Fog Computing Approach for Music Cognition System Based on Machine Learning Algorithm

Article · October 2018

DOI: 10.1109/TCSS.2018.2871694

Guoqiang Li
Shanghai Jiao Tong University

Hongming Cai
Shanghai Jiao Tong University

Fog Computing Approach for Music Cognition System Based on Machine Learning Algorithm

Lifei Lu, Lida Xu, *Fellow, IEEE*, Boyi Xu, Guoqiang Li, and Hongming Cai, *Senior Member, IEEE*

***Abstract*— With the wide spreading of mobile and Internet of Things (IoT) devices, music cognition as a meaningful task for music promotion has attracted a lot of attention around the world. How to automatically generate music score is an important part in music cognition, which acts as an important carrier so as to disposing huge quantity of music data in IoT networks or Internet. For the reason that the computers lack of the domain knowledge and cognitive ability, it is hard for computers to recognize the melody of music or write score while listening to the music. Therefore, a music cognition system is introduced to cognate music and automatically write score based on machine learning methods. First, considering large-scale data processing is needed by machine learning algorithms and a number of music devices are involved in the cognition system through Internet, fog computing is adopted in the proposed architecture to efficiently allocate computing resources. Then, the system can collect, preprocess, and store raw music data on the fringe nodes. Meanwhile, these data will be transmitted from fog nodes to cloud servers to form music databases. Then, machine learning algorithms, such as hidden Markov model and Gaussian mixture model, are performed in cloud servers to recognize music melody. Finally, a case study of music score generation demonstrates the proposed system. It is shown that the method provides an effective support to generate music score, and also proposed a promising way for the research and application of music cognition.**

***Index Terms*— Audio signal process, fog computing, Internet of Things (IoT), machine learning, music cognition.**

I. Introduction

MUSIC plays a more and more important role in the digital world. However, there are many limitations in the existing music appreciation and promotion field. As music analysis largely relies on the professional experts, it is not convenient for common people to appreciate the melody of music. Meanwhile, it is more difficult for computer to recognize the audio signals than to combine the lyrics and the album information, so it is hard for computer to understand the music. Moreover, due to the large number of the digital songs, it is emergent for a Service Oriented Architecture (SOA) [1] system to automatically analyze these music songs. Recently, clients of the application in Web or mobile devices, which are used to listen to music, download music, and purchase music in this domain, have higher level requirements, including music score generation, music similarity analysis, music retrieval, and recommendation, so it is urgently to develop a music cognitive platform.

Meanwhile, with the development of Internet of Things (IoT) [2], many studies focus on the industrial IoT applications in areas such as agriculture, food processing industry, environmental monitoring, security surveillance, and others [3]. The supporting technologies of IoT include radio frequency identification technology, barcodes, smart phones, social networks, and cloud computing [4]. Moreover, the concept of SIoT, which means applying the social networking principles to the IoT, has attracted the attention of many scholars [5]. Concentrating on music applications, there are many devices of client users. The perturbation between human and devices can influence the performance of the music application. As the style and timbre can be impacted by the devices, different client users may have a different aural reception of an identical device and the relationship between the users and their device environment can also result in different perceptions. How to combine devices and clients in the IoT platform [6] is significant in music application scenarios.

Moreover, with the development of fog computing [7], [8], the preprocessing of large-scale data in different formats can be deployed in near-user devices, which can reduce the load of cloud servers and distributed devices. As music data include structural ones and unstructured ones, how to combine the data in a standard format is an important job. Therefore, the music model establishment can be deployed in the fringe devices. Though the preprocessing of large-scale data is often introduced, the analysis of music application is still hard for developers without the help of machine learning and statistical technologies. With the help of these methods, the performance of calculation services can be improved. Considering the requirements of clients are variant and flexible, it is important to describe the requirements of clients in a standard format, which can be well resolved with the help of uniform interface.

Based on the above analysis, we proposed a system for automatic music cognition. Clients can express their requirements by the application running on various devices. They can also input audio files or record signals in real time as the raw testing data, while the system can analyze the requirement and deploy the task in different levels of services. With the help of algorithm optimization and historical knowledge enrichment, the performance can be improved gradually.

Manuscript received February 28, 2018; revised August 4, 2018; accepted September 8, 2018. This work was supported by the National Natural Science Foundation of China under Grant 61373030. *(Corresponding author: Hongming Cai.)*

L. Lu, G. Li, and H. Cai are with the School of Software, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: lulifei@sjtu.edu.cn; li.g@sjtu.edu.cn; hmcai@sjtu.edu.cn).

L. Xu is with Old Dominion University, Norfolk, VA 23529 USA (e-mail: lxu@odu.edu).

B. Xu is with the Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200052, China (e-mail: byxu@sjtu.edu.cn).

Digital Object Identifier 10.1109/TCSS.2018.2871694

There are many challenges in our research. As there are multiple formats of music data, including audio signals and lyrics and music score, how to extract and describe the features from different formats of data still needs a specification. Meanwhile, the cognition system which provides overall music services for clients is urgently needed for the state of the art. Moreover, better deployment of the devices involved in the system is needed.

Considering these challenges, the main contributions of this paper include the following.

1) A new music data model based on the features extracted from audio signals, lyrics, and some other information of the songs is established.
2) A system for music cognition based on the music model data sets is proposed. The system can provide services for clients with the help of machine learning technologies. Meanwhile, the fog computing methods are involved in our deployment phase to optimize the performance of our system.
3) A case study for music score recognition based on the cognitive system is implemented. After that, we compare our platform with others.

The rest of this paper is organized as follows. In Section II, we introduce the related work in this field. The overview of the proposed system and the interaction interface are explained in Section III. In Section IV, we explain the methodology of the music cognition in detail. Implementation and discussion are displayed in Section V. In Section VI, we make a conclusion and discuss some future work of this paper.

II. Related Work

Cognitive computing, which aims to develop a coherent, unified, and universal mechanism inspired by a collection of processes of sensation, perception, action, emotion, and cognition, has the ability to make decision and initiate sophisticated coordinated actions [9]. There are lots of studies in this filed, for example, Esser *et al.* [10] developed a set of abstractions, algorithms, and applications that are natively efficient for TrueNorth. In this area, audio signal process is a pregnant problem, which has attracted intensive research interest. Umapathy *et al.* [11] used the local discriminant bases technique to identify discriminatory time–frequency subspaces. Umapathy *et al.* [11] also proposed an audio feature extraction and a multigroup classification algorithm. Liang *et al.* [12] analyzed the features of audio signals, classified the audio signals into different classes, and proposed some new features for audio analysis. Rai *et al.* [13] applied audio feature extraction and support vector machine to classify bird species. Goecke *et al.* [14] enhanced signals with noisy audio features for the noise audio signal condition. Helwani *et al.* [15] used linear loudspeaker arrays to synthesize acoustic wave and also proposed an echo cancellers method for full-duplex massive multichannel systems.

