Expert Systems With Applications 209 (2022) 118190

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Review

A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends

Miguel Civit [a], Javier Civit-Masot [b,*], Francisco Cuadrado [a], Maria J. Escalona [c]

[a] Department of Communication, Education and Humanities, Universidad Loyola Andalucía, Seville, Spain
[b] Department Architecture and Computer Technology (ATC), E.T.S. Ingeniería Informática, Universidad de Sevilla, Avda. Reina Mercedes s/n, Seville, 41012, Spain
[c] Department of Computer Languages and Systems, E.T.S. Ingeniería Informática, Universidad de Sevilla, Avda. Reina Mercedes s/n, Seville, 41012, Spain

ARTICLE INFO

Keywords:
Automatic music generation
Assisted music composition
Artificial intelligence
Scoping review
Human–machine co-creation

ABSTRACT

Currently available reviews in the area of artificial intelligence-based music generation do not provide a wide range of publications and are usually centered around comparing very specific topics between a very limited range of solutions. Best surveys available in the field are bibliography sections of some papers and books which lack a systematic approach and limit their scope to only handpicked examples In this work, we analyze the scope and trends of the research on artificial intelligence-based music generation by performing a systematic review of the available publications in the field using the Prisma methodology. Furthermore, we discuss the possible implementations and accessibility of a set of currently available AI solutions, as aids to musical composition. Our research shows how publications are being distributed globally according to many characteristics, which provides a clear picture of the situation of this technology.

Through our research it becomes clear that the interest of both musicians and computer scientists in AI-based automatic music generation has increased significantly in the last few years with an increasing participation of mayor companies in the field whose works we analyze. We discuss several generation architectures, both from a technical and a musical point of view and we highlight various areas were further research is needed.

1. Introduction

On October 9th, 2021, Beethoven's previously unfinished Tenth Symphony was premiered in Bonn to celebrate the 250th anniversary of the composer's birth. The media announced that the work had been completed by AI.[1] This statement hardly reflects the reality of the compositional process: the final work was based on some sketches by the original composer, the first two movements having been pieced together from those fragments by British musicologist and composer Barry Cooper in 1988. The last two movements were composed with some help from AI tools, but still required a lot of work by human composers. The use of computer-based technologies to create or help create music is not new. As early as 1848, Ada Lovelace, in her notes on Babbage's Analytical Engine wrote, ''Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent'' (Menabrea & Lovelace, 1842). In recent years, the use of artificial intelligence in general and, more specifically, the use of Deep Learning-based technologies for music composition have become quite common in scientific publications. However, only events like the recreation of Beethoven's symphony or the use of AI to help generate the ''Tokyo 2020 beat'', the official anthem of the Tokyo Olympic games, have received attention in the media.[2]

There have been some very good analyses on the topic of automatic music generation (e.g., Briot, 2021; Briot, Hadjeres, & Pachet, 2020). These works provide a well-presented description of the topic supported by significant research papers, chosen specifically by the authors. In the current paper, we follow a completely different approach by performing first a systematic search of the available literature in the last five years and then analyzing the scope, trends, and future directions of this research. It is clear, that each approach has certain clear advantages. The analysis by topics and problems with author based selection of works, as done in Briot et al. (2020), is clearly very good for a textbook

* Correspondence to: E.T.S. Ingeniería Informática, Department of Architecture and Computer Technology (ATC), Avda. Reina Mercedes s/n, Office B1.46, 41012, Seville, Spain

E-mail addresses: mcivit@uloyola.es (M. Civit), mjavier@us.es (J. Civit-Masot), fjcuadrado@uloyola.es (F. Cuadrado), mjescalona@us.es (M.J. Escalona).

[1] https://www.udiscovermusic.com/classical-news/beethovens-10th-symphony-ai/

[2] https://youtu.be/smMVQ6C4Wqg

https://doi.org/10.1016/j.eswa.2022.118190
Received 8 January 2022; Received in revised form 16 June 2022; Accepted 15 July 2022
Available online 20 July 2022

as it provides very good representative examples. Our systematic search approach provides a wider coverage and tries to find most of the available alternatives. Thus, from a researcher point of view, both approaches are useful and complementary.

We specifically aim to answer the following research questions:

- RQ1: Is research in this field clearly increasing? Where is this research mainly being carried out? (geographical areas, private sector, universities, etc.)
- RQ2: Which are the AI-based techniques most used for music generation? Which problems do they aim to solve and are they style specific? Which are the most widely used datasets?
- RQ3: What are the possible uses from the point of view of a musician and are these solutions available to final users? Can they fit into the workflows currently used throughout the music industry?
- RQ4: Are user interface issues taken into account? Do the system designs analyzed consider emotion-related issues?

Thus, following a methodology that will be detailed in the next section, we analyzed the number of publications per year in the field, considering the affiliations of the authors, the type of institutions, where the research is published, the architecture of the proposed solutions, the availability of code and the existence of demonstrations, the systems' integration with DAWs (digital audio workstations) and availability as web-services, etc. We also specifically analyzed systems with available musical demonstrations to correlate them to the quality of their results as perceived by a professional musician. It is important to mention that two of this paper's authors are professional musicians and composers with experience in composing for media, live performance, and sound design, while the others are computer scientists with experience in intelligent systems. As a final point, and considering the fundamental importance of emotions in music (Williams & Lee, 2018), specially in the case of game and film related applications, we have analyzed if "emotion in music", as a topic, is specifically considered in the solution design.

2. Methodology

The methodology used corresponds to the PRISMA scoping review process (Peters et al., 2017, 2020) using google sheet macros for duplicate removal, filtering and data representation.

2.1. Database selection

To obtain a significant amount of good-quality works for analysis, our idea was originally to restrict our search to scientific papers published in international journals. Thus, as a first step, we performed an initial search in the Clarivate Web of Science (WOS) library. Although this approach produced an acceptable number of works, comparing these results to other studies on the topic (Briot, 2021; Briot et al., 2020; Liu & Ting, 2016) it was clear that some widely discussed systems, such as Deepbach (Hadjeres, Pachet, & Nielsen, 2017), Google's Magenta musicVAE (Roberts, Engel, Raffel, Hawthorne, & Eck, 2018) and Wavenet (Dieleman, Oord, & Simonyan, 2018) amongst many others were missing. Extending the search by allowing conference publications solved the problem for these systems but not for others like MuseGan (Dong, Hsiao, Yang, & Yang, 2018) or MidiNet (Yang, Chou, & Yang, 2017). Hence, the solution was to include IEEE Iexplore and the ACM digital library also as search sources. Even though this extended the search's scope to cover a wider range of conferences, it still did not provide an adequate solution. Eventually, Google Scholar was included, as an alternative additional source, as it integrates works from several platforms, and allowed us to include very highly cited references not available from other sources.

2.2. String development

Our search string consisted of the three main key terms Music, Generation and Artificial Intelligence. These terms, which where based on our research objectives, were complemented with alternatives and, in the case of artificial intelligence also with specific alternatives that have been widely used in paper titles. For the Clarivate Web of Science (WOS) library. The specific query used was:

TS=(Music AND (Generat* OR Composition) AND ((Artificial Intelligence) OR "AI" OR (neural net*) OR "CNN" OR "RNN" OR "Machine Learning" OR "LSTM"))

The search for IEEE and ACM was the same with minor syntactical variations to adapt to the specific characteristics of the database. For Google scholar we had to simplify it due to the way in which Boolean operators work on Google search engines.[3] Due to the large number of obtained results in this last source the information obtained was sorted by relevance and pruned to the top 400 works.

2.3. Inclusion/exclusion criteria

We only included works that met the following criteria:

- The work must have been published between 2017 and 2021.
- The work has to deal with "music generation" or "music composition" and "artificial intelligence" or any equivalent formulation of this area.
- The work must be a full text. The entire contents should be available through the data source.
- The article must be peer-reviewed. It can be published in a journal or peer-reviewed proceedings of a conference. To be able to include some of the most widely discussed systems e.g. Flow Machines (Pachet, Roy, & Carré, 2021) we accepted arxiv works for which both code and demos are publicly available and whose authors appear in other peer reviewed works in the study.
- The work should be written in English.

2.4. Article screening results

Applying the search criteria in the four data sources (Fig. 1), the results obtained were as follows:

- IEEE: 404 works.
- ACM: 223 works.
- WOS: 485 works.
- Scholar: 5070 works.

Most works would appear in several searches and we therefore had to remove duplicates and classify them considering their main source. Works from IEEE conferences and publications (appearing in IEE Explore) were classified as IEEE, those from AC conferences and publications (appearing in the ACM digital library) as ACM. Those appearing in WOS but not in the previous sources were classified as WOS, and those appearing only in Google Scholar as GS. The results of these searches included works outside the scope of this paper, so we had to perform a review process as described in Fig. 1. In this process, the 3 authors first reviewed the papers based on their titles. If any doubt remained about a paper's suitability for analysis, they checked its abstract and, as a last resource, the full body of the article. This process was repeated by another author and, in case of disagreement, by a third. After this process, the following works remained:

- IEEE: 66 works.
- ACM: 13 works.
- WOS: 31 works.
- Scholar: 29 works.

[3] https://t.ly/TrJN

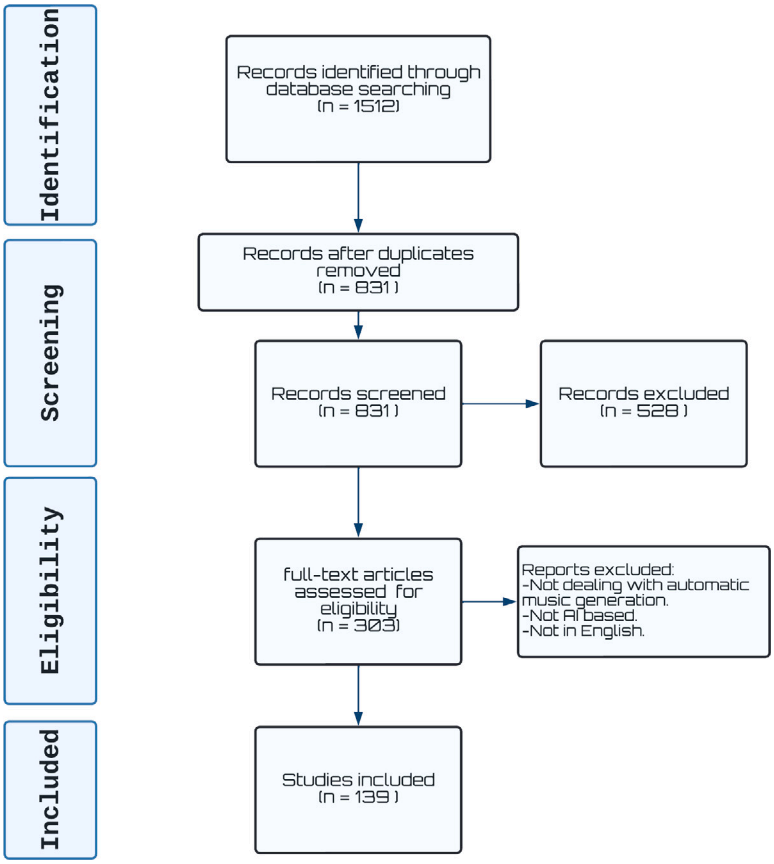
Identification
Screening
Eligibility
Included
Records identified through database searching (n = 1512)
Records after duplicates removed (n = 831)
Records screened (n = 831)
Records excluded (n = 528)
full-text articles assessed for eligibility (n = 303)
Reports excluded:
-Not dealing with automatic music generation.
-Not AI based.
-Not in English.
Studies included (n = 139)

Fig. 1. PRISMA scoping methodology.

Table 1
Publications summary.

