IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 30, 2021, accepted September 5, 2021, date of publication September 17, 2021,
date of current version September 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3113829

Monophonic Music Generation With a Given Emotion Using Conditional Variational Autoencoder

JACEK GREKOW AND TEODORA DIMITROVA-GREKOW
Faculty of Computer Science, Bialystok University of Technology, 15-351 Bialystok, Poland

Corresponding author: Jacek Grekow (j.grekow@pb.edu.pl)

This work was supported by the Ministry of Science and Higher Education through Bialystok University of Technology under Grant WZ/WI-IIT/2/2020.

ABSTRACT The rapid increase in the importance of human-machine interaction and the accelerating pace of life pose various challenges for the creators of digital environments. Continuous improvement of human-machine interaction requires precise modeling of the physical and emotional state of people. By implementing emotional intelligence in machines, robots are expected not only to recognize and track emotions when interacting with humans, but also to respond and behave appropriately. The machine should match its reaction to the mood of the user as precisely as possible. Music generation with a given emotion can be a good start to fulfilling such a requirement. This article presents the process of building a system generating music content of a specified emotion. As the emotion labels, four basic emotions: happy, angry, sad, relaxed, corresponding to the four quarters of Russell's model, were used. Conditional variational autoencoder using a recurrent neural network for sequence processing was used as a generative model. The obtained results in the form of the generated music examples with a specific emotion are convincing in their structure and sound. The generated examples were evaluated with two methods, in the first using metrics for comparison with the training set and in the second using expert annotation.

INDEX TERMS Generative models, music generation, music emotion, variational autoencoder.

I. INTRODUCTION

More and more devices and machines enter our everyday life. Nowadays, human-machine interaction can be encountered not only in industry. It started more than half a century ago with industrial robots [1]. Gradually, they were joined by increasingly complex and multifunctional information and vending machines, and today this interaction is almost everywhere, e.g. a great number of people are increasingly using e-assistants like Amazon Alexa[1] and Google Assistant.[2]

The importance of human-machine interaction on the one hand and customer expectations on the other set quality requirements for new machine generation. The continuously improving digital environment reflects the gradually progressing state of people. The implemented emotional intelligence machines and robots are expected not only to recognize and track emotions when interacting with humans but also to respond and behave appropriately to the actual human mood. One way to fulfill this requirement is through the appropriate creative behavior such as music generation with a given emotion.

The generation of content with a specific emotion [2] by intelligent machines is the next stage in the development of systems that deal with the emotions expressed by humans, its recognition and tracking. Expressing emotions by robots interacting with humans is quite an important issue if this interaction is to be successful. Generating music with a specific emotion is also part of this form of communication, as music transmits content in which emotions play a dominant role.

Deep learning techniques for music generation is a relatively new phenomenon [3] that enters the area of music composition, which is typically an area of human creativity and artistry. There are more and more music generating

The associate editor coordinating the review of this manuscript and approving it for publication was Luca Turchet.

[1] https://developer.amazon.com/en-IN/alexa

[2] https://assistant.google.com/

systems [4]–[6] that try to imitate human creativity, even compete with it, and learn from compositions created over the past centuries of human development. Music is an expression of human thoughts in the form of an organization of sounds in time. It can be compared to verbal expression, which is also spread over time in the form of words, creating sentences, conveying content and abstract concepts. Due to this similarity, also technological solutions to problems such as text generation and music generation have similar approaches.

Similarly to verbal expression, in addition to its content, music conveys emotions, which in music are evoked by musical elements spread over time. Depending on the changes over time in melody, timbre, dynamics, rhythm, or harmony, we can notice different emotions in music [7]. Song lyrics may also affect emotions [8], however in this study we focused on music files without lyrics.

The aim of this paper was to build a model generating monophonic music sequences with one of four basic emotions: happy, angry, sad, relaxed. The musical elements of the generated sequences should affect the emotion they contain. The model should recognize the emotion-affecting patterns in the training set and apply them to the generated examples.

The use of the four categories of emotions is of course a simplification of the possible emotional variations of the generated music, but it helps to start experiments on the problem of generating music with a specific emotion. This choice facilitates the labeling process with emotion music data as well as model building. A more advanced version of the music generation problem would be based on continuous values of emotion descriptions. A similar selection of four categories for generating emotional symbolic music was also selected in [9].

II. RELATED WORK

Studying human-machine interaction in industrial environments, a lot of research on the robot's perception and hardware for the recognition of human activities emerged [10], [11]. Being an important basis for human-machine interaction success, emotion recognition is also a popular field of exploration. Over the last decades, a wide range of deep learning techniques based on various models and databases [12], [13], research on feature extraction algorithms [14], [15], etc. have been conducted. However, the implementation of their results for human-machine interaction improvement has not yet been fully resolved.

