Neural Computing and Applications
https://doi.org/10.1007/s00521-024-10555-x

REVIEW

Artificial intelligence in music: recent trends and challenges

Jan Mycka[1] · Jacek Mańdziuk[1,2]

Received: 17 January 2024 / Accepted: 3 October 2024

Abstract
Music has always been an essential aspect of human culture, and the methods for its creation and analysis have evolved alongside the advancement of computational capabilities. With the emergence of artificial intelligence (AI) and one of its major goals referring to mimicking human creativity, the interest in music-related research has increased significantly. This review examines current literature from renowned journals and top-tier conferences, published between 2017 and 2023, regarding the application of AI to music-related topics. The study proposes a division of AI-in-music research into three major categories: music classification, music generation and music recommendation. Each category is segmented into smaller thematic areas, with detailed analysis of their inter- and intra-similarities and differences. The second part of the study is devoted to the presentation of the AI methods employed, with specific attention given to deep neural networks—the prevailing approach in this domain, nowadays. In addition, real-life applications and copyright aspects of generated music are outlined. We believe that a detailed presentation of the field along with pointing out possible future challenges in the area will be of some value for both the established AI-in-music researchers, as well as the new scholars entering this fascinating field.

Keywords Music generation · Music classification · Music recommendation · AI in music

1 Introduction

Music has been a fundamental aspect of human culture for centuries [1]. It is an universal language that can be understood by all, serving as a conduit for communication and the sharing of ideas [2]. Being an essential part of most people's everyday lives, music allows them to express themselves and convey their emotions, whether by composing, performing or listening [3].

With the development of artificial intelligence (AI), music has become one of the fields in which AI is being tested and developed. Initially, the problems of representing music data [4–6] or generating single notes (sounds) [7] were addressed. In the following years, the range of problems has been significantly extended, and the most popular problems have become generation of larger fragments of music or music classification [8–11]. In recent years, music recommendation or music search issues have also gained importance [12, 13]. The increased interest is related to the growing popularity of various music streaming services.

In all fields, the performance of AI is steadily improving, approaching or even sometimes surpassing the performance of humans. At the same time, especially in the case of music generation, there is a problem of the creation of objective tools for performance evaluation, which makes a comparison of AI models significantly more difficult [14]. Despite the continuous development of AI in the field of music, there are still unsolved problems or new challenges arising. Among others, generating long fragments of expressive music that can imitate humans even in the eyes of experts is an example of such an unsolved problem.

1.1 Motivation and goals

Music, during both creation and performance, requires people to use their creativity, as well as the ability to adjust to the musical rules developed over centuries. Solutions to

✉ Jan Mycka
jan.mycka.dokt@pw.edu.pl

Jacek Mańdziuk
jacek.mandziuk@pw.edu.pl

[1] Warsaw University of Technology, Warsaw, Poland

[2] AGH University of Krakow, Krakow, Poland

Published online: 16 November 2024

music-related problems that use AI methods help to illustrate the level at which the research on human-like AI is advanced.

The use of artificial intelligence in music has gained momentum and has been intensively explored in recent years. With this in mind, it is crucial to systematically describe the ongoing state of knowledge in this area to establish a basis for its further development. This review covers a wide range of music-related problems and includes topics from various areas of AI applications in music. The succinct presentation of music-related issues and their solutions enables efficient comparisons among various problems and methods and facilitates the identification of underexplored research areas or approaches. The scope of the review also encompasses possible research paths, open problems and challenges.

1.2 Scope

In recent years, a wide range of musical topics from different fields have been undertaken. This diversity makes it difficult to divide them into just a few categories. Generally, the following three groups that cover the majority of the topics can be distinguished:

- *Music classification* topics related to recognizing attributes of music (such as the performer, the composer, a music genre, etc.).
- *Music generation* topics related to the artificial creation of music, including the generation of both sheet music and generation of audio pieces of various types. The subject of copyright in relation to the generative models is also addressed.
- *Music recommendation* topics related to the recommendation of music based on identified preferences. Preferences can refer to the context in which the recommended music is located and/or music content.

When it comes to solution methods, the most popular technique, that clearly stands out, is neural networks, followed by genetic algorithms.

Comprehensive review papers focusing on the use of AI in music are relatively old and are dated at 2013 [9], 2012 [13] 2008 [12], 2007 [8], or focus on particular areas or methods [10, 11]. Since then, there have been published numerous papers worth noticing, presenting various applications of AI methods in the field. This review brings together papers published at leading conferences (i.e. *NeurIPS*, *AAAI*, *ICML*, etc.) from 2017–2023 and in leading journals (i.e. *Proceedings of the IEEE*, *Artificial Intelligence*, etc.) from 2005 to 2023. Relevant papers published in other venues are also included. Table 1 summarizes the collected papers relative to years, and Table 2 lists the conferences and journals considered.

This survey is continued by reviewing existing topics and problems in Sect. 2. To increase the practical value of the paper, the topic of real-world applications is briefly addressed in Sect. 3. Section 4 categorizes the most commonly used methods in solving the presented problems. Next, Sect. 5 describes challenges that have emerged in recent years. Finally, Sect. 6 concludes the paper.

1.3 Music representation

Before music data can be used by any algorithm, it must be represented in an appropriate way that determines which features can be extracted from a music piece and subsequently processed. For this reason, choosing the right representation can be crucial in solving problems. There are three most commonly used representations that allow to obtain different musical properties: *waveform* model, *MIDI* representation and *piano-roll* representation. Figures 1 and 2 depict different representations of the same piece of music.

1.3.1 Waveform

Waveform records signal amplitude as a function of time. An important parameter of the model is the sampling rate, which is the number of recorded amplitude values per second (usually, sampling rate is $44.1kHz$ or $48kHz$). Since a naturally emitted sound is actually composed of several superimposed sound waves, it is not possible to recover pitches directly from the signal. The most common representation derived from the model is the spectrogram, which is calculated using the Fourier transform. The model preserves the expressiveness of the music performance. Waveform and spectrogram representations are presented in Fig. 2 (bottom and middle part, respectively).

1.3.2 MIDI format

MIDI stores the musical data, not the audio recording itself. The basic information stored is the pitches of the notes, the duration of the notes, and when these notes started. In addition, MIDI can store a lot of detailed information about notes, such as *velocity value* (the intensity the note was played with). With all this information, the MIDI format also preserves the expressiveness of the performance.

1.3.3 Sheet representation

Sheet representation is the most common method of music representation. It is created directly by the composer and does not preserve the expressiveness of the performance in any way. In some cases, in addition to the notes themselves, performance guidelines that the musician should

Table 1 The number of considered papers grouped by their publication year. Only articles referring to music content are counted, i.e. works describing machine learning models and/or methods with no direct relation to music are excluded

2023	2022	2021	2020	2019	2018	2017	2011–2016	2005–2010
11	7	21	16	18	12	16	10	10

Table 2 Summary of leading conference and journal source venues. Only articles referring to music content are counted, i.e. works describing machine learning models and/or methods with no direct relation to music are excluded

Journals	Conferences
Proceedings of the IEEE	AAAI
Artificial Intelligence	IJCNN
Neurocomputing	IJCAI
Computer Music Journal	ICML
Journal of New Music Research	ISMIR
IEEE Transactions on Neural Networks and Learning Systems	ICLR
International Journal of Approximate Reasoning	ICONIP
Machine Learning	ICCS
Neural Computing and Applications	NeurIPS
IEEE Transactions on Evolutionary Computation	GECCO
ACM Transactions on Interactive Intelligent Systems	KDD
Fuzzy Sets and Systems	FUZZ-IEEE
EURASIP Journal on Audio, Speech, and Music Processing	
Advances in Data Analysis and Classification	
Cognitive Computation	

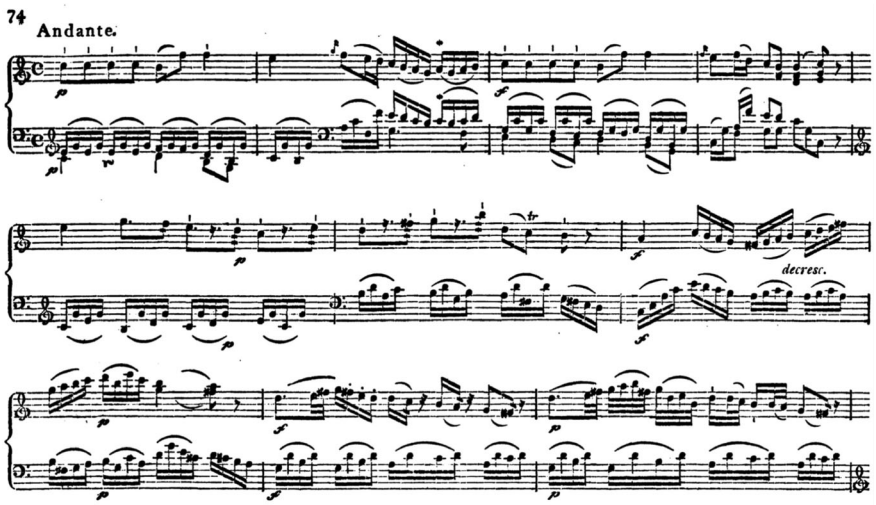


Fig. 1 Sheet representation of the beginning of *W. A. Mozart Piano Sonata No. 5, KV. 283, II. Andante*

follow are also included. However, the final interpretation of these instructions is up to the performer of the piece. An example of sheet representation is presented in Fig. 1.

1.3.4 Piano-roll

Piano-roll is a simple representation specifying only the pitches of notes and their arrangement in time. It can be presented as a two-dimensional graph with pitches of notes on the vertical axis and time on the horizontal axis. A played note is indicated by a horizontal line at the height of the corresponding pitch. In a piano-roll representation, an expressive aspect that can be preserved, if the resolution is high enough, is the note's duration and timing. On the other hand, too low resolution may corrupt the original data. An example of a piano-roll representation is illustrated in Fig. 2 (top part).

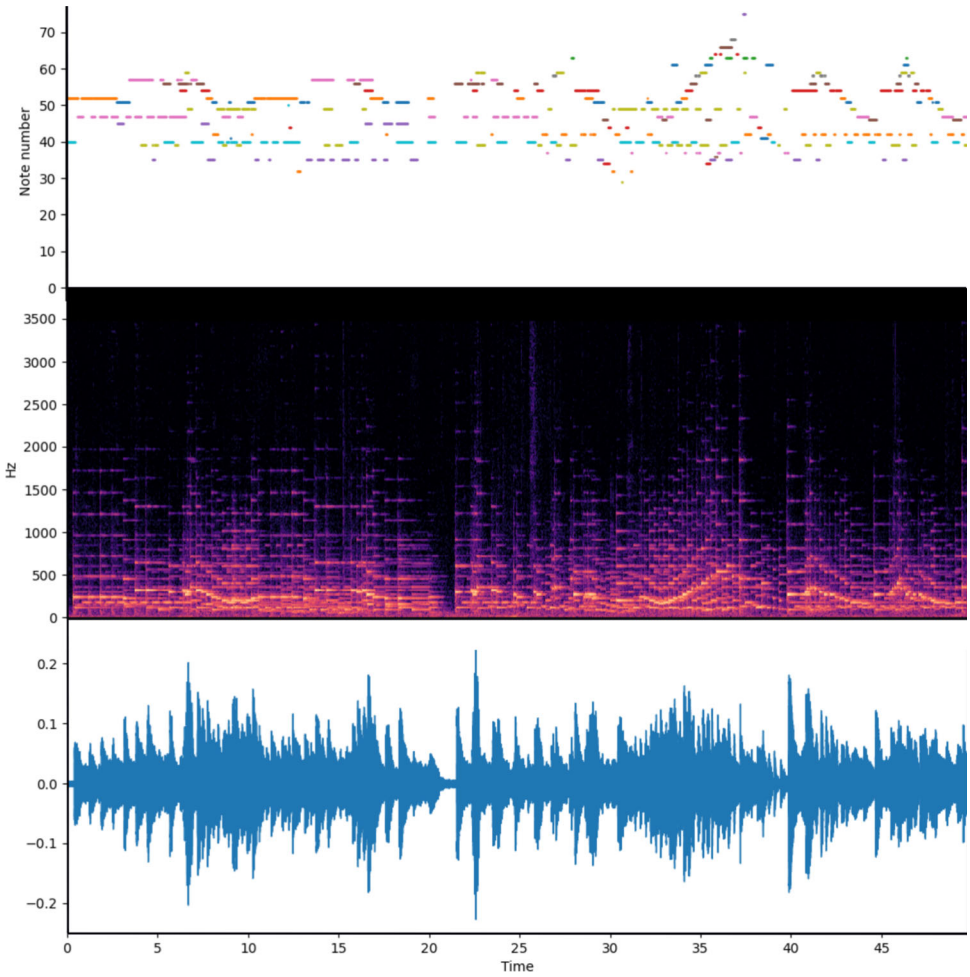


Fig. 2 Three different representations of the beginning of *W. A. Mozart Piano Sonata No. 5, KV. 283, II. Andante*: piano-roll, spectrogram, waveform (from top to bottom, respectively)

2 Music-related problems and solutions

One possible categorization of music-related problems is by the topic addressed. A classification of problems considered in this review is presented in Fig. 3. A summary of the number of papers considered for each problem type is presented in Table 3.

Table 3 The number of papers included in the review, classified according to the type of problem addressed

	Number of papers
Music generation (sect. 2.1)	53
Music classification (sect. 2.2)	26
Music recommendation (sect. 2.3)	21
Other problems (sect. 2.4)	5

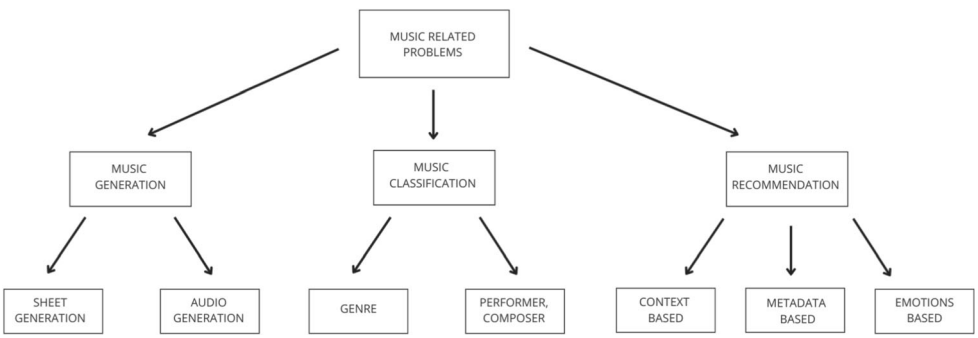


Fig. 3 Groups and subgroups of music-related problems

2.1 Music generation

The problem of music generation consists in creating music in any of its forms i.e. sound, notes or other representation. Music generation specifically relates to *creativity*—a *very human* ability. For this reason, the problem has attracted a lot of attention since the onset of AI application to music [9, 10, 15, 16]. Two aspects of the music generation should be distinguished. One is generation of new musical compositions (e.g. generation of sheet music), and the other one is generation of expressive performance (referring to already existing works). Both aspects emulate human creativity process. This section discusses papers that address various problems associated with the generation of music—see Table 4 for their summary.

2.1.1 Harmonization and counterpoint generation

This section includes papers revolving around the subject of harmonization and accompaniment generation. Problems considered in this area are strongly connected to music theory, and created solutions are significantly based on music theoretical principles.

Harmonization [17–19] consists in complementing one melodic line (sequence of notes) with three remaining lines. The newly created melodic lines complement the given one and are played simultaneously with this line. This makes it possible to consider the problem of harmonization in two perspectives. The first one (horizontal) refers to melodic lines, and the second one (vertical) to chords, i.e. notes located at the same time moment in all four melodic lines. It is crucial that both of these perspectives are internally coherent. Figure 4 presents an example.

The paper [17] approaches harmonization problem based on Markov Decision Process, where the reward value is calculated based on the fulfilment (positive reward) or violation (negative reward) of defined rules drawn from music theory and musical common knowledge. The reward is determined for the chord to complete the vertical plane under the currently considered note from the given melodic line (the highest one).

In [18] harmonization is split into two subproblems. The first of them is matching suitable harmonic functions (defining a possible chord) to the notes given in the lowest melodic line. The second one is completing the notes in the remaining melodic lines. The evaluation of both stages is also based on the principles of music theory.

In [20, 21] only the second part of the division presented in [18], i.e. creation of new melodic lines based on the harmonic functions, is considered. The solutions are also grounded in the music theory, which forms a basis for an objective function of the implemented evolutionary algorithm.

Harmonization is also addressed in [19] where, in contrast to [17, 18], the evaluation of the created solution is not only based on rules theory-related rules but also on a statistical analysis of Bach chords (the evaluation of the settled consecutive harmonic functions). This analysis allows determining the frequency of chord sequences in the chorales, and on this basis, numerical scores are assigned to these sequences.

A problem similar to harmonization is the *generation of counterpoint or accompaniment* [22, 23]. As in harmonization, the task is to create a new melody that complements the existing melody. In this case, however, a single melodic line is created, with greater focus on internal coherence and sound.

An interesting case combining counterpoint creation and music style transfer is described in [22], where the counterpoint (Western music style) is created for Chinese folk melodies. The generating model is based on two elements: evaluation of the Chinese music style (internal evaluation of the created melodic line) and evaluation of the counterpoint (evaluation of both melodic lines simultaneously). Similarly to [19], the evaluation is based on the analysis of Bach chorales (the counterpoint) and additionally the selected folk melodies (the Chinese style).

In [23], the problem of creating real-time (online) accompaniment is considered. As in [19, 22], the model that selects notes for the accompaniment is not based on music rules, but on a prior analysis of mono- and polyphonic music. This preserves both the internal consistency of the accompaniment and the consistency of the accompaniment with the leading melody.

Table 4 Summary of papers that address various aspects of the music generation problem

	Number of papers	Papers
Harmonization and counterpoint generation (sect. 2.1.1)	9	[17–25]
Non-expressive music generation (sect. 2.1.2, 2.1.3)	18	[26–43]
Expressive music generation (sect. 2.1.4)	10	[44–53]
Other problems (sect. 2.1.5)	10	[54–63]
Performance evaluation (sect. 2.1.6)	6	[33, 64–68]

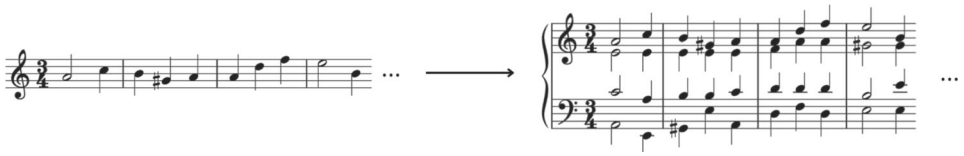


Fig. 4 Harmonization problem. Top melodic line is complimented by the three other lines

Harmonization is one of the central problems in music [17–19, 22, 23]. Consequently, there are established rules that define the correctness of the created harmonizations, which can be used when creating models that solve this problem [17–19]. On the other hand, models can also be based on the analysis of other works and the relationships between notes or chords that occur there [19, 22, 23].

A problem simpler than constructing entire melodic lines is the prediction and generation of the chord sequences [24, 25] (see Fig. 5). The problem can be regarded as a kind of harmonization subproblem (cf. [18]) since it does not deal with the arrangement of particular notes in melodic lines, but only with their definition.

The work [24] addresses the problem of creating a *chord progression*, that is, a smooth (in terms of sound) chord transition from one key to another. The final choice of the next chord in a sequence is up to the user, but the model, on the basis of the already created fragment, proposes several possibilities. The choice of the proposal is made based on the distances of chords in the created space, in which harmonically similar chords are located close to each other.

A similar approach is taken by [25], focusing on the prediction of the next chord based on the previous eight chords. In this case, the chord distance metric is constructed based on Tonnetz lattice [69].

2.1.2 Composing a music piece

A task more demanding than harmonization or accompaniment generation is to create music from scratch. Composing entire pieces additionally requires the creation of a main melody, which should be both good-sounding and engaging. Creating such a melody is a much more complex task than harmonization for both the human and algorithmic approaches. This section includes papers revolving around the subject of generating a single melody line or an entire music piece in musical notation.

The problem of creating a melodic line has been considered in many works, e.g. [26–29]. In [27], a melody is constructed using an evolutionary algorithm, where the matching function is based on created scoring tables for consecutive notes. In [26], on the other hand, in addition to generating a melody, an accompaniment (defined by harmonic functions) is matched to it. Additionally, the work attempted to generate music that reflects a particular emotion, specified by the user. Toward this end, a set of chords with associated emotions, based on people's feelings when they listen to them, was generated, based on expert knowledge.

An intriguing technique for creating melodic line with harmonic progression was introduced in [29], which employed Schenkerian analysis [70] as the foundation of the generation process and mimicked human composition process. In the initial step, the music's phrase structure is devised, and the rhythmic parameters are determined (either based on a statistical analysis of the musical literature or given by the user). Based on those elements, harmonic progression and melodic rhythm are crafted, which are then utilized to devise the melody contour and ultimately the melody itself. All stages are executed through heuristics (rhythm generation, final melody generation) or the employment of Probabilistic Context-Free Grammar (harmonic progression, melody contour), which is based on a conducted Schenkerian analysis of comparable examples.