Source	Publ. first Search	Publ. review	Journals	Conferences	Other
IEEE	404	66	13	53	0
ACM	223	13	2	11	0
WOS	485	31	22	9	0
GS	5080	29	2	18	8

These 139 works would be analyzed to find the main research topics and trends in automatic music generation. To summarize (see Table 1), we used WOS as our single source for the first search and obtained 485 works. After including IEEE and ACM, we had 1,112 candidate works, rising to 6,182 works when Google Scholar was considered. After pruning the Scholar search to the most relevant 300 works, removing duplicates, and performing a review to ensure that the papers were related to artificial intelligence or machine learning and music generation or composition, we finally had 139 works to analyze in detail.

In a first step, two authors screen the records considering only the title and the keywords. If at least one of the authors considers that the title and keyword fulfill the inclusion criteria the paper is kept for a full text eligibility check.

2.5. Extracted information

In this epigraph, the information initially extracted from the reviewed manuscripts is detailed. This information is essential to get the answers to the initial questions.

- Title: Title of the manuscript or work analyzed.
- DOI (Digital Object Identifier): Unique identifier to be able to retrieve the publication.
- Publication year: Year of publication of the work reviewed.
- Keywords: Words used during the search process.
- Search source: The source was searched in order to find the work.
- Magazine/book/conference: Where the manuscript was published.
- Localization: Countries where the research was carried out.
- Institution Type: Type of institution (academic or corporate) where the research was carried out.
- Paper type: Type of paper between generator, survey or other.
- Abstract: Paper summary.
- Keywords: Keywords included in the article.

For those papers that describe a generation system, we also include:

- System name: Only if the paper uses a specific name for the system.
- Dataset: Dataset used for training.
- Music Representation: The type of music representation used in the paper (symbolic or audio).
- Type of generation: Ex Nihilo, inpainting, harmonization, etc.
- Musical Type: Monophonic, polyphonic, multitrack, accompaniment, etc.
- Architecture: Architecture of the proposed generator (e.g., LSTM, VAE, GAN, Transformer ...)
- Code availability: if the code is available and where.
- Demo availability: if there is a demo of the generated output available.
- DAW integration: If the system is integrated into an available Digital Audio Workstation.
- Web availability: If the system can be used directly on the web.
- Commercial: If the system is commercially available.

2.6. Study limitations

There are three main limitations to our study:

- First, our review is limited to finding articles from three databases, mainly IEEE, ACM, and WOS, and complementing these results with a limited search in Google Scholar. While these three are the top databases in the fields of Artificial Intelligence and cover the music generation aspects well, the possibility of missed articles in this field clearly exists. In addition, our review is limited to articles written in English. Some articles in other languages, for example Spanish, French or Korean, were found in our search and excluded by inclusion criteria.
- There is also a limitation related to nonpublished work. Although we extended our initial criteria to include arxiv works with available code, demos, and peer-reviewed citations it is clear that important information related to commercial products may not be available through our search methodology.
- There is an impossible number of possible future applications of AI in music composition. We discuss only some of the possible uses that are starting to be implemented. Uses, such as composition for visual media, are still very much in their infancy and therefore beyond the scope of our review of possible applications.

3. Results

3.1. Publication distribution data

In Fig. 2 we present the distribution of the publications analyzed. In Fig. 2a we consider the publication type. It can be seen that approximately 64% (86 papers) were conference contributions, 30% (39 works) were journal articles, and the remaining 8 works were other types of publications. These include books and some highly cited arxiv preprints.

Fig. 2b shows the publication distribution according to the source database. We can see slightly less that half of the publications come from IEEE, while WOS and GS account for slightly over 20% each and the remaining 10% correspond to ACM,

Fig. 2c shows the distribution according to affiliation. We can see that around 16% of the publications come from commercial corporations, while the remaining come from academic institutions.

Fig. 2d shows the geographical distribution of the publications, we can see that about 40% of the publications come from Asia (40% of them from China), about 30% from Europe, around 25% from America (almost all from the US and Canada) and the remaining 5% from the rest of the world. Table 2 classifies the analyzed works by publication type (conference-C or journal-J), institution type (commercial-C or Academic-A) and geographical area.

Regarding the nature of the studied works, 118 papers present generators, 10 are some type of survey studies, and 12 are creation environments, evaluation metrics, specific datasets, evaluation of generated pieces, etc. This information is shown in Fig. 2e. It is worth mentioning that (Briot et al., 2020) has been counted as a survey study but also as a generator as it presents a specifically developed MiniBach as a basic generator example.

Fig. 2f shows the evolution of the publications over time. We can see a clear increase in scientific interest in this field. It should be clarified that the data for 2021 corresponds only to 9 months of the year; thus the expectation is that the final publication number for this year should be at least equal to that of 2020.

3.1.1. Citation data

Fig. 3A shows the total number of citations in Google Scholar as a function of the year of publication of the article. It is clear that the number of citations for a specific paper increases over time, hence the pattern shown in the figure. What is much more interesting is that the 2017 articles have a mean of almost 80 citations, while the 2018 articles have more than 55 and the 2019 ones more than 20.

Fig. 3B shows the distribution of citations among the articles analyzed. 44.3% of the papers are cited more than 15 times, while 30.9% are cited between 5 and 15 times. The rest are cited less than five times. This situation changes significantly (Fig. 3C) when we consider papers from commercial corporations. In this case, 66.7% of the papers are cited more than 15 times. Note that the five most cited articles (Dong et al., 2018; Hadjeres et al., 2017; Huang et al., 2018; Roberts et al., 2018; Yang et al., 2017) have between 323 and 247 citations. The most cited paper comes from Sony CSL Paris, while two other heavily cited papers come from research groups at Google. Two more are from Academia Sinica in Taiwan.

3.2. Datasets

Most automatic music generation systems have to be trained using a pre-existing music dataset. It should be clear that the selection of the dataset depends on several factors. The first is the type of music representation used by the system. Most of the systems analyzed in our study were symbolic, that is, expressed through scores or lead sheets or their digital equivalents, such as MIDI or piano roll. All of the systems described in the most cited papers mentioned above are symbolic generators. The other alternative is to represent the music as audio. This technique is much less popular in automatic music generation. If the articles analyzed are ordered by number of citations, the first audio-based system is Wavenet (Dieleman et al., 2018), which uses the same technology as the Google Assistant voice synthesizer and is 10th in the ranking with 105 citations. Interestingly, some articles, such as (Manzelli et al., 2018a, 2018b), used both audio and symbolic representations.

Another important aspect of the dataset selection is that the style of music included in the dataset clearly influences the style of music that will be generated. As an example, (Hadjeres et al., 2017) uses Bach chorales as training data, and clearly this helps the system compose in a style similar to these works. As we will discuss in Section 3.3 in some cases, including (Hadjeres et al., 2017), constraints can be imposed on the generated score so that the style is not conditioned exclusively by the training dataset. As a further example (De Felice et al., 2017), which uses an evolutionary generation algorithm and, therefore, uses constraints to optimize the evolution of the melody, uses a small dataset of Bach chorales for evaluation purposes.

In the systems analyzed, the datasets are widely varied. There is a wide selection of systems that for some reason-usually to produce music in a very specific style-developed their own dataset. As an example, (Tanberk & Tükel, 2021) uses a Turkish pop music dataset, while (Huang & Yang, 2020) uses a mix of Japanese anime, Korean pop and western songs. The most widely used dataset is Lakh[4] which is a collection of 176,581 deduped MIDI files. The second most used dataset is the Nottingham dataset; this is a collection of 1200 American and British folk songs that are initially in ABC format but are also available as MIDI files.[5] The piano-midi dataset, which includes 11,086 piano pieces.[6]

One dataset of particular interest, specifically designed for automatic music generation, is Maestro (MIDI and Audio Edited for Synchronous TRacks and Organization) (Hawthorne et al., 2018), which includes synchronized audio and symbolic information.

[4] https://colinraffel.com/projects/lmd/

[5] https://github.com/jukedeck/nottingham-dataset

[6] https://paperswithcode.com/dataset/adl-piano-midi

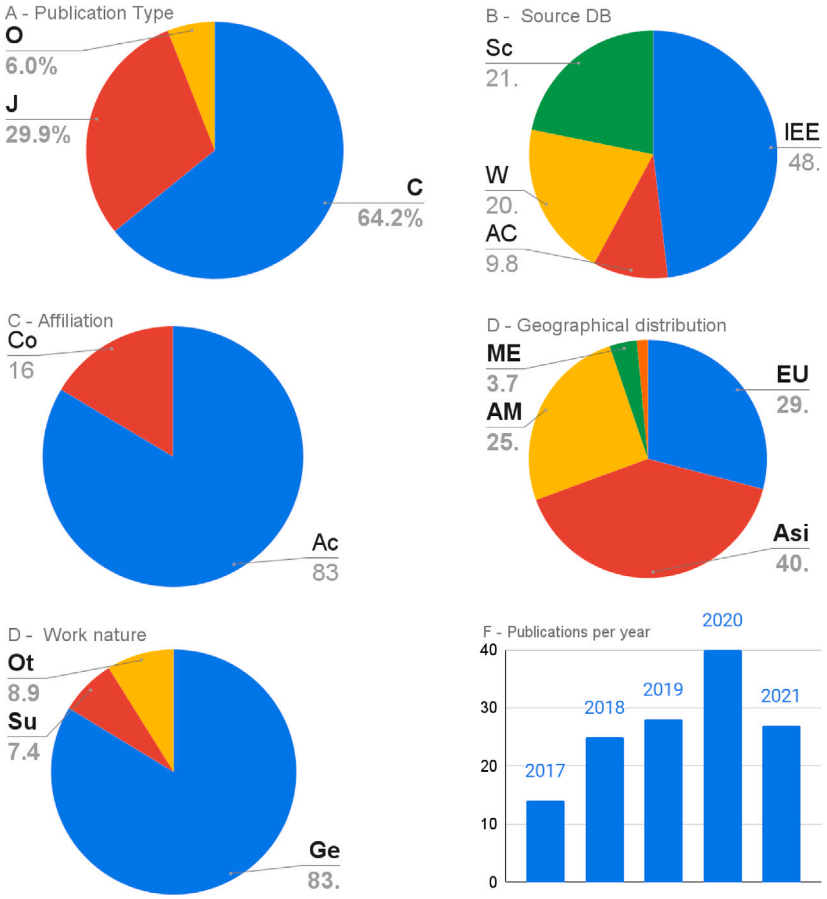

Fig. 2. Publication distribution.

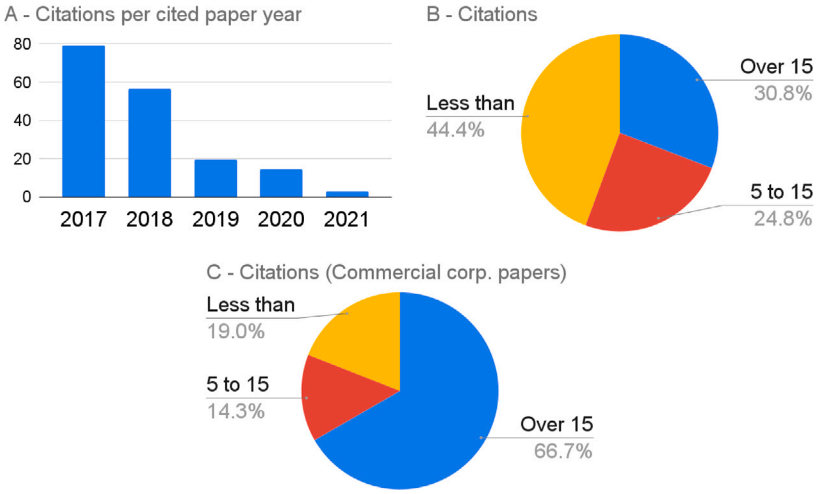

Fig. 3. Citation distribution.