In human-machine interaction, human emotions are seldom the central theme. Usually they are only a one-sided background. However, once the machine identifies and recognizes them, it would be quite nice if the machine responded appropriately. By combining emotion recognition with an appropriate machine reaction, the machine is expected to generate or create a response containing at least a partially human element in the audio or video domain. Williams *et al.* [16] demonstrated how real-time generated music can improve runners' performance considering an individual user's needs. In [17], Navarro-Cáceres *et al.* proposed melody generating under the supervision of the user. The user is supported by a mechanical device capturing the user's movements, and translates them into a melody.

After recognizing an emotion, the next step toward the development of the human-machine interaction, the intelligent machine should be able to generate at least an appropriate musical phrase. The resulting music could be set as a background or audio theme to support the ongoing interaction. In this way, additional psycho-physical comfort is provided without any additional commitment.

Division into categorical and dimensional approach can be found in papers devoted to music emotion recognition [18]. In the categorical approach, a number of emotional categories (adjectives) are used for labeling music excerpts [19]–[21]. In the dimensional approach, emotion is described using dimensional space, like the 2D model proposed by [22], where the dimensions are represented by arousal and valence [23]–[27]. In our work we will use categorical approach with four basic emotions: happy, angry, sad, relaxed.

A comprehensive overview of music generating systems such as recurrent neural networks, convolutional networks, generative adversarial networks, and autoencoders was presented by Briot *et al.* [3]. A functional taxonomy and state of the art in music generation systems includes work by Herremans *et al.* [28]. The main concepts, specific tasks, and open challenges of music generation were the topics of the work of Carnovalini and Rodà [29].

A review of systems for algorithmic composition with the intention of targeting specific emotional responses in the listener was presented by Williams *et al.* [30]. It described using sequencing, transformative and generative algorithms to create novel and emotionally satisfying music. Additionally, it also considered the use of various emotional models and musical features, which were employed by such systems. Scirea *et al.* [31], described a music generator for games, MetaCompose, which is based on evolutionary computation and creates music that can express different mood states in real-time. The authors evaluated the affective expression perceived in the music generated by the proposed system, based on human annotation. The idea of automatically generated music with a given sentiment (positive/negative) was presented in [32]. It developed the method used for generating textual product reviews with a sentiment [33] by using a single-layer multiplicative long short-term memory (mLSTM) network. The network is controlled by optimizing the weights of neurons found that are responsible for the sentiment signal. A variant of this network, where logistic regression uses the hidden states of the generative mLSTM to encode the labeled MIDI phrases, was used as a classifier of sentiment. The training dataset was extracted from video game soundtracks in MIDI format, a part of which was annotated according to a two-dimensional model that represents emotion using valence-arousal.

In [34], Hadjeres *et al.* proposed geodesic latent space regularization for the variational autoencoder, which enhances latent space navigation with the change of the attributes of the decoded sequences. The paper presents a music generation system using the proposed regulation that controls the number of notes generated by variations of a given monophonic melody. In [35], Valenti *et al.* presented the architecture for music generation that is based on an adversarial autoencoder. The conducted experiments show that the model can organize the latent space according to high-level genre information of the musical pieces, which allows you to modify the style of the input song. In [36], a generative VAE model to control tonal tension in generated music was used. For identifying latent tension variables, the labeled musical fragment positions in the latent space were calculated. The generated music is similar to the original music by keeping the rhythm and manipulating the pitches to match the tonal tension.

What distinguishes this work from others is that it uses a conditional variational autoencoder with the emotion parameter influencing the generated examples. The use of this model with four basic emotions has not yet been noted in the literature.

The rest of this paper is organized as follows. Section III describes the phases of building a music dataset and the emotion model used in the experiments. Section IV presents the representation of symbolic music, which is the data form used during the generated model training. Section V describes the concept of conditional variational autoencoder, its implementations, parameters, and training. Section VI presents the generated music samples as well as their evaluation. Finally, Section VII summarizes the main findings.

III. MUSIC DATASET

A. PREPARING OF SYMBOLIC MUSIC DATASET

The first phase of building a music generating system is building or selecting a database with musical compositions. In this study, the symbolic music library music21 [37] containing compositions by J.S. Bach was used. This collection mostly includes chorales (382) as well as several other compositions, 410 pieces in total. The full list of compositions in the MusicXML format is available in [38].

Due to the fact that the symbolic music library was to be annotated with emotion labels, the selection of the database was guided by the fact that the database should contain files with varying emotions. In [39], Dong *et al.* studied key mode distributions of different music datasets, among others (Lakh MIDI Dataset, Wikifonia Lead Sheet Dataset, Hymnal Dataset, J. S. Bach music21 Dataset). They found that key mode distributions (minor, major) in most databases were rather imbalanced, with the exception of the J. S. Bach music21 Dataset, where the occurrence of major compositions is equal to 56% in relation to the whole. A fairly even key mode distribution of compositions is important when creating a database in which emotions will be assessed, therefore the J. S. Bach music21 Dataset was selected as the starting database for building the training set. The database was accessed via the MusPy Toolkit [39] and imported into the MusPy format.