An even more advanced problem is addressed in [28]. The task is to generate melody and accompaniment (not as chord definitions, but in the form of melodic lines), based on a defined chord progression. The system initially generates a single melodic line (pitches and rhythm), and then, based on the created melody, an accompaniment for various instruments is generated.

Generating the main melody or a whole piece of music is a multi-aspect problem. The proposed solutions usually focus on some of these aspects, as it is hard to take them all into account simultaneously. An aspect often addressed is keeping the generated piece in a particular style (e.g. pop music, music imitating particular composer, etc.) [28, 30–33]. Another frequently encountered issue is coherence on the musical structure of the generated piece itself (the long-term internal consistency of the piece) [34–36].

[30] focuses on the problem of automatic generation of pop music (a melody). An important part of the problem is

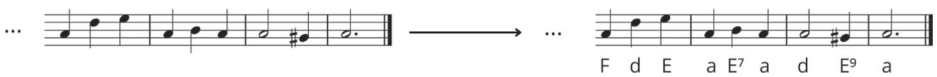


Fig. 5 Chord sequence generation problem. For a given melodic line a sequence of chords is generated

the initial generation of the description of the song structure. The structure description contains indications about repeated sequences in the song (repetition of rhythm or repetition of both rhythm and melody). Then, based on the created description and a given chord progression, a melody is generated.

[31] has at its core the extraction of information about a particular musical style (experiments have been conducted on Hungarian folk music). Various properties are extracted from the melodies of a particular style and based on them a fitness function is constructed to check the similarity of the subsequently generated melodies (their representations) to the selected style. For the created fitness function and the developed binary representation of the melody, an evolutionary algorithm is run, with the task of finding a representation that best reflects the given style.

Similarly to [31], music generation in [32] consists in adaptation to a particular style. Here, however, the constructed model is designed to work closely with the musician and be able to meet his/her expectations. Therefore, the style of music is determined by the musician-composer (a human), who provides a collection of other pieces that reflect his/her intentions regarding the style of the generated music. In addition, the task of the musician-composer is to provide the chord progression to which the created melody will be matched. From the indicated set of base songs, two sensibility models are induced, one corresponding to the rhythm and the other one to the flow (motion) of the melody. Using the evolutionary algorithm and the induced models, a musical template is then generated, consisting of a description of the rhythm and the flow of the melody. This template mimics the musical style of the collection of selected songs. The final step is to add the pitches themselves to the created template. These pitches are generated using a pitch selection strategy based on the chord progression provided by the musician-composer.

Generation in a specific style is also undertaken in [33]. Here, the model (variational autoencoder) is taught to imitate a specific musical genre or composer, according to the developed similarity measure (based on Mahalanobis distance) that allows to determine to what extent the generated pieces actually reflect the characteristic features of a particular style or genre.

Preserving the long-term structure of the generated music is the focus of [34–36]. This aspect is addressed through construction of an appropriate representation of music data and/or the use of an AI model (e.g. a recurrent neural network) that allows to notice structural nuances (also in the temporal context) of the presented data.

In [34], the music is regarded as a sequence of events, and each event contains information about one note (pitch, duration, corresponding chord). A neural network model is developed, which includes multiple hierarchical tiers. Each tier processes a unique range of event data, and the range widens as the tiers ascend. This technique effectively captures the long-term structure at higher tiers and predicts future events at the lowest tier with conditioning from higher tiers. The bottom tier implements 1D convolutions to process events, while all higher tiers utilize LSTM (Long Short-Term Memory) to handle grouped event data.

In [35] the problem of maintaining the long-term consistency of the generated music is addressed by selecting a specific architecture of the AI model, which is an autoencoder. An important part of the model falls on the decoder, where decoding process is performed in several steps. From the input data of the decoder, vectors corresponding to shorter fragments of the melody are produced. Specific notes are then reproduced from these vectors. The transferred vectors together with the information about the previous note are used to generate the current note.

[36] also considers the aspect of maintaining a coherent structure of a song. Here too, analogously to [34], music is considered to be a sequence of events, and the task is to generate successive events in the sequence (using a transformer neural network). An additional element is modelling the expression of the generated music through a set of appropriate tokens (e.g. representing the velocity of a note).

Looking at music as a sequence of events is a common approach [34, 36–39]. In [37, 38], the methods very similar to [36] are used, i.e. the model (transformer neural network) generates successive events that (sometimes after aggregation) correspond to notes in a sequence. The model proposed in [37] additionally allows several notes to occur simultaneously so that polyphonic music can be generated. In [38] the introduction of summary tokens related to entire bars, internally composed of token sequences, is a noteworthy addition. This adjustment facilitates a modification of the attention module such that only summary tokens are analysed for bars that are not structure-related (bars that are not likely to be repeated), rather than analysing individual tokens from each bar.

An interesting approach is proposed in [39] where the authors rely on the observation that music created by humans often refers to itself (is self-similar). Therefore, the prediction of the next sound in the sequence can be strongly based on already generated fragments (motifs).

2.1.3 Generating polyphonic music

A task more complicated than generating a single line (even with additional chord progression) is polyphonic music generation. Such music consists of multiple simultaneous melodic lines/tracks. Polyphonic music requires not only consistency within a single voice but also the concurrent compatibility of all voices. One of the

convenient data representations for this problem is the piano-roll, which allows for parallel, yet separate, descriptions of melodic lines of different voices/instruments. This section discusses papers revolving around the subject of generating polyphonic music.

In [40], the GAN (Generative Adversarial Network) model consisting of several parts/branches, each of them corresponding to one melodic line, is applied. The branches use the same input data (randomized noise), but process it separately. The compatibility of melodic lines is assured by the shared input data. The result is represented by means of a piano-roll format.

A GAN-based solution similar to [40] is presented in [41]. In this case, each bar for each instrument is generated independently. In order to maintain consistency (within instrument lines and between instruments), a series of vectors responsible for line consistency are generated before producing individual bars. While generating a single bar for each instrument, these vectors are combined with random vectors to maintain the consistency of the entire piece.

[42] considers the problem of generation based on an already existing one of the lines/tracks conditioned accompaniment generation. Similarly to [40, 41], the authors use a piano-roll representation. The variational autoencoder model is used, and in the encoding part, the tracks are processed separately in the initial layers to capture their internal characteristics. Then the results of the separate tracks are combined and processed together, capturing the features that bind them together into a single piece. Both the main track and the accompanying tracks are processed. An input to the decoding part is a vector from a latent space, from which the melodic lines are generated. This vector is combined with the embedding created earlier (via convolutional layers) of the conditioning line. The decoding method is the reverse of the encoding process.

The work [43] extends the problem considered [42] in by adding the aspect of dynamics of the generated piece. The standard piano-roll notation is extended to include information about the velocity of each note. In addition, the piece generation process can be influenced by the user by selecting the initial and the target pattern. Based on these two patterns, a vector is generated, from which the entire song is decoded.

In summary, generation of music (its representation) from scratch can be looked at from various perspectives, which is clearly reflected in the created models and proposed approaches. The main distinction seems to be between generation of a single melodic line (sometimes with added chord progression) [30–32, 34, 35, 39] and generation of multiple parallel lines [37, 40–43]. On the other hand, regardless of the number of melodic lines generated, emphasis is put on different aspects of the generated music. The aspects most commonly considered are generation in a specific style [30–33, 41, 43] and generation with preserving the long-term structural consistency of the music [34–36, 40–42]. There are also two views at the music itself. The first one treats music as sequences of successive events (the appearance or disappearance of a note), resulting in note-by-note generation [34, 36, 37, 39]. The other one focuses simultaneously on the entire generated fragment [30–32, 35, 40–43]. Each approach is different and requires an appropriate view of the music data itself (sometimes a significantly different representation) and a tailored model.

2.1.4 Generation of expressive music and modelling music expressiveness

Music creation, besides the composition of new pieces, in the form of sheet music, can also be regarded or manifested through music interpretation, i.e. translation of already created notation into sound with the elements of dynamics and expression added. It is also possible to consider those two creative aspects simultaneously, that is, to compose and perform music at the same time (improvisation). Deciding which aspect of music (from the point of view of AI) is more demanding is, to some extent, subjective. However, a frequently presented opinion is that attempts to imitate (human) improvisation are more challenging [71], than modelling dynamics and expression for existing notes (or their representations). Nevertheless, both of these tasks still require human-specific skills. This section considers papers focused on music improvisation (i.e. on music composing and interpreting simultaneously), as well as on modelling music expressiveness.

One way to model the dynamics of the generated piece is to adjust its sound volume on the fly. Such an approach is presented in [44]. The generating module consists of three parts: the first one deals with the chord prediction, the second one with a single note prediction, and the third one with the volume prediction. Each part draws on the results of the previous one. Thus, the prediction of a pitch (as well as of the instrument that is to play that sound) is made on the basis of the already predicted chord. Similarly, the loudness of an individual note is matched to the already generated pitch.

Another aspect of modelling the dynamics is its analysis of already created/played pieces. In [45] a tool for such an analysis of the recorded music is presented, allowing to obtain a description of various properties of a given piece/sound that are responsible for its expression. These described properties are divided into three groups (rubato, loudness and timbre) and allow to accurately describe the change of expression and piece dynamics over time.

In [46] the expression modelling is carried out using a set of functions corresponding to individual elements of the dynamics and expression (e.g. note volume, ornamentation markings, etc.). The functions created in this way also allow for an accurate description of the dynamic markings of the piece, contained in the scores themselves. Three models (linear, nonlinear and recurrent) are created to describe the variability of these functions throughout the piece. Parameters of the models are determined in the training process, based on the performances of the pieces. Trained models are able to model the dynamics of a piece based on its notes.

A similar approach to expression modelling is proposed in [47]. The process of modelling is based on the assumption that the expressive elements (volume and tempo) can be represented by curves, describing their variation over time. Each such compound curve can be decomposed into base quadratic curves (second-degree polynomials). Decomposition of a compound curve is done iteratively, and during this process parameters of the polynomials are selected. Prediction of the magnitudes of the expression elements is performed in the reverse process. Its input is a piece representation and the obtained polynomial functions. The output presents the values of the expression elements obtained after combining the polynomials.

An approach similar to [46, 47] is also proposed in [48]. For a pre-defined set of note features (describing expression), a set of rules is created to describe the mutual relationships between these notes. Each rule is represented as a binary vector, and the rules are generated using a sequential covering algorithm, until all training data are covered. Based on these rules, the expressive features of each note (and consequently of the entire piece) are then generated.

In [49], a novel representation of notes (in a piece) is used. The piece's musical score is represented as a graph that models connections between individual notes (played simultaneously, consecutively, etc.) and coded with Gated Graph Neural Network [72]. An autoencoder is used to model the expression, where the encoder part generates a latent vector based on a coded (embedded) score and expressive performance features (analogously to [46, 47]). At the same time, the decoding part is trained based on the coded (embedded) score, latent (style) vector from the encoder and initial performance features, which are inferred during the decoding to match the given style.

A different variant of the expression modelling problem is addressed in [50], where the task is to model the expression according to the user-provided preferences. A distinction is made between the two layers of expression: the first one comes from the song itself and the melody flow (neutral performance), while the second one stems strictly from the particular performance and depends on the performer's emotions that he/she wants to express. The system's task is to add the second layer of expression, having a recording that already contains the first layer as its input. The notes' expression is represented using the set of features analogous to [45–48]. Training of the system (selection of parameters of the function for feature changes relative to the neutral performance) is carried out based on the recordings provided by several musicians who played the same pieces with different emotions. Ultimately, the system has the ability to add an expressive (emotional) sound to the neutral (non-emotional) performance.

Generation of the expressive music is usually linked to its particular representation (e.g. MIDI). Rarely, as in the papers [51–53], models are created that operate with several representations.

In [53], an autoregressive model is proposed to generate a waveform. The approach allows to create stylistically consistent music fragments of about 25 s length (at 16kHz sampling).

Similarly, waveform generation is considered in [51], based on a melody representation grounded in MIDI format. Simultaneously, the model learns characteristics (timbres) of specific instruments. Once trained, the model is able to generate a waveform in the indicated timbre based on presented notes.

Rather than working with symbolic or raw audio representations, one may choose to operate on spectrograms. This approach (working essentially on the image) is utilized in [52]. The generating model consists of two modules. The first one, based on the piano-roll representation, performs a transformation to a spectrogram. The task of the second module is to fine-tune the spectrogram towards mimicking the natural sound.

In summary, generating expressive music is considered a human-specific ability. Expression in music can be divided into two layers, which is reflected in the considered problems. The first layer comes from the melody itself [44, 45, 47, 48, 71], whereas the second layer reflects the emotions of the performer (his/her individual expression) [50]. Another distinct feature is the approach to generation itself. Most of the papers deal with the problem of performance, that is, the creation of an expressive performance from the given notes [46, 47, 49, 51, 52]. A slightly more complicated problem, which is also addressed, is the problem of improvisation, that is, the simultaneous composition of both a melody and its expressive performance [44, 53, 71]. The last notable classification reflected in the discussed papers refers to a music representation they deal with. The most common are models that work with symbolic sound representation (e.g. MIDI) [46–50], for which the expression (and dynamics) of a piece is described by matching a set of additional features to the notes that define the expression (e.g.

loudness, onset time, etc.). The second option, less popular and more demanding, is to work in the raw audio domain [51, 53], where sound generation boils down to generation of a waveform. Such a representation (a waveform) contains expressive features by itself, so there is no need for attaching additional elements. The third possibility, related to a waveform representation, is generation of a spectrogram [52]. A waveform can be transformed into a spectrogram, and the reverse process is also possible. Spectrogram generation can also be looked at as an image generation process.

2.1.5 Other problems in the area of music generation

Besides the most common music generation problems discussed above, there are several other related problems that are too specific to be grouped into broader themes.

An example of such a problem is mixes/mashups creation, which involves combining several pieces of music into one piece. One possible way consists in adding smooth transitions between the pieces [54, 55]. Another possibility is to use several pieces simultaneously, so that at each moment the newly created piece is a compilation of a few pieces [56].

In [55], generation of a medley (combination of several pieces) is performed by arranging shorter fragments of mixed pieces in the appropriate order. The model is trained by sorting short fragments from a single piece, i.e. indicating whether two fragments are consecutive and which one of them occurred earlier in the piece. Generating a mix is performed by sorting fragments from more than one piece.

A similar approach to the mix-generation problem is presented in [54]. Generation of a mix is based on the analysis of pieces from an external database. Selecting and sorting the pieces in the order in which they will be arranged is based on their similarity (thematic, rhythmic and instrumental). Likewise, the selection of the point at which two pieces are combined is made through their structural analysis. The last step is the transition from one piece to another. In order to make it smooth, both pieces are played simultaneously for a certain period. Consequently, it may be necessary to adjust the tempo of the pieces (slowing down or speeding up one of them) and adjust the accents. Then the crossfading is carried out, i.e. one track is gradually muted and at the same time, the other one is gradually turning up.

In [56], the tracks to be combined are selected from a database created for this purpose. Each such piece is broken down into individual tracks (i.e. vocal, bass, percussion). The creation of a mashup begins with a random selection of the main track (vocal), to which the other two tracks are then matched. These two tracks (harmony and percussion) are selected from the database, and the merge occurs only if they are close enough (the key for the harmony track and the tempo for the percussion track) to the already selected main track. Merging is accomplished by aligning the tempo in all tracks and adjusting the key.

Yet another problem of music generation is music style transfer [57–60]. It usually does not lead to creation of completely new pieces, but to modification of existing pieces through rearrangement. Change in style can be carried out as a change in the timbre of the piece. In [57] an autoencoder model operates in the raw waveform domain. The task of the decoding part is to predict the next fragment of a piece based on the vector embedding of the previous fragment. The trained model is able to effectively morph between instruments, which can result in producing new sounds.

A similar approach is taken in [58], where the model also operates on a waveform. Using disentanglement techniques (adversarial training), a separation is made between the part responsible for the style/timbre of the instruments themselves and the part responsible for the pitch. Thus, it is possible to modify the style without affecting the pitch. This approach allows for a new instrumental arrangement of the pieces.

Music generation that adheres to a particular musical style is also undertaken in [59]. The model consists of an encoder and a decoder. The encoder allows creation of the embedding of MIDI representation, which includes expressive properties that define the performance style. Furthermore, the model allows completion of the performance embedding with a melody, facilitating creation of varied performance styles for the same melody.

In [60], the model operates on a waveform, from which certain properties describing the piece are extracted, especially its style/sound (such as mel-spectrogram, MFCC, spectral envelope). Style transfer is performed in the form of an image-to-image translation. The trained model is able to transfer, for instance, from piano solo to string quartet or guitar solo.

A problem similar to the style transfer is presented in [61], where the authors aim to generate new music that resembles the input music. The MIDI input is analysed in four aspects: structure, chord progression, melody and bass. The resulting music is created in four steps, corresponding to each analysed aspect, based on the statistical analysis of the input.

Another approach to the generation problem is presented in [62], where the melody is created based on lyrics. A GAN model is used in the generation process, with three modules that are responsible for creating individual components of the music: pitch sequence, duration sequence and rest sequence. Each module utilizes the previously

generated output and the embedding of the syllable for the current sound generated.

A slightly different problem is discussed in [63], where music generation is merged with hand movement analysis. The hand movements serve as input for defining the next chord within the melody. A selection of admissible chords is concurrently established, based on the existing part of the chord progression and music theory. In case an indicated chord is not within the permissible set, the melody retains the current chord set; otherwise, the chord was replaced by the one pointed with fingers.

2.1.6 Performance evaluation

A problem related to music generation is evaluation of model's performance. This topic is of particular importance when comparing generative models among themselves. Here, one of the biggest challenges is identification of an appropriate metric to assess both the correctness and creativity of the generated music. This subject has been intensively addressed in recent years [64], and the related papers are presented in this section.

A breakdown of existing methods for evaluating model performance was proposed in [64], where three categories of metrics were identified: subjective evaluation (listening and visual analysis), objective evaluation (music domain metrics, model-based metrics) and combined evaluation (heuristics).

One of the most popular subjective methods is human (expert) evaluation [65]. This method, however, does not scale well, as it requires a lot of time. Furthermore, even with the selection of experts with many years of experience and the ability to properly assess the piece under evaluation, the assessment, to some extent, is based on the subjective preferences of the expert. For this reason, there are attempts to create more objective measures for evaluating the generated results [33, 66].

An issue raised in [65] is human evaluation of generated music and the ability to its distinction from the one played by humans. The authors conducted a survey among more than 170 people, in which both music-educated and music-uneducated subjects rated 7 pieces of music (6 generated, 1 played by a human). The evaluations indicated that human-played music was not rated better than generated one.

If the purpose of the generated music is to affect the listener's emotions, it is possible to evaluate the results using two parameters: valence and arousal [66]. Valence describes whether the emotion felt is negative or positive, and arousal refers to the intensity of that emotion. However, while such an assessment can be made by any person (not just an expert), it is still prone to the subjectivity of the evaluating person.

The second (objective) path is to try to create formal measures of a quantitative nature that can be automatically calculated. In [33], the proposed measure is based on similarities to other pieces. It is assumed that training the model with pieces in a certain style should result in generating pieces in the same style. Accordingly, the degree of success of generation can be determined by the degree of similarity of the generated pieces to pieces from the learning set. To calculate such a similarity, a measure based on a dozen high-level properties, such as number of notes, polyphonic rate, pitch interval range, duration range, etc., is created. These properties are embedded as a vector, and the degree of similarity between pieces is determined using Mahalanobis distance.

The last option is to utilize a combined evaluation. In [67], four heuristic metrics are proposed for generated music evaluation, based on the intuition and musical experience of the authors. The quality of these metrics is tested by conducting a survey of over 1,000 respondents, each of whom was tasked with recognizing whether the piece presented was created by a human or an AI model. The validity of the metrics is confirmed by comparing their values with the results of the survey. It is observed that models that are rated higher by humans (and therefore less frequently identified as automatically generated music) are also rated higher in the proposed metrics.

The issue of the usefulness of the generated music is addressed in [68], where the authors describe a concert organized in 2017 with the music composed solely or partly (assisted) by AI.

2.1.7 Copyright in music generation problem

One aspect not directly related to the music generation methods, though closely linked to the results produced by these (AI) models, is the matter of a copyright for the generated content. Although relatively young, this problem is becoming increasingly important as generative models become more and more popular.

In [73], the authors conducted an analysis of the methods currently used to protect the copyright of a generated material. The most significant problem identified is the lack of unified methods to protect content that can be applied to various models, since current protection methods are often tailored to specific model architectures, lacking adaptability. A particular challenge is the development of effective methods to protect music copyright, where popular approaches such as watermarks are not applicable.

It is also important to consider that copyright legislation varies from country to country, making it challenging to develop universal protection methods. This concern is not only identified by researchers but also by state authorities. In [74], a legal case is presented to ascertain whether the

utilisation of in-copyright works as training data for generative AI systems constitutes a violation of copyright and whether the outputs of such systems infringe it.

In [75], the legal complexities surrounding AI-generated music are examined, including debates on copyright protection and the ownership of rights, as well as the use of copyrighted material in training AI models. In particular, the topics of plagiarism and copyright are considered based on FolkRNN project [76], which utilizes several AI models trained on tens of thousands of transcriptions of Irish folk music. In addition, concerns about disruption to musical traditions and historical links are raised; however, the potential for enrichment in that area is also acknowledged. It is noted that existing legal frameworks may need adjustment to facilitate creative activity and innovation of the current and prospect AI approaches in this domain.