In terms of music signal analysis, many studies focused on the music tagging, music similarity, music searching, and recognition. Choi *et al.* [16] proposed many time–frequency representations of signals and investigated the effect of audio preprocessing on music tagging with neural networks. Logan and Salomon [17] used k-means clustering of spectral features for analyzing the similarity of different songs based on the audio content. Viro [18] proposed a technique based on a search engine of music score, which created new strategy to explore notated music. Haitsma *et al.* [19] proposed a new technology, called robust audio hashing, to identify the audio content. The technology used a bit string to represent the original audios. By comparing the hash values of a received audio clip with the original one, it can identify the audio [19]. Clausen and Kurth [20] proposed a unified approach to fast index-based music recognition; the approach can be applied to the polyphonic (musical score-based) search in polyphonic score data and the identification of pulse-code modulation audio material from a given acoustic waveform.

Moreover, audio content of music songs has attracted multiple researchers; as the audio feathers can be used in different domains, there are many works that have been carried out. Dannenberg and Raphael [21] considered online and offline matching of music signals, focusing on music score matching, score following, and score alignment. Kolozali *et al.* [22] formed an automatic ontology generation method to analyze musical instruments by exploring concept analysis techniques in a semantic web environment. Chang and Su [23] proposed a method based on recurrent network. The method required a smaller time frame to estimate the pitch compared to other methods; it can also track the pitch of a pitch-varying signal or a quasi-periodic signal [23]. Bhalke *et al.* [24] used dynamic time wrapping techniques to recognize the six Indian musical instruments with the mel-frequency cepstral coefficients (MFCC) features.

The score of music, which is the visual expression of a song, has the propagative and preserved impact on the music domain. Ryynanen and Klapuri [25] used three acoustic features and a hidden Markov model (HMM) to represent the note event and then built note event modeling for polyphonic music transcription. Tamboli and Kokate [26] designed a method for musical note recognition based on the classification framework using an optimization-based neural network. de Jesus Guerrero-Turrubiates *et al.* [27] built a musical note recognition system based on harmonic modification and artificial neural network. Bietti *et al.* [28] introduced a new electromagnetic algorithm combined with HMM to deal with music segmentation problems.

Based on the review of former research, music analysis and cognition computing are popular research areas; however, there are many limitations for music application. On the one hand, there are still requirements for domain knowledge in many music research focused on the application and analysis of the music data. On the other hand, there is a lack of normalization of music data in large scale of music analysis. As a consequence, a system which can automatically perform the analysis of music is required.

III. Music Cognition System Outline

A. Research Framework

With the development of artificial intelligence and deep learning (DL), cognitive computing has become a hot topic.

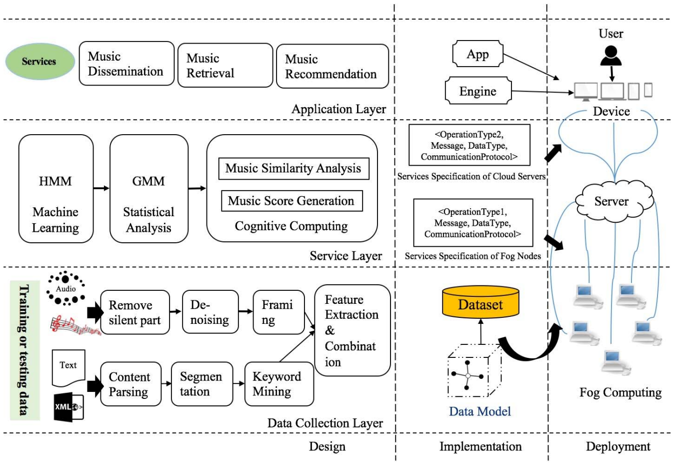


Fig. 1. Framework for music cognition system based on fog computing.

Cognition research especially audio cognition research is valuable for audio signal process and music recommendation. In order to improve the precision of music process and recommendation, services composition, services orchestration, and choreography should not only consider the functional/performance requirements of cognitive computing, but also consider the features of the IoT devices that are connected through Internet and the deployment of the services in the fog/cloud architecture. Therefore, in this paper, a research framework is proposed to design a music cognition system to help users to understand the meaning or style of the music downloaded on Web, as shown in Fig. 1.

In Fig. 1, the proposed framework includes three phases, which are design, implementation, and deployment.

In the design phase, three layers are included which are data collection layer, service layer, and application layer. In data collection layer, different formats of music can be collected and processed. In service layer, machine learning, statistical analysis, and cognitive computing technology are used to accomplish large-scale computing of music application. In application layer, music dissemination, music retrieval, and music recommendation are provided based on the analysis of users and devices.

In the implementation phase, data model is designed to establish data sets; meanwhile, service specification of fog nodes and cloud servers is implemented; after that, the applications embedded in the IoT devices and the search engines on servers are implemented to provide services for users.

In the deployment phase, fog/cloud architecture is used to allocate cognition tasks to the distributed computing resources. End users can use mobile IoT devices to satisfy their requirements of music cognition.

Based on the architecture presented above, the proposed system can improve the accuracy and efficiency of music application. The music cognition system can help the computer to understand the audio signals of the music as well as the lyrics and some other information of music. Meanwhile, it can unify the services included in the application more organically. Based on the computing of different layers, the system would help to increase the efficiency of cognitive computing in SOA-based scenarios. Clients can express their requirements without having to understand the inner structure of the server, while the developer can optimize the parameters of the services based on the historical logs, so applications in this area can become more and more convenient. Therefore, the system would cover activities of data modeling, services development, services composition, information systems deployment, and business process optimization based on mining.

B. Interaction Interfaces

The interaction between three layers of the proposed framework is well-organized. The data collection layer deals with the training data from collection terminals as well as from the Web, and the testing data, which may be input by client users, can be added to the training data sets afterward. The composition of this layer includes a large number of near-user edge devices, which carry out substantial amount of storage. These devices can be called fringe nodes of our system. Based on the interaction of fringe nodes, the preprocessing of data can be executed in the low level, which reduces the resource occupancy in higher levels. After the procedure, data collection layer establishes data sets of the raw data; moreover, it provides a message described in Web Services Description Language (WSDL) for the service layer.