The music generation system created in this work should generate monophonic sequences, therefore the original J. S. Bach music21 Dataset underwent several transformations (Fig. 1). First, the tempo of all songs was standardized to 120 BPM. The note values in songs with a tempo other than 120 BPM were adjusted so that only the note lengths (sixteenths, eighth notes, quarter notes, half notes, whole notes) affected the tempo.

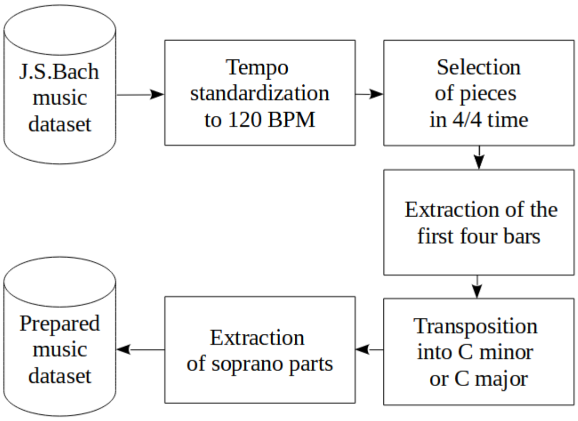

J.S.Bach music dataset
Tempo standardization to 120 BPM
Selection of pieces in 4/4 time
Extraction of the first four bars
Transposition into C minor or C major
Extraction of soprano parts
Prepared music dataset

FIGURE 1. Transformations of J. S. Bach music21 dataset.

Another transformation is the limitation of the music example length to four bars and the selection of pieces only in a 4/4 time signature, which prevail in the J. S. Bach music21 Dataset, but which resulted in a reduction in the number of examples in the dataset. Thus, the rhythmic structure of the examples was standardized, covering four bars with four quarter notes. The result was eight-second examples, each example having 16 beats at a tempo of 120 BPM.

Another transformation concerned the keys of the examples, which vary greatly in the J. S. Bach music21 Dataset. When generating simple musical sequences, distances between sounds and rhythmic values are important, the key does not play a significant role, and even examples in different keys could interfere with model training. All compositions were transposed into C minor or C major.

Our model is supposed to generate one-voice musical sequences, and therefore the next transformation concerned only the highest voice of the composition, the soprano part, which usually contains the main melody of the piece. After applying all the transformations, a unified set of 344 single-voice musical sequences was obtained, all examples of which have the same length (8 s), are in the key of C major or C minor, and are saved in the MIDI format.

B. DATASET ANNOTATION

During annotation of music samples, we used one of four basic emotions: happy, angry, sad, relaxed, which correspond to the four quarters of Russell's model (Fig. 2), which consists of two independent dimensions of arousal (vertical axis) and valence (horizontal axis). Happy, angry, sad, relaxed, these

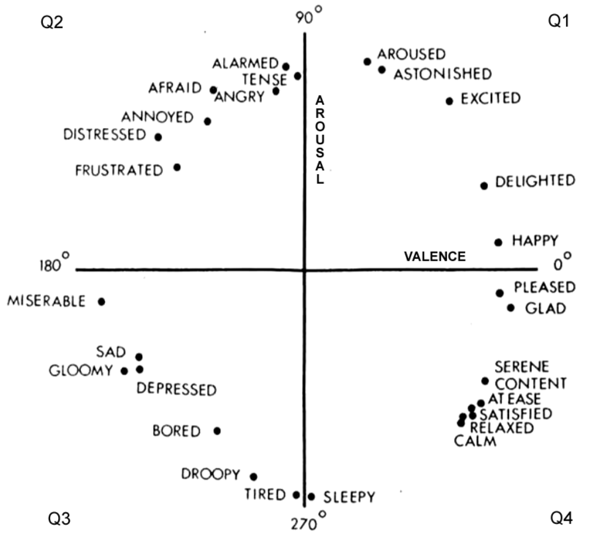


FIGURE 2. Russell's circumplex model [22].

are just labels representing the individual quarters of the emotion model. Under each label, there are secondary emotions from a given quarter of Russell's model, i.e. the happy label groups emotions with high arousal and high valence; angry, high arousal and low valence; sad, low arousal, low valence; and relaxed, low arousal, high valence. Similar divisions of emotions into categories were used in papers [19], [40], [41].

