features, MFCC features, GFCC features, and rhythm features, based on the property that PCA dimensionality reduction can effectively reduce noise and improve retrieval efficiency, this paper proposes vector fusion by dimensionality reduction of GFCC features and rhythm features. The matching retrieval of audio features is an important task in music retrieval. In this paper, the DTW algorithm is chosen as the main algorithm for retrieving music. The classification retrieval of music is also achieved by the *K*-nearest neighbor algorithm. In this paper, after implementing the research and improvement of algorithms, these algorithms are integrated into the system to achieve audio preprocessing, feature extraction, feature postprocessing, and matching retrieval. This article uses 100 different kinds of MP3 format music as the music library and randomly selects 4 pieces each time, and it tests the system under different system parameters, recording duration, and environmental noise. Through the research of this paper, the efficiency of music retrieval is improved and theoretical support is provided for the design of music retrieval software integration system.

1. Introduction

With the rapid development of computer technology, human civilization began to experience the third wave, and the age of information technology finally arrived. In the social production mainly based on information technology, the amount of information exploded and the variety of contents was colorful [1]. Particularly after the development of mature computer networks, multimedia resources other than text have been freely disseminated. As one of the important types of art in human life, music is naturally a necessary need for people [2]. With the open sharing nature of digital music on the Internet, its dissemination has reached the ultimate. Users need to search for the music they need. Usually, we can complete the search based on the song title, artist, or even lyrics information. This is the search method provided by most music websites and APPs, which is built on the premise that users know some song information and also basically meets their needs [3]. This traditional way of using text information for music search has been developed very maturely and is technically easy to implement, and the search efficiency and accuracy are relatively high, but this way also has its insurmountable shortcomings. Multimedia data is expanding at a massive rate every day, and it is not only costly to manually label the features of these data, but also difficult to complete the labeling work for such huge data. If too little information is manually annotated, it is easy to make the data impossible to retrieve by simple textual information [4].

Original audio fragment retrieval is closer to an exact match, because the user input audio fragment is intercepted or recorded from the original music file, which can be

regarded as a sample of the original music, avoiding the user's involvement in the music subsequence description, and less different from the original music, of course, provided that the user has or can record the information of the original music fragment. For this kind of retrieval task, the method based on music subsequence matching is very suitable [5]. A musical subsequence is a digital signature of music content, which is extracted by processing the music through certain algorithms and can usually represent important acoustic features of the music. The specific process is as follows: firstly, using subsequence extraction algorithm, all the subsequences of music in the music library are calculated separately, and these subsequences are associated with music information and build a subsequence library; then, the subsequences of samples are extracted by the same subsequence extraction algorithm and matched in the subsequence library. Finally, the music with the highest relevance is calculated as the result [6]. To speed up the retrieval, the subsequence library is usually optimized and designed to build an efficient index. Among them, subsequence extraction and matching algorithms are important. The good thing is that there are many mature algorithms, based on which there are more applications with higher accuracy and naturally good usage results [7].

This paper focuses on rhyme-based music retrieval techniques. Music retrieval systems generally include two main techniques, a feature extraction algorithm and a retrieval algorithm. The purpose of this paper is to investigate how to improve the accuracy and efficiency of retrieval through the research and improvement of algorithms. The goal of the research is that users can achieve the music retrieval function through music fragments without requesting aids. Music subsequence is an audio feature extracted from music that can be used as a unique identifier of music, just like a human subsequence, with a very low repetition rate. It is generally required to obtain the time–frequency energy map by Fourier transform with a short plus window and take the peak of time–frequency interval as the subsequence feature. The first section mainly describes the research background and significance of this paper, as well as a brief description of the research structure of this paper. Section 2 examines the current research status and applications as a theoretical basis for the next key sections. Section 3 investigates the preprocessing and feature extraction of music information while proposing an algorithmic model for subserial music retrieval, along with a detailed description of the system designed in this paper. Section 4 demonstrates the effectiveness and feasibility of the research in this paper through the analysis of the algorithm model, the test analysis of the music retrieval system, and the performance analysis of the matched subsequence music retrieval. Section 5 is the conclusion and outlook of this thesis. The next work plan is proposed along with the summary of the research of this thesis.

2. Related Work

The key to music subsequence retrieval is extracting representative features from a given audio query fragment and then comparing them quickly in a specified audio feature library to retrieve the audio with the highest similarity to the audio query fragment. This paper deals with the retrieval of speech and music. Jebbara et al. propose a matching algorithm based on the geometric similarity of pitch contours by first extracting pitches, drawing pitch curves according to time, and later comparing the geometric features of the pitch curves extends idea with the innovation of linearly extending the pitch curves in time before making comparisons, as a way to match. In the system completed, the number of songs in the library is 10,000 songs and the number of tests is 20 music subsequence fragments, and the results of each query can accurately get different versions of the songs [8]. Priya et al. have proposed a method that uses the statistical model Markov to compare melodic similarity, which is extremely different from the more popular approximate string matching and can allow for note omission, musical subsequence rhythmic errors, but is more sensitive to pitch [9]. Palpanas and Beckmann proposed adding rhythmic information to the music retrieval system to assist retrieval, thus improving the accuracy of the system retrieval, and developed a MELEDEX system, which applied the Gold–Rabiner algorithm to extract music subsequence song fundamental frequencies, then performed note slicing, and finally searched songs by matching with the notes in the database [10].

In matching retrieval, dynamic temporal regularization algorithms and various improved algorithms are used on a large scale to detect the relevance of subsequent query speech and speech-based documents in speech libraries. Liu et al. propose an unsupervised framework based on a fragment-based acoustic bagging framework to address the problem of discovering spoken terms in large speech databases. The temporal matching technique of DTW reorders the results and recovers the time series information [11]. The speech data is efficiently stored in a reverse index, which makes the retrieval very fast and thus makes the framework particularly efficient for searching large databases. In the task of speech keyword detection, the algorithm needs to find out the occurrence of specific speech keywords from longer speech audio documents, and thus the starting and ending points of the alignment paths in the dynamic time regularization algorithm are not known in advance. Liu et al. propose a segmented dynamic time regularization algorithm, which can find the speech segment with the most similar pronunciation to the speech keyword in a long speech document by sliding a window of equal length to the sequence of speech keyword feature vectors and calculating the DTW distance between the speech keyword and the speech segment within this window each time [12]. Due to the problem of low efficiency of DTW search performed on a large-scale corpus and the fact that the window to be detected and the query speech keyword are equal in time, this is not the same as the fact that the speaker pronounces the speech multiple times with a fast or slow speech rate [13]. The DTW algorithm based on local normalization of subsequence is proposed in the literature, and its biggest advantage over the subsequence-based DTW algorithm is the concept of local normalization, which allows the dynamic time

regularization algorithm to calculate the value in the distance accumulation matrix as the value of the original calculation of the accumulation distance divided by the length of its current matching path [14].