3.3. Music related characteristics

This subsection analyzes a set of features specifically related to the music generated by an automatic composition system.

Of the 118 analyzed generators, 113 work with symbolic music, 3 generate audio directly and 2 use mixed representation. It is worth mentioning that the 2 mixed representation papers are related to the same systems and that the 3 audio-based generators differ from each other very significantly. Gacela (Marafioti et al., 2020) is a short imprinting application that aims to restore missing audio fragments lasting up to a few seconds. Wavenet (Dieleman et al., 2018)is designed to generate music with humanized interpretation while Jukebox (Dhariwal et al., 2020) generates music, including rudimentary singing, in a variety of artistic styles. Table 3 specifies the generators that do not produce symbolic output according to the type of output produced.

Virtually all generators produce music in a specific style. In the vast majority of cases, this is done through the selection of a specific dataset for training. Some systems, however, impose a specific style either by making the generator follow a particular set of musical theory rules or by establishing specific constrains on the generator output. This

Table 2
Papers by publication type and geographical area.

	CC	CA	JC	JA	OC	OA
AM	Roberts et al. (2018) Huang et al. (2018) Jaques, Gu, Turner, and Eck (2017) Hawthorne et al. (2018) Muhamed et al. (2021)	Johnson, Keller, and Weintraut (2017) Chu, Urtasun, and Fidler (2016) Lopez-Rincon, Starostenko, and Ayala-San Martín (2018) Manzelli, Thakkar, Siahkamari, and Kulis (2018b) Manzelli, Thakkar, Siahkamari, and Kulis (2018a) Mao, Shin, and Cottrell (2018) Koh, Dubnov, and Wright (2018) Brown and Casey (2019) Donahue, Mao, Li, Cottrell, and McAuley (2019) Ferreira and Whitehead (2021) Stoltz and Aravind (2019) Chen, Zhang, Dubnov, Xia, and Li (2019) Chen, Xia, and Dubnov (2020) Delarosa and Soros (2020) Eisenbeiser (2020) Louie, Coenen, Huang, Terry, and Cai (2020) Huang, Hsieh, Qin, Liu, and Eirinaki (2020) Suh, Youngblom, Terry, and Cai (2021) Galajda, Royal, and Hua (2021) Azevedo, Silla Jr, and Costa-Abreu (2021) Lopes, Martins, Cardoso, and dos Santos (2017)	Huang, Cooijmans, Roberts, Courville, and Eck (2019) Oore, Simon, Dieleman, Eck, and Simonyan (2020)	Salas (2018) Hutchings and McCormack (2019) Plut and Pasquier (2020) Yang and Lerch (2020) Cunha, Subramanian, and Herremans (2018)	Payne (2019) Dhariwal et al. (2020)	Ens and Pasquier (2020)
EU	Hadjeres et al. (2017) Dieleman et al. (2018) Liang, Gotham, Johnson, and Shotton (2017) Lattner and Grachten (2019) Bazin and Hadjeres (2019)	Makris, Kaliakatsos-Papakostas, Karydis, and Kermanidis (2017) Colombo, Seeholzer, and Gerstner (2017) Brunner, Wang, Wattenhofer, and Wiesendanger (2017) Kaliakatsos-Papakostas, Gkiokas, and Katsouros (2018) Brunner, Konrad, Wang, and Wattenhofer (2018) Simões, Machado, and Rodrigues (2019) Ebrahimi, Majidi, and Eshghi (2019) Garoufis, Zlatintsi, and Maragos (2020) Frid, Gomes, and Jin (2020) Dervakos, Filandrianos, and Stamou (2021) Walter et al. (2021)	Hadjeres and Nielsen (2020) Grachten, Lattner, and Deruty (2020) Briot and Pachet (2020)	Williams et al. (2017) Lattner, Grachten, and Widmer (2018) Avdeeff (2019) Makris, Kaliakatsos-Papakostas, Karydis, and Kermanidis (2019) Herremans and Chew (2017) Goienetxea, Mendialdua, Rodríguez, and Sierra (2019) Harrison and Pearce (2020) Gioti (2020) Ycart and Benetos (2020) Tikhonov, Yamshchikov, et al. (2017) Briot (2021) Moura and Maw (2021) Anantrasirichai and Bull (2021) Marafioti, Majdak, Holighaus, and Perraudin (2020) Grekow and Dimitrova-Grekow (2021) De Felice et al. (2017) De Prisco, Zaccagnino, and Zaccagnino (2020)	Briot et al. (2020) Briot, Hadjeres, and Pachet (2017) Pachet et al. (2021) Hadjeres and Crestel (2021)	Cambouropoulos et al. (2021)

(*continued on next page*)

information is shown in Fig. 4A. Note that those systems that are based on genetic or evolutionary algorithms are in general completely theory-based and do not use specific style datasets for training in that style. As an example, (Wen & Ting, 2020) produces music in a bossa nova style defined explicitly by rules that are used to evaluate the candidate musical data (further argued in Section 4).

According to the number of melody lines generated, we can see that most of the generators are polyphonic. We consider a system as polyphonic when it is capable of generating multiple voices (multiple melodies that are correlated). These generators either do not specify the instrument that is going to play the melodies or generate their output assuming that the final instrument will be a polyphonic harmonic instrument like an organ, as in (Harrison & Pearce, 2020; Liang et al., 2017) or a piano, as in (Huang et al., 2018; Madhok et al., 2018; Mao et al., 2018). When the system produces outputs for several instruments we consider it as multitrack, as it is generating multiple MIDI tracks or

Table 2 (*continued*).

	CC	CA	JC	JA	OC	OA
RW	Chen, Xiao, and Yin (2019)	Yang et al. (2017), Lim, Rhyu, and Lee (2017) Evans, Munekata, and Ono (2017) Joshi, Nyayapati, Singh, and Karmarkar (2018) Mo, Wang, Li, and Qian (2018) Wiriyachaiporn, Chanasit, Suchato, Punyabukkana, and Chuangsuwanich (2018) Singh and Ratnawat (2018) Shukla and Banka (2018) Sun et al. (2018) Liu and Yang (2018) Dong et al. (2018) Agarwal, Saxena, Singal, and Aggarwal (2018) Masuda and Iba (2018) Madhok, Goel, and Garg (2018) Guan, Yu, and Yang (2019) Zhao, Li, Cai, Wang, and Wang (2019) Yang, Sun, Zhang, and Zhang (2019) Jia, Lv, Pu, and Yang (2019) Hung, Wang, Yang, and Wang (2019) Wang, Wang, and Cai (2019) Nadeem, Tagle, and Sitsabesan (2019) Qiu et al. (2019) Jiang, Xiao, and Yin (2019) Cheng, Lai, Chang, Chiou, and Yang (2020) Wang, Liu, Jin, Li, and Ma (2020) Wen and Ting (2020) Huang and Huang (2020) Kan and Sourin (2020) Shopynskyi, Golian, and Afanasieva (2020) Kurniawati, Suprapto, and Yuniarno (2020) Huang and Yang (2020) Lang, Wu, Zhu, and Li (2020) Lim, Chan, and Loo (2020b) Diéguez and Soo (2020) Hakimi, Bhonker, and El-Yaniv (2020) Zeng and Zhou (2021) Marsden and Ajoodha (2021) Suthaphan, Boonrod, Kumyaito, and Tamee (2021) Chen, Wei, Chao, and Li (2021) Tanberk and Tükel (2021) Sabitha et al. (2021) Makris, Agres, and Herremans (2021), Ma, Liu, Qiao, Cao, and Yin (2020)		Liu and Ting (2016) Ting, Wu, and Liu (2017) Li, Jang, and Sung (2019) Cai and Cai (2019) Wu, Hu, Wang, Hu, and Zhu (2019) Jin, Tie, Bai, Lv, and Liu (2020) Mor, Garhwal, and Kumar (2020) Wu, Liu, Hu, and Zhu (2020) Shi and Wang (2020) Dean and Forth (2020) Yeh et al. (2021) Choi, Park, Heo, Jeon, and Park (2021) Lim, Chan, and Loo (2020a) Yu, Srivastava, and Canales (2021) Li and Sung (2021) Jeong, Kim, and Ahn (2017)		

Table 3
Generator output format.

Raw Audio	Mixed
Dieleman et al. (2018)	Manzelli et al. (2018b)
Dhariwal et al. (2020)	Manzelli et al. (2018a)
Marafioti et al. (2020)	

audio files, one for each specific instrument. Tables 6 and 5 classify the works according to the produced output.

Most generators, whether seeded or unseeded, aimed at generating full pieces of music or new ideas for the composer to choose from, while inpainting generators are usually more focused on completing pieces or repairing damaged audio. This classification is further characterized in Section 4, where its implications for real-life applications are further discussed.

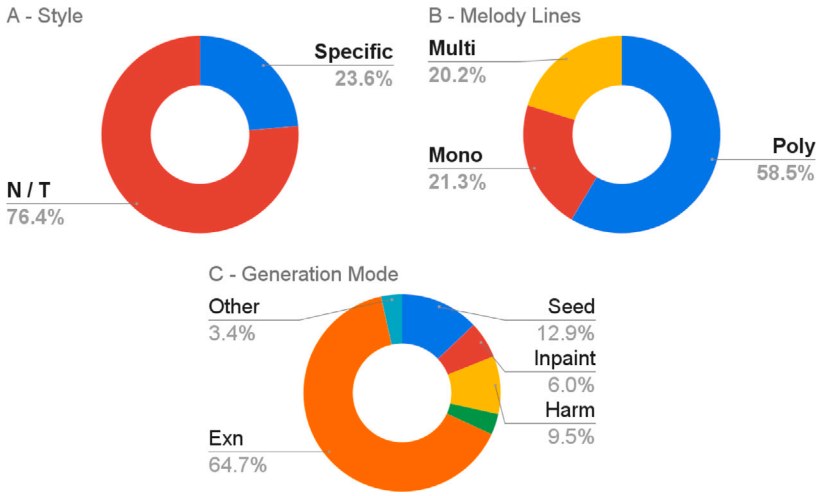

Fig. 4. Musical characteristics.

Table 4
Specific Style generators.

Rules	Constrains	Other
Chu et al. (2016)	Hadjeres and Nielsen (2020)	Lattner et al. (2018)
Mo et al. (2018)	Briot and Pachet (2020)	Hakimi et al. (2020)
Wiriyachaiporn et al. (2018)	Pachet et al. (2021)	Lim et al. (2020a)
Shukla and Banka (2018)	Lim et al. (2020b)	
Ting et al. (2017)	Choi et al. (2021)	
Stoltz and Aravind (2019)	Hadjeres et al. (2017)	
Garoufis et al. (2020)	Makris et al. (2021)	
Herremans and Chew (2017)	Choi et al. (2021)	
Sun et al. (2018)	Cunha et al. (2018)	
Wang et al. (2020)		
Azevedo et al. (2021)		
Sabitha et al. (2021)		
Jeong et al. (2017)		
Lopes et al. (2017)		
De Felice et al. (2017)		
De Prisco et al. (2020)		

3.4. Human factors

In this subsection, we study two aspects related to the interaction between a user (either the composer or a listener) and an automatic music generation system.

The first aspect is related to the user-system interface. To build a system that will be useful outside of the research community, the user interface must be clearly taken into account. However, our study shows that only 12 papers, such as (Hakimi et al., 2020; Suh et al., 2021), take this aspect into account. Thus, in their current state, the vast majority of the systems analyzed are very difficult to use in real-world music generation scenarios.