2.2 Music classification

The second key problem related to AI in music is music classification. This task can take different forms, but in general can be stated as determining a particular feature of a given piece (genre, composer, performer, etc.). Another, more specific form, is piece recognition, which usually refers to identifying the author and name of the piece, sometimes also the performer. An important aspect of this problem is dealing with distortions that can occur when music is recorded. Systems for classification/recognition should work properly despite these inefficiencies. Attempts to solve the problem have a long history, and include, for instance, creation of a set of simple rules for musical performance analysis [77]. As an example, genres classification problem is presented in Fig. 6. This section includes papers addressing various music classification problems, summarized in Table 5.

2.2.1 Classification of genres

Classification of a music genre is a very popular task, resulting in a variety of approaches and solutions published recently [103]. In addition, some number of related datasets have been created, such as GTZAN [104], Ballroom [105], Extended Ballroom [106] or FMA [107].

The work [78] proposes classification based on sound waveform representation. From this recording, various audio features (i.e. fast Fourier transform coefficient, real cepstral coefficients, zero-crossing rate) are obtained and used in classification. Importantly, classification is performed not on single frames of the input (about 46ms), but on features aggregated across a larger number of frames (up to 600), allowing a longer portion of the song to be considered at one time. The method achieved an average accuracy of about 82% on USPOP [108] dataset.

Another possibility is to classify music based on image. In [79], a special type of neural network architecture (ConditionaL Neural Networks, Masked ConditionaL Neural Networks) that preserves spatial locality of the learned features, is used for this purpose. In addition, the networks preserve temporal relations between frames by operating on a window rather than an isolated frame. Consequently, the model is frequency shift-invariant.

A model similar to [79] is presented in [80], also for classification based on spectrograms. The authors postulate that not every part of the classified music (so not every spectrogram fragment) is equally relevant; therefore, a key part of the model refers to the attention mechanism, which allows the relevance of the analysed fragment to be taken into account.

The use of neural networks and spectrograms for classification is also shown in [81]. An important part of the model is specially designed loss function, intended to allow the extraction of deep features from spectrograms. The proposed solution is tested not only on genre classification but also on music emotion classification (MER500 collection [109]).

In [82], the authors note that CRNNs (Convolutional Recurrent Neural Networks) [110] are often used for music

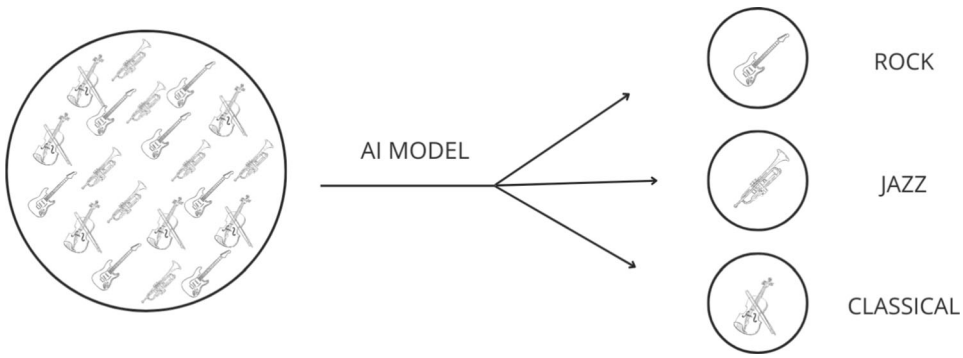


Fig. 6 Genres classification problem. AI model allows for recognition of specific music genres

Table 5 Summary of papers that address various aspects of the music classification problem

	Number of papers	Papers
Genres classification (sect. 2.2.1)	8	[78–85]
Performer/composer classification (sect. 2.2.2)	5	[78, 86–89]
Recognition of particular music pieces (sect. 2.2.3)	2	[90, 91]
Other problems (sect. 2.2.4)	11	[92–102]

classification, where the recursive part is responsible for temporal dependencies analysis in a piece (song). They conclude that the model should perform even better if it would additionally analyse the frequency and its variability. For this reason, a CRNN-TF network is proposed in [82]. In addition, the problem is extended to include the ability to predict the song tags (rather than just recognizing their genre). Experiments demonstrate that CRNN-TF network outperforms CRNN architecture in this task.

The use of frequency changes for classification is also considered in [83], where the Bottom-up Broadcast Neural Network (BBNN) model is created deal with genre classification problem. Each genre operates on distinct frequency bands and time intervals. BBNN allows for transfer of multiscale time-frequency information to the decision-making layers, which is essential for accurate genre classification.

A comparison of various machine learning (ML) models in the classification task is presented in [84]. Classification is approached from two different angles: (1) based on handcrafted features and classical ML algorithms (k-NN, SVM, etc.), and (2) using representation learning and neural networks. Classical ML algorithms used standard features (i.e. Mel-Frequency Cepstral Coefficients, Statistical Spectrum Descriptors), and neural networks used spectrograms (Convolutional Neural Network—CNN) or chord and word sequences (LSTM). The best by far accuracy is achieved by a CNN architecture. Another comparison of the models is shown in [85], where classification models utilizing different types of networks (CNNs, CRNNs, LSTMs) or their combinations (ensemble learning with voting) are compared. The experiments show that CNN models and a model combining CNN and RCNN perform best.

In summary, various ML models are used to classify genres of music pieces, but the most popular ones are neural networks. In this case, the input data are usually a spectrogram, from which (using an appropriate network architecture) different types of information are extracted. Traditional ML models (SVM, k-NN, RT, etc.) rely on standard features and are less effective than models implementing (deep) neural networks. Different types of approaches emphasize the use of different features for the purpose of classification. Sometimes these are features extracted directly from the sound waveform (e.g. [78, 84]), while at other times special attention is given to analysing the frequency changes in the musical piece (e.g. [82, 83]).

2.2.2 Performer/composer classification

The process of classification of composers (or performers) involves the assignment of a piece or its performance to the appropriate composer (or performer). Classification of performers may in some cases be equivalent to classification of genres, if the performers selected for classification perform different styles of music. However, in case they perform the same (or very similar) musical pieces, it is necessary to base the prediction on individual characteristics of each performer (evident in played music) in order to accurately identify an artist. Various attempts to performer or composer classification are presented in this section.

Due to certain similarities of the problem to music classification, the models that are used to classify performers are similar (and sometimes even identical) to those discussed in sect. 2.2.1 above. An attempt to use the same model to classify genres and performers is presented in [78]. The model, described in sect. 2.2.1, when applied to the performer classification task (and tested on the same sets as for genre classification) achieved the level of about 68% accuracy.

A relatively rare approach is MIDI-based classification [86]. The authors use a collection consisting of two works by F. Chopin (an étude and a ballad), where each piece is performed by 22 musicians. Classification features are based on three values: Inter-Onset Interval, Off-Time Duration and Dynamic Level. Classification is performed with an ensemble of small classifiers, where base-level classifiers are created with linear discriminant analysis and operate on different features.

A more standard approach is to rely on spectrograms. In [87] the task is to classify singers. The effect of separating the voice from the accompaniment and classification based on the voice alone is additionally tested, which leads to an improvement of results compared to the baseline case (no separation).

Similarly to [87], a spectrogram is also used as the input in [88], where an artist classification model is created based on the CRNN. The use of CRNN allows considering the frequency patterns (convolutional layers) and resulting patterns as temporal sequences (recurrent layers). In

addition, a visualization of the class separation (classified artists) is carried out using t-SNE [111].

A similar problem is classification of composers, so again, it can sometimes be reduced to genre classification (if each composer represents a different musical genre). The topic of composer classification (Bach, Beethoven and Haydn) is addressed in [89], with the help of features obtained from MIDI representations. Several ML classifiers (decision trees, logistic regression, SVM) are built and tested based on these features, with the best results achieved by the SVM classifier.

Since the composer classification problem is similar to the problem of genre classification, similar (or the same) methods and models are applied to both problems. Most often the models are built based on neural networks [87, 88] (e.g. spectrogram analysis), or (less frequently) based on properties obtained directly from MIDI representation [86, 89] or waveform representation [78].

2.2.3 Recognition of particular musical pieces

Another music-related ML task is song recognition [90, 91], which involves identifying (i.e. matching the title of) a musical piece on the basis of its characteristic features.

In [90], the problem of determining the similarity of a sung song, generally affected by errors (inaccurate singing, etc.), to a potential corresponding song in a song database is addressed. A Hidden Markov Model is constructed to express various types of errors, e.g. transposition, tempo changes and local errors. This makes it possible to determine the similarity between songs (and thus a possible recognition of a given song) even with inaccurate singing. [91] deals with spectrogram-based recognition (as opposed to [90] which uses MIDI). Based on the spectrogram of the input, an audio fingerprint is created and compared with the track prints in the database. The use of the audio fingerprint allows for efficient recognition, despite distortions that the input audio signal (and its spectrogram) may have undergone.

2.2.4 Other problems in music classification

In addition to the leading music classification tasks, i.e. classification of genres or performers, or recognition of pieces, other related problems are also addressed though with lesser popularity. Several such topics are considered in this section.

One such problem is determining the degree of similarity between two (or more) pieces of music. This makes it possible to automatically perform clustering of larger music collections, which is beyond human capabilities [92, 93].

In [92], categorization refers to the leading melodic line and is performed based on a waveform representation (raw audio), from which pitches representing the leading melodic line are extracted. Then, based on the obtained pitches, similarity matrices are determined for each pair of songs, and based on them a supervised classification (using k-NN) or an unsupervised clustering (using spectral clustering algorithm) is performed. The constructed model can be used for both inter- and intra-style categorization tasks.

The work [93] also focuses on determining similarities between pieces and answers the following question: "to which of two (other) pieces a given piece is more similar to?". The solution model employs the Restricted Boltzmann Machine [112].

A slightly different problem is considered in [94]. The task is to divide a piece into segments, and the segment boundaries are determined by changes in musical properties, e.g. instruments, tonality, tempo, etc. Based on these properties, it is possible to define various boundaries between segments, so the constructed model performs multi-objective optimization when detecting the boundaries.

An auxiliary problem, being often part of other problems (e.g. [87]), is separation of different tracks in audio. The problem relates to complex signals resulting from simultaneous recording of sound from several sources (instruments, instruments and vocal, etc.). Signal separation and instrument recognition are approached in [95], where the task is divided into 3 stages: pitch extraction and frequency information acquiring, obtaining feature vector from gathered data, and lastly, instrument recognition. Neural networks, trained with feature vectors extracted from sound, are used for instrument recognition. The same issue is addressed in [96], where spectrogram, tempogram [113] and *modgdgram* [114] are utilized as inputs for the Swin-Transformer model [115] to identify musical instruments. An analogous problem of audio signal separation per instrument is of interest in [97], where a model describing the spectral properties of the instruments, independent of pitch is proposed. The algorithm matches recorded discrete frequency spectra with continuous spectra from the model. Since parameters characterizing individual instruments are not known in advance, a related dictionary containing them is created (learned).

Other attempts to solve the source separation problem are presented in [98–100]. In the first paper, a specifically designed Deep Representation-Decoupling Neural Network is used for separation. With a spectrogram given as input, the network uses CNN and LSTM to obtain the features of audio. The separation module consists of an MLP group that divides the extracted features into specific sources. The reverse process is then performed, and the separated features are independently processed back into

spectrograms, individual for each source. Another spectrogram-based approach is presented in [100]. Similar to [98], a single input spectrogram is split into individual spectrograms that correspond to the sources. To tackle the issue, a U-Net structure that combines CNN (high-resolution detail recovery) and Transformer (analysis of vertical and horizontal dependencies) is utilized. [99] also uses spectrogram as the input data, which is processed using convolutional layers, dilated-GRU blocks [116] and again convolutional layers to obtain spectrograms for individual audio sources.

Another task is addressed in [101], where the authors conduct the main melody extraction. An important assumption is made regarding the possible existence of multiple main melodies, which contrasts with the main melody's uniqueness, frequently assumed in other approaches. The prepared model utilizing MIDI format and based on Transformer accomplished exceptional efficiency, classifying notes belonging to the primary melody with over 97% accuracy. A comparable problem of melodic line extraction is addressed in [102], where the task is to extract a sung melodic line from polyphonic music. The solution utilized both waveform and a combination of frequency and periodicity [117] as dual representations. The CNN and LSTM blocks first process these representations individually, and subsequently, after feature extraction, merge them for the final sung pitch prediction.

A related task, situated at the border between audio recognition and speech recognition, is audio event recognition (the identification of distinctive sounds, such as those of animals, vehicles, people or nature). A YamNet[1] model was developed and pretrained for such a task. The model was constructed using MobileNet [118] and trained on the AudioSet [119], which comprises 521 distinct audio events. The resulting mean average precision was 0.306.

2.3 Music recommendation

Creating recommendation systems is the third major music-related topic (see Fig. 7). This problem has recently gained much popularity [12, 13], partly due to the growing interest in various music-related streaming services. The basic task of recommendation is to suggest music (usually songs) to the user in accordance with his/her preferences. Thus, the problem is to determine the preferences of the user, both general and moment-specific, and then to select songs from a pool of music resources that best match these preferences. The essence of the problem is that the recommended songs should not only be selected from the songs that the user has already listened to, but especially from the songs that are available in the pool and with which the user is not familiar yet. This section comprises papers that address a variety of problems associated with the music recommendation, as summarized in Table 6.

2.3.1 Context-based recommendation

Context-based recommendation acknowledges that individuals' preferences are subject to change over time, being influenced by a multitude of factors, including mood, time of day and other external stimuli. One approach to taking advantage of those moment-specific preferences is to make recommendations by analysing the context in which the recommendation is to be made (e.g. playlist, other users' behaviour, etc.). This section covers papers related to music recommendation based on the contextual information.

This type of approach was proposed in [120], where the main assumption was that recommendation systems are limited by a small amount of user-music interaction data. Therefore, a heterogeneous information graph was designed based on contextual information: playlists, song metadata and the above-mentioned interactions. The resulting graph enabled generation of song embeddings as a basis for a recommendation system that exploits global and contextual user preferences. [121] continues [120] this line of research by introducing a similar hybrid system that relies on music metadata (tags, text) as well as on context, which in this case is the playlist and the behaviour of other users. This time, a heterogeneous information network (HIN) is applied to combine contextual data and music metadata. Specifically, [121] proposes a Content- and Context-Aware Music Embedding (CAME) method, which extracts a feature representation of songs from HIN. This allows for highlighting different aspects of a song based on the surrounding context. CAME obtains feature vectors from HIN using CNNs with attention mechanism and creates user preferences that are used for making a recommendation.

An interesting approach using contextual data is presented in [122], which addresses the cold-start problem by analysing information from other users. The problem, which usually occurs during the first uses of a recommendation system, consists in poor quality of recommendations provided to a user due to the small amount of information collected about his/her preferences. An essential aspect of the system is to perform embedding that depicts the preferences of long-term users. A trained model can predict such preferences for new users based on data collected from their first day of using the system, which facilitates appropriate recommendations.

[1] https://github.com/tensorflow/models/tree/master/research/audioset/yamnet.

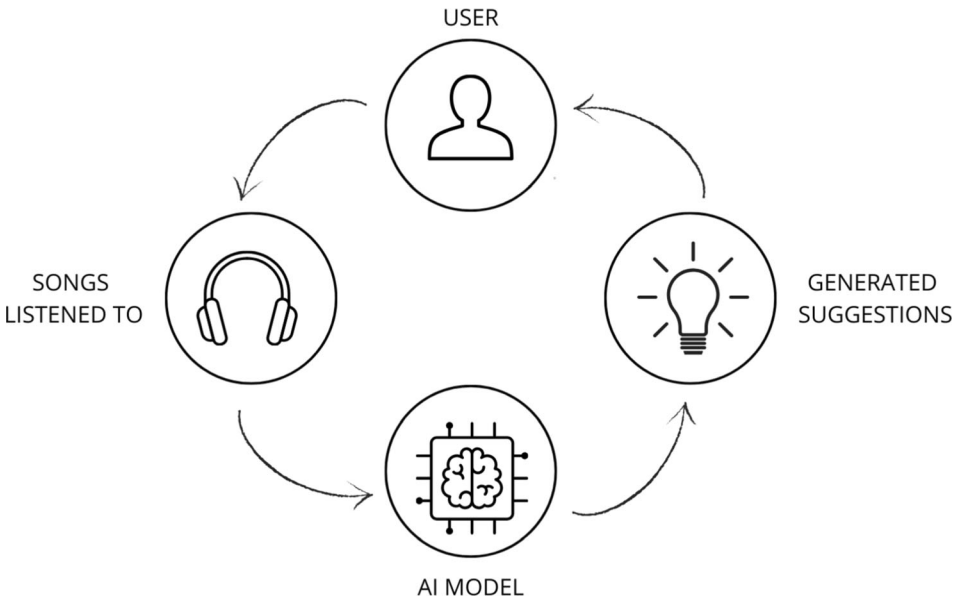


Fig. 7 Basic flow of the recommendation process. Based on user listening history new recommendations are proposed

Table 6 Summary of papers that address various aspects of the music recommendation problem

	Number of papers	Papers
Context-based (sect. 2.3.1)	7	[120–126]
Emotions recognition (sect. 2.3.2)	10	[127–136]
Metadata-based (sect. 2.3.3)	4	[137–140]

Some works on recommendation systems rely primarily on playlists [123, 124]. [123] describes an approach to recommend consecutive songs dynamically. Users can indicate their dissatisfaction with a recommended song by skipping it, and the next song is selected based on several heuristics that depend on whether the user accepts or skips the recommendation after listening. In order to create these heuristics, fuzzy sets are used, which provide both formal and intuitive definitions. Authors of [124] perform an approach similar to [123], claiming that song playlist is a valuable source of information, providing abundant data for recommendations, which is often underutilized. The similarities between songs are determined by creating word embedding techniques based on lyrics, and the recommendation itself is based on matrix factorization.

Song content can be an equally interesting source of information when building recommendation systems and is often used together with contextual information, as shown in [120, 121]. Some systems are built entirely around this source of information. For instance, in [125] a convolutional network is proposed that allows both music classification and recommendation based on the content, where content features are extracted from spectrograms. For the recommendation task, a vector of features is extracted from the spectrograms and averaged over the entire song, with song similarity evaluated using cosine distance. Another example of the use of content for music recommendation is [126]. In this case, to make recommendations, a deep neural network is trained using the song's MIDI data and the user's preferences embedding.

In summary, recent years have seen the development of recommendation systems based on broad sources of information, as presented in the referenced papers. To achieve this goal, recommendations utilize not only song metadata but also song contents [125, 126], playlists [123, 124] and details of user-system interactions [122], which are sometimes combined [120, 121] to enhance recommendation quality.

2.3.2 Recognition of emotions

Every song invokes a specific emotional response in its listeners. Identifying such emotional responses opens up many possibilities for music recommendation, as the recommended songs can be selected based on eliciting similar emotions. Therefore, recently, several systems have been developed to recognize [127] and suggest music based on users' presumed emotional responses. Various optimization and machine learning approaches have been applied to classification and recognition of emotions in music (including evolutionary algorithms [128]), but neural networks have been by far the most popular one. This section is devoted to works that address the problem of making recommendation based on recognition and analysis of expressed emotions in a music piece.

The representation of emotions commonly uses Thayer's two-dimensional model [141]. The first dimension, also known as arousal, corresponds to the energy conveyed by the music, which permits categorizing music into stimulating emotions such as joy or anger, as well as calming ones like contentment or sadness. The second dimension is valence, which describes the temper associated with the music. Valence allows classifying emotions into positive ones like joy or contentment and negative ones like anger or fear.

This (or similar) model was considered in various research papers. For instance, the authors of [129] attempted to build an interpretable emotion recognition model. Selecting appropriate song features to train the classifier is a crucial element in building such a model, so features describing the four primary dimensions of music, intensity, timbre, rhythm and tonality, were extracted. Each of these dimensions was then defined by a set of individual characteristics taken from the music piece, which were further selected to determine the most significant features that dictate emotionality. Emotions were classified using different models, which included k-NN, SVM and logistic regression. The experiments show that only a small subset of analysed characteristics significantly affects arousal determination, while the vast majority of characteristics make a minor contribution to the determination of valence.

Another approach that uses the above-described two-dimensional model was presented in [130], where triplet neural networks (TNN) were implemented to generate relevant representations while reducing the number of features simultaneously. The resulting representation could then be classified via standard classifiers such as SVM [142] or GBM [143]. Overall, the results show that TNNs are more effective in feature dimensionality reduction than PCA [144] or autoencoder [145].

Application of the two-dimensional emotion model is also demonstrated in [131]. The initial step involves analysing a photograph of the user's face to extract their emotional state. Then, recommendations are made based on a dedicated dataset [146] that assigns an arousal-and-valence value to each song. Recommended songs are chosen using k-NN with Euclidean distance in the arousal-and-valence space.

The two-dimensional model for describing emotions can be extended by adding a third dimension, resonance, corresponding to the characteristics of the music itself, including continuity-discontinuity and melodic-harmonic contrast [132]. In this work, distinctive features were extracted from the music for each of these three dimensions, and regression models were then trained to identify each of them. Consequently, a system for song recommendation based on the predicted emotional state of the listener was developed. For a particular sequence of songs listened to, a prediction of the listener's current emotional state was made, and then a song corresponding to that emotional state was recommended.