Audio-based subsequence retrieval technology is still in the exploration and research stage, and although researchers in various countries have carried out a lot of work on it, audio-based subsequence retrieval technology started late and faces a lot of difficulties in the practical field, so there is a lack of commercial audio retrieval systems so far. Although most of the existing audio retrieval systems with good practical performance are based on speech recognition technology, which achieves relatively good retrieval results in a quiet environment, their processing speed is still limited, the complexity of their algorithms is still difficult when dealing with large amounts of speech data, and a large amount of manual annotation is required [15]. In contrast, the theoretical processing speed of a subsequence-based music retrieval system is much faster than that of a speech recognition-based system, making it possible to handle large amounts of Internet audio data. Existing research has achieved certain results for music retrieval, and if the retrieval system for music can achieve better results for speech retrieval at the same time, then it can make speech retrieval and audio retrieval be applied in the same system without the need for two systems and two algorithms [16]. In a real-time system, how to retrieve the information users want quickly is one of the most important problems facing retrieval. Whether it is music or speech, the slow retrieval speed is an inevitable problem in the case of large data volume [17]. It is necessary to retrieve the audio information that users need more quickly while ensuring a certain range of error rate [18].

3. Research on the Matching Subsequence Music Retrieval System

3.1. Subsequence Music Retrieval Feature Processing. Ordinary formats of music are not suitable for direct analysis and processing, such as MP3 which requires signal processing and the use of Mel-Frequency Cepstral Coefficients (MFCC) to obtain the cepstral spectrum, but Musical Instrument Digital Interface (MIDI) as a special format, which is not a realistic recorded audio [19]. To facilitate music feature extraction, MIDI files are used here for music feature analysis. A MIDI file is a set of instructions that belong to a binary file, such as pitch, loudness, and other elements, and usually consists of multiple tracks that can use a thousand times less disk space than the equivalent recorded audio. The standard MIDI format is a representation of music designed to be replayed through electronic instruments [20]. The melody line is the main character in the accompaniment composition. The MIDI elements are constructed as tracks. Usually, each track contains a single track played by a note instrument and the melody line is usually stored in a single track [21].

The master track information is processed to decode the duration, intensity, and pitch of each timestamped note. Using the MIDI parsing file from the previous section, you can see that the note events consist of three 16-bit binary numbers, where 90 represents a note on, and the note duration can be calculated based on the note on to note off. The MIDI note code table is shown in Table 1.

In this section, an audio fingerprint extraction algorithm based on wavelet transform will be introduced, which represents the music information by a language spectrogram and performs a series of calculations such as wavelet transform based on the language spectrogram to obtain a compact audio fingerprint. Most formats of audio files have some header file information added in addition to the bare audio data saved. What the system needs to extract from the fingerprint of audio files is the audio bare data, so the first step of this algorithm is to preprocess the audio files and extract the audio bare data as shown in Figure 1.

Usually, the tone is not periodic, and the music features extracted on its basis are not continuous [22]. To solve this problem, the method of overlapping split I post is often used, i.e., allowing a certain overlap between the previous frame and the next frame. Parton is implemented by weighing the audio signal with a movable window of finite length, multiplying the audio signal AS (m) by some window function $f(m)$ to obtain the windowed audio signal as follows:

$$\mathrm{AS}_f = \mathrm{AS}(m) \cdot f(m). \tag{1}$$

The rectangular window is a time-invariant zero curtain window that has a more concentrated main flap, but often has high-frequency interference, leakage, and negative frequency. Its expression is shown in equation (2). When using a computer to realize engineering test signal processing, it is impossible to measure and calculate an infinitely long signal, but take a limited time segment for analysis. The method is to intercept a time segment from the signal and then use the intercepted signal time segment to perform period extension processing to obtain a virtual infinite signal, and then the signal can be subjected to mathematical processing such as Fourier transform and correlation analysis. After the infinite signal is truncated, its frequency spectrum is distorted, and the energy originally concentrated at $f(0)$ is dispersed into two wider frequency bands.

$$f(m) = \begin{cases} 1, & 0 \le m \le N, \\ 0, & m > N; m < -N, \\ -1, & -N \le m < 0. \end{cases} \tag{2}$$

Wavelet transform is a new transform analysis method, which inherits and develops the idea of localization of short-time Fourier transform, while overcoming the shortcomings such as window size does not change with frequency and can provide a time–frequency window that changes with frequency, which is an ideal tool for signal time–frequency analysis and processing [16]. In this algorithm, we use the Hal wavelet transform to process the subspeech spectrogram. The mother wavelet of the Haar wavelet is as follows:

Table 1: MIDI note code table.

Note code	Binary code	Decimal encoding	Hexadecimal encoding	Scale
do	00000000	0	0x00	−1
ri	00000001	1	0x01	−1
mi	00000010	2	0x02	−1
fa	00000011	3	0x03	−1
so	00000100	4	0x04	−1
la	00000101	5	0x05	−1
xi	000000110	6	0x06	−1
7	000000111	7	0x07	−1