Based on the data sets and the process of the data collection layer, the service layer can use machine learning methods, statistical analysis methods, and cognitive computing technologies to perform large amount of computation task. The devices in this layer are distributed computing servers and some cloud computing servers. Due to the high complexity of machine learning and large-scale computing, this layer learns from the historical logs offline to reduce the computing time of real tasks. Meanwhile, it records the completed results and learns from these useful logs. Finally, it provides an interface described in WSDL for the application layer.

The application layer aims to interact with end users. Clients can describe their requirements based on the graphics user interface of application and search engine. After parsing the requirements' content, this layer builds a specification of the requirement, which defines the needs of the client. Based on the historical logs of the system, this layer decides the task allocation strategy and estimates the execution time of the task. After that, it sends messages to the service layer and waits for the response. After calculation of the service layer, the application gets the results of the requirement. The application layer forms the final results, meets clients' need, and displays the results visually.

IV. Methodology of Music Cognition

A. Music Cognition Model Definition

In the music cognitive computing scenarios in SOA shown in Fig. 1, the establishment of acoustic models of the music is the basis and crucial procedure. We propose a new acoustic model of music based on our research on the audio signals in the SOA systems. After that, we can use the acoustic models to analyze and recognize music signals.

Audio signal is our main consideration when trying to cognize songs, while lyrics may influence the emotion of the songs, and album and the artist also have big impact on the music appreciation and propagation area. If we can build a model to combine all the important factors of a song automatically, we can analyze the models rather than deal with heterogeneous information, and the interaction between different services will be become easier, which can help services in SOA to perform these services by different components dynamically.

The goal of the establishment of our proposed acoustic model is to help automatical music cognition, which includes music score generation, music similarity analysis, music retrieval, and music recommendation.

The following elements are involved in the proposed acoustic model.

Definition 1 (Note (N)): Notes are the basic unit of music score. Duration and pitch are the basic parameters of a note. The duration can be calculated by the end time of a note minus the start time of the note, and it represents the elapsed time of a note. Usually, the duration of a quarter note is regarded as the unit time of a note. There are many other kinds of notes, such as half note and eighth note. The pitch of a note can be obtained through the frequency calculation, which usually utilizes the fast Fourier transform (FFT) or other methods. As there may be more than one note at a time, the start time is added as one of the basic parameters of a note. Meanwhile, a note may be played by different instruments; it is also important to describe the instrument information. Notes are defined as tuples as follows:

$$\text{N} = \langle \text{N\_ID, N\_ST, N\_Duration, N\_Pitch, I\_ID} \rangle.$$

N_ID is the serial number of the note to be performed. N_ST corresponds to the start time of the note. N_Duration records the time of duration. N_Pitch represents the pitch of the note, while I_ID represents the instrument id of the note.

Definition 2 (Note Sequence): Note sequences describe the order of notes, which can be used to represent the melody of a song. Let N_ID represents a note, we can get the order of the notes by comparing the start time of the notes. When multiple notes appear at the same time, i.e., a chord appears, we sort notes according to the instrument priority table. Note sequences can be defined as follows:

$$\text{NS} = \langle \text{NS\_ID}, \{\text{N\_ID1, N\_ID2, N\_ID3}, \ldots\} \rangle.$$

NS_ID is the serial number of the note sequence to be recognized. N_IDi corresponds to the specific note.

Definition 3 (Instrument (I)): It is generally accepted that different instruments have different tones, and with the fusion of multiple instruments, symphony becomes one of the most beautiful music formats. In such scenarios, in which different instruments play together, priorities should be set to ensure that the major instrument has the main auditory sensitivity. Instrument is defined as follows:

$$\text{I} = \langle \text{I\_ID, I\_Name, I\_Priority, I\_BaseInformation} \rangle.$$

I_ID is the serial number of the instrument to be recognized, I_Name corresponds to the name of the instrument, and I_Priority represents the priority of the instrument. I_BaseInformation consists of the base information of the instrument, such as base frequency of a key for piano and the note feature of an instrument.

Definition 4 (Lyrics (L)): Most songs have lyrics including different languages and different versions. We define lyrics of a song as follows:

$$\text{L} = \langle \text{L\_ID, I\_Language, L\_Content} \rangle.$$

L_ID is the serial number of the lyrics of the song. L_Language corresponds to the language of the lyrics and L_Content represents the content of the lyrics, which usually has the format of text.

Definition 5 (Album and Artist Information): Different albums and different artists also have a different expression. For example, a new version of a song by a popular artist can promote the propagation of the song. The album and artist information (AAI) can be defined as follows:

$$\text{AAI} = \langle \text{AAI\_ID, Album\_ID, Album\_Name, Artist\_ID, Artist\_Name} \rangle.$$

AAI_ID is the serial number of the AAI of the song. Album_ID corresponds to the id of the album of the song belongs to, and Album_Name represents the name of the album. Artist_ID represents the id of the artist of the song, and Artist_Name represents the name of the artist.

Definition 6 (Song (S)): A song can be inferred through the above definitions, which can be used to analyze the similarity of the songs as well as generating the score of the music. The song can be defined as follows:

$$\text{S} = \langle \text{S\_ID, L, AAI}, \{\text{NS1, NS2, NS3}, \ldots\} \rangle.$$

S_ID is the serial number of the song. L corresponds to the lyrics of the song. AAI represents the AAI of the song, and NSi represents the note sequence of the part of the song, so the whole song can be expressed as the sequence of the NSi.

B. Music Data Process

Data collection is the most basic operation of the system. This layer deals with multiple input data formats, such as: 1) audio signal files, which may be in .wav or .mp3 format; 2) music scores, which may be represented in numbered musical notation format; 3) text, which represents the lyrics of the songs; and 4) xml, which contains the AAI of the songs. All the data formats can be classified into two categories, unstructured and structured, while the semistructured category, which can be handled according to the above two categories, is not included in this paper.

The unstructured category, which includes audio signal files and music score files, can be handled in the following process.