A more comprehensive approach to the problem was presented in [133], where the authors constructed a recommendation model based not only on the predicted emotions of the users but also on their predicted personality. The data were gathered from the WeChat system, which enabled the examination of several elements that might be linked to the user's personality, such as demographics, social behaviour (which includes WeChat activity) and the topics of the articles read. The emotional description itself was inferred in three steps. The first step determined whether an emotion was positive or negative, the second one determined the general emotional state (seven possibilities), and the final one detected a specific emotion (21 possibilities). Furthermore, a song description was created, encompassing metadata, acoustic features, lyrics and an emotional nature (determined primarily from the lyrics or, in the absence of, also from the acoustic features). The collected data were used to train a recommendation model in the form of a deep neural network with an attention mechanism.

An alternative approach to emotion recognition was presented in [134]. Instead of relying on a two-dimensional arousal and valence model, specific emotions (anger, fear, joy, sadness and surprise) were classified. The recognition model was based on a fuzzy neural network (the ANFIS model [147]). The results were promising, especially for the subsets consisting of two or three emotions.

A related problem, the identification of emotions' category, was addressed in [135], where the analysis was conducted not only on audio but also on video data. Two distinct CNN models were developed to process the input material separately. One model classified the input based solely on audio data, represented as a spectrogram, while the other model processed the input relying solely on a sequence of images. In the initial phase, the models were trained separately and then combined using a Softmax classifier to form a multimodal classifier. This integrated model exceeded 88% accuracy in recognizing the class of emotions.

A distinctly different perspective on emotion recognition was presented in [136]. The emphasis was on the subjectivity of emotions, leading to uncertain perceptions. The multi-task learning model was crafted to leverage this uncertainty, instead of evaluating the emotional state and perception of uncertainty apart from each other.

To sum up, predicting and recognizing emotions in music is an intriguing topic as emotions are, by definition, subjective feelings unique to each individual. Therefore, the created models tend to focus on predicting specific properties of emotions (e.g. arousal and

valence) [129, 130, 132] and less frequently specific emotions directly [133, 134].

2.3.3 Metadata-based recommendation

Using a song's metadata as a basis for recommendation is a distinct alternative option. The most commonly used metadata in song recommendations are tags, which are descriptive keywords that provide information such as the artist's name and music genre, among other song details. In this section, articles approaching the recommendation problem based on metadata included in a musical piece are discussed.

An example of a song metadata-focused recommendation system, where aspects such as duration and loudness are also considered, is presented in [137]. The model is a combination of logistic regression and eXtreme Gradient Boosting (XGB).

Typically, traditional recommendation models employ tags to describe songs. Yet, a challenge for large song databases is adding proper tags to all items, which is often beyond the capabilities of a human. Thus, to support tag-based systems, models have been developed to automatically generate song tags [138–140]. A model for automatic generation of song tags based on their content is presented in [138]. A song is represented as raw audio, preprocessed using a scattering transform. The tag prediction is performed by a deep neural network (Recurrent Neural Network—RNN with Gated Recurrent Unit – GRU), and the output of the network determines to which of the 50 possible tags the song should be assigned. It is noteworthy that in this case the automatic tagging problem becomes a song classification problem, where specific tag-represented classes are assigned to the song.

A similar problem of predicting song tags is considered in [138]. The proposed model uses modulation filter banks representation within a convolutional neural network framework, which facilitates the retrieval of relevant song features.

Another method for solving the automatic tagging problem is presented in [140]. The main goal is to improve the efficiency of recommendation systems for songs with a limited number of tags, in order to address the cold-start problem. Therefore, pseudo-tags created based on the song's content are introduced for songs with a low number of real tags. For creation of the pseudo-tags the Mel-Frequency Spectrum is used, which enables an implicit representation of content inside tags. As experiments show that it is possible to alleviate the cold-start problem with a hybrid recommendation method that leverages both pseudo-tags and real tags, weighted differently based on the number of real tags.

In summary, due to the popularity of tag-based recommendation, one of the key issues is automation of song tagging [138–140]. Two main approaches to this topic can be distinguished: assignment of standard tags belonging to a pre-defined set of tags, based on the song's content [138, 139] or creation of novel, unique tags to substitute (or complement) standard tags [140].

2.4 Other problems

Apart from the three main above-mentioned trends, i.e. generation (sec. 2.1), classification (sec. 2.2) and recommendation (sec. 2.3), there are several other topics that do not fit under any of these categories.

One of the less popular, though interesting, research avenues involves both human vision and hearing [148, 149]. In [148], an endeavour is made to synchronize music (hearing) with its notation (vision) automatically. The system relies solely on pitch-based representation, and the solution model considers both discrete (notation position) and continuous (song tempo tracking) variables. The task is to find the correct position in a musical score given an audio frame. The estimation of a score position is carried out using tree search algorithm. In [149] another pairing of visual and auditory senses is presented, where the goal is to associate images with a given song. Separate sets of features are extracted from both song (MFCC) and image (colour and texture) and input together into the system.

The detection of music plagiarism is an example of another less popular music-related topic. The system proposed in [150] uses fuzzy vector-based similarity to identify plagiarism. The detection process starts with a transformation of the suspected piece into a vector representation, then the search for similar pieces is performed, and finally, the fuzzy similarity score is calculated. The system has demonstrated high detection accuracy when tested on a collection of musical pieces containing confirmed instances of plagiarism.

An interesting initial task that can serve as a foundation for further analysis or data processing is automatic feature detection. An example of such is proposed in [151], where extraction is performed using unsupervised training of a feedforward network with Hebbian learning.

Another topic, the orchestration problem, is quite similar to the sound source separation, described in sect. 2.2.4. The objective of the task is to choose a group of musical instruments with which a music piece would be played. In the problem, formulation considered in [152], the orchestration is established based on a reference sound. Hence, for a specific sound (timbre) an ensemble of instruments is chosen, which are intended to recreate the sound as closely as possible. The problem is developed as an optimization

task in which the distance between the reference sound and the selected instrument combination is minimized.

2.5 Summary

Music-related topics form an extensive research field in which computer science, and artificial intelligence in particular, can be employed in numerous ways. Each of the three fundamental areas, i.e. creation, classification and recommendation, comprises several more specific topics. Besides these three primary sectors, other specific issues, not belonging to the mainstream research, can be dealt with, as well.

Problems, whose solving requires the highest degree of creativity from a human perspective, are typically found in the generation area, which can be further divided into two smaller sub-areas, namely sheet music generation and audio generation. Sheet music generation is commonly used for the purpose of generating accompaniment or harmonization. The accompaniment of a melody is a critical problem in music, as it can enhance and enrich the melody's sound, and its automatic generation can be helpful, especially for inexpert musicians. The second subcategory, audio generation, addresses another human factor, i.e. the performer(s), allowing the audio to be acquired quicker than in a traditional way (asking the performer to record the music). It also provides the ability, if needed, to adjust the music automatically in real time to specific demands or a context for which the music is intended.

Music classification has multiple applications too. The most apparent use case is the automatic organization (categorization) of extensive music collections. Application of AI techniques allows for effective and time-efficient grouping, based on selected parameters like composers, genres, performers, etc. Other challenges in this field, such as automatic plagiarism detection, can also be more accurately solved by AI systems than humans, particularly when addressing large datasets with pressing time constraints.

Likewise, solving recommendation problems has clear benefits. In order to satisfy the users recommendation tools must be very accurate, due to the massive amounts of music in streaming services and databases. When it comes to the quality of recommendations, considering not only the global preferences of the user but also his/her temporary, moment-based preferences related to the current listening session is crucial. One way to approach the aspect of local preferences in AI recommendation systems is by recognizing or assessing the emotions conveyed by a piece of music.

On a general note, the diversity of issues constituting the fusion of AI and music has led to the intensive music-related research that has been conducted in recent years; however, there is still ample potential for new ideas and methods in this area.

3 Real-life applications of AI in music

In the field of using AI in music, the majority of considered tasks remain within the realm of theoretical considerations, with few examples of possible real-life applications of the proposed models. In the context of classification, models have been developed that allow for automatic clustering of large music data sets [92, 93]. Similarly, attempts have been made to perform concerts of AI-created music in the domain of music generation [68]. Also, there are emerging works that combine different fields and aspects of real life. The focus in this section is on these real-life applications of AI in music.

In [153], an analysis of physiological signals in the context of a human body's response to the reception of different musical genres is presented. The analysis was conducted with 24 participants listening to 12 pieces from three musical genres. Four signals (electrodermal activity, blood volume pulse, skin temperature and pupil dilation) were measured during the listening session. From each of these signals, 34 features were extracted, including the average value of the power, average amplitude change and fuzzy entropy. Based on these extracted features, an animation (named Gingerbread animation) was created to illustrate the change in signal values over time, where each signal was represented by a different colour. Subsequently, a classification of this animation was conducted using the CNN model, resulting in a classification accuracy of a musical genre exceeding 99%.

A framework for retrieving knowledge from radio music was created in [154]. There are four primary objectives of the system: to accommodate the continuous arrival of audio data (an immanent aspect of radio station music), to obtain records from the indicated time interval, to enable filtering based on music content (speech, genre) and to visualize this information. A significant component of the system is a layer that enables the user to categorize radio broadcast segments into specified classes within three areas: speech/music content, music genre and emotion. Audio features (spectral centroid, zero-crossing rate, MFCC and others) are extracted from a given segment, and classification is performed (with SVM or neural network) using these features. The system allows the user to create reports based on a set of filters of his/her choice. Furthermore, the data visualization mechanism was designed to facilitate intuitive analysis of the generated reports. The accuracy of visualizations was validated through a comparison with the

calculated Jaccard similarity measure between songs played at particular radio stations.

One of the most prevalent applications of automatically generated music is in computer games. An overview of such methods is presented in [155] and encompasses not only the use of music as a background for a gameplay but also the generation of sound effects. Furthermore, a review of papers dealing with the impact of generated music on a game assessment by the users is included.

The topic of music generation tested in computer games is considered in [156], in the contexts of the game of checkers [157]. The generation system is based on three steps: chord progression generation, main melody line generation and accompaniment generation, all of them implemented using an evolutionary algorithm. The system allows the adjustment of the sentiment of the generated music through selection of several parameters, such as timbre (choice of instruments), rhythm and number of dissonances. In the test of the system presented in [157] the users were asked to assess the impact of using the system on their perception of the game.

Another possibility is to use AI to support the music teaching. In [158] the AI-assisted learning system represented as a neural network is introduced. The network is trained on a dataset comprising exercises performed by musicians and various music-related tasks, accompanied by feedback from experienced teachers. The system is integrated with the flipped classroom approach, where the experimental group of students engages with it, while the control group receives materials via the MOODLE platform. Following the completion of exercises, students from the experimental group receive detailed feedback on their work, aiding in error identification and correction. The system assesses their performance, suggests exercises and corrects the errors, guiding students towards improvement and preparing them for subsequent lessons. The results of this study demonstrate the effectiveness of AI integration in the flipped classroom model, leading to significantly higher learning outcomes and increased student engagement in music education (compared to the usage of MOODLE), based on scores achieved during the final test held once the tutoring cycle had been completed.

4 Solution methods

A method-oriented perspective presents an alternative approach to reviewing the current AI research in music. In the following sections, the extensive and varied use of different methods in this field is summarized and divided into several categories, with the main ones being neural networks (sect. 4.1), evolutionary methods (sect. 4.2) and others (sect. 4.3). A breakdown of the number of included papers according to the methods used is presented in Table 7.

4.1 Neural networks

The widest and the most popular group are artificial neural networks, whose undeniable advantage is potential applicability to virtually any problem formulation among those presented in sect. 2. In addition, many neural network-based approaches turned out to be state-of-the-art solutions for many music-related tasks. In what follows, we group and discuss the papers according to the particular neural network models they applied. A summary of these papers is presented in Table 8.

4.1.1 Convolutional neural networks (CNNs)

CNNs (Fig. 8) are well-known to be highly effective in image processing. Therefore, when applying CNN, the initial phase usually involves acquiring image data from the raw music data. One straightforward approach is to represent the music data as a spectrogram, an image that depicts the frequency spectrum over time (see Fig. 2). This section is devoted to CNN-based solutions to various music-related problems.

Application of CNNs to genre classification was investigated in [84]. The authors conducted a comparative analysis of different models' classification performance levels, distinguishing those with handcrafted features (k-NN, RF, SVM) and those based on representation learning (CNN, LSTM). Besides, various combinations of the above algorithms were tested, with the final scores obtained either by means of the sum or product of individual model's results, or through a majority vote. The best results were obtained by a single CNN (using spectrograms) and a combined product of CNN with either SVM or RF.

In [87] CNN was applied to a singer classification task. The study evaluated the effect of three loss functions: triplet loss [161], prototypical network loss [162] and generalized end-to-end loss [163] on the network's performance. The CNN model was built based on the ResNet architecture, and the experiments yielded the prototypical network loss to be an optimal choice for this task.

Table 7 The number of papers included in the review, classified according to the type of method used

	Number of papers
Neural networks (sect. 4.1)	52
Evolutionary algorithms (sect. 4.2)	7
Other (sect. 4.3)	33

Table 8 Summary of papers that uses various neural networks for music-related problems

	Number of papers	Papers
Convolutional Neural Networks (sect. 4.1.1)	12	[51, 56, 81–84, 87, 88, 121, 125, 139, 159]
Recurrent Neural Networks (sect. 4.1.2)	7	[25, 34, 44, 71, 99, 99, 138]
Autoencoders (sect. 4.1.3)	10	[33, 35, 42, 43, 49, 53, 57–59, 80]
Complex models (sect. 4.1.4)	6	[28, 30, 52, 85, 98, 100]
Other models (sect. 4.1.5)	17	[26, 37, 38, 40, 41, 55, 60, 62, 79, 93, 95, 96, 102, 122, 130, 134, 160]

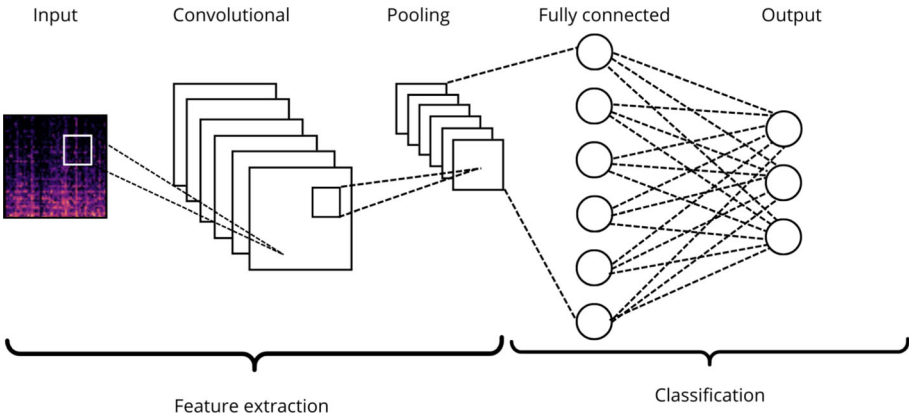


Fig. 8 CNN model scheme. Data processing is divided into two stages: feature extraction (convolutional and pooling layers) and classification (fully connected layers)

A similar problem was discussed in [81] with a common ResNet architecture altered by incorporating a specially designed loss function to facilitate improved music feature extraction from spectrograms. Instead of a typical softmax loss, an Angular Margin [164] and Cosine Margin [165] softmax loss was applied, enabling both intra-class compactness and inter-class discrepancy for genre classification problems.

A study in [125] employed a basic CNN architecture to create a classification and recommendation system. The CNN utilized a spectrogram as input, processed by four convolutional blocks, followed by a global average pooling layer and two fully connected layers. Depending on the task, the approach was executed in two distinct ways: for classification, an additional output layer of the size equal to the number of classes was appended; for a recommendation, the data from each analysed spectrogram of a music piece were averaged, and then the resultant vector was compared with vectors of other tracks using cosine similarity.

Yet another approach was presented in [139], where an important aspect of the auto-tagging problem in the preprocessing of the spectrogram data, prior to their input into the network, was considered. A set of learnable sinc filters (i.e. removal of all frequencies higher than cut-off value) parameterized by their cut-off frequencies was used to preprocess the spectrograms. Once the data had undergone this preprocessing, they were passed through the CNN/ResNet classifier.

In [56], the CNN model was utilized to classify mashups (i.e. mixing two or more songs). The objective was to train a network to discern whether a produced mashup is pleasant or unpleasant to hear. The training data included genuine songs as affirmative samples and inaccurately developed mashups, regarding the key and tempo matching between mashup tracks, as negative examples. Two models were developed to address this problem: the first one took three separate tracks as input to create the mashup, while the second one used an already assembled mashup. In the latter case, the spectrogram underwent processing by several convolutional blocks and subsequently, fully connected layers. For the model that dealt with mashup tracks independently, each track underwent separate initial processing by two convolutional blocks and the outcomes from these blocks were merged and further processed by another convolutional block and fully connected layers. Both models performed very well, achieving over 98% accuracy when classifying mashups.

In [51], a music generation model using the WaveNet model [166] was presented. Skip connections were excluded and SeLU was utilized as the activation function in the final layer. Furthermore, each input channel was processed by a distinct filter set.

CNN was also utilized in [121], where the model combined a heterogeneous information network (HIN) with

CNN embeddings to address the recommendation problem. The HIN network integrated songs' data such as tags and lyrics with contextual information such as user playback sessions and global preferences. A content- and context-aware music embedding (CAME) model was developed to learn feature vectors from the HIN network. To create these embeddings, CAME applied CNNs with an attention mechanism. CAME aimed to achieve two objectives: a structural objective referred to measuring the likelihood of an edge and a text objective (was related to incorporation of tags, lyrics, etc.). The model enabled the creation of embeddings for global and contextual user preferences, which were further utilized for recommendations.

Besides the use of standard models, extended or modified CNN models are often utilized, which should, by design, provide better performance in handling music data. One example is the convolutional recurrent neural network (CRNN) employed in [88]. The created model includes 4 convolutional blocks (CNN part) and 2 GRU blocks (RNN part) and performs exceptionally well in the task of artist classification.

An analogous extension of the CNN network is used in [82], where the CRNN in Time and Frequency dimension (CRNN-TF) model is introduced. This model combines the advantages of both convolutional and recurrent networks while exploring spatial dependencies in the time and frequency dimensions in various directions. An activation map obtained from the CNN part is converted into eight sequences by scanning all possible combinations of the following directions: top to bottom/bottom to top, left to right/right to left, column-/row-wise. These sequences are then presented as input into separate GridLSTMs [167]. The resulting outputs from all LSTMs are merged and processed by a fully connected layer. This model, when used as a classifier, delivers strong results in tag prediction task.

Another specific model, presented in [83, 159], is the Bottom-Up Broadcast Neural Network (BBNN). The model comprises four components: shallow feature extraction layers, a Broadcast Module (BM), transition layers and decision layers. Broadcast Module is created by stacking Inception Blocks [168] on top of each other, which enables the development of the learning features from multiple receptive fields, while the single Inception Block allows merging convolutions with varying kernel sizes. All Inception Blocks are also densely connected, allowing for bottom-up transmission of extracted features to all blocks in BM, which reduces the network's sensitivity to frequency shifts in the data. The model is utilized for the music genre classification problem. In [83], a modified Inception Block is proposed, and the BM that uses 6 of these blocks instead of 3 as proposed in [159]. This modification leads to even higher classification accuracy, exceeding 97%.

Generally, the employment of convolutional networks is frequently linked with image processing, and the same applies to music-related topics, where a crucial component involves transforming the music data into a spectrogram image. Nevertheless, unlike regular image data, one of the dimensions of a spectrogram is time, which retains the data's sequential nature. For this reason, when building CNN models for music-related topics, efforts are frequently made to modify these models to handle the time dimension, such as employing a dedicated loss function [81] or incorporating a recurrent component to deal with sequential data [82, 88].

4.1.2 Recurrent neural networks (RNNs)

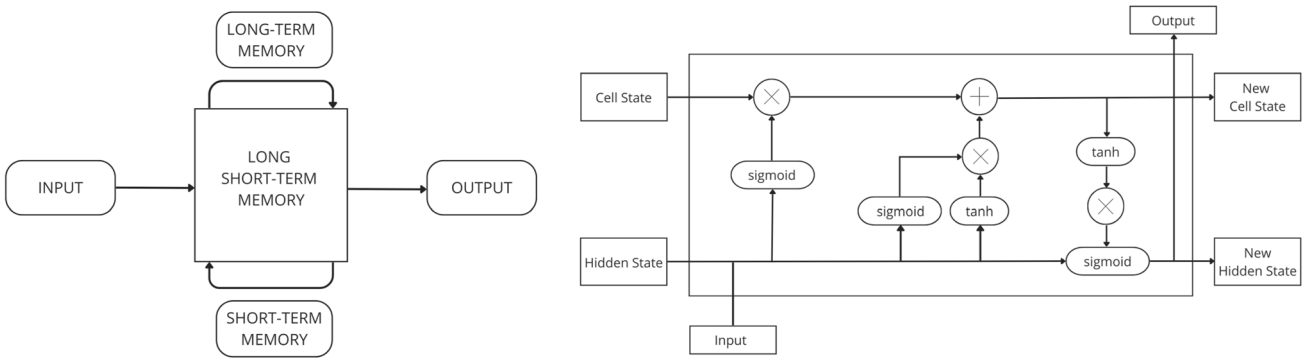
Music data, regardless of its representation, are sequential. Therefore, models developed to solve music-related problems are typically adjusted to accommodate the temporal aspect of the data. Among neural network models, the foremost examples are recurrent neural networks, in which the output from certain neurons impacts the future input to those same neurons. The most popular RNN architectures are Long Short-Term Memory—see Fig. 9 and Gated Recurrent Unit—see Fig. 10. This section covers various RNN-based approaches in the field of music.