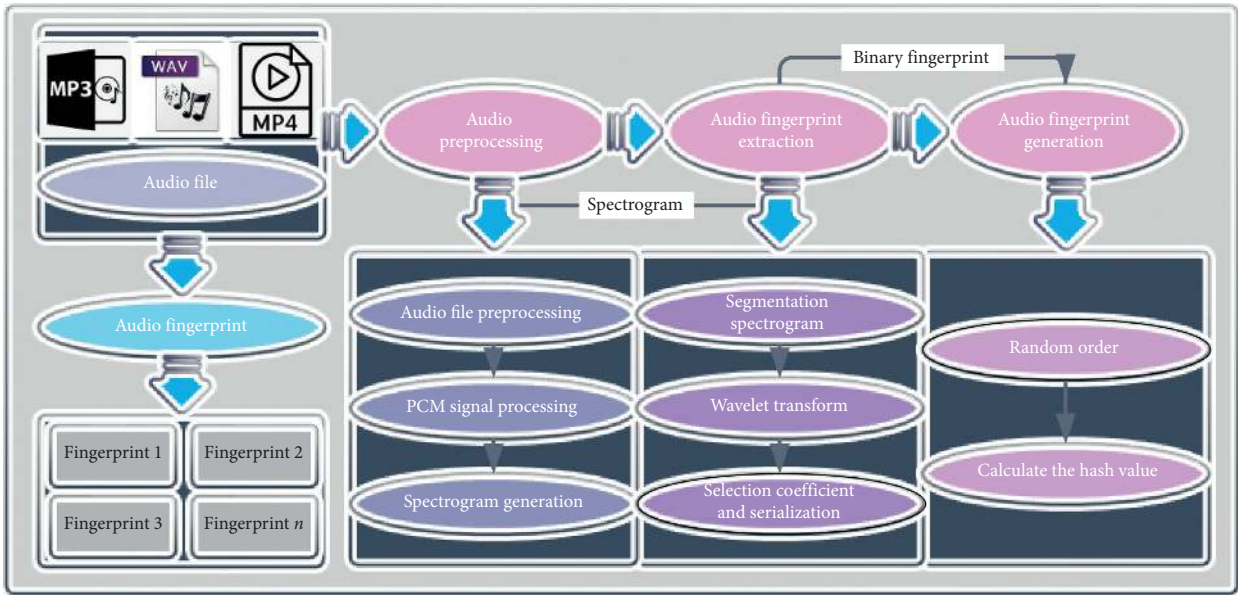


Figure 1: Flowchart of audio fingerprint extraction algorithm.

$$\psi(t) = \begin{cases} 1, & 0 \le t \le 1, \\ 0, & t < -1; t > 1, \\ -1, & -1 \le t < 0. \end{cases} \tag{3}$$

3.2. Subsequence Music Retrieval Algorithm Model. Generally speaking, the frame-based melody matching algorithm is based on the original fundamental frequency sequence and uses a relatively accurate fundamental frequency sequence for matching, and its retrieval accuracy is relatively higher, but it also has a major drawback that it compares and calculates with each frame of data, which leads to the matching process consuming a lot of time and a slow matching rate, which affects the overall system user experience [23]. The note-based melody matching algorithm first has to turn the audio into notes one by one, which introduces a certain error in this note slicing process and generates more errors in the subsequent note processing, which leads to a decrease in the retrieval accuracy. However, it also has an advantage that retrieval matching is faster, because compared with frame-based matching processing, note-based matching has less redundancy, and matching uses more concise features [24].

For the case of two sequences with different lengths in the dimension of time, it is possible to use the DTW algorithm to calculate their similarity, which can be used not only for speech recognition but also for the matching of subsequence fundamental frequency sequences and the template of fundamental frequency sequences in the database, indicating the similarity of subsequence songs and songs in the database.

First, a matrix of size $n \times m$ is constructed, and the element of the matrix at position (x, y) is D_{xy}, which represents the distance between the size A_x of the x-value in sequence B and the size B_y of the y-value in sequence A. The square of the Euclidean distance is generally used. The Euclidean distance represents the true distance between two points in the m-dimensional space, or the natural length of a vector. The Euclidean distance in two-dimensional and three-dimensional space is the actual distance between two points, as in the following equation:

$$D_{xy} = \sqrt{\sum_{i=1}^{m} \left(A_x - B_y\right)_i^2}. \tag{4}$$

After calculating the elements at each position in the matrix, the idea of dynamic programming can be used to

find a path in the matrix that satisfies the requirements from the lower left to the upper right. This path is represented by the following equation and is called a regularized path, which reflects a kind of mapping relationship between a time series A and B:

$$G(k) = \sum_{k=0}^{m\times n} g(k). \tag{5}$$

In practice, just satisfying the requirement two time series is not enough to compare the similarity, because many paths satisfy the condition, so the one that needs to be found should satisfy the requirement of the following equation; that is, the computational cost of this regularized path is minimized [25]:

$$D(x, y) = \text{Max}\left\{\frac{\sqrt{\sum_{k=0}^{m} G(k) + \sum_{k=0}^{n} G(k)}}{m \cdot n}\right\}. \tag{6}$$

In equation (6), K is used to compensate for paths of different lengths. Thus, the ϖ algorithm is to locally scale two time series of different degrees to get a minimum similarity distance, and with this minimum similarity distance, an optimal regularization path is found. ϖ accumulates the distance ϖ (x, y) on the regularization path, which can be obtained from equation (7), where ϖ (x, y) is the sum of the distance D (A_x, B_y) of the current point of the regularization path and the accumulation distance of the previous point of the path and the sum of the cumulative distances of the previous point of the path.

$$\omega(x, y) = D\left(A_x, B_y\right) + \text{Max}\left\{\sum_{i=0}^{x} \omega(i) - \sum_{j=0}^{y} \omega(j), \omega(x-1, y-1)\right\}. \tag{7}$$

In the subsequence retrieval system, which belongs to the note-based matching algorithm, the following model is built using bulldozer distance: If the fundamental frequencies are extracted from the composite music data and transformed into a sequence of notes as $A = \{(a_1, w_1), (a_2, w_2), (a_3, w_3)...(a_m,)\}$, then $B = \{(b_1, w_1), (b_2, w_2), (b_3, w_3)...(b_n, w_n)\}$, where A_i and B_i are the ith note of the note sequence A and B, respectively, and w_i is the duration of this note. Considering A and B as the source and target distributions in the EMD model, respectively, the duration of each note is considered as the weight in the EMD model, and the distance between notes is used as the surrogate value in the EMD model. Thus, the following objective function can be established:

$$F(A, B) = \frac{\sum_{i=0}^{n} f(i) \cdot D\left(A_i, B_i\right)}{\sum_{i=0}^{n} f(i)} + \frac{\sum_{j=0}^{n} f(j) \cdot D\left(A_j, B_j\right)}{\sum_{j=0}^{n} f(j)}. \tag{8}$$

In practical applications of subsequence retrieval systems, the construction of the score function is important, and it directly affects final similarity size results. Therefore, usually, the construction of the score function requires a combination of both the true distance of the notes and the statistical results. Through the above analysis, although both the DTW algorithm and the string alignment algorithm are based on dynamic programming algorithms, the time complexity of the string alignment algorithm is much lower than that of the DTW algorithm under the same conditions, which is the biggest advantage of the cozy note string alignment algorithm. The string alignment algorithm also has an artificially caused disadvantage, because the string alignment algorithm is based on note features, but the extraction of the note features itself introduces human error, which leads to a decrease in the retrieval accuracy of the algorithm [26–28].