The other factor of human interaction that we take into account is the emotion-related aspects of music generation. Regardless of whether a human composer is involved, emotions undoubtedly play a very significant role in the composition of music (Juslin & Sloboda, 2001). This can be particularly relevant when composing for films and games, as music usually needs to complement the emotion suggested by the other media. However, this aspect has not been adequately addressed in automatic music generation systems. In fact, only 18 of the works analyzed in this study consider this topic. Two of those that do are (Cai & Cai, 2019; Shi & Wang, 2020).

3.5. Code and demo availability and integration

As mentioned in Section 3.1, in this work we analyzed 112 music generators and 10 works dealing with creation environments, work evaluation, datasets, and other topics. Of the works that present generators, 4 could potentially include analyzable code and musical demos. Another work that analyzed an AI-generated musical album also included musical demos. In total, then, 116 works could potentially include available code, and 117 could include musical demos. However, only 40 of these possible candidates have publicly available code and only 49 include musical demos. Most human composers currently use digital audio workstations (DAWs) to help them with their daily work. If a generator is to be used as a daily aid tool by composers, integration into DAWs is essential, but only 3 of the 112 generators (Bazin & Hadjeres, 2019; Hadjeres & Crestel, 2021; Roberts et al., 2018) currently include this feature. One particular case is that of Flow Machines (Pachet et al., 2021). In this case, the tool is deployed as an app that is a full assisted composition environment, that is, a small DAW. This tool is currently only available for Apple IPads in some markets. There are three more cases where Web versions of the generators are available. However, these examples are quite different, since (Oore et al., 2020) is a JavaScript-based interface for the generator. This generator, as well as (Huang et al., 2018; Jaques et al., 2017; Roberts et al., 2018), are part of Google's Magenta framework, which allows developers to build their own music production applications in a flexible fashion. Cococo (Louie et al., 2020) is a very nice, mostly educational, co-creation environment. The web demo provided with Open AI's Jukebox is just a Colab notebook to show the possibilities of using this generator as a development tool. Table 8 shows which works have publicly available code or demos.

Table 5
Works with Polyphonic output.

Hadjeres et al. (2017), Liang et al. (2017)
Brunner et al. (2017), Lim et al. (2017)
Chen et al. (2021), Ting et al. (2017)
Manzelli et al. (2018a), Roberts et al. (2018)
Mao et al. (2018), Mo et al. (2018)
Madhok et al. (2018), Wiriyachaiporn et al. (2018)
Kaliakatsos-Papakostas et al. (2018), Koh et al. (2018)
Agarwal et al. (2018), Lattner et al. (2018)
Huang et al. (2019), Hung et al. (2019), Zhao et al. (2019)
Nadeem et al. (2019), Wang et al. (2019), Wu et al. (2019)
Chen, Zhang, et al. (2019), Stoltz and Aravind (2019)
Ebrahimi et al. (2019), Herremans and Chew (2017)
Harrison and Pearce (2020), Huang and Yang (2020)
Cheng et al. (2020), Grachten et al. (2020)
Ma et al. (2020), Wang et al. (2020)
Briot et al. (2020), Chen et al. (2020)
Briot and Pachet (2020), Delarosa and Soros (2020)
Kan and Sourin (2020), Lim et al. (2020a)
Eisenbeiser (2020), Ycart and Benetos (2020)
Kurniawati et al. (2020), Tikhonov et al. (2017)
Lang et al. (2020), Oore et al. (2020)
Lim et al. (2020b), Wu et al. (2020)
Choi et al. (2021), Dean and Forth (2020)
Diéguez and Soo (2020), Suh et al. (2021)
Yeh et al. (2021), Zeng and Zhou (2021)
Azevedo et al. (2021), Hadjeres and Crestel (2021)
Choi et al. (2021), Li and Sung (2021)
Muhamed et al. (2021), Walter et al. (2021)
De Felice et al. (2017), De Prisco et al. (2020)
Cunha et al. (2018)

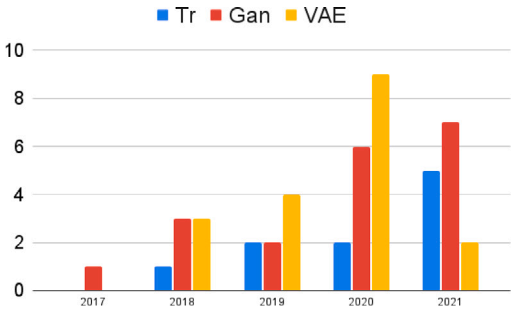

Fig. 5. Architecture evolution.

3.6. System architecture

In this section, we discuss the main architectures used to implement the music generators studied in this paper. The purpose of this study is not to discuss in depth the different architectural implementations that can be used to design an automatic music generator. A good description of most architectures can be found in (Briot et al., 2020). However, for the sake of completeness, we consider a basic description of the alternatives to be necessary.

Table 6
Works according to melody lines.

Mono	Multitrack
Jaques et al. (2017)	Roberts et al. (2018)
Johnson et al. (2017)	Yang et al. (2017)
Colombo et al. (2017)	Huang et al. (2018)
Williams et al. (2017)	Chu et al. (2016)
Yang et al. (2019)	Liu and Yang (2018)
Ferreira and Whitehead (2021)	Dong et al. (2018)
Hadjeres and Nielsen (2020)	Lattner and Grachten (2019)
Goienetxea et al. (2019)	Simões et al. (2019)
Huang and Huang (2020)	Makris et al. (2019)
Jin et al. (2020)	Guan et al. (2019)
Huang et al. (2020)	Donahue et al. (2019)
Shopynskyi et al. (2020)	Hutchings and McCormack (2019)
Hakimi et al. (2020)	Payne (2019)
Suthaphan et al. (2021)	Chen, Xiao, and Yin (2019)
Shi and Wang (2020)	Jia et al. (2019)
Grekow and Dimitrova-Grekow (2021)	Ens and Pasquier (2020)
Tanberk and Tükel (2021)	Frid et al. (2020), Pachet et al. (2021)
Sabitha et al. (2021)	Wen and Ting (2020)
Makris et al. (2021)	
Jeong et al. (2017)	
Lopes et al. (2017)	

First, we will briefly address the systems that are not based on neural networks. These systems cannot be directly considered as artificial intelligence systems, although they are used in several generators to condition the output of other "intelligent" systems. A Markov chain is a model that is used to describe a sequence of possible events. This sequence should be such that the probability of the next state depends only on the previous state. In general, Markov chains are a good way of imposing rules: we can consider, for example, that after a chord the probability of certain other chords is zero, while the transition to another chord is possible with a specific degree of probability. We may, of course, consider that our states should specify full bars instead of chords. In this case, our system will generate a bar based on the previous bar. Clearly, much more sophisticated options are also possible. As an example, (Chen et al., 2020) generates 8 bars based on the previous 8 vars using a generator that includes a neural network and a Markov model.

Evolutionary algorithms use an artificial version of Darwin's evolution theory to try to improve an original piece of music so that it adheres more to the desired style. The approach always starts with a set of chromosomes, which are themselves a list of genes. In the field of musical generation, chromosomes are pieces of music or fragments of pieces of music, and genes can be notes, chords, bars, or groups of bars. Initially, we apply a mutation function to the chromosomes. In many cases, the mutation function is based on harmonic rules with the assigned objective of producing new melodies based on the initial ones without breaking those harmonic rules. After the mutation, the crossover function is applied to be able to mix genes from different chromosomes into new chromosomes. The fitness function, which evaluates the quality of the resulting melodies, is then used to decide which genes survive. As an example, (Wen & Ting, 2020) uses a simple genetic algorithm in which genes are pitches or "tenuto" to hold the previous pitch and the chromosomes are the final output of the generator. (Zeng & Zhou, 2021) uses a similar approach for traditional Chinese music. In Table 9 we can see that genetic architectures are widely used by themselves or as components of more complex systems that also include convolutional networks (e.g. Shi & Wang, 2020). Out of the 118 generators analyzed, 13 are based on evolutionary algorithms.

At least in principle, the simplest type of neural network is a feedforward network. A feedforward neural network is an artificial

Table 7
Generation objective.

Seeded	Inpainting	Harmonization
Donahue et al. (2019)	Roberts et al. (2018)	Huang et al. (2018)
Huang et al. (2019)	Yang et al. (2017)	Liang et al. (2017)
Herremans and Chew (2017)	Ens and Pasquier (2020)	Shukla and Banka (2018)
Goienetxea et al. (2019)	Bazin and Hadjeres (2019)	Liu and Yang (2018)
Chen et al. (2020)	Brunner et al. (2018)	Huang et al. (2019)
Delarosa and Soros (2020)	Hadjeres and Crestel (2021)	Lattner and Grachten (2019)
Dhariwal et al. (2020)	Diéguez and Soo (2020)	Garoufis et al. (2020)
Grachten et al. (2020)		Jia et al. (2019)
Payne (2019)		Pachet et al. (2021)
Pachet et al. (2021)		Yang et al. (2019)
Huang and Yang (2020)		Sabitha et al. (2021)
Shi and Wang (2020)		Yeh et al. (2021)
Dean and Forth (2020)		De Prisco et al. (2020)

Table 8
Code and Demo availability.

Code and Demo	Code but no Demo	Demo but no Code
Hadjeres et al. (2017)	Briot et al. (2017, 2020)	Johnson et al. (2017)
Yang et al. (2017)	Roberts et al. (2018)	Chu et al. (2016)
Liang et al. (2017)	Muhamed et al. (2021)	Manzelli et al. (2018b)
Brunner et al. (2017)	Harrison and Pearce (2020)	Colombo et al. (2017)
Lim et al. (2020a, 2020b, 2017)	Diéguez and Soo (2020)	Ting et al. (2017)
Roberts et al. (2018)		Manzelli et al. (2018a)
Jaques et al. (2017)		Lattner and Grachten (2019)
Mao et al. (2018)		Simões et al. (2019)
Payne (2019), Salas (2018)		Herremans and Chew (2017)
Liu and Yang (2018)		Agarwal et al. (2018)
Dong et al. (2018)		Delarosa and Soros (2020)
Huang et al. (2019, 2018)		Pachet et al. (2021)
Brunner et al. (2018)		Jin et al. (2020)
Donahue et al. (2019)		Wu et al. (2020)
Ferreira and Whitehead (2021)		Dean and Forth (2020)
Stoltz and Aravind (2019)		Jeong et al. (2017)
Bazin and Hadjeres (2019)		De Felice et al. (2017)
Briot et al. (2020)		Cunha et al. (2018)
Harrison and Pearce (2020)		De Prisco et al. (2020)
Hadjeres and Nielsen (2020)		
Louie et al. (2020)		
Dhariwal et al. (2020)		
Ens and Pasquier (2020)		
Huang and Yang (2020)		
Suh et al. (2021)		
Oore et al. (2020)		
Hakimi et al. (2020)		
Azevedo et al. (2021)		
Marafioti et al. (2020)		
Hadjeres and Crestel (2021)		
Makris et al. (2021)		
Yu et al. (2021)		
Lopes et al. (2017)		

Table 9
Generator architectures.

	Total	Not GAN	GAN
RNN	52	46	6
VAE	18	14	4
Transformer	10	9	1
FF	24	16	8
Rule based	7	6	1
Evolutionary	13	13	0
other	12	12	0

neural network in which the connections between nodes do not form a cycle. These networks are widely used in image processing applications. Convolutional neural networks (CNN) such as Inception (used as part of the generator in Li & Sung, 2021) and dense fully connected networks (e.g., the Minibach toy example network introduced in (Briot et al., 2020) or the simple network used to produce traditional Indonesian music in (Kurniawati et al., 2020). These types of networks are widely used as part of more complex systems. In Table 9 it can be seen that more than 20% of the generators analyzed use FF networks as part of their architecture.