1) Preprocessing: We get audio files or record audio signals in real time as our input. We choose a monophonic signal among all input signals. First, we should recognize the start point and endpoint of the whole audio files. We use short-time energy as the feature to recognize the endpoint, and we remove the silent part of the audio files, which can be calculated by comparing with the threshold delta, refer to the following:

$$E(i) = S(i)^2 > \text{delta}. \tag{1}$$

As devices and environment noise may affect the results of our further research, we use a method for noise reduction and signal emphasis. After denoising, we get greater quality audio signals. In order to get stable short-time audio signals, we use the Hamming window function to add windows. Through the smooth moving on the audio signals of the window function, the signal is split into frames.

2) Feature Extraction: In order to better analyze the audio files, we should choose specific features [29]. As every music note may have different number of frames, the variation of a note signal is always enhanced at first and decays afterward; we can combine frames of the same note. Within a note, we set a weight for different frames to reduce the affection of overlap of different notes. The process of our feature extraction can be divided into three steps: 1) music note endpoint recognition; 2) feature selection; and 3) feature combination. We introduce the steps briefly as follows.

a) Music note endpoint recognition: As every note may have different durations and a note may have multiple frames, we should recognize the endpoint of different notes. We use the short-time energy to separate the frames

$$\text{endpoint} = \lim_{\Delta x \to 0} \frac{\Delta \text{energy}}{\Delta x}. \tag{2}$$

Equation (2) shows the calculation of endpoint. The endpoint is calculated as the derivation of the energy difference, and we choose the maximum derivation of the energy as the endpoint of our notes.

b) Feature selection: There are many features of audio signals, such as zero-crossing rate, base frequency, short-time energy, short-time range, and MFCC. As for music note recognition, we choose base frequency, which represents the music pitch, as our main feature. As shown in (3), the frequency is calculated through the FFT algorithm of the note signal. We choose the max intensity frequency as the base frequency of the note

$$\text{frequency} = \text{fft}(\text{note signal}). \tag{3}$$

Meanwhile, the duration of a note is represented by the multiply of the amount of frames and the length of each frame. As shown in (4), the fs represents the sampling frequency of the signal

$$\text{duration} = \text{the amount of frames/fs}. \tag{4}$$

We get the features of each frame. Furthermore, the timbre of the musical instruments is represented by the MFCC, which is regarded as the perceptual features of audio signals.

c) Feature combination: After steps 1 and 2, we use weight function to combine the frames of a note. As the start point and endpoint of a note may be affected by the neighboring notes and the middle frames are more stable and representative, we use the Gaussian distribution function as our weight function, as shown in (5). After the calculation, we get the feature vector of notes, consisting of base frequency and MFCC, and the duration feature is utilized in the recognition of note range

$$\text{vector} = (\text{feature1}, \text{feature2}, \ldots) * \frac{1}{\sqrt{2\pi}} e^{\left(-\frac{x^2}{2}\right)}. \tag{5}$$

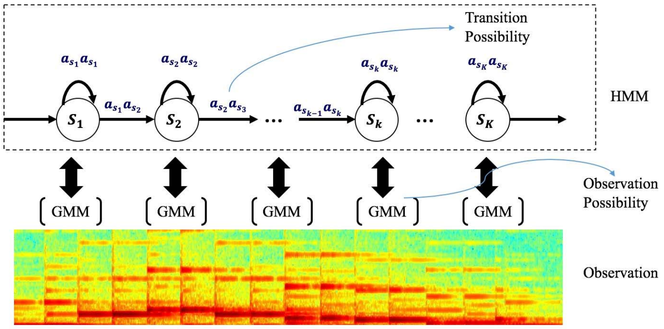


Fig. 2. Disposing procedure of music recognition.

After the above process, we can establish an acoustic model from audio signal input. As for the music score format input, we first play the music score in a normal instrument, such as piano; we can get audio signals afterward; and then, we can deal with the process similar to the audio signals input.

The structured category, which includes text and xml format, can be handled in the following process. First, the content is parsed; after that the word segmentation is executed to split the lyrics; and then the keyword mining method is used to get the keyword of the lyrics. As for the xml, we parse the xml files to get the corresponding information. Both keywords of the lyrics and the corresponding information are used to label the combined features. After all the unstructured and structured processes, we get the data model of our input. The data model is used as the data sets of our analysis procedure. The devices and users of the system can exchange data through specific interfaces.

C. Music Cognition

Service layer is the most important part of the system; it analyzes the data sets in Section IV-B; after that, it can provide application layer with the basic services of cognitive requires. The technologies used in server layer include machine learning, statistical analysis, and cognitive computing.

1) Matching and Recognition: We choose the HMM [30] as the model to analyze the music state transition; for each note can be treated as a state, we use the Gaussian mixture model (GMM) [31], [32] to analyze every note. After training the parameters of the GMM, we input all the parameters into the HMM; after that, we can get that the likelihood of a sequence is identical to the model. The basic principle of the recognition procedure can be described in Fig. 2.

As the data sets may have many dimensions, the parameters of the GMM can be very complex, which can be optimized by the DL method to get the better fitting degree of the sequence.

The cognitive computing requirements, such as music similarity analysis and music score generation, can be obtained through the above methods. We choose the maximum possible template model for each input file. As shown in (6), the similarity of musical sequences is calculated as the average distance of every feature. We choose the minimum similarity

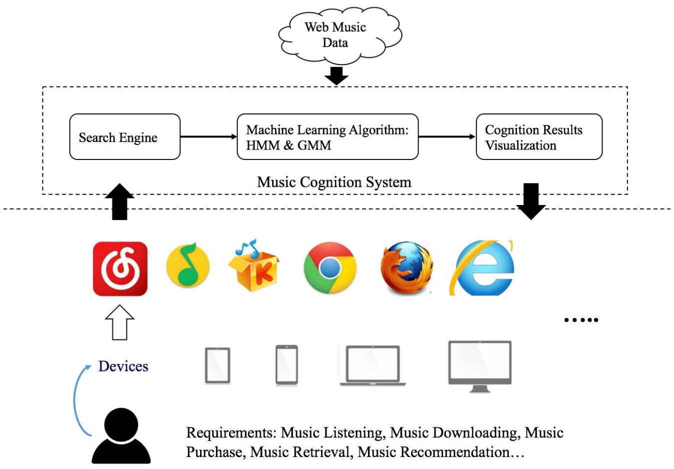

Web Music Data
Search Engine
Machine Learning Algorithm: HMM & GMM
Cognition Results Visualization
Music Cognition System
.....
Devices
Requirements: Music Listening, Music Downloading, Music Purchase, Music Retrieval, Music Recommendation…