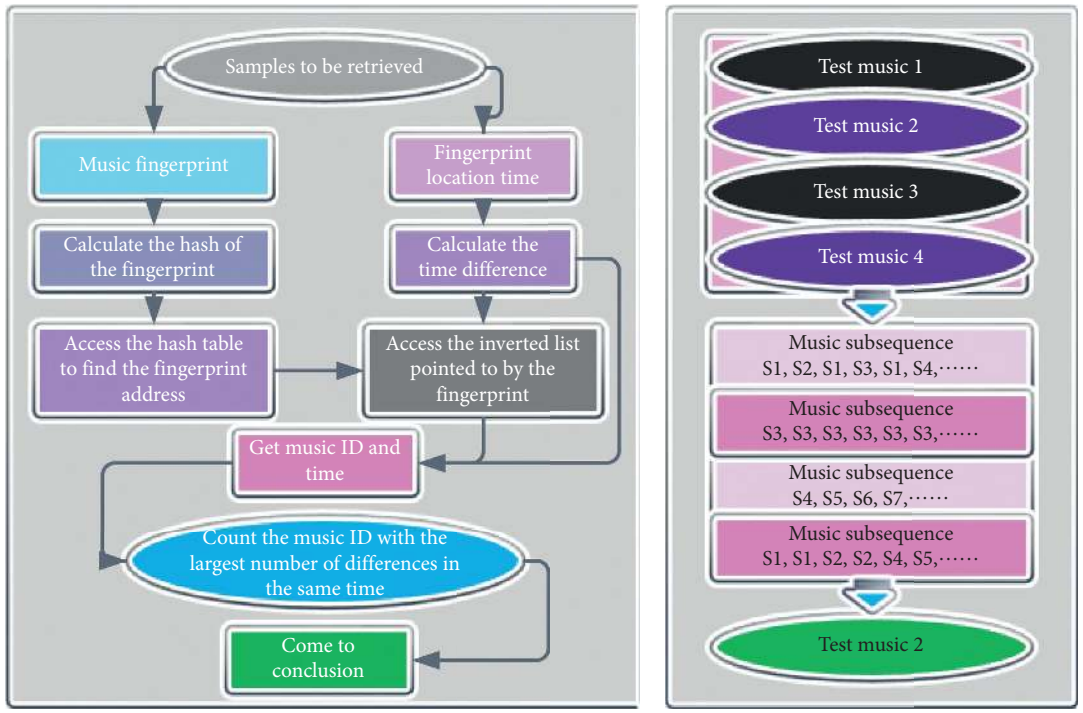
3.3. Software Integration System Design. For the retrieved samples, they are also processed like the music library files using the fingerprint extraction module and then combined with the fingerprint library for matching, and the matching results are counted to derive the best matching original music ID [29–31]. Firstly, all the fingerprints and their accompanying time information are extracted from the samples. Then, a hash is calculated for each fingerprint in turn and the fingerprint storage location is found in the fingerprint library, and then the inverted alignment table is accessed according to the address. Again, the music ID and the time location of the fingerprint are obtained from the inverse table, and the time of the fingerprint is subtracted from the time of the sample, and the music ID and the time difference are combined and stored in a suitable data structure. Finally, after performing steps (2) and (3) for all fingerprints, all the music IDs associated with the samples are obtained, and each music ID may correspond to a series of time differences, and the music ID with the most identical time differences is found. The general process of this module is shown in Figure 2(a).

After computing the time difference of all fingerprints of a sample, a series of music IDs will be obtained, and each music ID may correspond to a series of time differences. For example, after extracting all the fingerprints of a sample, the fingerprints are matched with 4 pieces of music in the fingerprint database, among which Music 2 corresponds to the largest number of the same time differences S3, so Music 2 is judged to be the best match with the sample, as shown in Figure 2(b).

There are two entities in the database, music and fingerprints. Music contains more than one fingerprint; a fingerprint may also appear in more than one piece of music. For the inclusion relationship, a relationship table is needed, which contains the location time attribute of the fingerprint in the corresponding music. The database needs to design three tables, namely, the music table, fingerprint table, and the relation-containing table; the specific design is shown in Table 2.

4. Results and Analysis

4.1. Subsequence Music Retrieval Algorithm Model Analysis. A test set was used to compare the original algorithm with the improved feature extraction algorithm, and the different



(a) (b)

Figure 2: Music subsequence matching and similarity determination.

Table 2: Database tables.

Table name	Field name	Type	Whether nullable	Primary key
Music sheet	MusicID	Int (64)	No	Yes
	MusicName	Varchar (256)	No	No
	MusicPath	Varchar (256)	No	No
Fingerprint table	FingerPrintID	Int (64)	No	Yes
	FingerPrint	Int (64)	No	No
	InclusionID	Int (64)	No	No
Relation-containing table	MusicID	Varchar (256)	No	Yes
	FingerPrintID	Varchar (256)	No	Yes
	Position	Int (64)	No	No

algorithms were compared based on the retrieval error rate and time. The improved algorithm first extracts feature points from a range of one frame and compares them with the original algorithm in terms of error rate, and then it extends the comparison to a range of multiple frames, but the performance in a range of one frame is less satisfactory, so the experimental data are not written in the following table. This indicates that the improved feature extraction algorithm has a lower error rate, and the effect of the energy point ratio of different rectangular regions on the retrieval time and error rate is also studied.

As can be seen from Figure 3, the improved feature extraction algorithms in terms of retrieval error rate have all decreased to different degrees compared to the original algorithm, but the improvement of retrieval error rate is different for different rectangular regions. By cross-sectional comparison, the retrieval error rate has no direct linear relationship with the ratio size of the energy sum in the rectangular region. The retrieval error rate of Algorithm 1 at 8 s has the highest retrieval error rate, which is 0.31% lower than the original algorithm, with a relative decrease of 7.92%; the retrieval error rate of Algorithm 2 has the lowest retrieval error rate, which is 2.12% lower than the original algorithm, with a relative decrease of 55.3%. The highest retrieval error rate was found in the 10 s segment (algorithm search fragment 5), which decreased 0.51% and 23.18% relative to the original algorithm; the lowest retrieval error rate was found in Algorithm 3, which decreased 1.76% and 81.05% relative to the original algorithm. By further vertical comparison, the retrieval error rate is still lower than that of the original algorithm, but not as low as that of the horizontal equivalent rectangular area; for example, the retrieval error rates of Algorithm 2 and Algorithm 11 are higher.