Recurrent neural networks (RNNs) were introduced to deal with sequences and time series. RNNs use their internal state to be able to process variable-length sequences of inputs and are therefore clearly the most popular option when generating music. More specifically, long-short-term memory(LSTM) networks, which include memory cells that can remember values over arbitrary time intervals and many of their variations, are the most widely used networks in music generation. As an example of widely cited LSTM-based systems, (Chu et al., 2016) uses a set of hierarchical LSTM systems to generate melodies. In (Hadjeres et al., 2017), three RNNs and an FF CNN are used to produce harmonized melodies. In Table 9 it can be seen that most of the analyzed generators use some type of RNN as a component of their generators.

Generative adversarial networks (GANs) are generative models: They create new data instances that resemble the training data. GANs are implemented by pairing two networks: a generator, which learns to produce the target output, and a discriminator, which learns to distinguish the true data from the generator output. The generator tries to fool the discriminator, and the discriminator tries to avoid being fooled. GANS are very widely used in automatic music generation because the required task is to produce melodies that cannot be easily distinguished from those of the training dataset. In GANs, the generator and the discriminator can be any of the networks already discussed. In (Dong et al., 2018; Yang et al., 2017)], for example, GANs are implemented using feedforward networks, while (Liu & Yang, 2018) uses an RNN as part of the generator. Of the 118 analyzed generators, 19 are GAN-based. Table 9 shows the types of architectures used in these GAN-based generators.

Variational autoencoders (VAE) are designed to compress input information into a constrained multivariate latent distribution (encoding) in order to reconstruct it as accurately as possible (decoding). Thus, this type of encoder can learn the fundamental characteristics of a training dataset and exclude unconventional possibilities, thereby compressing information about a piece of music into a reduced amount of information in the latent space. As the latent space tends to cluster similar examples close together, the information in this space can later be suitably altered to generate new pieces of music. These models have been widely used in automatic composition. As an example, Google's MusicVAE (Roberts et al., 2018) is capable of learning long-term structure using a hierarchical VAE. This model allows custom modifications to the latent space and, thus, is used in other works, e.g. (Diéguez & Soo, 2020) to implement alternative generators. Open AI Jukebox (Dhariwal et al., 2020) uses a VAE for raw audio compression. VAEs can also be used as components of GAN-based generators, as in (Wang et al., 2020). In Table 9 it can be seen that of the 113 works analyzed that present generators, 18 of them are VAE based, including some of the best known and most widely used generators.

Transformer networks are encoder–decoder architectures based on attention layers. These architectures, which also use positional encoding techniques to improve their long-term behavior, are very suitable for processing data sequences. As a result, this type of architecture has been extremely successful in automatic music generation systems. Transformer-based systems can be used by themselves (Choi et al., 2021; Huang et al., 2018; Huang & Yang, 2020) or as part of GAN-based generators (Muhamed et al., 2021). In Table 9 it can be seen that 10 out of the 118 analyzed systems use transformers as part of their architecture.

Table 10 shows the architectures of the generators studied. To reduce the size of the table and facilitate its usage, RNN architectures have not been included.

In Fig. 5 we see the growth of the different architectures studied per year. We have not included FFs or RNNs in the figure because currently they are mostly used as components of more complex generators. The figure shows that the use of GAN-based architectures has continued to grow steadily over the past five years. However, it is important to remember that these architectures can use other architectures as components. VAE-based systems have increased their usage, but, according to current data, seem to have stopped growing in 2021. Transformers, which are the newest of the architectures considered, have clearly become increasingly popular.

Considering the architecture used in the five most cited papers, they use a wide variety of architectures: DeepBach (Hadjeres et al., 2017) uses a combination of RNNs and FF networks, MidiNet and MuseGan (Dong et al., 2018; Yang et al., 2017) use GANs based on feedforward networks, Magenta Transformer (Huang et al., 2018) uses a Transformer Network, and Magenta MusicVAE (Roberts et al., 2018) uses a variational autoencoder. Considering the architecture used in the five most cited papers, they use a wide variety of architectures: DeepBach (Hadjeres et al., 2017) uses a combination of RNNs and FF networks, MidiNet and MuseGan (Dong et al., 2018; Yang et al., 2017) use GANs based on feed forward networks, Magenta Transformer (Huang et al., 2018) uses a Transformer Network and Magenta MusicVAE (Roberts et al., 2018) uses a Variational Autoencoder.

4. Real-world application

4.1. Analysis

When discussing the musical output of the different models, we decided to only take into consideration those systems that had music examples that could be analyzed (e.g., Colombo et al., 2017; Simões et al., 2019) or actual demonstrations of the software with which we could experiment with (e.g Donahue et al., 2019). The musicians involved in the research went over 52 different works, evaluating whether those systems could have real-life applications in the fields of live performance, music production, composing for media or as aids to musical composition. The evaluation took into account the current state of each system, the music it produced, its ease of use when code/demos were available, and the overall likelihood of professional implementation in daily workflows.

Overall, most of the systems developed are still very much in their inception stages and show little consideration for user experience. Works such as Jambot (Brunner et al., 2017) and MidiNet (Yang et al., 2017) could potentially produce symbolic music faster than it is reproduced, making them theoretically useful for live performance applications, but they nevertheless lack a user interface that can also facilitate workflows that require almost real-time speed both for the system and for the musician/composer. DeepBach (Hadjeres et al., 2017) is another capable system, although very style-specific. It creates up to four-part Bach chorales and is not particularly user-friendly, but has been used in other works with better UIs, such as NONOTO (Bazin & Hadjeres, 2019), a platform created using DeepBach and which facilitates its use and interactivity. Another approach is to integrate these systems (DeepBach and NONOTO) into other popular commercial music writing and DAW (Digital Audio workstations platforms for recording, mixing, and editing audio) applications by turning them into plugins for Musescore and Ableton, respectively. This particular method of implementation is shared by Magenta MusicVAE and PIA, the piano inpainting application (Hadjeres & Crestel, 2021; Roberts et al., 2018), which both can be used as plugins for the popular Ableton DAW. These implementations work particularly well, and therefore they make for a much more consistent user experience when composing and performing live.

Analyzing the music produced by these systems requires some contextualization because, as with most artistic expressions, it is not easy to judge the overall quality of the results. We compared the overall complexity of the music produced together with the consistency of the musical structure and the motivic development that occurs during the pieces. Works based on transformer architectures, such as Magenta Transformer (Huang et al., 2018), Musenet (Payne, 2019) or PIA (Hadjeres & Crestel, 2021), appear to offer the best performance. However, this is not always the case. LAKHNES (Donahue et al., 2019), another transformer-based system, is very good at style specificity, but the music it produces may lack strong long-term structure, even while being

Table 10
Works by network architecture.

	Not GAN based	GAN based
VAE	Brunner et al. (2018), Roberts et al. (2018) Lattner and Grachten (2019) Masuda and Iba (2018) Hung et al. (2019), Jia et al. (2019) Chen et al. (2020) Dhariwal et al. (2020) Grachten et al. (2020) Tikhonov et al. (2017) Grekow and Dimitrova-Grekow (2021) Lim et al. (2020a, 2020b) Diéguez and Soo (2020)	Qiu et al. (2019) Cheng et al. (2020) Wang et al. (2020) Huang and Huang (2020)
Transformer	Huang et al. (2018) Donahue et al. (2019), Payne (2019) Ens and Pasquier (2020) Huang and Yang (2020) Hadjeres and Crestel (2021) Choi et al. (2021), Makris et al. (2021)	Muhamed et al. (2021)
Evolutionary	Stoltz and Aravind (2019) Masuda and Iba (2018), Mo et al. (2018) Wen and Ting (2020) Azevedo et al. (2021) Sabitha et al. (2021), Zeng and Zhou (2021) Shi and Wang (2020) Jeong et al. (2017), Lopes et al. (2017) De Felice et al. (2017) De Prisco et al. (2020)	
Rule based	Manzelli et al. (2018b) Wiriyachaiporn et al. (2018) Cunha et al. (2018)	Jin et al. (2020)

coded with such objective taken into consideration. Other systems-like MCNN based on a GAN with LSTM applied rules-also offer good long-term structure but may not develop such complex or surprising musical content as those of the transformer-based mentioned before, probably due to their use of very strict rules. Jukebox (Dhariwal et al., 2020) is another transformer-based system that is very distinctive for its ability to work directly with sound (instead of symbolic music) and to generate every part of the song from music and rhythm to lyrics and singing. However, music generated working directly with .wav files, like the one from Jukebox (Dhariwal et al., 2020) or Wavenet (Dieleman et al., 2018), still presents some artifacts that are very recognizable to our ears and are therefore much harder to implement in a professional production than easily modifiable MIDI files. It would be very interesting to see the music produced by such systems applied in a heavily processed electronic mix to gauge the public acceptance of such acoustic artifacts in a distortion-rich context.

Another possible approach to evaluating the music produced by the studied works would be to consider how well it complies with the rules and characteristics of their specific musical style. The NONOTO and COCOCO systems mentioned above adapt very well to the sound and perceived characteristics of the four-voice Baroque compositions they aim to create. They do so by carefully choosing datasets for training and by using some manner of musical theory rules. EvoComposer (De Prisco et al., 2020), an evolutionary algorithm that also aims to compose four-part Bach chorales, may be better at complying with theoretical music rules, as we spotted very few mistakes in the counterpoint of the provided scores. Other evolutionary systems, like BOSSA (Wen & Ting, 2020), do not seem to generate melodies that are particularly representative of the style in question. The BOSSA system seems to use post-processing, fitting the melody to a prearranged bossa nova guitar part, so that the final result will be perceived as matching that genre. With no available code and having to rely exclusively on prerecorded audio demonstrations, it is very difficult to identify what the systems are actually generating and what is done by a human. Deep-J (Mao et al., 2018), an RNN based system capable of adapting to specific styles, clearly does this, but sometimes portrays the typical characteristics of the different styles so obviously that its compositions can potentially be perceived as very predictable.

Having analyzed all these works, it can be argued that most of the systems described are style-specific, albeit unintentionally. This is due to the datasets selected during training. This in itself does not necessarily present a problem when the authors are aware. However, there can be a problem when authors present their works as non-specific-style generators. Such works usually are biased towards the style of their training datasets, and therefore are likely to produce better results in said style. There are very few works, if any, that are trained with non-western music, meanwhile, there are many works aimed at creating generators useful for any particular style (with over 74% of the works analyzed not suggesting a particular style). We decided not to mention any particular proposal, as this is a common trend, and we believe that more in-depth research is needed to measure the cultural impact of dataset choice when training algorithms for music generators.

4.2. Compositional problems

As further discussed in Section 4.3 in paper (Avdeeff, 2019) there is a categorization between music generation and pop music generation, in which the latter tries to facilitate and speed the processes of music creation rather than attempting to create the best possible music. When utilizing different solutions for music generation we have to take into consideration the specific musical problem we are faced with. The guitar solo generator (Cunha et al., 2018) is a genetic generator that combines pre-composed 1 bar sections of guitar solos using combinatorial optimization methods to generate a 12 bar blues guitar solo. This particular approach lacks the flexibility and inventiveness of other methods and require some previous compositions. Nevertheless, when facing a pop music generating scenario its reliability may compensate for it. We have found out that evolutionary generating algorithms as (Jeong et al., 2017) as well as other rule based systems such as (Wang et al., 2019) are very good at generating for specific styles although their main caveat is that they require a very good understanding of the music theory relevant to the particular style in order to create them, and as (Jeong et al., 2017) articulates it, they are somehow limited by the math with which they are created. A very good example of it is the EvoComposer (De Prisco et al., 2020) which by the works provided seems as capable as the most cited work of this kind the DeepBach (Hadjeres et al., 2017) at following music theory rules for creating baroque style 4-part harmonizations.