Fig. 3. Application pattern of music cognition system.

of the input note with all the preserved data sets

$$\mathrm{sim}(a,b) = \mathrm{sqrt}\sum_{1}^{n}(a_i - b_i)\text{\textasciicircum}2. \tag{6}$$

2) Training: We use the data sets established in Section IV-B to train our analysis models. During the training procedure, the parameters of the GMM are trained. The algorithm we used is as follows. First, we estimate the likelihood of data from each component, and each component means a single Gaussian model. For each data x_j, it is generated by the kth component that is calculated by the following equation:

$$\gamma(i,k) = \frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\Sigma_{j=1}^{k} \mathcal{N}(x_i|\mu_j, \Sigma_j)} \tag{7}$$

As μ_k and $\sum_k$ also need to be estimated, an iterative method is used. μ_k and $\sum_k$ are constant when calculating $\gamma(i,k)$. We use the data gained from the last iteration (or initialization). Second, we estimate the parameters of each component. Suppose we have obtained the correct $\gamma(i,k)$, the probability of data x_i. is generated by component k, which can also be treated as the contribution of the component in the data generation. Considering all the data, the component generates such data, $\gamma(i,k)x_1, \ldots, \gamma(N,K)x_N$. As each component can be treated as a standard Gaussian distribution, we can get the parameters of the most likelihood easily

$$\mu_k = \frac{1}{N_k}\sum_{i=1}^{N}\gamma(i,k)x_i \tag{8}$$

$$\sum_k = \frac{1}{N_k}\sum_{i=1}^{N}\gamma(i,k)(x_i - \mu_k)(x_i - \mu_k)^{\mathrm{T}}. \tag{9}$$

As shown in (8) and (9), $N_k = \sum_{i=1}^{N}\gamma(i,k)$, and π_k can be estimated as N_k/N. Finally, we repeat the above two phases until the convergence of the probability density function

$$p(x) = \sum_{k=1}^{K} p(k)(p(x|k)) = \sum_{k=1}^{K}\pi_k N\left(x_i|\mu_k, \sum_k\right). \tag{10}$$

After the training algorithm, we get the template of different notes of the same instrument, and every note has a model. After that, we use the base frequency of the instrument of the same note online to revise the model. As for the duration feature, which is represented as the length of time, we calculate it as follows:

$$\mathrm{duration} = \frac{\text{endpoint–startpoint}}{\text{length of a quarter note}}. \tag{11}$$

V. Case Study and Discussion

A. Scenario

The proposed system can be used in the following scenario. With the development of IoT, clients can use multiple devices, including mobile phones and laptops; the devices can also be treated as clients. The requirements of the clients contain music retrieval, music recommendation, etc. The proposed music cognition system can analyze the requirements of clients from the search engine and application. After calculation by the algorithm of machine learning based on the web music data, it returns the cognition results to clients. The process is shown in Fig. 3.

As shown in Fig. 3, clients can express their requirements through the devices. With the help of all the application and search engine in portable equipment and desktops, the requirements of clients can be passed into our computing model. Based on the requirement analysis, the system first normalizes the message passed from the application and search engine. After that, it collects music data on Web and allocates the computing resources.

The computing model aims to allocate the tasks of the requirements based on the analysis results. As there are many distributed computing resources, the model deals with the coordination of all the servers in different regions. Due to the required data of large scale, machine learning algorithms, such as HMM and GMM, are used to accomplish the tasks.

Based on the above process, the cognition result visualization model can display the results based on the desired format of the requirements. After passing the results to the application, the model terminates the tasks and recovers the resource; finally, it writes the finish time of the requirement to the log.

B. Deployment of Music Cognition System

Fig. 4 shows the deployment of music cognition system. As shown in Fig. 4, we use RESTful Application Programming Interface for decoupling of the front end and back end. Clients can use mobile phone or desktop computer to use our system; we use the Hypertext Transport Protocol to contact with the server. After the response of the application server, tasks can read or write the data server, which we use MySQL.

C. Case Study

In this section, we will take music score generation as a case to show the running procedure of our system. We use an audio signal as the input. An onset detect method is used to segment different notes. After analyzing the model similarity, we get

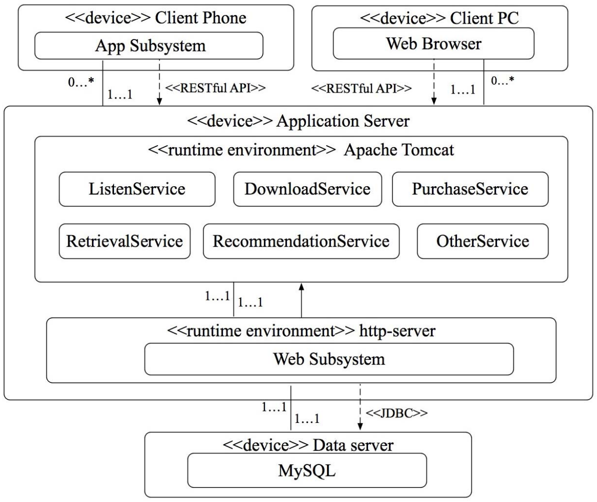


Fig. 4. Deployment structure of music cognition system.

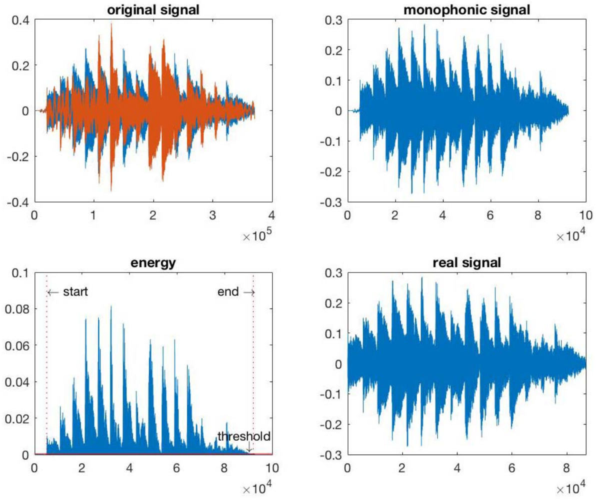


Fig. 5. Preprocessing results.

the maximum likelihood sequence of the notes. By adding the duration parameters, we get the results of the music scores. The experimental results are shown as follows.