We use the normal random sequence generated by the MATLAB tool function as Gaussian noise, the roadside car running noise as ambient noise, and the collected breathing

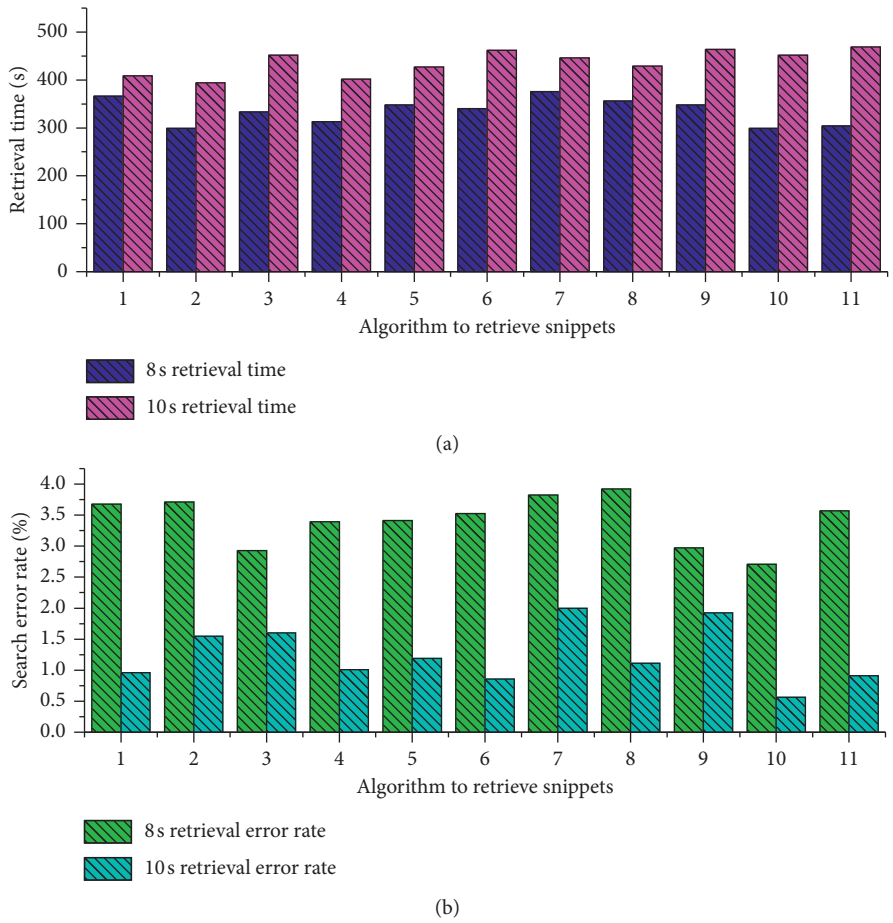


Figure 3: Effect of different comparison algorithms on retrieval time and error rate.

noise and mix them into the audio signal of the song fragment performed by a single person at a quiet time, respectively, to generate noisy subseries audio, all with a signal-to-noise ratio of −10 dB, and perform systematic subseries retrieval. The system's subsequence retrieval is performed, the song list is returned, and the hit rate is calculated. Through a large number of experiments, the experimental data results are shown in Figure 4. From the experimental data results in Figure 4, it can be seen that there is no significant change in the accuracy of subsequence retrieval compared with the accuracy of subsequence retrieval in the noise-free environment, where Gaussian noise and ambient noise have very little effect on subsequence. The retrieval accuracy decreases more significantly for breathing noise compared to the other two noise environments.

After analysis, we found that the original algorithm is to extract the peak energy point of each frame; the energy point of general noise is smaller than the energy of the peak point, which will not affect the extraction of the peak point, so the impact on the error rate will not be great. And the improved feature extraction algorithm is different, it is to select the threshold point of energy of local rectangular area in the time–frequency domain, and the energy point of some features is not the peak point, but only the threshold point of local energy; then the energy point of strong noise will affect the threshold point, so it will have a great impact on the retrieval error rate.

4.2. Music Retrieval System Test Analysis. In this test, the wavelet transform-based audio retrieval algorithm was applied to the subsequence-based audio retrieval system and the subsequence retrieval system, respectively, under the premise that the retrieval algorithm used the LSH-based audio fingerprint retrieval algorithm, and the final retrieval test obtained the search completion rate, search accuracy rate, and retrieval time as shown in Figure 5. The current test discusses the ability to retrieve different quality music subsequences. The retrieval capabilities of three different types of music subsequence systems, namely, sample retrieval, professional music subsequence retrieval, and amateur music subsequence retrieval, are available. The comparison conditions of retrieval capabilities are fully checked, accuracy, precision, and retrieval time. From Figure 5, we can see that the wavelet transform-based audio retrieval algorithm has a good retrieval effect when applied to the subsequence-based retrieval system. When the wavelet transform-based audio retrieval algorithm is applied to the subsequence retrieval system, the search completion rate and the search accuracy rate are reduced, and the retrieval time is

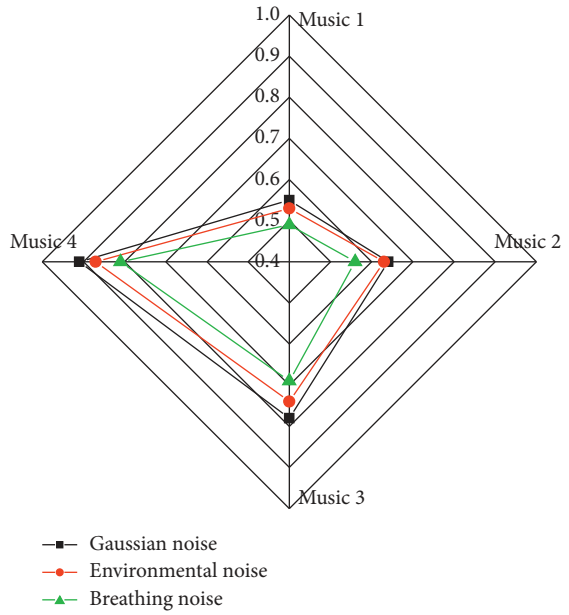


Figure 4: Effect of different noise environments on the success rate of subsequence retrieval.

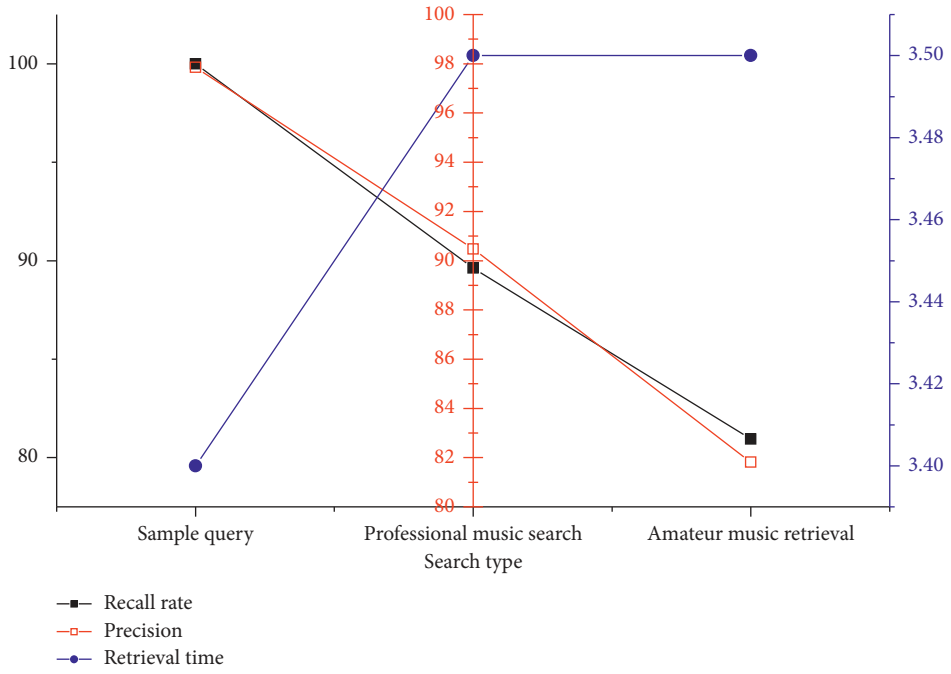


Figure 5: Search completion rate, search accuracy rate, and search time of the retrieval system.