The other main issue that can be differential between architectures is what they are actually generating. When generating some of the most recurrent aims for a generator are (Tables 6 and 5): single line melodies (non-polyphonic), polyphonic accompaniment (aimed at generating homophony), polyphonic harmonization (where the individual melodies are as important as the relations between them), and drum generation. There are other implementations as well, as there are many possible musical problems that could be resolved by AI generating techniques, one of the most recognizable is JukeBox from OpenAI (Dhariwal et al., 2020) aimed at creating a full song or an inpainting between two stylistically different musical fragments in the audio realm. We speculate that the most relevant aspects for determining which AI architectures should be used for a specific purpose are the reliability of the produced outcome and the possible necessity to create completely new musical material with good long-term structure. Nevertheless, further research is needed to have a more definitive position on the matter. There seems to be evidence that systems such as transformer-based LAkhNes (Donahue et al., 2019) and Magenta Transformer (Huang et al., 2018), which are focused on the long-term structure of the generated pieces, are very good at generating short to medium musical ideas that could become the basis of new compositions. Nowadays, they require a human to choose which ones to use, as they are prone to generate inconsistent musical material. On the other hand, systems where musical rules have more weight such as Flow Machines (Pachet et al., 2021) or other statistical and evolutionary systems tend to be more consistent in their output even when the musical ideas they generate may sometimes be perceived as overly predictable. Regarding style specificity, such a compositional problem can be tackled using different strategies. Whether harmonizing Bach Chorales or creating modern jazz "improvisations", the use of datasets specific to the style is the most common solution. Very rarely systems are trained with a dataset of undesirable outputs (for example, different and undesired music genres). This may lead to the classic problem with Deep Learning of generating good music that can nevertheless be identified with many genres beyond the one originally intended. The other route to achieve style specificity is through music rules or constraints specific to the style, which requires a very good understanding of musical theory.

4.3. Usability and user interface issues

In a review of Hello World by SKYGGE, the first album created using multiple AI music generation techniques (with the FLOW MACHINES system) (Avdeeff, 2019), Avdeeff differentiates between traditional AI music generation and AI pop music generation. By making this distinction, the author reflects on how AI pop music generators are not meant to create the best possible music for themselves, but rather to create music and processes that accelerate and facilitate efforts to compose and produce music. We have come to the conclusion that current AI-based generators are more than capable of being integrated into a very wide range of music creation routines. As many of these generators are incredibly fast, they can free musicians from routine chores, allowing them to focus on selecting and piecing together music and sounds. As already mentioned, many commercial players in the field are beginning to pay attention to user interfaces and DAW integration. We believe that to be fully relevant to the scientific community and to make the knowledge obtained transferable to the music industry, most research into music generators should at least incorporate audio samples, if possible providing open source code or interactive demos (details on available codes and demos for the analyzed works can be found in Section 3.5).

As discussed in Section 3.4 emotions play a very significant role in the composition of music. This is reflected in the fact that 18 of the 118 generators analyzed take this aspect into account. However, while user interface issues are considered in some very highly cited papers (e.g. Roberts et al., 2018) papers considering emotional issues generally have a small number of citations. It should be mentioned that some of the commercial AI-based music services such as AIVA[7] consider emotional generation an essential part of their business.

5. Discussion and conclusions

Regarding RQ1, Section 3.1 and especially Table 2 and Figs. 2 and 3, show that there is growing interest in this field, as shown by the increase in the number of publications, and that although it may still not be the main focus of AI research, it is taken very seriously. Publications are geographically widely distributed, clearly showing a global interest in the field. Publications also come from both the academic and private sectors, with important contributions from key players in the fields of AI (Google, OpenAI, Amazon) and music (Sony, Spotify).

Regarding RQ2, Section 3.6 and especially Table 9 and Fig. 5 show that there is a wide variety of AI solutions implemented for music generators. Although more than 70% of the works are based on deep learning, evolutionary-based solutions are also widely represented. Among deep-leaning-based solutions, transformers and GAN-based solutions seem to be gaining popularity.

Table 5, Table 6 and Fig. 4B indicate the melody lines produced by the generator, while Table 7 and Fig. 4C show the specific generation problem aimed at by the generator. Table 4 and Fig. 4A indicate the generators that target a specific style by explicit design restrictions. It can be seen that most generators are polyphonic, while monophonic and multi-instrument generators are still popular. We also find a wide range of solutions for seeded melodies, inpainting and harmonization, and that in many cases the style is obtained mainly through the training dataset as discussed in Section 4.2. The wide variety of datasets used is discussed in Section 3.2.

Regarding RQ3, in Section 4 we discuss several systems for music generation and their many possible applications, from polyphonic seeded generation to inpainting in the audio realm. There is a growing number of solutions available, and we emphasize the importance of having codes and demos for research purposes while DAW integration or careful consideration of the UI is a must if they are intended to be implemented in professional workflows. We discuss how style specificity can be both an objective for the generator as well a tool for determining how good the output of such a generator is. Nevertheless, further research is needed in order to create a robust methodology for judging the AI musical generations based on style-specific criteria. Such a tool would help music industry professionals to pick the correct AI tool for a particular endeavor.

Regarding RQ4, Section 3.4 shows the relatively small effort that has been paid to human interface and emotion-related issues. Nevertheless, it is worth mentioning that several highly cited papers (Avdeeff, 2019; Payne, 2019; Roberts et al., 2018) already include DAW integration aimed at non-research usage scenarios. Section 4.3 further expands the topic of why such integration can be relevant.

Briefly summarizing our finding related to future research trends and needs, it is clear from the number of publications, the number of citations and the academic and commercial institutions involved

[7] https://www.aiva.ai/

that research related to automatic music generation systems and applications will continue to increase steadily in the near future. As in other applications based on artificial intelligence (Raschka, Patterson, & Nolet, 2020), from the available data, it seems that transformer-based architectures will increase their popularity and usage. Although most of the systems analyzed are symbolic, the number of citations for audio-based generators (especially Dieleman et al., 2018 and Dhariwal et al., 2020) suggests that interest in this type of system will also increase, with further implementations being developed as issues with audio artifacts are resolved. It is also clear that generators are starting to become usable products for a musician's daily life, but an important research effort on interfacing issues and human–machine co-creation is still needed. Although it cannot be deduced from our review data, considering current commercial systems with no associated publications, we think that the interest in emotion-aware generators will also increase in the future. We have been able to show that the interest in automatic music composition is increasing and that most of the main players, both in the AI and music industries, are involved (Google, OpenAI, Amazon, Sony, Spotify, etc.).

As a final point though, during our testing of the different systems, we came to realize how different the approach to composition is when using these automatic generators. In our experience, the composer became more of an arranger of different melodies, something like a producer from the 70 s rock and roll scene trying to order the wild creativity of some misbehaving rock stars. Although sometimes frustrating, it is a very creative, fruitful process, one with an endless flow of new ideas from the generators, and we firmly believe that further research needs to be carried out into the relationship between human and AI composers in order to provide a framework in which each of them can make use of their very best qualities.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Maria J. Escalona reports financial support was provided by Spain Ministry of Science and Innovation. Maria J. Escalona reports financial support was provided by Government of Andalusia Ministry of Economic Transformation Industry Knowledge and Universities.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported by the NICO project (PID2019-105455GB-C31) from Ministerio de Ciencia, Innovación y Universidades (Spanish Government) and by the NDT 4.0 project (US-1251532) from Consejería de Economía y Conocimiento (Junta de Andalucia).

References

Agarwal, S., Saxena, V., Singal, V., & Aggarwal, S. (2018). Lstm based music generation with dataset preprocessing and reconstruction techniques. In *2018 IEEE symposium series on computational intelligence* (pp. 455–462). IEEE.

Anantrasirichai, N., & Bull, D. (2021). Artificial intelligence in the creative industries: A review. *Artificial Intelligence Review*, 1–68.

Avdeeff, M. (2019). Artificial intelligence & popular music: SKYGGE, flow machines, and the audio uncanny valley. In *Arts, vol. 8, no. 4* (p. 130). Multidisciplinary Digital Publishing Institute.

Azevedo, L. R., Silla Jr, C. N., & Costa-Abreu, M. (2021). A methodology for procedural piano music composition with mood templates using genetic algorithms.

Bazin, T., & Hadjeres, G. (2019). Nonoto: A model-agnostic web interface for interactive music composition by inpainting. arXiv preprint arXiv:1907.10380.

Briot, J.-P. (2021). From artificial neural networks to deep learning for music generation: history, concepts and trends. *Neural Computing and Applications*, *33*(1), 39–65.

Briot, J.-P., Hadjeres, G., & Pachet, F.-D. (2017). Deep learning techniques for music generation–a survey. arXiv preprint arXiv:1709.01620.

Briot, J.-P., Hadjeres, G., & Pachet, F.-D. (2020). *Deep learning techniques for music generation*. Springer.

Briot, J.-P., & Pachet, F. (2020). Deep learning for music generation: Challenges and directions. *Neural Computing and Applications*, *32*(4), 981–993.

Brown, H., & Casey, M. (2019). Heretic: Modeling anthony braxton's language music. In *2019 International workshop on multilayer music representation and processing* (pp. 35–40). IEEE.

Brunner, G., Konrad, A., Wang, Y., & Wattenhofer, R. (2018). MIDI-VAE: Modeling dynamics and instrumentation of music with applications to style transfer. arXiv preprint arXiv:1809.07600.

Brunner, G., Wang, Y., Wattenhofer, R., & Wiesendanger, J. (2017). JamBot: Music theory aware chord based generation of polyphonic music with LSTMs. In *2017 IEEE 29th international conference on tools with artificial intelligence* (pp. 519–526). IEEE.

Cai, L., & Cai, Q. (2019). Music creation and emotional recognition using neural network analysis. *Journal of Ambient Intelligence and Humanized Computing*, 1–10.

Cambouropoulos, E., et al. (2021). Cognitive musicology and artificial intelligence: Harmonic analysis, learning, and generation. In *Handbook of artificial intelligence for music* (pp. 263–281). Springer.

Chen, M.-Y., Wei, W., Chao, H.-C., & Li, Y.-F. (2021). Robotic musicianship based on least squares and sequence generative adversarial networks. *IEEE Sensors Journal*.

Chen, K., Xia, G., & Dubnov, S. (2020). Continuous melody generation via disentangled short-term representations and structural conditions. In *2020 IEEE 14th international conference on semantic computing* (pp. 128–135). IEEE.

Chen, H., Xiao, Q., & Yin, X. (2019). Generating music algorithm with deep convolutional generative adversarial networks. In *2019 IEEE 2nd international conference on electronics technology* (pp. 576–580). IEEE.

Chen, K., Zhang, W., Dubnov, S., Xia, G., & Li, W. (2019). The effect of explicit structure encoding of deep neural networks for symbolic music generation. In *2019 International workshop on multilayer music representation and processing* (pp. 77–84). IEEE.

Cheng, P.-S., Lai, C.-Y., Chang, C.-C., Chiou, S.-F., & Yang, Y.-C. (2020). A variant model of TGAN for music generation. In *Proceedings of the 2020 asia service sciences and software engineering conference* (pp. 40–45).

Choi, K., Park, J., Heo, W., Jeon, S., & Park, J. (2021). Chord conditioned melody generation with transformer based decoders. *IEEE Access*, *9*, 42071–42080.

Chu, H., Urtasun, R., & Fidler, S. (2016). Song from PI: A musically plausible network for pop music generation. arXiv preprint arXiv:1611.03477.

Colombo, F., Seeholzer, A., & Gerstner, W. (2017). Deep artificial composer: A creative neural network model for automated melody generation. In *International conference on evolutionary and biologically inspired music and art* (pp. 81–96). Springer.