The use of LSTM was, among others, demonstrated in [25], for chord prediction. The noteworthy aspect is data preprocessing and establishing the 20-dimensional chord embeddings. The authors utilized a model with 2 uni-directional LSTM hidden layers and a linear output layer to forecast the subsequent chord based on past chord sequences.

A model for music generation based on LSTM was also constructed in [44] and was composed of three modules focusing on the sequential prediction of chord, note and volume, respectively. Each module consisted of an embedding layer that generated a vector representation of the input, LSTM layers with batch normalization and a fully connected layer incorporating Softmax activation. These features enabled the prediction of sequences of module-specific events.

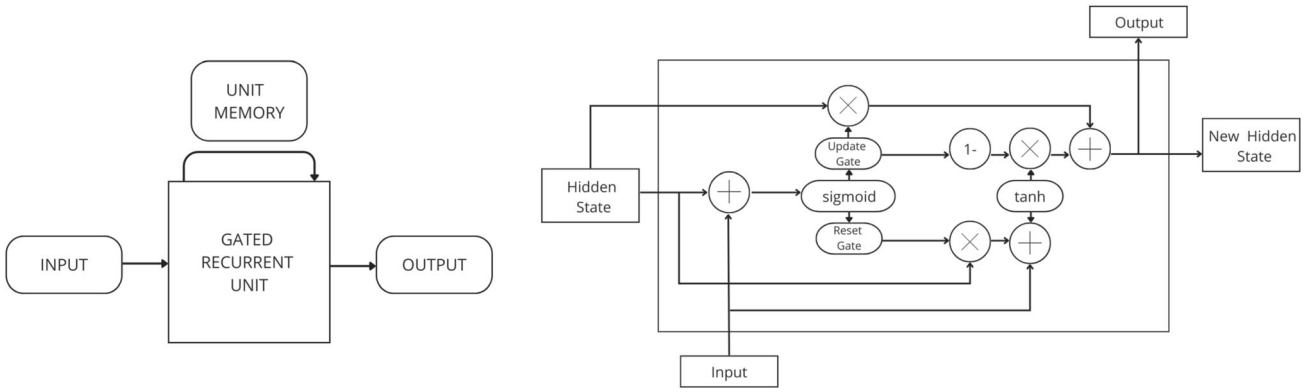
Yet another LSTM-based model was proposed in [71] for expressive music generation by means of predicting a sequence of events. The developed model was relatively simple, comprising three 512-node LSTM layers with the input and output representing 413 events in the form of one-hot encoded vectors.

A more complex model was created in [34], which utilized a hierarchical recurrent neural network (HRNN) for event generation. The model was composed of three hierarchical levels, with the top two employing LSTM cells



(a) LSTM general data flow. (b) LSTM detailed data flow.

Fig. 9 A scheme of an LSTM model. A general flow of the data (Fig. 9a) and a detailed scheme (Fig. 9b)



(a) GRU general data flow. (b) GRU detailed data flow.

Fig. 10 A scheme of a GRU model. A general flow of the data (Fig. 10a) and a detailed scheme (Fig. 10b)

to capture dependencies over a larger time window and the bottom one directly focused on event prediction, relying on the data from the two higher levels and a number of preceding events. To ensure that the lowest level took into account the sequentiality of the data, 1D convolution was used, and the top two levels utilized a 2-layer LSTM, each with a memory unit size of 256. The prediction module, which was integrated in the lowest level, contained fully connected layers for pitch and duration prediction. The resulting model was able to capture dependencies across larger time windows (utilizing the upper levels and LSTM layers) as well as more local dependencies (with the lowest level).

Models relying on GRU were presented in [99, 138]. In [138], the objective was to predict tags from the mel-spectrogram data preprocessed with the scattering transform [169] to increase the timescale without an excessive loss of information. The preprocessed data were then fed into a deep GRU-inspired network [170]. A 5-layer RNN, consisting of five stacked GRU layers, was employed, with the final sigmoid layer performing tag classification.

The authors of [99] drew inspiration from the WaveNet architecture [166], which employs stacked dilated convolutions, to devise an architecture with dilated GRU [116], where units receive information from a certain k preceding steps before, rather than only from the previous step. The proposed model aimed to integrate recursive layers and dilated convolutions into a block called a Dilated GRU-Dilated convolution block (D2 block). The ultimate architecture was applied to sound source separation task and consisted of five elements: a convolutional layer, three D2 blocks and another convolutional layer.

Recurrent architectures are well suited to processing of music data due to their inherent time dimension. Recurrent models considered in the literature include mainly LSTM [25, 34, 44, 71] and GRU [116, 138], two most popular architectures of this type. From the problem type point of view, their application includes music generation [34, 44], tag prediction [138] or separation of audio sources [116].

4.1.3 Autoencoders

Another frequently employed neural network model is the autoencoder (AE), which is composed of two components: an encoder and a decoder (see Fig. 11a). The encoder generates an efficient representation of the input data, while the decoder converts the received representation back into the form of the encoder's input. A significant aspect of the autoencoder's design entails selecting an appropriate architecture for both these components to ensure that the resulting representation of the input data will be as close to the original input as possible. In addition, the generation task commonly employs a variant of autoencoder, known as variational autoencoder (VAE), presented in Fig. 11b. To properly handle a time dimension of the music data, an RNN architecture is often utilized for both the encoder and decoder. Applications of the third popular type of deep models, i.e. autoencoders or variational autoencoders, are presented and discussed in this section.

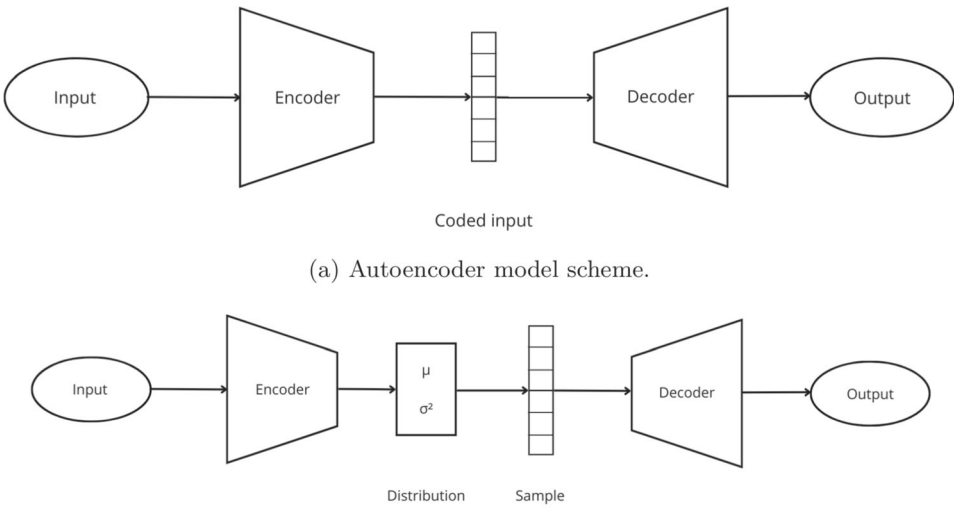
In [57], the authors use the WaveNet architecture to develop an autoencoder that facilitates the production of raw audio. WaveNet allows the prediction of the next piece of audio based on the input, but the generated audio lacks long-term structure, which may be enforced through conditioning on temporally aligned features. An autoencoder eliminates the need for external conditioning, making it a plausible solution. The encoder produces an encoding of the input. The decoder is fed with the same input (audio chunk) as the encoder and with data produced by the encoder (audio encoding) in order to predict the next fragment of audio.

VAE is used for generating music in [33], where the authors apply Deep Recurrent Attentive Writer [171] (without attention mechanism) relying on the LSTM-RNN architecture [172]. The output is computed sequentially, enabling local changes to be made in successive time steps.

Another application of VAE was presented in [42] for generating accompaniment in piano-roll representation. A coupled latent variable model was proposed to simultaneously capture the structure of each track and the joint distribution of multiple tracks through weight-sharing. The initial encoder layers process each track, including the condition track, separately to capture intra-track dependencies. The track outputs are then combined and processed by subsequent layers, to capture inter-track dependencies. The final two layers of the encoder consisted of fully connected layers which calculated the latent variable. The decoder's structure was symmetrical to that of the encoder, i.e. initially, the data were processed together and then divided into separate tracks and processed individually. The decoder's input was a combination of the latent variable and the embedded condition track.

In [35], the authors use the model initially presented in [173], in which the encoding stage is based on a two-layer bidirectional LSTM network and the decoder's architecture is in the form of a hierarchical RNN. At the beginning of the decoding process, the encoded data are split with first LSTM network into non-overlapping sequences, which are then separately decoded using a second two-layer LSTM.

[49] emphasises the importance of the initial processing of the data by the Gated Graph Neural Network (GGNN) [72] in the expressive music performance generation. The GGNN is used to encode the musical score, while the LSTM encoder is used to construct the performance representation. The decoder generates the desired performance features based on a coded (embedded) score, a style vector (generated by the encoder) and initial performance features.



(a) Autoencoder model scheme.

(b) Variational autoencoder model scheme.

Fig. 11 Schemes of autoencoder (Fig. 11a) and variational autoencoder (Fig. 11b)

Besides classical RNN models as part of an encoder or decoder, other architectures are also utilized. In [59], a Transformer network [174] is employed to build an autoencoder for music creation in various performance styles. The encoder consists of six layers comprising a multi-head relative attention mechanism and a position-wise fully connected feed-forward network [175]. The decoder has an identical construction, but with an extra layer of attention over the encoder outputs. Additionally, it turned out to be feasible to integrate a second encoder specialized in melody encoding, excluding the performance features. Only a combination of the outputs from these two encoders allows for the generation of the desired results.

In [53], the authors employed an autoregressive discrete autoencoder. The model includes a quantization module situated between the encoder and decoder, which processes the continuous output data from the encoder. A modulator and autoregressive model build up the decoder, which utilizes the quantized query to replicate the input data. Each of the three networks (encoder, modulator, decoder) utilizes the WaveNet architecture.

The Wasserstein Autoencoder [176] composed of an encoder and a decoder, both of which utilize the DCGAN architecture [177] is discussed in [43]. The middle layers use 256 filters of size 8x8. Quite surprisingly, the model enables good reconstruction of the training examples despite a small size of the latent vector, composed of 3 elements only.

An interesting encoder/decoder architecture is presented in [58], where the underlying concept is to separate latent factors for a pitch and timbre that could be useful in style transfer. For this purpose, two models are proposed. In the first one, based on [178], the encoding is divided between two distinct encoders. One handles the pitch while the other manages the timbre and the resulting data form a matrix, with time being one of the dimensions. The second model, inspired by [179], utilizes a solitary encoder for the timbre and skip connections between encoder and decoder to enable the transmission of pitch data. Since the encoding focuses solely on the timbre, skip connections are utilized to convey the pitch information. Moreover, two additional decoders are trained, to deal with the instrument and pitch information, respectively.

Yet another encoder/decoder architecture is presented in [80]. The encoder's objective is to create a meaningful representation of a spectrogram, and the decoder's goal is to utilize this encoding to determine the music genre label. The encoder consists of two stacked bidirectional RNN layers, along with shortcut connections that bypass one hidden layer. Additionally, an attention mechanism, implemented with a CNN, is included in the encoder, parallel to RNN. An attention score vector is then merged with RNN output and passed to the decoder, which is built as a softmax classifier that points to the recognized (classified) music genre.

4.1.4 Complex models

A common approach to challenging problems (whether in the area of music or other domains) is to generate complex models often composed of various modules, spanning multiple architectures like CNN, RNN and others. Such an approach usually facilitates division of the entire problem into a set of subproblems, each of them addressed with a distinct model or technique. The other general approach to challenging tasks is a mixture-of-experts paradigm, where several models are used to solve the problem and then their results are combined for the final outcome. This section includes papers revolving around solutions based on combination of multiple models.

The effectiveness of various complex models for music genre classification was tested in [85]. Two methods were implemented to assess the efficacy. The first one involved voting, where the problem was tackled by individual models and the final category was determined by a majority vote. The second one used the fully connected layer to combine the model outputs, and classification was executed using the Softmax function. Complex models based on the combination of CNNs and CRNNs (using 1D or 2D convolution) were proven to be more effective than individual model instances.

Combining models for the generation problem was presented in [30] and divided into two subproblems. First, a convolutional network was used to generate the music's structure (rhythmic and melodic relationships between bars); then, the second module employed the RNN to produce the melody itself. The structure-generating module was built with the Wasserstein GAN within a gradient penalty framework [180]. A generator, which utilized a fully connected layer and three 2D convolutional layers, created the structures from a vector, and a critic (discriminator, whose architecture mirrored the generator's one) judged whether a given structure is real or fake. The melody generation module was based on a bidirectional GRU and conditioned on the structure generated by the previous module and chord progression.

The task of generating music was also split into two subproblems in [28]. The first step involved creation of a melody (pitches and rhythm) to match a given chord progression, while the second step entailed producing an accompaniment for the previously generated melody. While the solution to the first part involved a model based on GRU, the second part—generating an accompaniment—was implemented as the task of decoding a single sequence into multiple sequences (multiple instruments). Initially, the GRU processed a single sequence, which was

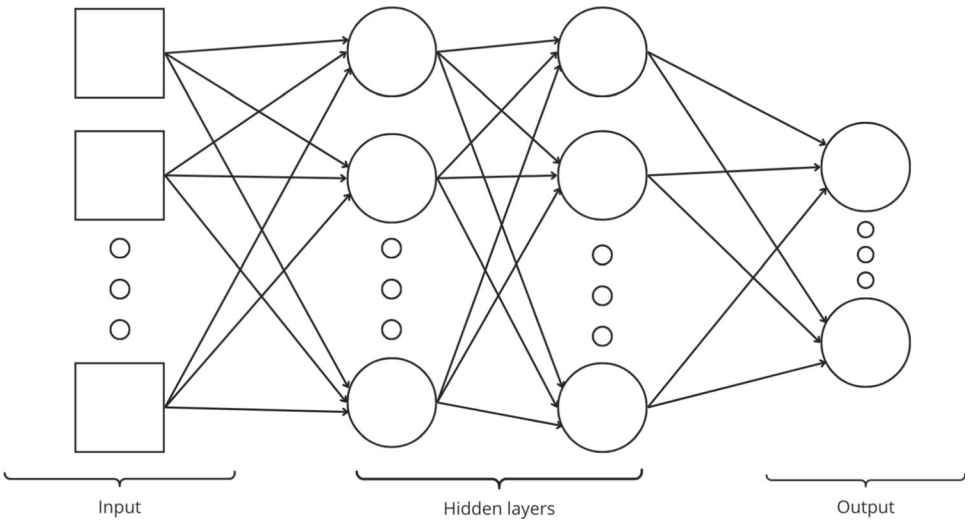


Fig. 12 An example of a multilayer perceptron model with two hidden layers

then followed by a designed attention mechanism and a specifically designed MLP Cell to generate multiple sequences that were mutually related.

In [52], a music generation task was approached using spectrograms and image generation techniques. The problem was separated into two stages: firstly, an initial spectrogram from a piano-roll depiction was produced, and secondly, the spectrogram was refined. The first stage was resolved by employing an encoder/decoder architecture that utilized convolutional networks, with skip connections included between the encoder and decoder. Both the encoder and decoder were composed of 5 layers and implemented 1D convolutions. The second component of the model, used for the spectrogram refinement, consisted of four layers of Multi-Band Residual Blocks (MRBR). Each MRBR divided the spectrogram into 4 parts and processed them with additional convolutional layers. The spectrogram refinement component was trained with residual learning [181].

A complex architecture was also used in [98] for audio source separation. The music data, represented in the form of a spectrogram, was initially divided into overlapping batches. Each batch was processed by CNN layers (dimension reduction) followed by LSTM (feature extraction), akin to CRNN architecture. To break down the extracted features into separate sources, a decoupling component consisting of a collection of multi-layer perceptrons was employed. After decoupling, the opposite process occurred, in which distinct features were processed by LSTM and CNN layers in order to regenerate the spectrogram for each individual audio source.

An analogous audio source separation issue was addressed in [100] utilizing a complex model, with U-Net-like architecture that merged CNN and Transformer. The model's input was a single spectrogram with mixed sources, and the resulting output generated a set of spectrograms, each of them representing one source. Due to the large size of the input spectrogram, three CNN blocks (each consisting of two convolutional layers, a leaky ReLU and a Batch Normalization) were first used for its downsampling. Then, it was processed by 5 Transformer blocks, followed by a CNN for a high-dimensional data reconstruction. Each Transformer block employed a specifically designed stripe-wise self-attention mechanism for capturing long-term data dependencies.

4.1.5 Other models

In addition to the continued use of the most prevalent neural network models, a variety of other models have also been employed. These, less typical solutions are discussed in this section.

One of the simplest and oldest neural network models is a feed-forward network (a Multilayer Perceptron—MLP), depicted in Fig. 12. Despite its simplicity, the model can be effectively applied to solving even complex tasks, such as those related to music [95]. MLP was used in [122] for music recommendation, where embeddings of various attributes related to music preferences (such as searched songs, artists and played songs), as well as user demographics (age, country of origin, etc.) were collected and concatenated into a single vector. An MLP network was then developed to predict new user preferences based on this vector. The network comprised 3 hidden layers featuring 400, 300 and 200 neurons, respectively, and the output layer included an embedded representation of the predicted preferences.

In [93], a Restricted Boltzmann Machine (RBM) [112] was utilized. RBM is a classical two-layer model that predicts the probability distribution over a set of inputs.

The constructed model addressed the inquiry "is x more akin to y than z," in which x, y and z represented musical compositions. Two extensions of the standard RBM model: ConditionaL Neural Network (CLNN) and Masked CLNN, used to tackle the problem of frequency shifts in spectrograms, were proposed in [160] for audio classification. A Masked CLNN model was also used in [79] to solve the problem of music genre classification.

Another model, a Transformer network [174], has recently gained popularity due to its ability to process sequential data, rendering it useful for music-related problems. In a generation problem explored in [26], a Transformer network was adapted with an additional 4 self-attention layers and 8 multi-head attention modules.

Another Transformer-based approach was described in [37] where the authors' primary objective was to employ compound words instead of individual tokens to represent a song. Tokens that defined a single music event (such as pitch and duration) were grouped into a single supertoken and fed as input to the network, allowing for a condensed representation of the entire song. Additionally, distinct feed-forward heads for various token types were used as part of the network construction.

A comparable technique utilizing the Transformer network and token aggregation was introduced in [38]. The original Transformer model underwent alterations to effectively grasp the underlying relationships between bars through modifying the attention module. The tokens present in a single bar were amalgamated into a summary token, with the attention module using either individual tokens (fine-grained attention) or summary tokens (coarse-grained attention) to the structure-related and other bars, respectively. Structure-related bars were determined based on the likelihood of a bar repetition during creation. Another Transformer-based model (the Swin-Transformer [115]) was utilized in [96] for instrument recognition based on three representations: spectrogram, tempogram [113] and *modgdgram* [114]. The input was divided into non-overlapping patches, from which an embedding was created. These embedded patches were subsequently processed by two Swin-Transformer blocks and merged prior to processing by the last two Dense Layers that were employed for instrument classification.

A Generative Adversarial Network [182], presented in Fig. 13, is another commonly used architecture in music generation problems. Using a GAN involves parallel training of two networks: one (a generator) focusing on the generation task and the other one (a discriminator) learning to differentiate between real and fake objects. Typically, both networks have symmetric architectures. The generator's objective is to create objects that would deceive the discriminator, while the discriminator's task is to differentiate between the actual and artificial objects (generated by the generator network) with the highest possible accuracy.

GANs were applied in [41] to a problem of generating multi-track piano-roll representations. The authors introduced three approaches. The first one utilized a jamming model based on several generators, each dealing with one track and being completely independent of the others. The second one used a single generator simultaneously for all tracks, while the last one was a hybrid model that utilized multiple generators that not only took the intra-track but also the inter-track information as its input.

An extension of the GAN architecture was presented in [62] as a Conditional-Hybrid GAN. The model's purpose was to generate a sequence of successive sounds conditioned on lyrics. Three separate modules with the same structure generated a pitch, a duration and a rest information, respectively. A module utilized the previously generated element and the next syllable embedding (conditioning) to process it through the Relational Memory Core (RMC) [183], which enabled modelling of long-term dependencies. The discriminator analysed the combined pitch, duration, rest and syllable embedding, also with a single RMC block.