Fig. 5 shows the results of preprocessing stage. We use a song in the solo piano version as our input. We first choose a monophonic signal from the original signal, which has two channels, called "left channel" and "right channel." After that, we calculate the short-time energy of the signal. As the noise signal usually has a low energy, we can reduce noise as well as recognizing the endpoints of the music signal. As we can see from Fig. 5, we use a threshold value to determine the start point and endpoint of the signal, and then we cut the silent part of the signals.

Fig. 6 shows the spectrum and endpoint results of the signal. Fig. 6 (left) shows the spectrum result, of which the horizontal axis represents time parameters, while the vertical axis represents the frequency parameters. As we can see from Fig. 6 (left), the distribution of the frequency is changing with the time variant. The gradation of color shows the intensity of the frequency at that time. Fig. 6 (right) shows the detected endpoints of our algorithm, and the vertical red lines represent the start time and end time of a note.

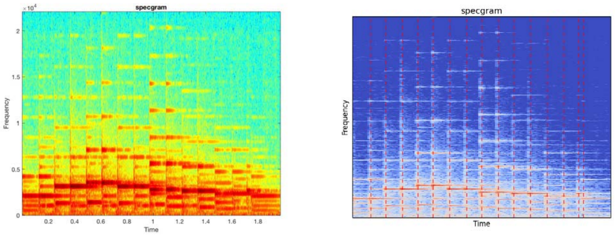


Fig. 6. Left: spectrum. The darker the color, the stronger the intensity. Right: endpoint results. Dashed lines: endpoints of notes. (X-axis represents time and Y-axis represents the frequency intensity.)

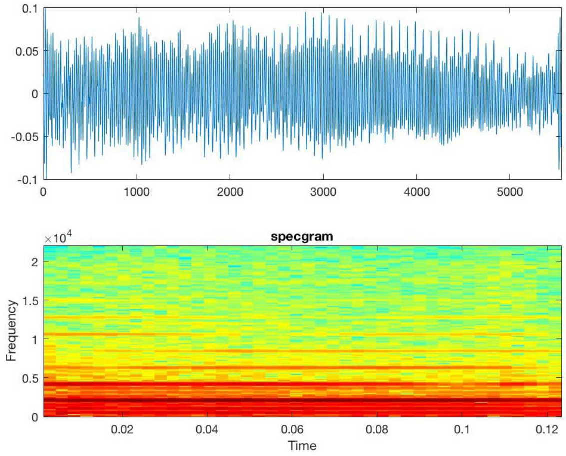


Fig. 7. Top: signal of a note. X-axis represents time and Y-axis represents the signal intensity that has dynamic range (−0.1–0.1). Bottom: frequency of a note. X-axis represents time and Y-axis represents the frequency intensity, the darker the color, the stronger the intensity.

After detecting the endpoint of a note, we continue the analyzing of notes. Take a note as example, shown in Fig. 7, we use the start point and endpoint of the note, and the signal is shown in Fig. 7 (top), while the frequency is shown in Fig. 7 (bottom). We can calculate the average frequency of the duration, which can be defined as the end time minus the start time. As we can see from Fig. 7, the frequency is 2100 Hz and the time coordinate of the note is 5550. As the sample frequency is 441 000 Hz, the duration of the note can be calculated as 0.12 s. Other notes can be analyzed identically.

During the training phase, we use labeled audio files of different notes to train our note model, which frequency is the main parameter, and we get the training results as the input of our final models. Meanwhile, we use the base frequency of piano as a parameter to revise our model [33], which is shown in Table I. We use a weighted average of training frequency and base frequency, as shown in the following equation:

$$\text{frequency} = a\sum_{1}^{n} f0_i/n + b * f1. \tag{12}$$

The frequency is the final result of our note model, and f0 represents the base frequency of a note, while f1 represents the training frequency, and a and b represent the weights of frequency, which sums to 1. After the calculation, we save the results of different note models in our databases.

In the test phase, we use the Euclidean distance of notes to show the possibility of a test note being the base note.

TABLE I
NOTES FREQUENCY OF THE PIANO

Notes	Octaves O1	O2	O3	O4	O5	O6	O7	O8
A	27.500	55.000	110.000	220.000	440.000	880.000	176.000	3520.000
B^b	29.135	58.270	116.541	233.082	466.164	932.328	1864.655	3729.310
B	30.868	61.735	123.471	246.942	493.883	987.767	1975.533	3951.066
C	32.703	65.406	130.813	261.626	523.251	1046.502	2093.004	4186.009
C#	34.648	69.296	138.591	277.183	554.365	1108.731	2217.461	
D	36.708	73.416	146.832	293.665	587.330	1174.659	2349.318	
E^b	38.891	77.782	155.563	311.127	622.254	1244.598	2489.016	
E	41.203	82.407	164.814	329.629	659.255	1318.520	2637.020	
F	43.654	87.307	174.614	349.228	698.456	1396.913	2793.826	
F#	46.249	92.499	184.997	369.994	739.989	1479.978	2959.955	
G	48.999	97.999	195.998	391.995	783.991	1567.982	3135.437	
G#	51.913	103.826	207.653	415.305	830.609	1661.219	3322.437	
	O1	O2	O3	O4	O5	O6	O7	O8

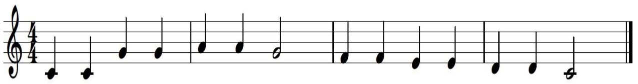


Fig. 8. Music score of the audio signal.

We choose the minimum distance of a note, and we consider the testing note as the identical note. Finally, we get the score of the music signals (as shown in Fig. 8).

The above algorithm is based on the hypothesis that there is only a single note at the same time. Considering the chord fragment, which is common in the symphony or chorus scene, the variety of musical instruments are hard to recognize; meanwhile, there are more than one note at a time, which needs to be separated from different instruments and different notes simultaneously. As there may be reverberation of different instruments and the overlap of notes, we will further enhance the methods to optimize the performance.

D. Discussion

There are some relevant music platforms. Pandora is the largest online radio station, which provides users with customized and high-quality recommendation [34]. Pandora brings unexpected surprise for users without specific requirements, but may not be suitable for other users who know what they want. Last.fm is a music service platform, focusing on music social networking, and it offers multiple services to the clients [35]. The main feature of Last.fm is the establishment of similar tastes; users can share lists with others. There are many other platforms such as Spotify [36] which offers a large music library. Pandora focuses on the gene of music, and Last.fm focuses on social networking. As our system has similar requirements with Pandora and Last.fm, the comparison between the Pandora, Last.fm, and our system may help a lot in our future work. We give a comparison of the Pandora, Last.fm, and our system in Table II.