Cunha, N. d. S., Subramanian, A., & Herremans, D. (2018). Generating guitar solos by integer programming. *Journal of the Operational Research Society*, *69*(6), 971–985.

De Felice, C., De Prisco, R., Malandrino, D., Zaccagnino, G., Zaccagnino, R., & Zizza, R. (2017). Splicing music composition. *Information Sciences*, *385*, 196–212.

De Prisco, R., Zaccagnino, G., & Zaccagnino, R. (2020). Evocomposer: An evolutionary algorithm for 4-voice music compositions. *Evolutionary Computation*, *28*(3), 489–530.

Dean, R. T., & Forth, J. (2020). Towards a deep improviser: A prototype deep learning post-tonal free music generator. *Neural Computing and Applications*, *32*(4), 969–979.

Delarosa, O., & Soros, L. B. (2020). Growing MIDI music files using convolutional cellular automata. In *2020 IEEE symposium series on computational intelligence* (pp. 1187–1194). IEEE.

Dervakos, E., Filandrianos, G., & Stamou, G. (2021). Heuristics for evaluation of AI generated music. In *2020 25th International conference on pattern recognition* (pp. 9164–9171). IEEE.

Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., & Sutskever, I. (2020). Jukebox: A generative model for music. arXiv preprint arXiv:2005.00341.

Diéguez, P. L., & Soo, V.-W. (2020). Variational autoencoders for polyphonic music interpolation. In *2020 International conference on technologies and applications of artificial Intelligence* (pp. 56–61). IEEE.

Dieleman, S., Oord, A. v. d., & Simonyan, K. (2018). The challenge of realistic music generation: modelling raw audio at scale. arXiv preprint arXiv:1806.10474.

Donahue, C., Mao, H. H., Li, Y. E., Cottrell, G. W., & McAuley, J. (2019). LakhNES: Improving multi-instrumental music generation with cross-domain pre-training. arXiv preprint arXiv:1907.04868.

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 *Thirty-second AAAI conference on artificial intelligence*.

Ebrahimi, M., Majidi, B., & Eshghi, M. (2019). Procedural composition of traditional Persian music using deep neural networks. In *2019 5th Conference on knowledge based engineering and innovation* (pp. 521–525). IEEE.

Eisenbeiser, L. (2020). Latent walking techniques for conditioning GAN-generated music. In *2020 11th IEEE annual ubiquitous computing, electronics & mobile communication conference* (pp. 0548–0553). IEEE.

Ens, J., & Pasquier, P. (2020). Mmm: Exploring conditional multi-track music generation with the transformer. arXiv preprint arXiv:2008.06048.

Evans, B. L., Munekata, N., & Ono, T. (2017). Using a human-agent interaction model to consider the interaction of humans and music-generation systems. In *Proceedings of the companion of the 2017 ACM/IEEE international conference on human-robot interaction* (pp. 115–116).

Ferreira, L. N., & Whitehead, J. (2021). Learning to generate music with sentiment. arXiv preprint arXiv:2103.06125.

Frid, E., Gomes, C., & Jin, Z. (2020). Music creation by example. In *CHI '20*, (pp. 1–13). New York, NY, USA: Association for Computing Machinery.

Galajda, J. E., Royal, B., & Hua, K. A. (2021). Deep composer: A hash-based duplicative neural network for generating multi-instrument songs. In *2020 25th international conference on pattern recognition* (pp. 7961–7968). IEEE.

Garoufis, C., Zlatintsi, A., & Maragos, P. (2020). An LSTM-based dynamic chord progression generation system for interactive music performance. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing* (pp. 4502–4506). IEEE.

Gioti, A.-M. (2020). From artificial to extended intelligence in music composition. *Organised Sound*, *25*(1), 25–32.

Goienetxea, I., Mendialdua, I., Rodríguez, I., & Sierra, B. (2019). Statistics-based music generation approach considering both rhythm and melody coherence. *IEEE Access*, *7*, 183365–183382.

Grachten, M., Lattner, S., & Deruty, E. (2020). BassNet: A variational gated autoencoder for conditional generation of bass guitar tracks with learned interactive control. *Applied Sciences*, *10*(18), 6627.

Grekow, J., & Dimitrova-Grekow, T. (2021). Monophonic music generation with a given emotion using conditional variational autoencoder. *IEEE Access*, *9*, 129088–129101.

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* (pp. 1–6). IEEE.

Hadjeres, G., & Crestel, L. (2021). The piano inpainting application. arXiv preprint arXiv:2107.05944.

Hadjeres, G., & Nielsen, F. (2020). Anticipation-RNN: Enforcing unary constraints in sequence generation, with application to interactive music generation. *Neural Computing and Applications*, *32*(4), 995–1005.

Hadjeres, G., Pachet, F., & Nielsen, F. (2017). Deepbach: A steerable model for bach chorales generation. In *International conference on machine learning* (pp. 1362–1371). PMLR.

Hakimi, S. H., Bhonker, N., & El-Yaniv, R. (2020). Bebopnet: Deep neural models for personalized jazz improvisations. In *Proceedings of the 21st international society for music information retrieval conference*.

Harrison, P. M., & Pearce, M. T. (2020). A computational cognitive model for the analysis and generation of voice leadings. *Music Perception*, *37*(3), 208–224.

Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, C.-Z. A., Dieleman, S., et al. (2018). Enabling factorized piano music modeling and generation with the MAESTRO dataset. arXiv preprint arXiv:1810.12247.

Herremans, D., & Chew, E. (2017). Morpheus: Generating structured music with constrained patterns and tension. *IEEE Transactions on Affective Computing*, *10*(4), 510–523.

Huang, C.-Z. A., Cooijmans, T., Roberts, A., Courville, A., & Eck, D. (2019). Counterpoint by convolution. arXiv preprint arXiv:1903.07227.

Huang, T.-M., Hsieh, H., Qin, J., Liu, H.-F., & Eirinaki, M. (2020). Play it again IMuCo! music composition to match your mood. In *2020 Second international conference on transdisciplinary AI* (pp. 9–16). IEEE.

Huang, C.-F., & Huang, C.-Y. (2020). Emotion-based AI music generation system with CVAE-GAN. In *2020 IEEE eurasia conference on IOT, communication and engineering* (pp. 220–222). IEEE.

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., et al. (2018). Music transformer. arXiv preprint arXiv:1809.04281.

Huang, Y.-S., & Yang, Y.-H. (2020). Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *Proceedings of the 28th ACM international conference on multimedia* (pp. 1180–1188).

Hung, H.-T., Wang, C.-Y., Yang, Y.-H., & Wang, H.-M. (2019). Improving automatic jazz melody generation by transfer learning techniques. In *2019 Asia-Pacific signal and information processing association annual summit and conference* (pp. 339–346). IEEE.

Hutchings, P. E., & McCormack, J. (2019). Adaptive music composition for games. *IEEE Transactions on Games*, *12*(3), 270–280.

Jaques, N., Gu, S., Turner, R. E., & Eck, D. (2017). Tuning recurrent neural networks with reinforcement learning.

Jeong, J., Kim, Y., & Ahn, C. W. (2017). A multi-objective evolutionary approach to automatic melody generation. *Expert Systems with Applications*, *90*, 50–61.

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* (pp. 1–6). IEEE.

Jiang, T., Xiao, Q., & Yin, X. (2019). Music generation using bidirectional recurrent network. In *2019 IEEE 2nd international conference on electronics technology* (pp. 564–569). IEEE.

Jin, C., Tie, Y., Bai, Y., Lv, X., & Liu, S. (2020). A style-specific music composition neural network. *Neural Processing Letters*, *52*, 1893–1912.

Johnson, D. D., Keller, R. M., & Weintraut, N. (2017). Learning to create jazz melodies using a product of experts..

Joshi, G., Nayapati, V., Singh, J., & Karmarkar, A. (2018). A comparative analysis of algorithmic music generation on GPUs and FPGAs. In *2018 Second international conference on inventive communication and computational technologies* (pp. 229–232). IEEE.

Juslin, P. N., & Sloboda, J. A. (2001). *Music and emotion: Theory and research*. Oxford University Press.

Kaliakatsos-Papakostas, M., Gkiokas, A., & Katsouros, V. (2018). Interactive control of explicit musical features in generative LSTM-based systems. In *Proceedings of the audio mostly 2018 on sound in immersion and emotion* (pp. 1–7).

Kan, Z. J., & Sourin, A. (2020). Generation of irregular music patterns with deep learning. In *2020 International conference on cyberworlds* (pp. 188–195). IEEE.

Koh, E. S., Dubnov, S., & Wright, D. (2018). Rethinking recurrent latent variable model for music composition. In *2018 IEEE 20th international workshop on multimedia signal processing* (pp. 1–6). IEEE.

Kurniawati, A., Suprapto, Y. K., & Yuniarno, E. M. (2020). Multilayer perceptron for symbolic Indonesian music generation. In *2020 International seminar on intelligent technology and its applications* (pp. 228–233). IEEE.

Lang, R., Wu, S., Zhu, S., & Li, Z. (2020). SSCL: Music generation in long-term with cluster learning. In *2020 IEEE 4th information technology, networking, electronic and automation control conference, Vol. 1* (pp. 77–81). IEEE.

Lattner, S., & Grachten, M. (2019). High-level control of drum track generation using learned patterns of rhythmic interaction. In *2019 IEEE workshop on applications of signal processing to audio and acoustics* (pp. 35–39). IEEE.

Lattner, S., Grachten, M., & Widmer, G. (2018). Imposing higher-level structure in polyphonic music generation using convolutional restricted Boltzmann machines and constraints. *Journal of Creative Music Systems*, *2*, 1–31.

Li, S., Jang, S., & Sung, Y. (2019). Melody extraction and encoding method for generating healthcare music automatically. *Electronics*, *8*(11), 1250.

Li, S., & Sung, Y. (2021). INCO-GAN: Variable-length music generation method based on inception model-based conditional GAN. *Mathematics*, *9*(4), 387.

Liang, F. T., Gotham, M., Johnson, M., & Shotton, J. (2017). Automatic stylistic composition of bach chorales with deep LSTM.

Lim, Y.-Q., Chan, C. S., & Loo, F. Y. (2020a). ClaviNet: Generate music with different musical styles. *IEEE MultiMedia*, *28*(1), 83–93.

Lim, Y.-Q., Chan, C. S., & Loo, F. Y. (2020b). Style-conditioned music generation. In *2020 IEEE international conference on multimedia and expo* (pp. 1–6). IEEE.

Lim, H., Rhyu, S., & Lee, K. (2017). Chord generation from symbolic melody using BLSTM networks. arXiv preprint arXiv:1712.01011.

Liu, C.-H., & Ting, C.-K. (2016). Computational intelligence in music composition: A survey. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *1*(1), 2–15.

Liu, H.-M., & Yang, Y.-H. (2018). Lead sheet generation and arrangement by conditional generative adversarial network. In *2018 17th IEEE international conference on machine learning and applications* (pp. 722–727). IEEE.

Lopes, H. B., Martins, F. V. C., Cardoso, R. T., & dos Santos, V. F. (2017). Combining rules and proportions: A multiobjective approach to algorithmic composition. In *2017 IEEE congress on evolutionary computation* (pp. 2282–2289). IEEE.

Lopez-Rincon, O., Starostenko, O., & Ayala-San Martín, G. (2018). Algoritmic music composition based on artificial intelligence: A survey. In *2018 International conference on electronics, communications and computers* (pp. 187–193). IEEE.

Louie, R., Coenen, A., Huang, C. Z., Terry, M., & Cai, C. J. (2020). Novice-AI music co-creation via AI-steering tools for deep generative models. In *Proceedings of the 2020 CHI conference on human factors in computing systems* (pp. 1–13).