increased. When the wavelet transform-based audio retrieval algorithm is applied to the subsequence retrieval system, the requirements for subsequence are relatively high. When the subsequence quality is high, the system can have better retrieval performance.

In this test, the improved LSH-based audio fingerprint retrieval matching algorithm is applied to the subsequence-based music retrieval system with the subsequence music data set as the retrieval input and the parameters m and T taken as 5 and 3. The results of this test are shown in Figure 6. From Figure 6, we can see that the improved LSH-based audio fingerprint retrieval matching algorithm has superiority in retrieving music of similar music styles.

In the experimental setting of this paper, the present test can illustrate that the wavelet transform-based fingerprint extraction algorithm can be applied among the subsequence retrieval systems, but the requirements for subsequence are very high. When the improved LSH-based audio fingerprint retrieval matching algorithm is applied to the subsequence-based music retrieval and subsequence retrieval systems with the combination of parameters m and T of $5+3$ in the secondary voting process, the detection rate of the two systems is slightly

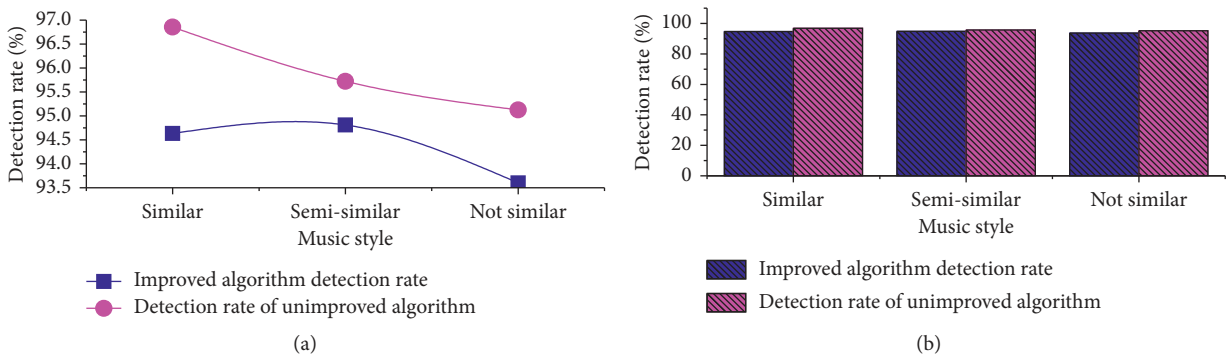


Figure 6: Comparison of retrieval performance of LSH algorithm.

improved. The improved fingerprint retrieval matching algorithm has superiority in retrieving music of similar musical styles because of the increased set of candidate sequences.

4.3. Matching Subsequence Music Retrieval Performance Analysis. The squared Euclidean distance, Manhattan distance, Log function method distance, and Arctan function method distance are applied to calculate the performance of the algorithm on the database, respectively. The experimental results are shown in Figure 7. The comparison of the experimental results in Figure 7 shows that the distance calculation of the LS algorithm using the absolute value function method has a better matching effect.

In the second experiment, when a single LS algorithm and DTW algorithm are combined (the threshold value is set to 2, and the LS algorithm distance and DTW distance are combined in such a way that the smallest value is selected as the final similarity estimation distance), their retrieval accuracy and retrieval time are compared, and the experimental results are shown in Figure 8. The above experimental results show that the matching algorithm using the combination of the distance calculated by LS and the DTW distance can significantly reduce the retrieval matching time compared with the DTW algorithm; meanwhile, the top-n hit rate has an excellent performance in terms of the retrieval matching accuracy, although the MRR is unchanged.

Next, we will test the matching algorithm combining LS distance and DTW distance. In the experiment, we use four ways to compare the minimum, maximum, sum, and product of the distance calculated by the LS algorithm and the DTW distance, respectively. The experimental results are shown in Figure 9. It can be seen from Figure 9 that combining the distance calculated by the LS algorithm and the DTW distance based on the minimum value principle can improve the MRR and top-n retrieval success rate of the matching engine.

In this paper, we study the matching algorithm in a threnody-based sub-equence retrieval system, which is the core part of the system and has a very important

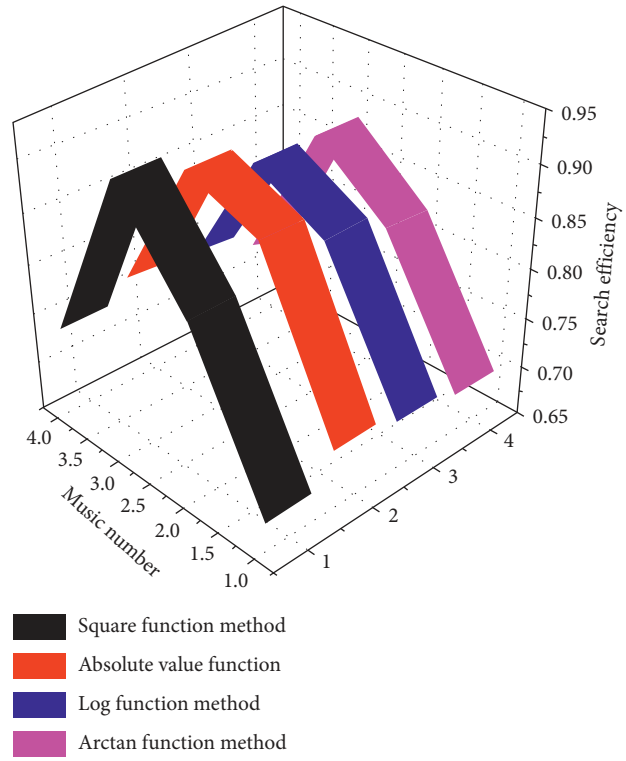


Figure 7: The retrieval performance of LS algorithm with different distance calculation methods.

impact on the performance of the system and the impact on the user experience. Combining the requirements of the system for the matching algorithm and the study of typical algorithms, the matching algorithm combining LS and DTW is proposed. Through the experiments, we find that the matching engine proposed in this paper has good performance and achieves a good usage effect. Combining the above experimental results, we can conclude that the retrieved subsequence fragments are pure audio or in an environment with less indoor noise, and with the improved feature extraction algorithm, the system has a lower retrieval error rate.