TABLE II
COMPARISON WITH OTHER MUSIC PLATFORM

Aspects		Pandora	Last.fm	Ours
Data	Collection	Experts analysis	Users sharing	User input and historical data
	Data Model	Not mention	Not mention	Music Data Model, combining all the information of songs
Service	Listen	Yes	Yes	Yes
	Purchase	Free	Free	Free
	Download	Yes	Yes	Yes
	Retrieval	No	No	Yes
	Score generation	No	No	Yes
	Recommendation	Yes, based on gene	Yes, based on user behavior	Yes, based on multiple strategies
Clients Interaction		Professional connoisseurship	Social purposes	Customized requirements
Optimization		Better gene analysis algorithm	Audience and listen behaviors increment	Datasets expanding improvement in algorithm and distribution of servers

As shown in Table II, we compare the three systems in the following aspects.

1) Data: The data used in these platforms are collected in different ways. The collection of Pandora is made by experts analyzing the music gene and labeling the music with the tags. However, the data that Last.fm used are collected by users sharing and discussion. The data in our system are the combination of user input and historical data. Meanwhile, our system defines a data model for music songs, which can help the further research.

2) Service: The services of the three platforms include music listening, music purchase, music downloading, music retrieval, music score generation, and music recommendation. Pandora and Last.fm cannot provide the retrieval and score generation services, which our system provided. As for the recommendation strategies, Pandora relies on the analysis of music gene, and Last.fm concerns more about user behavior, while our system combines multiple strategies, which enhances the results of recommendation.

3) Clients Interaction: Clients of these platforms involve a different people. Pandora provides high-quality analysis, which can satisfy the requirements of clients with professional connoisseurship. Last.fm is suited for the clients with social purposes. By discussing with other people and expressing their views on the system, the clients can have better experience. Our system, which can provide customized services, is designed for different types of clients. By expressing their requirements through the GUI, our system can interact with clients conveniently.

4) Optimization: The platform can be optimized through different aspects. Pandora focuses on the analysis of music gene, while Last.fm can be optimized with the increment of audience and the number of songs that the clients have listened. Our system can be improved through the expanding of data sets, and the inner algorithms and the distribution of the servers can be improved as well.

From the results, our system can provide better music services for clients. The data model can contribute to the research of music filed definitely.

VI. Conclusion

In this paper, a system is proposed to cognize the music songs automatically. We define a data model to represent a song, which is the basis of our system. The data collection and model establishment are deployed on the fringe nodes, which reduce the load of middle center of the server; then, machine learning and statistical analysis methods are performed to carry out the task of music applications in service layer. A case study is given to demonstrate the availability of our system. After that, we make a comparison between our platform and other platforms. The results show the effectiveness of our system.

However, there are still things to be done to optimize our system. First, the work we have done is focused on single musical instrument, which means that it cannot support the scenarios of multiple instruments, which are common situations in band management; meanwhile, the recognition of chord in the audio signals is still a serious problem. Second, the experiment largely depends on the sample data; the parameters we calculated also rely on the data we collected. Therefore, if incomplete data sets are used, we may get inaccurate results.

The system can be further optimized in the following directions. First, the performance of the system could be optimized in the multiple instruments and chord scenarios by redesign our algorithms. Second, we can expand the data sets to make the analysis more representative, and the data sets of training could also be promoted with the help of experts. Furthermore, the structure of the system can be optimized to achieve better performance.

References

[1] L. Jiang, J. Wang, N. Shah, H. Cai, C. Huang, and R. Farmer, "A process-mining-based scenarios generation method for SOA application development," *Service Oriented Comput. Appl.*, vol. 10, no. 3, pp. 303–315, 2016.

[2] L. Da Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[3] A. Krylovskiy, M. Jahn, and E. Patti, "Designing a smart city Internet of Things platform with microservice architecture," in *Proc. 3rd Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2015, pp. 25–30.

[4] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, "IoT-based big data storage systems in cloud computing: Perspectives and challenges," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 75–87, Jan. 2017.

[5] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social Internet of Things (SIoT)—When social networks meet the Internet of Things: Concept, architecture and network characterization," *Comput. Netw.*, vol. 56, no. 16, pp. 3594–3608, 2012.

[6] S. K. Datta, C. Bonnet, and J. Haerri, "Fog computing architecture to enable consumer centric Internet of Things services," in *Proc. IEEE Int. Symp. Consum. Electron. (ISCE)*, Jun. 2015, pp. 1–2.

[7] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st MCC Workshop Mobile Cloud Comput.*, 2012, pp. 13–16.

[8] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in *Proc. Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2014, pp. 464–470.

[9] D. S. Modha *et al.*, "Cognitive computing," *Commun. ACM*, vol. 54, no. 8, pp. 62–71, Aug. 2011.

[10] S. K. Esser *et al.*, "Cognitive computing systems: Algorithms and applications for networks of neurosynaptic cores," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2013, pp. 1–10.

[11] K. Umapathy, S. Krishnan, and R. K. Rao, "Audio signal feature extraction and classification using local discriminant bases," *IEEE Trans. Audio, Speech, Language Process.*, vol. 15, no. 4, pp. 1236–1246, May 2007.

[12] B. Liang, H. Yaali, L. Songyang, C. Jianyun, and W. Lingda, "Feature analysis and extraction for audio automatic classification," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 1, Oct. 2005, pp. 767–772.

[13] P. Rai, V. Golchha, A. Srivastava, G. Vyas, and S. Mishra, "An automatic classification of bird species using audio feature extraction and support vector machines," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, vol. 1, Aug. 2016, pp. 1–5.

[14] R. Goecke, G. Potamianos, and C. Neti, "Noisy audio feature enhancement using audio-visual speech data," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 2, May 2002, pp. II-2025–II-2028.

[15] K. Helwani, S. Spors, and H. Buchner, "Spatio-temporal signal preprocessing for multichannel acoustic echo cancellation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 93–96.

[16] K. Choi, G. Fazekas, K. Cho, and M. Sandler. (2017). "A comparison on audio signal preprocessing methods for deep neural networks on music tagging." [Online]. Available: https://arxiv.org/abs/1709.01922

[17] B. Logan and A. Salomon, "A music similarity function based on signal analysis," in *Proc. Null*, Aug. 2001, p. 190.