Ma, D., Liu, B., Qiao, X., Cao, D., & Yin, G. (2020). Coarse-to-fine framework for music generation via generative adversarial networks. In *Proceedings of the 2020 4th high performance computing and cluster technologies conference & 2020 3rd international conference on big data and artificial intelligence* (pp. 192–198).

Madhok, R., Goel, S., & Garg, S. (2018). Sentimozart: Music generation based on emotions. In *ICAART (2)* (pp. 501–506).

Makris, D., Agres, K. R., & Herremans, D. (2021). Generating lead sheets with affect: A novel conditional seq2seq framework. arXiv preprint arXiv:2104.13056.

Makris, D., Kaliakatsos-Papakostas, M., Karydis, I., & Kermanidis, K. L. (2017). Combining LSTM and feed forward neural networks for conditional rhythm composition. In *International conference on engineering applications of neural networks* (pp. 570–582). Springer.

Makris, D., Kaliakatsos-Papakostas, M., Karydis, I., & Kermanidis, K. L. (2019). Conditional neural sequence learners for generating drums' rhythms. *Neural Computing and Applications*, *31*(6), 1793–1804.

Manzelli, R., Thakkar, V., Siahkamari, A., & Kulis, B. (2018a). Conditioning deep generative raw audio models for structured automatic music. arXiv preprint arXiv:1806.09905.

Manzelli, R., Thakkar, V., Siahkamari, A., & Kulis, B. (2018b). An end to end model for automatic music generation: Combining deep raw and symbolic audio networks. In *Proceedings of the musical metacreation workshop at 9th international conference on computational creativity*. Salamanca, Spain.

Mao, H. H., Shin, T., & Cottrell, G. (2018). DeepJ: Style-specific music generation. In *2018 IEEE 12th international conference on semantic computing* (pp. 377–382). IEEE.

Marafioti, A., Majdak, P., Holighaus, N., & Perraudin, N. (2020). GACELA: A generative adversarial context encoder for long audio inpainting of music. *IEEE Journal of Selected Topics in Signal Processing*, *15*(1), 120–131.

Marsden, M., & Ajoodha, R. (2021). Algorithmic music composition using probabilistic graphical models and artificial neural networks. In *2021 Southern African Universities power engineering conference/robotics and mechatronics/pattern recognition association of South Africa* (pp. 1–4). IEEE.

Masuda, N., & Iba, H. (2018). Musical composition by interactive evolutionary computation and latent space modeling. In *2018 IEEE international conference on systems, man, and cybernetics* (pp. 2792–2797). IEEE.

Menabrea, L. F., & Lovelace, A. (1842). Sketch of the analytical engine invented by Charles Babbage.

Mo, F., Wang, X., Li, S., & Qian, H. (2018). A music generation model for robotic composers. In *2018 IEEE international conference on robotics and biomimetics* (pp. 1483–1488). IEEE.

Mor, B., Garhwal, S., & Kumar, A. (2020). A systematic literature review on computational musicology. *Archives of Computational Methods in Engineering*, *27*(3), 923–937.

Moura, F. T., & Maw, C. (2021). Artificial intelligence became beethoven: how do listeners and music professionals perceive artificially composed music? *Journal of Consumer Marketing*.

Muhamed, A., Li, L., Shi, X., Yaddanapudi, S., Chi, W., Jackson, D., et al. (2021). Symbolic music generation with transformer-GANs. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 35, no. 1* (pp. 408–417).

Nadeem, M., Tagle, A., & Sitsabesan, S. (2019). Let's make some music. In *2019 International conference on electronics, information, and communication* (pp. 1–4). IEEE.

Oore, S., Simon, I., Dieleman, S., Eck, D., & Simonyan, K. (2020). This time with feeling: Learning expressive musical performance. *Neural Computing and Applications*, *32*(4), 955–967.

Pachet, F., Roy, P., & Carré, B. (2021). Assisted music creation with flow machines: towards new categories of new. In *Handbook of artificial intelligence for music* (pp. 485–520). Springer.

Payne, C. (2019). Musenet. *OpenAI Blog*, *3*.

Peters, M. D., Godfrey, C. M., Khalil, H., McInerney, P., Parker, D., & Soares, C. B. (2017). Guidance for conducting systematic scoping reviews. *JBI Evidence Implementation*, *13*(3), 141–146.

Peters, M. D., Marnie, C., Tricco, A. C., Pollock, D., Munn, Z., Alexander, L., et al. (2020). Updated methodological guidance for the conduct of scoping reviews. *JBI Evidence Synthesis*, *18*(10), 2119–2126.

Plut, C., & Pasquier, P. (2020). Generative music in video games: State of the art, challenges, and prospects. *Entertainment Computing*, *33*, Article 100337.

Qiu, Z., Ren, Y., Li, C., Liu, H., Huang, Y., Yang, Y., et al. (2019). Mind band: A crossmedia AI music composing platform. In *Proceedings of the 27th ACM international conference on multimedia* (pp. 2231–2233).

Raschka, S., Patterson, J., & Nolet, C. (2020). Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *Information*, *11*(4), 193.

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.

Sabitha, R., Majji, S., Kathiravan, M., Kumar, S. G., Kharade, K., & Karanam, S. R. (2021). Artificial intelligence based music composition system-multi algorithmic music arranger (MAGMA). In *2021 Second international conference on electronics and sustainable communication systems* (pp. 1808–1813). IEEE.

Salas, J. (2018). Generating music from literature using topic extraction and sentiment analysis. *IEEE Potentials*, *37*(1), 15–18.

Shi, N., & Wang, Y. (2020). Symmetry in computer-aided music composition system with social network analysis and artificial neural network methods. *Journal of Ambient Intelligence and Humanized Computing*, 1–16.

Shopynskyi, M., Golian, N., & Afanasieva, I. (2020). Long short-term memory model appliance for generating music compositions. In *2020 IEEE international conference on problems of infocommunications. Science and technology* (pp. 239–242). IEEE.

Shukla, S., & Banka, H. (2018). An automatic chord progression generator based on reinforcement learning. In *2018 International conference on advances in computing, communications and informatics* (pp. 55–59). IEEE.

Simões, J. M., Machado, P., & Rodrigues, A. C. (2019). Deep learning for expressive music generation. In *Proceedings of the 9th international conference on digital and interactive arts* (pp. 1–9).

Singh, J., & Ratnawat, A. (2018). Algorithmic music generation for the stimulation of musical memory in Alzheimer's. In *2018 4th international conference on computing communication and automation* (pp. 1–4). IEEE.

Stoltz, B., & Aravind, A. (2019). MU_PSYC: Music psychology enriched genetic algorithm. In *2019 IEEE congress on evolutionary computation* (pp. 2121–2128). IEEE.

Suh, M., Youngblom, E., Terry, M., & Cai, C. J. (2021). AI as social glue: Uncovering the roles of deep generative AI during social music composition. In *Proceedings of the 2021 CHI conference on human factors in computing systems* (pp. 1–11).

Sun, Z., Liu, J., Zhang, Z., Chen, J., Huo, Z., Lee, C. H., et al. (2018). Composing music with grammar argumented neural networks and note-level encoding. In *2018 Asia-Pacific signal and information processing association annual summit and conference* (pp. 1864–1867). IEEE.

Suthaphan, P., Boonrod, V., Kumyaito, N., & Tamee, K. (2021). Music generator for elderly using deep learning. In *2021 Joint international conference on digital arts, media and technology with ECTI northern section conference on electrical, electronics, computer and telecommunication engineering* (pp. 289–292). IEEE.

Tanberk, S., & Tükel, D. B. (2021). Style-specific Turkish pop music composition with CNN and LSTM network. In *2021 IEEE 19th world symposium on applied machine intelligence and informatics* (pp. 000181–000185). IEEE.

Tikhonov, A., Yamshchikov, I. P., et al. (2017). Music generation with variational recurrent autoencoder supported by history. arXiv preprint arXiv:1705.05458.

Ting, C.-K., Wu, C.-L., & Liu, C.-H. (2017). A novel automatic composition system using evolutionary algorithm and phrase imitation. *IEEE Systems Journal*, *11*(3), 1284–1295.

Walter, S., Mougeot, G., Sun, Y., Jiang, L., Chao, K.-M., & Cai, H. (2021). MidiPGAN: A progressive GAN approach to MIDI generation. In *2021 IEEE 24th international conference on computer supported cooperative work in design* (pp. 1166–1171). IEEE.

Wang, T., Liu, J., Jin, C., Li, J., & Ma, S. (2020). An intelligent music generation based on variational autoencoder. In *2020 International conference on culture-oriented science & technology* (pp. 394–398). IEEE.

Wang, J., Wang, X., & Cai, J. (2019). Jazz music generation based on grammar and lstm. In *2019 11th international conference on intelligent human-machine systems and cybernetics, vol. 1* (pp. 115–120). IEEE.

Wen, Y.-W., & Ting, C.-K. (2020). Composing bossa nova by evolutionary computation. In *2020 International joint conference on neural networks* (pp. 1–8). IEEE.

Williams, D., Kirke, A., Miranda, E., Daly, I., Hwang, F., Weaver, J., et al. (2017). Affective calibration of musical feature sets in an emotionally intelligent music composition system. *ACM Transactions on Applied Perception (TAP)*, *14*(3), 1–13.

Williams, D., & Lee, N. (2018). *Emotion in video game soundtracking*. Springer.

Wiriyachaiporn, P., Chanasit, K., Suchato, A., Punyabukkana, P., & Chuangsuwanich, E. (2018). Algorithmic music composition comparison. In *2018 15th International joint conference on computer science and software engineering* (pp. 1–6). IEEE.

Wu, J., Hu, C., Wang, Y., Hu, X., & Zhu, J. (2019). A hierarchical recurrent neural network for symbolic melody generation. *IEEE Transactions on Cybernetics*, *50*(6), 2749–2757.

Wu, J., Liu, X., Hu, X., & Zhu, J. (2020). PopMNet: Generating structured pop music melodies using neural networks. *Artificial Intelligence*, *286*, Article 103303.

Yang, L.-C., Chou, S.-Y., & Yang, Y.-H. (2017). Midinet: A convolutional generative adversarial network for symbolic-domain music generation. arXiv preprint arXiv:1703.10847.

Yang, L.-C., & Lerch, A. (2020). On the evaluation of generative models in music. *Neural Computing and Applications*, *32*(9), 4773–4784.

Yang, W., Sun, P., Zhang, Y., & Zhang, Y. (2019). CLSTMS: A combination of two LSTM models to generate chords accompaniment for symbolic melody. In *2019 International conference on high performance big data and intelligent systems* (pp. 176–180). IEEE.

Ycart, A., & Benetos, E. (2020). Learning and evaluation methodologies for polyphonic music sequence prediction with LSTMs. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *28*, 1328–1341.

Yeh, Y.-C., Hsiao, W.-Y., Fukayama, S., Kitahara, T., Genchel, B., Liu, H.-M., et al. (2021). Automatic melody harmonization with triad chords: A comparative study. *Journal of New Music Research*, *50*(1), 37–51.

Yu, Y., Srivastava, A., & Canales, S. (2021). Conditional lstm-gan for melody generation from lyrics. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, *17*(1), 1–20.

Zeng, Z., & Zhou, L. (2021). A memetic algorithm for Chinese traditional music composition. In *2021 6th International conference on intelligent computing and signal processing* (pp. 187–192). IEEE.

Zhao, K., Li, S., Cai, J., Wang, H., & Wang, J. (2019). An emotional symbolic music generation system based on lstm networks. In *2019 IEEE 3rd information technology, networking, electronic and automation control conference* (pp. 2039–2043). IEEE.