GAN architecture was also presented in [60] for music style transfer. The Relativistic Average GAN [184] was utilized as a model's skeleton. In a standard GAN setup, the generator was trained to deceive the discriminator,

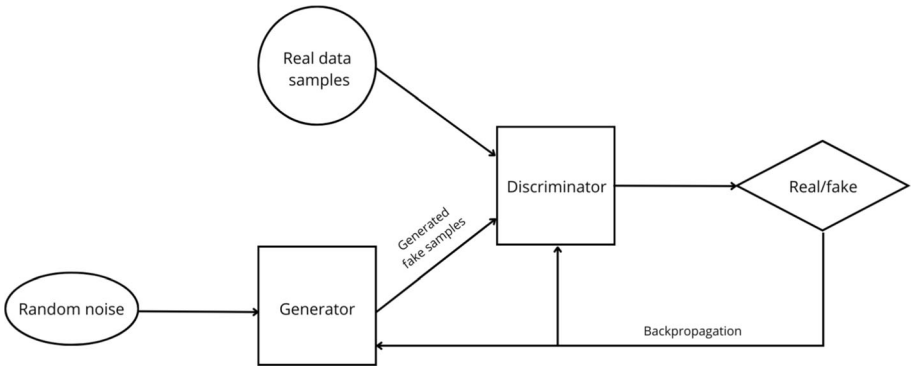


Fig. 13 A scheme of GAN model with a generator producing new data and a discriminator trying to distinguish the real data from the artificially produced

while the discriminator was trained to determine if music belongs to a desired style (target domain).

In [40], the authors endeavoured to address the challenge of extracting temporal features from GANs, an issue particularly salient in music. To this end, an attention mechanism was incorporated into the model. The music was generated in a piano-roll format, separately for each track, in a distinct branch, based on a shared vector produced by the core generator. The attention mechanism was added to the end of each branch, and the resulting outputs were combined to create a multi-track representation.

An interesting method for addressing an emotion prediction in music was outlined in [130], wherein a triplet neural networks (TNN) [185] architecture was used to construct a meaningful low-level representation that was applied directly to the prediction task. TNN utilizes three inputs: an anchor, positive samples and negative samples, with the positive input belonging to the same class as the anchor, and the negative input assigned to a distinct class. The network's task is to produce a representation that minimizes the distance between the anchor and the positive examples and maximizes the distance between the anchor and the negative examples. By doing so, vectors from one class are positioned closely together in the created space, while also ensuring that different classes are separated from each other. In [130], these representations were utilized by classic regression algorithms.

The same topic of emotion recognition was presented in [134] and was addressed using an Adaptive Neuro-Fuzzy Inference System (ANFIS) network [147]. The ANFIS model comprised five layers, utilizing both adaptive nodes (parameters that were tuned during training) and fixed nodes (parameters that were not altered during training) [186]. The model aimed to perform classification, based on a set of properties extracted from *.mp3* files, through a fuzzy inference.

In [55], a similarity embedding network was utilized to create music medleys, i.e. combinations of several music pieces, by determining the chronological order of two music fragments. The model was based on the Siamese network [187], which permits two inputs to be processed simultaneously. In the first segment of the model, the Siamese network extracted attributes from the inputs and generated a similarity matrix. This matrix was then processed using a series of convolution layers, to ascertain whether the input fragments were in the correct order. The model was trained by providing positive examples (in the correct order) and negative examples (incorrect order or examples not adjacent in the timeline) as the input.

An interesting model that simultaneously operates on two representations was introduced in [102]. A waveform and a combined frequency and periodicity representations [117] were utilized for the extraction of the sung melody. Initially, both representations were processed separately using convolutional layers and an LSTM block. These representations were then combined and jointly processed by a specially developed Gated Attentive Fusion Module, consisting of a GRU and self-attention. Finally, an MLP classifier was used for a sung pitch classification.

4.2 Evolutionary algorithms

A popular class of algorithms, also applied to music-related problems, are Evolutionary Algorithms (EAs)[188]. EAs are primarily implemented in optimization tasks, seeking to transfer processes observed in natural evolution to artificial intelligence. EA operates by maintaining a number of possible solutions to a specific problem and applies an evaluation function to determine if solution x surpasses solution y. Then, with the processes of selection, mutation and crossover (suitably chosen for the problem) applied to the maintained solutions, a search through the space of all solutions is conducted to pursue the top-rated solution. EAs have been used in the field of music for many years [16] to solve both engineering (such as sound synthesis) as well as creative problems (like music generation). This section includes various examples of EA application to musical problems that can be defined as an optimization problem.

In [48], an algorithm for modelling expressive music performance is presented, where the fundamental concept is to devise a collection of rules that precisely depict deviations from the music score in expressively played music. The performance is evaluated by different criteria, including note length ratios (longer, shorter, much shorter), pitch and Narmour's group [189]. A specific coding system is used to create rules, which are then represented in a binary form. An evolutionary algorithm is employed to search for highly accurate rules over the training dataset. Using the sequential covering algorithm, a set of rules to describe expressive performance is created. One rule, created with EA, is added at a time, and the rule selection is repeated until the entire training set is covered.

The work [27] presents the use of an EA to produce music, where the goal is to create 16-measure pieces of music. However, the solution representations (chromosomes) vary in length: each chromosome contains a sequence of pitches in the melody, with each pitch having an assigned duration. Therefore, the representation of multiple shorter musical notes is lengthier than that of fewer, longer music notes, although the overall duration of all pieces remains constant at 16 measures. To address this challenge, a geometric crossover [190] is applied. The evaluation function utilizes tables to score pairs of pitches relative to the chords. The sum of scores for all pairs determines the overall score for each chromosome (candidate solution).

One possibility for EA application is to use it for multi-objective optimization, where the objective is to optimize more than one goal. In [19] this method is applied to the 4-voice music generation problem. This approach involves the simultaneous optimization of two aspects: the melodic flow and the harmonic quality. The former is optimized based on the relationships between notes, and the latter is evaluated with the function derived from the principles of the harmony theory.

In [94], a multi-objective optimization is utilized for the selection of features that enable proper segmentation of a music piece. The paper considers three possible types of boundaries: instrumental change, tempo change and key change. To address this topic, different measures including accuracy, recall and F-measure are used for the algorithm's performance evaluation. Based on these measures, several scenarios are defined which consist in optimizing two particular measures (e.g. maximizing the F-measure for one boundary and minimizing the recall for the other boundaries, to increase the likelihood of predicting a specific boundary and simultaneously reduce the chances of detecting boundaries of a different type). For the set of pre-defined music features, a multi-objective EA is applied to select the subset most suitable for music segmentation.

A certain extension of EAs is Memetic Algorithms (MAs), which additionally incorporates a local optimization mechanism as a component of a space search strategy. This method was applied in [18] for the unfigured bass harmonization. The harmonization process was divided into two main stages: the first one involved generating harmonic functions for a given melodic line (the figuration step) and the second one consisted in its harmonization based on the generated functions. In the first step, dynamic programming was used with tables defining the quality of connections between specific chords. The second step was divided into two sub-steps. Firstly, an EA was used to search the space of four-voice compositions and identify a suitable melody subspace. Then, the identified subspace was searched using local optimization techniques such as tabu search [191], simulated annealing [192] and variable neighbourhood search [193]. A solution was assessed based on the analysis of the associations between chords and notes, using a specific set of rules.

An effort to develop an algorithm that can reflect the requirements of the user in the music generation problem was made in [32]. Based on a training dataset (created by the user), a collection of patterns (both rhythmic and melodic) was developed and utilized, with use of symbiotic evolution [194]. One notable distinction in the symbiotic evolution algorithm is the preservation of two populations: one that contains individuals with partial solutions and another that contains completed solutions, which are combinations of partial solutions. Looking at a piece of music as a combination of motifs, the symbiotic evolution was easily applicable. Through evolution melodic templates were generated, describing the melodic and rhythmic properties, but not specific pitches, which were generated using a specifically designed heuristic algorithm.

4.3 Other approaches

In addition to the most commonly used methods (neural networks and evolutionary algorithms), there are yet other approaches to solve music-related problems. This section presents examples of such less popular methods.

A population-based technique similar to EAs is the Artificial Immune System (AIS). Two examples of using AIS have been presented in [24, 152]. Both systems were based on opt-aiNet [195], which can generate multiple solutions for a multimodal fitness function optimization. This property has been taken advantage of in [24], where the authors aimed to create a chord progression, in collaboration with the user. Based on the defined fitness function, the two previous chords, and using chord distances in Tonal Interval Space [196], various chords were generated by the system, as a pool of candidates for the final selection made by the user. In a very similar way (by means of generating a pool of possible solutions), opi-aiNet was used in [152] for the problem of creating music orchestrations.

Another interesting approach is ensemble learning, which relies on employing multiple algorithms simultaneously to tackle a given problem. It is common to use either a parallel or sequential approach for ensemble learning, where either the learning of individual algorithms is independent of each other or the relationships between algorithms are exploited. Ensemble learning has been applied to classification tasks in [78] and [86]. In [78], the topic concerned the classification of performers and music genres, with small decision trees (in the extreme case with only two leaves) serving as individual algorithms. In [86], classification of performers was addressed through ensemble learning, where a single classifier was created using linear discriminant analysis based on selected music features.

Some algorithms are better suited for certain types of problems. An instance of such an algorithm is k-Nearest Neighbours (k-NN) [197], which is usually used for solving classification or recommendation problems [92, 140]. In [140], a suitable music representation was established, and the recommendation method relied on k-NN retrieval with Cosine similarity used as a distance measure. Similarly, in [92], the k-NN algorithm was utilized to categorize music. The collection of songs was classified according to an extracted melodic line, and pairwise similarities were determined, resulting in a similarity matrix.

The quality of the similarity matrix was assessed by through the N-fold k-NN classification [198].

Certain topics can be formulated as regression problems. In [46], linear regression was proposed as a means of modelling expressive features (in addition to nonlinear models such as feed-forward and recurrent neural networks). Expressive properties were modelled using a linear function also in [50], where the differences between neutral performance and performance with specific intentions (soft, hard, dark, light, etc.) were considered. Created models, which described alterations in selected expressive qualities, enabled the possibility to convert neutral performance to performance with a specific intention. The use of logistic regression in classification was demonstrated in [129], where among various simple classifiers, the penalized logistic regression model produced the best results for emotion classification in music. Logistic regression was also applied in [89] for composer classification, but, in this problem setup, was outperformed by the SVM [142].

In several works [17, 22, 23, 90], reinforcement learning (RL) [199] is utilized, especially Markov Decision Processes (MDPs) [200] and Hidden Markov Models (HMMs) [201]. In [17], a MDP was constructed to model chord progression for automatic harmonization, with the reward function either derived from classical harmony theory or inferred from a specific set to achieve a certain style of harmonization. Specifically, in the conducted experiments, the reward function was based on the studied expectations of the chord progression [202].

Similarly, the authors of [90] utilized HMMs to assess the similarity of sung music to its potential original composition. Since human singing (or humming) is susceptible to errors, it was necessary to take this into account during model development. The probability of a selected target matching a given sung query was defined as the likelihood of generating that query from the target, which was calculated as a product of probabilities of transitions between successive hidden states defined by different types of possible query alterations. For the real-time creation of an accompaniment [23], deep reinforcement learning was employed. The use of RL in accompaniment production facilitated generation of extended and coherent phrases, as the agent aimed to maximize long-term rewards instead of optimizing individual steps. The model comprised two parts: the generation agent, which utilized the actor-critic model with generalized advantage estimator [203] and neural network-based reward model. The generation agent sequentially produced notes, aided by a human input and the reward agent allocated a score for each note considering both the horizontal and vertical perspective.

In [22], a specific style transfer task problem was undertaken: for a provided Chinese folk-style musical piece, an automatic Western-style counterpoint was generated using RL. The developed model comprised a generator module adapted from [203], focused on maximizing long-term rewards. The reward module was composed of two sub-modules: the inter-rewarder, which assessed the quality of Western style counterpoint, and the style-rewarder, which modelled the patterns of Chinese folk songs. The style-rewarder was also updated through inverse reinforcement learning [204] during generator training, while inter-rewarder remained unchanged during the whole process.

Algorithms utilizing various distance or similarity measures are commonly employed for classification purposes. In [149], a system illustrating sound through images used Dynamic Time Wrapping [205] to assess the similarity of extracted music (MFCC) and image (colour and texture) features.

A similar technique was employed in [150] for detecting plagiarism, where music was first transformed into a vector and then a fuzzy similarity was computed to determine whether a melody was plagiarized. A system developed for objective assessment of automatically generated music also relied on distance measures [14]. Based on the characteristics of the produced music and the music from the training dataset, the quality of the generated music was assessed by examining the distances within the produced set (generation consistency) and between the produced and training sets (generation efficiency).

Yet another possible approach to various music-related tasks is to utilize heuristic or probabilistic algorithms [29, 54, 61, 123]. The benefit of such solutions lies in their adaptability to any problem type.

In [123], a heuristic was developed to suggest songs based on fuzzy sets. [54] focused on generating musical mixes. The model created mixes by blending songs from the database, using techniques such as beat tracking, selecting cue points and matching.

A music generation problem is addressed in [29] by combining a heuristic with a probabilistic approach, grounded in statistical analysis of a set of classical musical pieces. A music construction process involves several steps, analogous to human music composition. Firstly, the phrase structure and rhythmic parameters are selected. Next, a melodic rhythm is created based on these selections. These steps are implemented by drawing one of the possibilities present in the set on which the generation was based. Harmonic progression and melodic contour are created with the aid of a Probabilistic Context-Free Grammar based on conducted Schenkerian analysis [70] of other pieces in the dataset. Lastly, the pitch selection for the melody is determined using heuristics. A probabilistic approach is also presented in [61]. The task of generating music as an imitation of a given input is divided into 4

parts: structure creation, chord progression, melody and bass. Structure creation and chord progression are based on statistical analysis of the input and the aforementioned classical music dataset. Melody and bass are created by recognizing patterns in the input using heuristics and applying them to a new song.

Finally, there are a number of works that apply particular methods that are tailored to a specific problem representation. In [45] fuzzy techniques were applied to assess the music expressivity. The evaluation involved examining rubato (changes in rhythm), loudness and timbre. A tree search algorithm utilizing only pitch content was employed in [148] to solve the challenge of aligning audio with sheet music. An intriguing solution to the problem of recommendation was presented in [120], where music data were represented using a heterogeneous information graph. From this constructed graph, it became feasible to obtain low-dimensional representations of music, from which the user's preferences (both global and contextual) were inferred, and used for making a recommendation afterwards. Yet another approach to the recommendation, based on matrix factorization algorithm [206], was presented in [124], where songs were coded using Song2Vec method [207].

5 Summary and challenges

According to the presented categorization of topics (generation, classification, recommendation), gauging the quality of the outcomes, and comparing the efficacy of the methods across various models is the most extensive in classification-related tasks. Regardless of a particular subproblem, such as classification of genres or composers, evaluating the method's effectiveness is commonly based on labelling performance.

Generally speaking, it is reasonable to conclude that the problem of classifying music genres has been tackled quite successfully. Representative results from multiple experiments on various datasets are presented in Table 9. It should be underlined that the datasets include varying numbers of genres, ranging from 6 in the BRMD dataset to 16 in the FMA dataset. The results demonstrate that the problem of genre classification can be solved quite effectively with an accuracy exceeding 90% (93% [80], 98% [83]), even when dealing with numerous genres (13 and 10, respectively).

Similarly, the quality of composer/performer classification was evaluated in several papers. For the artists' classification the artist20 dataset [210] was used in [88], with an F1 score of 0.937. Likewise, in [78], the artists were classified on the USPOP dataset [108] (about 59% accuracy). In [86], the performers were classified using the Vienna4x22 Piano Corpus collection [211] with around 70% accuracy.

While the presented results can be generally considered as promising, regardless of the classification problem, there is still room for improvement, especially in the context of composer/performer classification. Furthermore, it is worth noting that a relatively rarely addressed topic is the classification directly related to music expression (e.g. recognition of performers of purely instrumental pieces) or compositional style (e.g. recognition of composers of purely instrumental music of a similar character).

Generally, it is more challenging to evaluate the results obtained for music generation-related tasks. For this reason, attempts are usually made to asses generated music in two ways: statistically or human-based. When conducting statistical or numerical evaluations, it is typical to utilize standard metrics or sometimes develop custom ones. If music generation is based on modelling specific parameters (such as loudness or note duration), or if the notes/chords themselves are generated without audio [18], it is possible to use recall and precision metrics (to evaluate these selected parameters) in relation to a test dataset. Sometimes, statistical analysis is also performed (in both audio and note generation settings), through comparison of selected statistics (e.g. pitch range, note density, number of notes per bar, rhythmic properties, etc.) [14, 22, 26, 28, 30, 33, 59].

The other popular way of evaluation involves human users/experts and can be carried out in two ways:

Table 9 Genre classification accuracy from different articles, referring to various datasets: Ballroom [105], Extended Ballroom [106], Homburg [208], GTZAN [104], FMA [107], BRMD [209]

	Ballroom	Extended Ballroom	Homburg	GTZAN	FMA	BRMD
[160]	~90%		~60%			
[80]		~93%		~90%		
[82]				~72%	~65%	
[81]				~75%		
[83]				~98%	~74%	
[84]						~78%

Table 10 A breakdown of the various methods used in evaluation of the results of generative models

Evaluation method		Number of papers	Papers
Objective		8	[14, 18, 22, 26, 28, 30, 33, 59]
Subjective	Theoretical	4	[18, 20, 21, 24]
	Listening	19	[17, 18, 22, 23, 26, 30, 34, 35, 42, 44, 51–53, 58–60, 66, 68, 71]

theoretical evaluation (more common in note generation) and listening-based evaluation. The main challenge of human evaluation is the subjectivity of the process, whether executed by an expert or an amateur. To increase objectivity, the assessment is usually conducted by dozens of people, and their ratings are averaged. Theoretical evaluation is primarily used for generating notes, including accompaniment, harmonization and chord progressions [18, 20, 21, 24]. Its purpose is to assess (usually by a music expert) how well the generated music adheres to the theoretical principles of the music. However, it is challenging to thoroughly carry out the evaluation and consider all the theoretical nuances, given the number and diversity of rules and potential deviations from them, resulting from the overall primacy of the sound over the theoretical correctness. For this reason, supplementary listening evaluations are also often conducted [17, 18, 22, 23, 26, 30, 34, 35, 42, 44, 51–53, 58–60, 66, 68, 71]. Most commonly, individuals with varying levels of expertise in music are presented with a set of questions, typically answered on a Likert scale [212]. For amateur evaluations, background information pertaining to the respondent's musical knowledge (such as whether they have experience playing an instrument) may also be collected before the essential questions are asked [22, 23]. The questions themselves typically focus on aspects such as musicality, fidelity, quality, smoothness, reality, integrity, etc., of the generated music. Table 10 presents a breakdown of the papers by the evaluation method used.

Evaluating computer-generated music, particularly when comparing different models is also challenging. On the one hand, constructing and utilizing metrics that are as objective as possible facilitates the straightforward evaluation and comparison of various generating models [14]. However, metrics based on statistical properties and distance-based metrics relative to the training and test datasets fail to provide a comprehensive assessment, e.g. in terms of musicality and integrity. On the other hand, evaluations made by humans, which are inherently subjective, are lacking a standardized reference point (such as fragments of music to compare), which, for obvious reasons, is practically impossible to define. It is also worth noting that, while objective evaluation methods are sometimes employed, the majority of the papers still utilize subjective/human evaluation techniques, as evidenced in Table 10.

An area that has witnessed significant development in recent years is the generation of expressive music, which aims to imitate human musical performance. However, current expressive music models are mainly developed without considering the unique aspects of an individual's performance style, which presents an intriguing challenge. Furthermore, there are occasional mentions in the literature about combining or transferring various musical genres, but such topics remain underexplored and present an interesting research avenue [23, 58, 60].

For the recommendation task, the commonly used metrics are precision, recall, F1 and hit rate [120–122, 124, 126, 132, 133, 140], which are calculated based on the coverage of the recommended tracks to listened tracks. The AUC-ROC metric [137–139] is also used for the auto-tagging problem. For the recognition/prediction of emotions in music, which is closely linked to recommendation problem, standard metrics can be used quite effectively. If a given emotion (or a group of emotions) is recognized [128, 147], one can apply the accuracy measure to assess the model's quality. Classic regression evaluation metrics (such as R^2, explained variance score, etc.) are often used [129, 130, 132, 136] when making emotion prediction with a regression approach, especially when it is based on Thayer's emotion model [141].

While evaluation of recommendation models using popular metrics is feasible, comparing them with each other is not always straightforward. In the case of recommendations based on the recognition of emotions, one of the problems is identification of specific musical emotions that could be used to train the models. Such a selection is highly subjective, as it is usually performed by people. The absence of standardized and acknowledged datasets makes comparing models even more challenging.

A similar issue, i.e. the absence of standardized datasets, is apparent in the traditional recommendation problem, as well, particularly in the task of forming a dynamic playlist.

High-quality recommendations are definitely needed in the face of ever-growing music databases (e.g. streaming services). While classical recommendation problems have been studied quite extensively, the problem of recommending music based not only on the listener's global preferences but also on the context of a playlist sequence is

gaining popularity and undoubtedly deserves further exploration [120, 121, 124].

6 Conclusions

Music has been a fundamental aspect of human culture throughout history. As one of the Arts, it has consistently been inextricably linked to the creativity of both the artist-creator and the artist-performer. Therefore, with the development of artificial intelligence, research has begun on the possibilities of its application in music-related problems. In this paper a categorization of the researched problems (classification, generation, recommendation) is provided, along with its further division into subgroups, and in each of these subgroups the currently addressed topics are discussed. For each of these topics, a description of the problem and proposed solutions is presented, along with a discussion of the relevant papers addressing the topic. Among all the considered topics, the problem of expressive music generation seems to be the most challenging, mainly due to its strong connection with creativity, which is generally perceived as a typically human trait. The associated problem of objective evaluation of the generated music also poses a significant challenge with regard to the assessment of results of different models and the possibility of their cross-comparison. Another area that has gained importance in recent years is the problem of copyright in the context of generative models. In the field of music recommendation, an intriguing approach is the utilization of contextual information (e.g. recently listened songs) to generate targeted recommendations that account for both the user's needs and their current mood. Presented problems are further supplemented with examples of real-life applications, including those related to music education.