[18] V. Viro, "Peachnote: Music score search and analysis platform," in *Proc. ISMIR*, 2011, pp. 359–362.

[19] J. Haitsma, T. Kalker, and J. Oostveen, "Robust audio hashing for content identification," in *Proc. Int. Workshop Content-Based Multimedia Indexing*, vol. 4, Sep. 2001, pp. 117–124.

[20] M. Clausen and F. Kurth, "A unified approach to content-based and fault-tolerant music recognition," *IEEE Trans. Multimedia*, vol. 6, no. 5, pp. 717–731, Oct. 2004.

[21] R. B. Dannenberg and C. Raphael, "Music score alignment and computer accompaniment," *Commun. ACM*, vol. 49, no. 8, pp. 38–43, 2006.

[22] S. Kolozali, M. Barthet, G. Fazekas, and M. Sandler, "Automatic ontology generation for musical instruments based on audio analysis," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 10, pp. 2207–2220, Oct. 2013.

[23] W.-C. Chang and A. W. Y. Su, "A novel recurrent network based pitch detection technique for quasi-periodic/pitch-varying signals," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 1, May 2002, pp. 816–821.

[24] D. G. Bhalke, C. B. R. Rao, and D. S. Bormane, "Dynamic time warping technique for musical instrument recognition for isolated notes," in *Proc. Int. Conf. Emerg. Trends Elect. Comput. Technol. (ICETECT)*, Mar. 2011, pp. 768–771.

[25] M. P. Ryynanen and A. Klapuri, "Polyphonic music transcription using note event modeling," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, Oct. 2005, pp. 319–322.

[26] A. I. Tamboli and R. D. Kokate, "An effective optimization-based neural network for musical note recognition," *J. Intell. Syst.*, to be published.

[27] J. de Jesus Guerrero-Turrubiates, S. E. Gonzalez-Reyna, S. E. Ledesma-Orozco, and J. G. Avina-Cervantes, "Pitch estimation for musical note recognition using artificial neural networks," in *Proc. Int. Conf. Electron., Commun. Comput. (CONIELECOMP)*, Feb. 2014, pp. 53–58.

[28] A. Bietti, F. Bach, and A. Cont, "An online em algorithm in hidden (semi-)Markov models for audio segmentation and clustering," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 1881–1885.

[29] H. X. Li and L. D. Xu, "Feature space theory—A mathematical foundation for data mining," *Knowl.-Based Syst.*, vol. 14, nos. 5–6, pp. 253–257, 2001.

[30] A. Perez-Carrillo and M. M. Wanderley, "Indirect acquisition of violin instrumental controls from audio signal with hidden Markov models," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 5, pp. 932–940, May 2015.

[31] P. Patel, A. Chaudhari, R. Kale, and M. Pund, "Emotion recognition from speech with Gaussian mixture models & via boosted GMM," *Int. J. Res. Sci. Eng.*, vol. 3, no. 2, pp. 47–53, 2017.

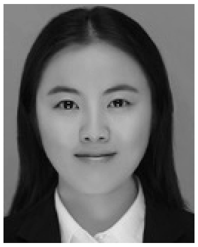
[32] D. Povey *et al.*, "The subspace Gaussian mixture model—A structured model for speech recognition," *Comput. Speech Lang.*, vol. 25, no. 2, pp. 404–439, 2011.

[33] *The Pitch and Corresponding Frequency of Piano*. Accessed: Sep. 17, 2018. [Online]. Available: https://wenku.baidu.com/view/68ca9643cf84b9d528ea7adf.html

[34] *Pandora Internet Radio—Listen to Free Music You'll Love*. Accessed: Sep. 17, 2018. [Online]. Available: https://www.pandora.com/

[35] *Last.fm | Play Music, Find Songs, and Discover Artists*. Accessed: Sep. 17, 2018. [Online]. Available: https://www.last.fm/

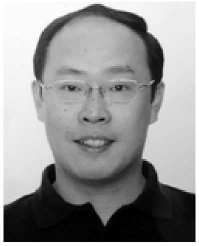
[36] *Music for Everyone—Spotify*. Accessed: Sep. 17, 2018. [Online]. Available: https://www.spotify.com/us/



Lifei Lu was born in Anyang, China, in 1993. She received the B.E. degree in computer science and technology from Wuhan University, Wuhan, China, in 2016. She is currently pursuing the master's degree with the School of Software, Shanghai Jiao Tong University, Shanghai, China.

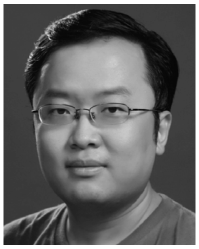
Lida Xu (M'86–SM'11–F'16) received the B.S. degree in information science and engineering and the M.S. degree in information science and engineering from the University of Science and Technology of China, Hefei, China, in 1978 and 1981, respectively, and the Ph.D. degree in systems science and engineering from Portland State University, Portland, OR, USA, in 1986.

Dr. Xu is an Academician of the European Academy of Sciences and the Russian Academy of Engineering. He is a 2016, 2017, and 2018 Highly Cited Researcher in the field of engineering named by Clarivate Analytics.



Boyi Xu received the B.S. degree in industrial automation and the Ph.D. degree in management science from Tianjin University, Tianjin, China, in 1987 and 1996, respectively.

He is currently an Associate Professor with the College of Economics and Management, Shanghai Jiao Tong University, Shanghai, China. His current research interests include enterprise information systems, Internet of Things, and business intelligence.



Guoqiang Li received the B.S. degree from the Taiyuan University of Technology, Taiyuan, China, in 2001, the M.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 2005, and the Ph.D. degree from the Japan Advanced Institute of Science and Technology, Nomi, Japan, in 2008.

He was an Assistant Professor with the School of Software, Shanghai Jiao Tong University, from 2009 to 2013, where he is currently an Associate Professor with the School of Software.

Hongming Cai (M'08–SM'15) was born in Liupanshui, Guizhou, China, in 1975. He received the B.S., M.S., and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, China, in 1996, 1999, and 2002, respectively.

He is currently a Professor with the School of Software, Shanghai Jiao Tong University, Shanghai, China.

Dr. Cai is a Senior Member of ACM and China Computer Federation. He is a Standing Director of the China Graphics Society, Beijing, China.