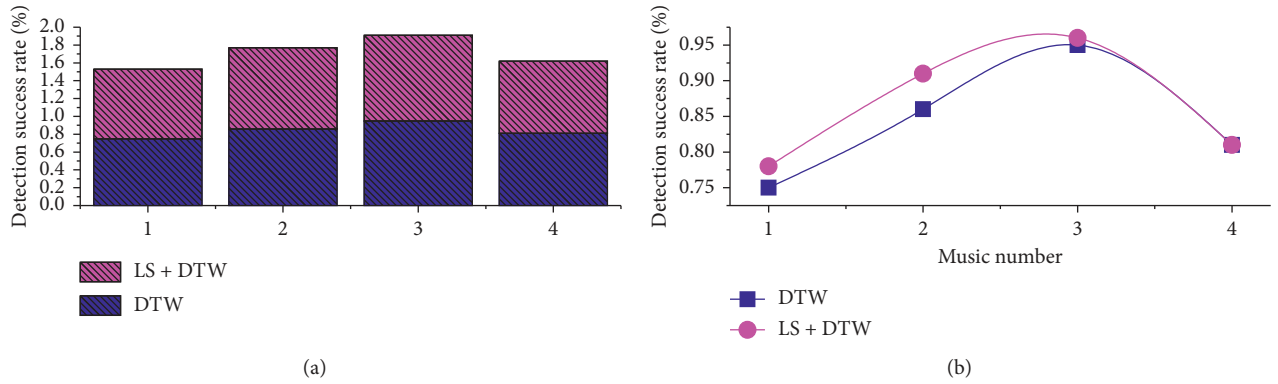


Figure 8: Comparison of DTW and combined LS and DTW matching performance.

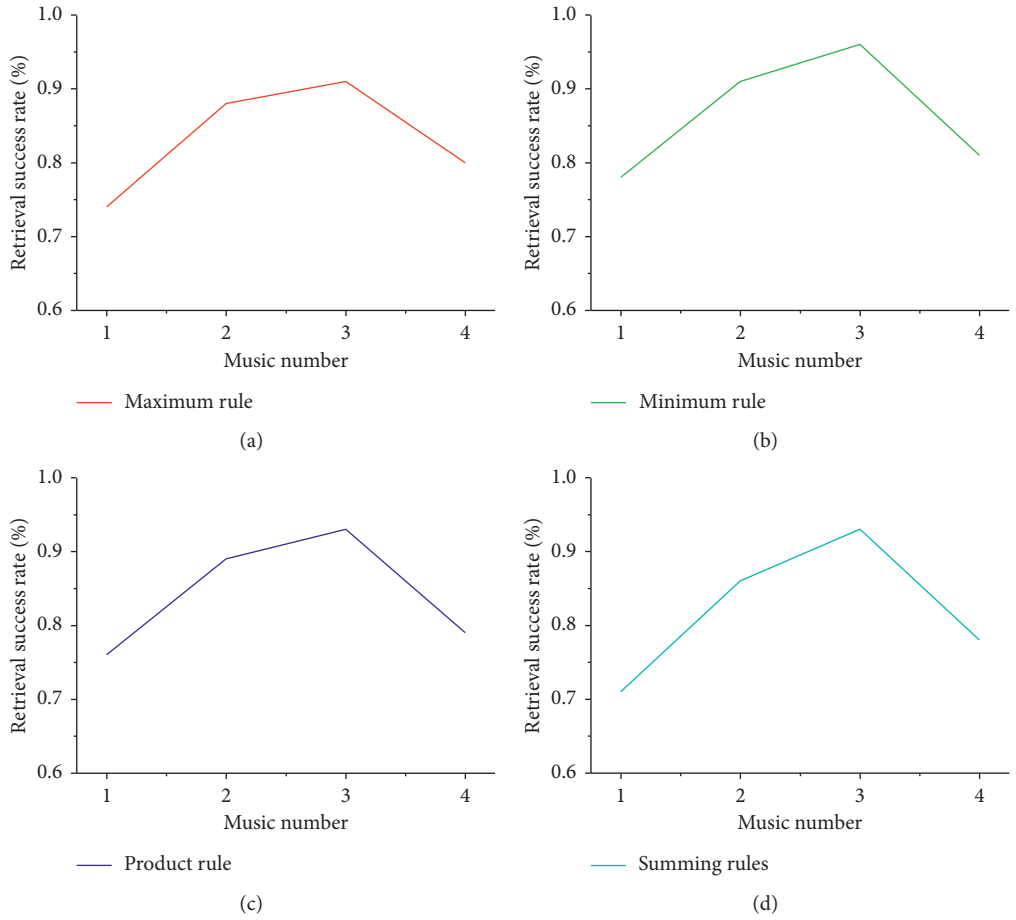


Figure 9: Matching performance at different ways of combining LS and DTW distances.

5. Conclusion

In this paper, we focus on music retrieval techniques based on subsequence and introduce several music feature extraction algorithms and matching algorithms. Based on the existing techniques, several feature improvement algorithms are proposed to address the problem of low matching accuracy, including vector fusion of rhythmic and GFCC features and PCA dimensionality reduction to achieve an improved retrieval rate. The implementation of the system function module mainly relies on the current music retrieval algorithms, including various preprocessing algorithms, feature extraction algorithms, and matching algorithms for audio signals. The system focuses on analyzing the impact of different feature extraction algorithms on retrieval accuracy and proposes a feature fusion method based on the advantages and disadvantages of various algorithms. On this basis, the PCA principal component analysis is used to

minimize the time complexity of the algorithm while ensuring the accuracy rate, to improve the efficiency of the music retrieval system in real-life applications. Due to the objective conditions, the system was experimentally validated in the MATLAB platform for local music database, including each function of music retrieval. The development of mobile Internet and the richness of multimedia resources make the research of human–computer interaction incandescent, and the massive music data will promote the development of music retrieval technology, there are still many problems in music retrieval, and this paper will continue the research in the above aspects to solve the existing problems.