Additionally, the classification of applied methods (presented independently of the addressed tasks) illustrates that neural networks, particularly deep learning models, are the most popular and the most effective approaches. It is also worth noting that the employed methods utilize state-of-the-art solutions derived from other fields, e.g. Transformer models are used in the problems of music generation or music classification.

The problems, methods and outcomes discussed in the paper demonstrate the range of research in the field of music AI applications, emphasizing its diversity, and showing that artificial intelligence has the potential to surpass humans in many of these research tasks. At the same time, there is a vast range of issues that still require innovative approaches and pose a challenge for contemporary music AI research, especially in the field of music expressivity and contextual music recommendation.

Availability of data and materials The paper contains all of the data.

Declarations

Conflict of interest Statement The authors report there is no conflict of interest to declare.

References

1. Barton G (2018) Music learning and teaching in culturally and socially diverse contexts: implications for classroom practice. Springer
2. Miell D, MacDonald RAR, Hargreaves DJ (2005) Musical communication. Oxford University Press
3. Robinson J, Hatten RS (2012) Emotions in music. Music Theory Spectr 34:71–106
4. Wiggins GA (1995) Understanding music with AI – perspectives on cognitive musicology. In: Balaban M, Ebcioglu K, Laske O (eds) Artificial intelligence 79: 373–385
5. Camurri A, Catorcini A, Innocenti C, Massari A (1995) Music and multimedia knowledge representation and reasoning: the HARP system. Comput Music J 19(2):34–58
6. Balaban M (1996) The music structures approach to knowledge representation for music processing. Comput Music J 20(2):96–111
7. Miranda ER (1995) An artificial intelligence approach to sound design. Comput Music J 19(2):59–75
8. Weihs C, Ligges U, Mörchen F, Müllensiefen D (2007) Classification in music research. Adv Data Anal Classif 1:255–291
9. Fernández JD, Vico F (2013) AI methods in algorithmic composition: a comprehensive survey. J Artif Intell Res 48:513–582
10. Kaliakatsos-Papakostas M, Floros A, Vrahatis MN (2020) Artificial intelligence methods for music generation: a review and future perspectives. Nature-inspired computation and swarm intelligence, pp 217–245
11. Ndou N, Ajoodha R, Jadhav A (2021) Music genre classification: a review of deep-learning and traditional machine-learning approaches. In: 2021 IEEE international IOT, electronics and mechatronics conference (IEMTRONICS), pp 1–6. https://doi.org/10.1109/IEMTRONICS52119.2021.9422487
12. Casey MA, Veltkamp R, Goto M, Leman M, Rhodes C, Slaney M (2008) Content-based music information retrieval: current directions and future challenges. Proc IEEE 96(4):668–696
13. Song Y, Dixon S, Pearce M (2012) A survey of music recommendation systems and future perspectives
14. Yang L-C, Lerch A (2020) On the evaluation of generative models in music. Neural Comput Appl 32:4773–4784

15. Cope D (1991) Computers and musical style. Computer music and digital audio series, A-R Editions. https://books.google.pl/books?id=SkoZAQAAIAAJ
16. Miranda ER (2004) At the crossroads of evolutionary computation and music: self-programming synthesizers, swarm orchestras and the origins of melody. Evol Comput 12(2):137–158. https://doi.org/10.1162/106365604773955120
17. Yi L, Goldsmith J (2010) Decision-theoretic harmony: a first step. Int J Approx Reason 51(2):263–274
18. Muñoz E, Cadenas JM, Ong YS, Acampora G (2016) Memetic music composition. IEEE Trans Evol Comput 20(1):1–15. https://doi.org/10.1109/TEVC.2014.2366871
19. De Prisco R, Zaccagnino G, Zaccagnino R (2020) EvoComposer: an evolutionary algorithm for 4-voice music compositions. Evol Comput 28(3):489–530. https://doi.org/10.1162/evco_a_00265
20. Mycka J, Żychowski A, Mańdziuk J (2022) Human-level melodic line harmonization. In: Groen D, Mulatier C, Paszynski M, Krzhizhanovskaya VV, Dongarra JJ, Sloot PMA (eds) Computational science-ICCS 2022. Springer, Cham, pp 17–30
21. Mycka J, Żychowski A, Mańdziuk J (2023) Toward human-level tonal and modal melody harmonizations. J Comput Sci 67:101963. https://doi.org/10.1016/j.jocs.2023.101963
22. Jiang N, Jin S, Duan Z, Zhang C (2020) When counterpoint meets Chinese folk melodies. Adv Neural Inf Process Syst 33:16258–16270
23. Jiang N, Jin S, Duan Z, Zhang C (2020) Rl-duet: online music accompaniment generation using deep reinforcement learning. Proc AAAI Conf Artif Intell 34:710–718. https://doi.org/10.1609/aaai.v34i01.5413
24. Navarro-Cáceres M, Caetano M, Bernardes G, de Castro LN (2019) Chordais: an assistive system for the generation of chord progressions with an artificial immune system. Swarm Evol Comput 50:100543. https://doi.org/10.1016/j.swevo.2019.05.012
25. Aminian M, Kehoe E, Ma X, Peterson A, Kirby M (2020) Exploring musical structure using Tonnetz lattice geometry and lstms. In: Krzhizhanovskaya VV, Závodszky G, Lees MH, Dongarra JJ, Sloot PMA, Brissos S, Teixeira J (eds) Computational science - ICCS 2020. Springer, Cham, pp 414–424
26. Makris D, Agres KR, Herremans D (2021) Generating lead sheets with affect: a novel conditional seq2seq framework. In: 2021 international joint conference on neural networks (IJCNN), pp 1–8
27. Nam Y-W, Kim Y-H (2017) Melody composition using geometric crossover for variable-length encoding. GECCO '17, pp 37–38. Association for computing machinery, New York, NY, USA. https://doi.org/10.1145/3067695.3082041
28. Zhu H, Liu Q, Yuan NJ, Qin C, Li J, Zhang K, Zhou G, Wei F, Xu Y, Chen E (2018) Xiaoice band: a melody and arrangement generation framework for pop music. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining. KDD '18, pp 2837–2846. Association for computing machinery, New York, NY, USA. https://doi.org/10.1145/3219819.3220105
29. Hahn S, Zhu R, Mak S, Rudin C, Jiang Y (2023) An interpretable, flexible, and interactive probabilistic framework for melody generation. In: Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining. KDD '23, pp 4089–4099. Association for computing machinery, New York, NY, USA. https://doi.org/10.1145/3580305.3599772
30. Wu J, Liu X, Hu X, Zhu J (2020) Popmnet: generating structured pop music melodies using neural networks. Artif Intell 286:103303. https://doi.org/10.1016/j.artint.2020.103303
31. Sulyok C, Harte C, Bodó Z (2019) On the impact of domain-specific knowledge in evolutionary music composition. In: Proceedings of the genetic and evolutionary computation conference. GECCO '19, pp 188–197. Association for computing machinery, New York, NY, USA. https://doi.org/10.1145/3321707.3321710
32. Otani N, Okabe D, Numao M (2018) Generating a melody based on symbiotic evolution for musicians' creative activities. In: Proceedings of the genetic and evolutionary computation conference. GECCO '18, pp 197–204. Association for computing machinery, New York, NY, USA. https://doi.org/10.1145/3205455.3205479
33. Sabathé R, Coutinho E, Schuller B (2017) Deep recurrent music writer: memory-enhanced variational autoencoder-based musical score composition and an objective measure. In: 2017 international joint conference on neural networks (IJCNN), pp 3467–3474. https://doi.org/10.1109/IJCNN.2017.7966292
34. Guo Z, Makris D, Herremans D (2021) Hierarchical recurrent neural networks for conditional melody generation with long-term structure. In: 2021 international joint conference on neural networks (IJCNN), pp 1–8
35. Roberts A, Engel J, Raffel C, Hawthorne C, Eck D (2018) A hierarchical latent vector model for learning long-term structure in music. In: International conference on machine learning, pp 4364–4373. PMLR
36. Muhamed A, Li L, Shi X, Yaddanapudi S, Chi W, Jackson D, Suresh R, Lipton ZC, Smola AJ (2021) Symbolic music generation with transformer-gans. Proc AAAI Conf Artif Intell 35(1):408–417
37. Hsiao W-Y, Liu J-Y, Yeh Y-C, Yang Y-H (2021) Compound word transformer: learning to compose full-song music over dynamic directed hypergraphs. Proc AAAI Conf Artif Intell 35:178–186
38. Yu B, Lu P, Wang R, Hu W, Tan X, Ye W, Zhang S, Qin T, Liu T-Y (2022) Museformer: transformer with fine- and coarse-grained attention for music generation. In: Oh AH, Agarwal A, Belgrave D, Cho K (eds) Advances in neural information processing systems
39. Walder C, Kim D (2018) Neural dynamic programming for musical self similarity. In: International conference on machine learning, pp 5105–5113. PMLR
40. Guan F, Yu C, Yang S (2019) A Gan model with self-attention mechanism to generate multi-instruments symbolic music. In: 2019 international joint conference on neural networks (IJCNN), pp 1–6. https://doi.org/10.1109/IJCNN.2019.8852291
41. Dong H-W, Hsiao W-Y, Yang L-C, Yang Y-H (2018) MuseGAN: multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In: Proceedings of the AAAI conference on artificial intelligence, vol 32
42. Jia B, Lv J, Pu Y, Yang X (2019) Impromptu accompaniment of pop music using coupled latent variable model with binary regularizer. In: 2019 international joint conference on neural networks (IJCNN), pp 1–6. https://doi.org/10.1109/IJCNN.2019.8852373
43. Borghuis V, Angioloni L, Brusci L, Frasconi P et al (2020) Pattern-based music generation with wasserstein autoencoders and prc descriptions. In: Proceedings of the twenty-ninth international joint conference on artificial intelligence, pp 5225–5227. Christian Bessiere
44. Samuel D, Pilát M (2019) Composing multi-instrumental music with recurrent neural networks. In: 2019 international joint conference on neural networks (IJCNN), pp 1–8. https://doi.org/10.1109/IJCNN.2019.8852430
45. Arcos JL, Guaus E, Ozaslan TH (2013) Analyzing musical expressivity with a soft computing approach. Fuzzy Sets Syst 214:65–74. https://doi.org/10.1016/j.fss.2012.01.019

46. Cancino-Chacón CE, Gadermaier T, Widmer G, Grachten M (2017) An evaluation of linear and non-linear models of expressive dynamics in classical piano and symphonic music. Mach Learn 106:887–909
47. Tobudic A, Widmer G (2006) Relational IBL in classical music. Mach Learn 64(1–3):5–24. https://doi.org/10.1007/s10994-006-8260-4
48. Ramirez R, Maestre E, Serra X (2012) A rule-based evolutionary approach to music performance modeling. IEEE Trans Evol Comput 16(1):96–107. https://doi.org/10.1109/TEVC.2010.2077299
49. Jeong D, Kwon T, Kim Y, Nam J (2019) Graph neural network for music score data and modeling expressive piano performance. In: International conference on machine learning, pp 3060–3070. PMLR
50. Canazza S, De Poli G, Drioli C, Roda A, Vidolin A (2004) Modeling and control of expressiveness in music performance. Proc IEEE 92(4):686–701. https://doi.org/10.1109/JPROC.2004.825889
51. Schimbinschi F, Walder C, Erfani SM, Bailey J (2019) Synthnet: learning to synthesize music end-to-end. In: International joint conferences on artificial intelligence organization, pp 3367–3374
52. Wang B, Yang yh (2019) Performancenet: score-to-audio music generation with multi-band convolutional residual network. Proc AAAI Conf Artif Intell 33:1174–1181. https://doi.org/10.1609/aaai.v33i01.33011174
53. Dieleman S, Van Den Oord A, Simonyan K (2018) The challenge of realistic music generation: modelling raw audio at scale. In: Advances in neural information processing systems 31
54. Vande Veire L, De Bie T (2018) From raw audio to a seamless mix: creating an automated DJ system for drum and bass. EURASIP J Audio Speech Music Process 2018(1):13. https://doi.org/10.1186/s13636-018-0134-8
55. Huang Y-S, Chou S-Y, Yang Y-H (2017) Generating music medleys via playing music puzzle games https://doi.org/10.48550/ARXIV.1709.04384
56. Huang J, Wang J-C, Smith JBL, Song X, Wang Y (2021) Modeling the compatibility of stem tracks to generate music mashups. Proc AAAI Conf Artif Intell 35(1):187–195
57. Engel J, Resnick C, Roberts A, Dieleman S, Norouzi M, Eck D, Simonyan K (2017) Neural audio synthesis of musical notes with Wavenet autoencoders. In: International conference on machine learning, pp 1068–1077. PMLR
58. Hung Y-N, Chiang I-T, Chen Y-A, Yang Y-H (2019) Musical composition style transfer via disentangled timbre representations. In: Proceedings of the twenty-eighth international joint conference on artificial intelligence, pp 4697–4703
59. Choi K, Hawthorne C, Simon I, Dinculescu M, Engel J (2020) Encoding musical style with transformer autoencoders. In: Proceedings of the 37th international conference on machine learning, pp 1899–1908
60. Lu C-Y, Xue M-X, Chang C-C, Lee C-R, Su L (2019) Play as you like: timbre-enhanced multi-modal music style transfer. Proc AAAI Conf Artif Intell 33:1061–1068. https://doi.org/10.1609/aaai.v33i01.33011061
61. Dai S, Ma X, Wang Y, Dannenberg R (2023) Personalised popular music generation using imitation and structure. J New Music Res 51:1–17. https://doi.org/10.1080/09298215.2023.2166848
62. Yu Y, Zhang Z, Duan W, Srivastava A, Shah R, Ren Y (2023) Conditional hybrid Gan for melody generation from lyrics. Neural Comput Appl 35(4):3191–3202. https://doi.org/10.1007/s00521-022-07863-5
63. Bian W, Song Y, Gu N, Chan TY, Lo TT, Li TS, Wong KC, Xue W, Alonso Trillo R (2023) Momusic: a motion-driven human-AI collaborative music composition and performing system. Proc AAAI Conf Arti Intell 37(13):16057–16062. https://doi.org/10.1609/aaai.v37i13.26907
64. Xiong Z, Wang W, Yu J, Lin Y, Wang Z (2023) A comprehensive survey for evaluation methodologies of AI-generated music. arXiv preprint arXiv:2308.13736
65. Schubert E, De Poli G, Roda A (2017) Algorithms can mimic human piano performance: the deep blues of music. J New Music Res. https://doi.org/10.1080/09298215.2016.1264976
66. Scirea M, Eklund P, Togelius J, Risi S (2017) Can you feel it? Evaluation of affective expression in music generated by metacompose. In: Proceedings of the genetic and evolutionary computation conference. GECCO '17, pp 211–218. Association for computing machinery, New York, NY, USA. https://doi.org/10.1145/3071178.3071314
67. Dervakos E, Filandrianos G, Stamou G (2021) Heuristics for evaluation of AI generated music. In: 2020 25th international conference on pattern recognition (ICPR), pp 9164–9171. IEEE
68. Sturm BL, Ben-Tal O, Monaghan Collins N, Herremans D, Chew E, Hadjeres G, Deruty E, Pachet F (2019) Machine learning research that matters for music creation: a case study. J New Music Res 48(1):36–55. https://doi.org/10.1080/09298215.2018.1515233
69. Euler L (1739) Tentamen Novae Theoriae Musicae Ex Certissismis Harmoniae Principiis Dilucide Expositae, Opera Omnia, Series 3, vol 1
70. Pankhurst T (2008) SchenkerGUIDE: a brief handbook and website for Schenkerian analysis, Routledge
71. Oore S, Simon I, Dieleman S, Eck D, Simonyan K (2020) This time with feeling: learning expressive musical performance. Neural Comput Appl 32(4):955–967. https://doi.org/10.1007/s00521-018-3758-9
72. Li Y, Tarlow D, Brockschmidt M, Zemel RS (2016) Gated graph sequence neural networks. CoRR **abs/1511.05493**
73. Ren J, Xu H, He P, Cui Y, Zeng S, Zhang J, Wen H, Ding J, Liu H, Chang Y et al (2024) Copyright protection in generative AI: a technical perspective. arXiv preprint arXiv:2402.02333
74. Samuelson P (2023) Generative AI meets copyright. Science 381(6654):158–161
75. Sturm BL, Iglesias M, Ben-Tal O, Miron M, Gómez E (2019) Artificial intelligence and music: open questions of copyright law and engineering praxis. In: Arts, vol 8, p 115. MDPI
76. Sturm BL, Santos JF, Ben-Tal O, Korshunova I (2016) Music transcription modelling and composition using deep learning. arXiv preprint arXiv:1604.08723
77. Widmer G (2003) Discovering simple rules in complex data: a meta-learning algorithm and some surprising musical discoveries. Artif Intell 146(2):129–148. https://doi.org/10.1016/S0004-3702(03)00016-X
78. Bergstra J, Casagrande N, Erhan D, Eck D, Kégl B (2006) Aggregate features and adaboost for music classification. Mach Learn 65:473–484. https://doi.org/10.1007/s10994-006-9019-7
79. Medhat F, Chesmore D, Robinson J (2017) Music genre classification using masked conditional neural networks. In: Liu D, Xie S, Li Y, Zhao D, El-Alfy E-SM (eds) Neural information processing. Springer, Cham, pp 470–481
80. Yu Y, Luo S, Liu S, Qiao H, Liu Y, Feng L (2020) Deep attention based music genre classification. Neurocomputing 372:84–91. https://doi.org/10.1016/j.neucom.2019.09.054
81. Li J, Han L, Wang Y, Yuan B, Yuan X, Yang Y, Yan H (2022) Combined angular margin and cosine margin softmax loss for music classification based on spectrograms. Neural Comput Appl 34(13):10337–10353. https://doi.org/10.1007/s00521-022-06896-0
82. Wang Z, Muknahallipatna S, Fan M, Okray A, Lan C (2019) Music classification using an improved CRNN with multi-

directional spatial dependencies in both time and frequency dimensions. In: 2019 international joint conference on neural networks (IJCNN). https://doi.org/10.1109/IJCNN.2019.8852128
83. El Achkar C, Couturier R, Atéchian T, Makhoul A (2021) Combining reduction and dense blocks for music genre classification. In: Mantoro T, Lee M, Ayu MA, Wong KW, Hidayanto AN (eds) Neural information processing. Springer, Cham, pp 752–760
84. Pereira RM, Costa YMG, Aguiar RL, Britto AS, Oliveira LES, Silla CN (2019) Representation learning vs. handcrafted features for music genre classification. In: 2019 international joint conference on neural networks (IJCNN). https://doi.org/10.1109/IJCNN.2019.8852334
85. Kostrzewa D, Kaminski P, Brzeski R (2021) Music genre classification: looking for the perfect network. In: Paszynski M, Kranzlmüller D, Krzhizhanovskaya VV, Dongarra JJ, Sloot PMA (eds) Computational science - ICCS 2021. Springer, Cham, pp 55–67
86. Stamatatos E, Widmer G (2005) Automatic identification of music performers with learning ensembles. Artif Intell 165(1):37–56. https://doi.org/10.1016/j.artint.2005.01.007
87. Hu S, Liang B, Chen Z, Lu X, Zhao E, Lui S (2021) Large-scale singer recognition using deep metric learning: an experimental study. In: 2021 international joint conference on neural networks (IJCNN). https://doi.org/10.1109/IJCNN52387.2021.9533911
88. Nasrullah Z, Zhao Y (2019). Music artist classification with convolutional recurrent neural networks. https://doi.org/10.1109/IJCNN.2019.8851988
89. Herremans D, Sörensen K, Martens D (2015) Classification and generation of composer-specific music using global feature models and variable neighborhood search. Comput Music J 39(3):71–91
90. Meek CJ, Birmingham WP (2004) A comprehensive trainable error model for sung music queries. J Artif Int Res 22(1):57–91
91. Williams D, Pooransingh A, Saitoo J (2017) Efficient music identification using orb descriptors of the spectrogram image. EURASIP J Audio Speech Music Process 2017(1):17. https://doi.org/10.1186/s13636-017-0114-4
92. Kroher N, Díaz-Báñez J-M (2018) Audio-based melody categorization: exploring signal representations and evaluation strategies. Comput Music J 41(4):64–82. https://doi.org/10.1162/comj_a_00440
93. Tran SN, Ngo S, Garcez Ad (2020) Probabilistic approaches for music similarity using restricted Boltzmann machines. Neural Comput Appl 32(8):3999–4008. https://doi.org/10.1007/s00521-019-04106-y
94. Vatolkin I, Ostermann F, Müller M (2021) An evolutionary multi-objective feature selection approach for detecting music segment boundaries of specific types. GECCO '21, pp 1061–1069. Association for computing machinery, New York, NY, USA. https://doi.org/10.1145/3449639.3459374
95. Kostek B (2004) Musical instrument classification and duet analysis employing music information retrieval techniques. Proc IEEE 92(4):712–729. https://doi.org/10.1109/JPROC.2004.825903
96. Cr L, Rajan R (2022) Transformer-based ensemble method for multiple predominant instruments recognition in polyphonic music. EURASIP J Audio Speech Music Process. https://doi.org/10.1186/s13636-022-00245-8
97. Schulze S, King EJ (2021) Sparse pursuit and dictionary learning for blind source separation in polyphonic music recordings. EURASIP J Audio Speech Music Process 2021(1):6. https://doi.org/10.1186/s13636-020-00190-4
98. Li Z, Wang H, Zhao M, Li W, Guo M (2018) Deep representation-decoupling neural networks for monaural music mixture separation. In: Proceedings of the AAAI conference on artificial intelligence, vol 32
99. Liu JY, Yang YH (2019) Dilated convolution with dilated gru for music source separation. In: Proceedings of the twenty-eighth international joint conference on artificial intelligence, pp 4718–4724. https://doi.org/10.24963/ijcai.2019/655
100. Qian J, Liu X, Yu Y, li W, (2023) Stripe-transformer: deep stripe feature learning for music source separation. EURASIP J Audio Speech Music Process. https://doi.org/10.1186/s13636-022-00268-1
101. Zhao J, Taniar D, Adhinugraha K, Baskaran V, Wong K (2023) Multi-MMLG: a novel framework of extracting multiple main melodies from midi files. Neural Comput Appl 35:1–18. https://doi.org/10.1007/s00521-023-08924-z
102. Yu S, Yu Y, Sun X, Li W (2023) A neural harmonic-aware network with gated attentive fusion for singing melody extraction. Neurocomputing 521:160–171. https://doi.org/10.1016/j.neucom.2022.11.086
103. Ramírez J, Flores MJ (2020) Machine learning for music genre: multifaceted review and experimentation with audioset. J Intell Inf Syst 55(3):469–499
104. Tzanetakis G, Cook P (2002) Musical genre classification of audio signals. IEEE Trans Speech Audio Process 10(5):293–302. https://doi.org/10.1109/TSA.2002.800560
105. Gouyon F, Klapuri A, Dixon S, Alonso M, Tzanetakis G, Uhle C, Cano P (2006) An experimental comparison of audio tempo induction algorithms. IEEE Trans Audio Speech Lang Process 14(5):1832–1844. https://doi.org/10.1109/TSA.2005.858509
106. Marchand U, Peeters G (2016) The extended ballroom dataset
107. Benzi K, Defferrard M, Vandergheynst P, Bresson X (2016) FMA: a dataset for music analysis. CoRR, abs/1612.01840 6, 39
108. Berenzweig A, Logan B, Ellis DP, Whitman B (2004) A large-scale evaluation of acoustic and subjective music-similarity measures. Comput Music J 28:63–76. https://doi.org/10.1162/014892604323112257
109. Velankar M (2020) MER500 dataset. https://www.kaggle.com/datasets/makvel/mer500. Accessed: 2023-11-13
110. Keren G, Schuller B (2016) Convolutional RNN: an enhanced model for extracting features from sequential data. In: 2016 international joint conference on neural networks (IJCNN), 3412–3419
111. Maaten L, Hinton G (2008) Visualizing data using t-sne. J Mach Learn Res 9:2579–2605
112. Fischer A, Igel C (2012) An introduction to restricted Boltzmann machines, pp 14–36. https://doi.org/10.1007/978-3-642-33275-3_2
113. Tian M, Fazekas G, Black D, Sandler M (2015) On the use of the tempogram to describe audio content and its application to music structural segmentation, pp 419–423.https://doi.org/10.1109/ICASSP.2015.7178003
114. Murthy HA, Yegnanarayana B (2011) Group delay functions and its applications in speech technology. Sadhana 36(5):745–782. https://doi.org/10.1007/s12046-011-0045-1
115. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: hierarchical vision transformer using shifted windows. 2021 IEEE/CVF international conference on computer vision (ICCV), 9992–10002
116. Chang S, Zhang Y, Han W, Yu M, Guo X, Tan W, Cui X, Witbrock M, Hasegawa-Johnson M, Huang T (2017) Dilated recurrent neural networks. Advances in neural information processing systems 2017-December, 77–87. 31st annual conference on neural information processing systems, NIPS 2017 ; Conference date: 04-12-2017 Through 09-12-2017
117. Lu WT, Su L (2018) Vocal melody extraction with semantic segmentation and audio-symbolic domain transfer learning. In: International society for music information retrieval conference

118. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861
119. Gemmeke JF, Ellis DP, Freedman D, Jansen A, Lawrence W, Moore RC, Plakal M, Ritter M (2017) Audio set: an ontology and human-labeled dataset for audio events. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 776–780. IEEE
120. Wang D, Xu G, Deng, S (2017) Music recommendation via heterogeneous information graph embedding. In: 2017 International joint conference on neural networks (IJCNN), pp 596–603. https://doi.org/10.1109/IJCNN.2017.7965907
121. Wang D, Zhang X, Yu D, Xu G, Deng S (2021) Came: content- and context-aware music embedding for recommendation. IEEE Trans Neural Netw Learn Syst 32(3):1375–1388. https://doi.org/10.1109/TNNLS.2020.2984665
122. Briand L, Salha-Galvan G, Bendada W, Morlon M, Tran V-A (2021) A semi-personalized system for user cold start recommendation on music streaming apps. Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining
123. Bosteels K, Kerre EE (2009) A fuzzy framework for defining dynamic playlist generation heuristics. Fuzzy Sets Syst 160(23):3342–3358. https://doi.org/10.1016/j.fss.2009.05.013
124. Cheng Z, Shen J, Zhu L, Kankanhalli M, Nie L (2017) Exploiting music play sequence for music recommendation. In: Proceedings of the twenty-sixth international joint conference on artificial intelligence, IJCAI-17, pp 3654–3660. https://doi.org/10.24963/ijcai.2017/511
125. Mao Y, Zhong G, Wang H, Huang K (2020) MCRN: a new content-based music classification and recommendation network. In: Yang H, Pasupa K, Leung AC-S, Kwok JT, Chan JH, King I (eds) Neural information processing. Springer, Cham, pp 771–779
126. Yadav N, Kumar Singh A, Pal S (2022) Improved self-attentive musical instrument digital interface content-based music recommendation system. Comput Intell 38(4):1232–1257. https://doi.org/10.1111/coin.12501
127. Nguyen VD, Nguyen QH, Freedman RG (2023) Predicting perceived music emotions with respect to instrument combinations. Proc AAAI Conf Artif Intell 37(13):16078–16086. https://doi.org/10.1609/aaai.v37i13.26910
128. Zhang K, Sun S (2013) Web music emotion recognition based on higher effective gene expression programming. Neurocomputing 105:100–106. https://doi.org/10.1016/j.neucom.2012.06.041
129. Zhang J, Huang X, Yang L, Nie L (2016) Bridge the semantic gap between pop music acoustic feature and emotion: build an interpretable model. Neurocomputing 208:333–341. https://doi.org/10.1016/j.neucom.2016.01.099
130. Cheuk KW, Luo Y-J, Balamurali BT, Roig G, Herremans D (2020) Regression-based music emotion prediction using triplet neural networks. In: 2020 international joint conference on neural networks (IJCNN), pp 1–7
131. Tran H, Le T, Do A, Vu T, Bogaerts S, Howard B (2023) Emotion-aware music recommendation. Proc AAAI Conf Artif Intell 37(13):16087–16095. https://doi.org/10.1609/aaai.v37i13.26911
132. Deng JJ, Leung CHC, Milani A, Chen L (2015) Emotional states associated with music: classification, prediction of changes, and consideration in recommendation. ACM Trans Interact Intell Syst 5(1):1. https://doi.org/10.1145/2723575
133. Shen T, Jia J, Li Y, Ma Y, Bu Y, Wang H, Chen B, Chua T-S, Hall W (2020) Peia: personality and emotion integrated attentive model for music recommendation on social media platforms. Proc AAAI Conf Artif Intell 34(01):206–213. https://doi.org/10.1609/aaai.v34i01.5352
134. Conceição Moreira PS, Tsunoda DF (2021) Recognition of emotions in music through the adaptive-network-based fuzzy (ANFIS). J New Music Res 50(4):342–354. https://doi.org/10.1080/09298215.2021.1977339
135. Pandeya YR, Lee J (2021) Deep learning-based late fusion of multimodal information for emotion classification of music video. Multimed Tools Appl 80(2):2887–2905
136. Han J, Zhang Z, Ren Z, Schuller B (2021) Exploring perception uncertainty for emotion recognition in dyadic conversation and music listening. Cogn Comput 13(2):231–240. https://doi.org/10.1007/s12559-019-09694-4
137. Tian H, Cai H, Wen J, Li S, Li Y (2019) A music recommendation system based on logistic regression and extreme gradient boosting. In: 2019 international joint conference on neural networks (IJCNN), pp 1–6. https://doi.org/10.1109/IJCNN.2019.8852094
138. Song G, Wang Z, Han F, Ding S, Iqbal MA (2018) Music auto-tagging using deep recurrent neural networks. Neurocomputing 292:104–110. https://doi.org/10.1016/j.neucom.2018.02.076
139. Vahidi C, Saitis C, Fazekas G (2021) A modulation front-end for music audio tagging. In 2021 international joint conference on neural networks (IJCNN), pp 1–7. https://doi.org/10.1109/IJCNN52387.2021.9533547
140. Horsburgh B, Craw S, Massie S (2015) Learning pseudo-tags to augment sparse tagging in hybrid music recommender systems. Artif Intell 219:25–39. https://doi.org/10.1016/j.artint.2014.11.004
141. Thayer RE (1989) The biopsychology of mood and arousal, Oxford University Press USA
142. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
143. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29(5):1189–1232
144. FRS, KP (1901) LIII. on lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine. J Sci (11): 559–572 https://doi.org/10.1080/14786440109462720
145. Wang Y, Yao H, Zhao S (2016) Auto-encoder based dimensionality reduction. Neurocomputing 184:232–242
146. Chu E, Roy DK (2017) Audio-visual sentiment analysis for learning emotional arcs in movies. In: 2017 IEEE international conference on data mining (ICDM), pp 829–834
147. Jang J-SR (1993) ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans Syst Man Cybern 23(3):665–685. https://doi.org/10.1109/21.256541
148. Raphael C (2006) Aligning music audio with symbolic scores using a hybrid graphical model. Mach Learn 65(2):389–409. https://doi.org/10.1007/s10994-006-8415-3
149. Li X, Tao D, Maybank SJ, Yuan Y (2008) Visual music and musical vision. Neurocomputing 71(10):2023–2028. https://doi.org/10.1016/j.neucom.2008.01.025
150. De Prisco R, Malandrino D, Zaccagnino G, Zaccagnino R (2017) Fuzzy vectorial-based similarity detection of music plagiarism. In: 2017 IEEE international conference on fuzzy systems (FUZZ-IEEE), pp 1–6. https://doi.org/10.1109/FUZZ-IEEE.2017.8015655
151. Das S, Kolya AK (2020) Detecting generic music features with single layer feedforward network using unsupervised Hebbian computation. Int J Distrib Artif Intell (IJDAI) 12(2):1–20
152. Caetano M, Zacharakis A, Barbancho I, Tardón LJ (2019) Leveraging diversity in computer-aided musical orchestration with an artificial immune system for multi-modal optimization. Swarm Evol Comput 50:100484. https://doi.org/10.1016/j.swevo.2018.12.010

153. Rahman JS, Gedeon T, Caldwell S, Jones R, Jin Z (2021) Towards effective music therapy for mental health care using machine learning tools: human affective reasoning and music genres. J Artif Intell Soft Comput Res 11(1):5–20
154. Furner M, Islam MZ, Li C-T (2021) Knowledge discovery and visualisation framework using machine learning for music information retrieval from broadcast radio data. Expert Syst Appl 182:115236
155. Yang T, Nazir S (2022) A comprehensive overview of AI-enabled music classification and its influence in games. Soft Comput 26(16):7679–7693
156. Scirea M, Togelius J, Eklund P, Risi S (2017) Affective evolutionary music composition with metacompose. Genet Program Evolvable Mach 18:433–465
157. Scirea M, Eklund P, Togelius J, Risi S (2018) Towards an experiment on perception of affective music generation using metacompose. In: Proceedings of the genetic and evolutionary computation conference companion, pp 131–132
158. Lv HZ (2023) Innovative music education: using an AI-based flipped classroom. Educ Inf Technol 28(11):15301–15316
159. Liu C, Feng L, Liu G, Wang H, Liu S (2021) Bottom-up broadcast neural network for music genre classification. Multimed Tools Appl 80(5):7313–7331. https://doi.org/10.1007/s11042-020-09643-6
160. Medhat F, Chesmore D, Robinson JA (2017) Masked conditional neural networks for audio classification. In: ICANN
161. Parkhi O, Vedaldi A, Zisserman A (2015) Deep face recognition. In: Proceedings of the British machine vision conference 2015. British Machine Vision Association
162. Wang J, Wang K-C, Law MT, Rudzicz F, Brudno M (2019) Centroid-based deep metric learning for speaker recognition. In: 2019 IEEE international conference on acoustics, speech and signal processing, pp 3652–3656. https://doi.org/10.1109/ICASSP.2019.8683393
163. Wan L, Wang Q, Papir A, Moreno IL (2017) Generalized end-to-end loss for speaker verification. https://arxiv.org/abs/1710.10467
164. Liu W, Wen Y, Yu Z, Li M, Raj B, Song L (2017) Sphereface: deep hypersphere embedding for face recognition. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 6738–6746. https://doi.org/10.1109/CVPR.2017.713
165. Wang H, Wang Y, Zhou Z, Ji X, Li Z, Gong D, Zhou J, Liu W (2018) Cosface: large margin cosine loss for deep face recognition. 2018 IEEE/CVF conference on computer vision and pattern recognition, pp 5265–5274
166. van den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K (2016) WaveNet: a Generative Model for Raw Audio. In: Proceeding of 9th ISCA workshop on speech synthesis workshop (SSW 9), p 125
167. Kalchbrenner N, Danihelka I, Graves A (2015) Grid long short-term memory
168. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR), pp 1–9. https://doi.org/10.1109/CVPR.2015.7298594
169. Andén J, Mallat S (2011) Multiscale scattering for audio classification. In: International society for music information retrieval conference
170. Graves A, Mohamed A-R, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing, pp 6645–6649. https://doi.org/10.1109/ICASSP.2013.6638947
171. Gregor K, Danihelka I, Graves A, Rezende D, Wierstra D (2015) Draw: a recurrent neural network for image generation. In: Bach F, Blei D (eds) Proceedings of the 32nd international conference on machine learning. Proceedings of machine learning research, vol 37, pp 1462–1471. PMLR, Lille, France
172. Eck D, Schmidhuber J (2002) Finding temporal structure in music: blues improvisation with lstm recurrent networks. In: Proceedings of the 12th IEEE workshop on neural networks for signal processing, pp 747–756. https://doi.org/10.1109/NNSP.2002.1030094
173. Bowman SR, Vilnis L, Vinyals O, Dai A, Jozefowicz R, Bengio S (2016) Generating sentences from a continuous space. In: Proceedings of the 20th SIGNLL conference on computational natural language learning, pp 10–21. Association for Computational Linguistics, Berlin, Germany. https://doi.org/10.18653/v1/K16-1002 . https://aclanthology.org/K16-1002
174. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Lu, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, vol 30
175. Huang C-ZA, Vaswani A, Uszkoreit J, Simon I, Hawthorne C, Shazeer NM, Dai AM, Hoffman MD, Dinculescu M, Eck D (2019) Music transformer: generating music with long-term structure. In: International conference on learning representations
176. Tolstikhin I, Bousquet O, Gelly S, Schoelkopf B (2017) Wasserstein auto-encoders. arXiv preprint arXiv:1711.01558
177. Radford A, Metz L, Chintala S (2016) Unsupervised representation learning with deep convolutional generative adversarial networks. CoRR **abs/1511.06434**
178. Liu J-Y, Yang Y-H, Jeng S-K (2019) Weakly-supervised visual instrument-playing action detection in videos. IEEE Trans Multimed 21(4):887–901. https://doi.org/10.1109/TMM.2018.2871418
179. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF (eds) Medical image computing and computer-assisted intervention - MICCAI 2015. Springer, Cham, pp 234–241
180. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville AC (2017) Improved training of Wasserstein gans. In: Advances in neural information processing systems, vol 30
181. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 770–778. https://doi.org/10.1109/CVPR.2016.90
182. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in neural information processing systems, vol 27
183. Santoro A, Faulkner R, Raposo D, Rae JW, Chrzanowski M, Weber T, Wierstra D, Vinyals O, Pascanu R, Lillicrap TP (2018) Relational recurrent neural networks. In: Neural information processing systems
184. Jolicoeur-Martineau A (2019) The relativistic discriminator: a key element missing from standard gan. ArXiv **abs/1807.00734**
185. Schroff F, Kalenichenko D, Philbin J (2015) Facenet: a unified embedding for face recognition and clustering. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR), pp 815–823. https://doi.org/10.1109/CVPR.2015.7298682
186. Tien Bui D, Pradhan B, Lofman O, Revhaug I, Dick OB (2012) Landslide susceptibility mapping at hoa binh province (Vietnam) using an adaptive neuro-fuzzy inference system and gis. Comput Geosci 45:199–211. https://doi.org/10.1016/j.cageo.2011.10.031
187. Bromley J, Bentz J, Bottou L, Guyon I, Lecun Y, Moore C, Sackinger E, Shah R (1993) Signature verification using a

"siamese'' time delay neural network. Int J Pattern Recognit Artif Intell 7:25. https://doi.org/10.1142/S0218001493000339
188. Vikhar PA (2016) Evolutionary algorithms: a critical review and its future prospects. In: 2016 international conference on global trends in signal processing, information computing and communication (ICGTSPICC), pp 261–265. https://doi.org/10.1109/ICGTSPICC.2016.7955308
189. Narmour E (1990) The analysis and cognition of basic melodic structures: the implication realization model, University of Chicago Press
190. Yoon Y, Kim Y-H, Moraglio A, Moon B-R (2012) Quotient geometric crossovers and redundant encodings. Theor Comput Sci 425:4–16. https://doi.org/10.1016/j.tcs.2011.08.015
191. Glover F (1989) Tabu search-part I. ORSA J Comput 1(3):190–206
192. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680
193. Mladenović N, Hansen P (1997) Variable neighborhood search. Comput Oper Res 24(11):1097–1100. https://doi.org/10.1016/S0305-0548(97)00031-2
194. Moriarty DE, Miikkulainen R (1996) Efficient reinforcement learning through symbiotic evolution. Mach Learn 22(1):11–32. https://doi.org/10.1007/BF00114722
195. Castro LN, Timmis J (2002) An artificial immune network for multimodal function optimization. Proceedings of the 2002 congress on evolutionary computation. CEC'02 (Cat. No.02TH8600) 1: 699–7041
196. Bernardes G, Cocharro D, Caetano M, Guedes C, Davies M (2016) A multi-level tonal interval space for modelling pitch relatedness and musical consonance. J New Music Res 45(4):281–294. https://doi.org/10.1080/09298215.2016.1182192
197. Altman NS (1992) An introduction to kernel and nearest-neighbor nonparametric regression. Am Stat 46(3):175–185
198. Díaz-Báñez JM, Rizo J-C (2014) An efficient dtw-based approach for melodic similarity in flamenco singing. In: Similarity search and applications
199. Sutton RS, Barto AG (2018) Reinforcement learning: an introduction, MIT press
200. BELLMAN R (1957) A Markovian decision process. J Math Mech 6(5):679–684
201. Baum LE, Petrie T (1966) Statistical inference for probabilistic functions of finite state Markov chains. Ann Math Stat 37(6):1554–1563
202. Loui P, Grent-'t-Jong T, Torpey D, Woldorff M (2005) Effects of attention on the neural processing of harmonic syntax in western music. Cogn Brain Res 25(3):678–687
203. Schulman J, Moritz P, Levine S, Jordan M, Abbeel P (2016) High-dimensional continuous control using generalized advantage estimation. In: Proceedings of the international conference on learning representations (ICLR)
204. Ziebart BD, Maas A, Bagnell JA, Dey AK (2008) Maximum entropy inverse reinforcement learning. In: Proceedings of the 23rd national conference on artificial intelligence, vol 3, pp 1433–1438
205. Berndt DJ, Clifford J (1994) Using dynamic time warping to find patterns in time series. In: KDD Workshop
206. Hu Y, Koren Y, Volinsky C (2008) Collaborative filtering for implicit feedback datasets. In: 2008 eighth IEEE international conference on data mining, pp 263–272. https://doi.org/10.1109/ICDM.2008.22
207. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, vol 26
208. Homburg H, Mierswa I, Möller B, Morik K, Wurst M (2005) A benchmark dataset for audio classification and clustering. In: International society for music information retrieval conference
209. Pereira RM, Silla CN (2017) Using simplified chords sequences to classify songs genres. In: 2017 IEEE international conference on multimedia and expo (ICME), pp 1446–1451. https://doi.org/10.1109/ICME.2017.8019531
210. Ellis DPW (2007) Classifying music audio with timbral and chroma features. In: International society for music information retrieval conference
211. Goebl W The Vienna 4x22 Piano Corpus. https://doi.org/10.21939/4X22
212. Likert R (1932) A technique for the measurement of attitudes, vol 136–165. Arch Psychol. https://books.google.pl/books?id=9rotAAAAYAAJ