Data Availability

Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

Conflicts of Interest

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

References

[1] H. Wang, J. Ye, Z. Yu, J. Wang, and C. Mao, "Unsupervised keyword extraction methods based on a word graph network," *International Journal of Ambient Computing and Intelligence*, vol. 11, no. 2, pp. 68–79, 2020.

[2] N. Firoozeh, A. Nazarenko, F. Alizon et al., "Keyword extraction: issues and methods[J]," *Natural Language Engineering*, vol. 26, no. 3, pp. 1–33, 2019.

[3] K. Gurjar and Y. S. Moon, "A comparative analysis of music similarity measures in music information retrieval systems," *Journal of Information Processing Systems*, vol. 14, no. 1, pp. 32–55, 2018.

[4] D. Jannach, B. Mobasher, and S. Berkovsky, "Research directions in session-based and sequential recommendation," *User Modeling and User-Adapted Interaction*, vol. 30, no. 4, pp. 609–616, 2020.

[5] A. Onan, S. Korukoğlu, and H. Bulut, "A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification," *Information Processing & Management*, vol. 53, no. 4, pp. 814–833, 2017.

[6] R. Kannao and P. Guha, "Segmenting with style: detecting program and story boundaries in TV news broadcast videos," *Multimedia Tools and Applications*, vol. 78, no. 22, pp. 31925–31957, 2019.

[7] S. Singh and M. S. Aswal, "Semantic web mining: survey and analysis," *Journal of Web Engineering and Technology*, vol. 5, no. 3, pp. 20–31, 2018.

[8] S. Jebbara, V. Basile, E. Cabrio et al., "Extracting common sense knowledge via triple ranking using supervised and unsupervised distributional models," *Semantic Web*, vol. 10, no. 1, pp. 139–158, 2019.

[9] V. Priya and K. Umamaheswari, "Aspect-based summarisation using distributed clustering and single-objective optimisation," *Journal of Information Science*, vol. 46, no. 2, pp. 176–190, 2020.

[10] T. Palpanas and V. Beckmann, "Report on the first and second interdisciplinary time series analysis workshop (itisa)," *ACM SIGMOD Record*, vol. 48, no. 3, pp. 36–40, 2019.

[11] 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.

[12] C. Liu, W. Huang, F. Sun et al., "LDS-FCM: A linear dynamical system based fuzzy C-means method for tactile recognition," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 1, pp. 72–83, 2018.

[13] C. H. Liu, J. Xu, J. Tang et al., "Social-aware sequential modeling of user interests: a deep learning approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 11, pp. 2200–2212, 2018.

[14] H. Zhou and K. Hirasawa, "Evolving temporal association rules in recommender system," *Neural Computing and Applications*, vol. 31, no. 7, pp. 2605–2619, 2019.

[15] K. Gallardo, "Competency-based assessment and the use of performance-based evaluation rubrics in higher education: challenges towards the next decade," *Problems of Education in the 21st Century*, vol. 78, no. 1, pp. 61–79, 2020.

[16] H. Darabian, A. Dehghantanha, S. Hashemi et al., "A multiview learning method for malware threat hunting: windows, IoT and android as case studies," *World Wide Web*, vol. 23, no. 2, pp. 1241–1260, 2020.

[17] F. Ravat and J. Song, "A unified approach to multisource data analyses," *Fundamenta Informaticae*, vol. 162, no. 4, pp. 311–359, 2018.

[18] F. A. Khan, F. Khelifi, M. A. Tahir et al., "Dissimilarity Gaussian mixture models for efficient offline handwritten text-independent identification using SIFT and RootSIFT descriptors," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 289–303, 2018.

[19] P. Khanh, D. T. Tr An, V. T. Duong et al., "The new design of cows' behavior classifier based on acceleration data and proposed feature set," *Mathematical Biosciences and Engineering: MBE*, vol. 17, no. 4, pp. 2760–2780, 2020.

[20] G. Yang, S. Qi, T. Yu et al., "SVDTWDD method for high correct recognition rate classifier with appropriate rejection recognition regions," *IEEE Access*, vol. 8, no. 99, pp. 47914–47924, 2020.

[21] K. R. Singh and S. Chaudhury, "Comparative analysis of texture feature extraction techniques for rice grain classification," *IET Image Processing*, vol. 14, no. 11, pp. 2532–2540, 2020.

[22] A. Onan, "Mining opinions from instructor evaluation reviews: a deep learning approach," *Computer Applications in Engineering Education*, vol. 28, no. 1, pp. 117–138, 2020.

[23] A. Falah, L. Pan, S. Huda, S. R. Pokhrel, and A. Anwar, "Improving malicious PDF classifier with feature engineering: a data-driven approach," *Future Generation Computer Systems*, vol. 115, pp. 314–326, 2021.

[24] K. Akyol, "Comparing of deep neural networks and extreme learning machines based on growing and pruning approach," *Expert Systems with Application*, vol. 140, no. Feb., pp. 1–7, 2020.

[25] M. M. Hittawe, S. Afzal, T. Jamil, H. Snoussi, I. Hoteit, and O. Knio, "Abnormal events detection using deep neural networks: application to extreme sea surface temperature detection in the Red Sea," *Journal of Electronic Imaging*, vol. 28, no. 02, pp. 1–8, 2019.

[26] A. B. Kanase-Patil, A. P. Kaldate, S. D. Lokhande, H. Panchal, M. Suresh, and V. Priya, "A review of artificial intelligence-based optimization techniques for the sizing of integrated renewable energy systems in smart cities," *Environmental Technology Reviews*, vol. 9, no. 1, pp. 111–136, 2020.

[27] S.-B. Son, J.-U. Jung, H.-S. Oh, and Y.-C. Jung, "DeepMask: face masking system using deep neural networks on real-time streaming," *Journal of Institute of Control, Robotics and Systems*, vol. 26, no. 6, pp. 423–428, 2020.

[28] H. H. Bu, N. C. Kim, B. J. Yun et al., "Content-based image retrieval using multi-resolution multi-direction filtering-based CLBP texture features and color autocorrelogram features," *Journal of Information Processing Systems*, vol. 16, no. 4, pp. 991–1000, 2020.

[29] B. Li, P. Chen, H. Liu et al., "Random sketch learning for deep neural networks in edge computing," *Nature Computational Science*, vol. 1, no. 3, pp. 221–228, 2021.

[30] D. Peer, S. Stabinger, and A. Rodríguez-Sánchez, "conflicting_bundle.py-A python module to identify problematic layers in deep neural networks," *Software Impacts*, vol. 7, no. 8, Article ID 100053, 2021.

[31] A. Zielonka, A. Sikora, M. Wozniak, W. Wei, Q. Ke, and Z. Bai, "Intelligent Internet of things system for smart home optimal convection," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4308–4317, 2021.