The annotated set of MIDI files was played with one volume and timbre (MIDI instrument: Grand Piano), these elements in our experiment will not affect the emotions. What affects the emotions of a music fragment is the musical content: sounds, their number, the pitch, rhythmic values, organization, minor/major scale [20], [42], [43].

The psychologist Gabrielsson in his work [44] made a distinction between emotion perception into perceived and felt (induced) emotions. In the case of the former, we can perceive emotional expression in music without necessarily being affected ourselves; while in the latter, we have an actual emotional response to the music. Perceived emotion is the emotion recognized in the music, and induced emotion is the emotion experienced by the listener. The music expert's task was to annotate the MIDI files with the perceived emotion.

Data annotation was done by three music experts with a university musical education. The musical education of the experts, people who deal with the creation and analysis of emotions in music on a daily basis, enables to trust the quality of their annotations. The musicians involved in the annotation are practitioners. They play in music bands, compose, give concerts, express emotions through music, i.e. they specialize not only in perceiving emotions but also in creating them, which makes them more competent in the subject of perceiving musical emotions than people who only listen to music.

Each music expert heard all the examples, 344 eight-second MIDI files, as a result of which each annotator was able to notice all the shades of emotions in the music, which is not always the case in databases with the emotions determined. This had a positive effect on the quality of the received data, which was emphasized by Aljanaki *et al.* [45]. The data collected from the three music experts was averaged. Considering the internal consistency of the collected data, Cronbachs α [46] obtained a value of 0.90. The amount of obtained examples is presented in Table 1. The collected data set with MIDI files annotated with four basic emotions can be found at link.[3]

TABLE 1. Amount of MIDI files annotated with 4 basic emotions.

Emotion	Abbreviation	Quarter in Russell's model	Arousal-Valence	Amount of examples
happy	e1	Q1	high-high	80
angry	e2	Q2	high-low	79
sad	e3	Q3	low-low	93
relaxed	e4	Q4	low-high	92

IV. REPRESENTATION OF SYMBOLIC MUSIC

Data from the MIDI files must be processed before being used to train the model to be understandable for the neural network. Since the music generation system will learn using monophonic melodies, all MIDI files from the dataset have been encoded into pitch-based representation using the MusPy Toolkit. The pitch-based representation represents music as a sequence of pitch, rest, and hold tokens. The output shape is $T \times 1$, where T is the number of time steps. The values in the sequence indicate whether the current time step is a pitch (0-127), a rest (128), or a hold (129). Hold tokens are used to hold the duration of a note when the note is longer than the selected resolution, in our case the resolution was sixteenth notes.

Details of the transformation are presented in Fig. 3. The first note, a quarter note with pitch E4, was coded with MIDI number 64, and therefore its length is four times the length of the sixteenth note; it was supplemented with three hold values (129). The next note (an eighth note E4) was coded similarly. An eighth note is two times longer than a sixteenth note and therefore was coded with two values: MIDI number 64 and hold value 129. The coding of subsequent notes followed the same rules.

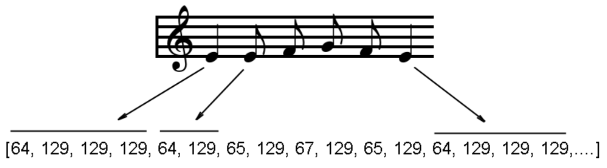


FIGURE 3. Example of creating pitch-based representation.

The length of each example from the dataset corresponds to four bars in a 4/4 time signature, which is four quarter notes per bar, making a total of 16 quarter notes. The shortest note value in the dataset is sixteenth notes, and therefore examples with sixteen notes were discretized. There are four

[3] https://github.com/grekowj/musgenvae

sixteenth notes for each quarter note, dividing the segment with the shortest note (sixteenth note) we get 64 time steps, $4(bar) \times 4(quarter\ \ note) \times 4(sixteenth\ \ note)$. Thus, each MIDI file from the dataset was encoded into a pitch-based representation with 64 time steps.

After processing the MIDI dataset, the number of different pitch notes was reduced to 29, which after adding rest and hold tokens gives a total of 31 different tokens in a sequence, which were additionally one-hot encoded. The shape of the target output tensor for one example was $64(time\ \ step) \times 31(token)$.

V. CONDITIONAL VAE

A generative model based on variational autoencoder (VAE) [47] was used to generate the musical sequences, which encodes the input data into latent space with Gaussian distribution and then decodes samples from the latent vector to a similar form as the input. The advantage of VAE is the ability to move in the continuous latent space of trained VAE, which allows to generate new musical sequences. In order to add the possibility of controlling the type of emotions in the generated musical sequences, the model was extended to conditional VAE (CVAE) [48]. What makes CVAE different from VAE is the addition of a condition, which in our case is an emotion label (Fig. 4). The condition is added on both the encoder and decoder inputs.

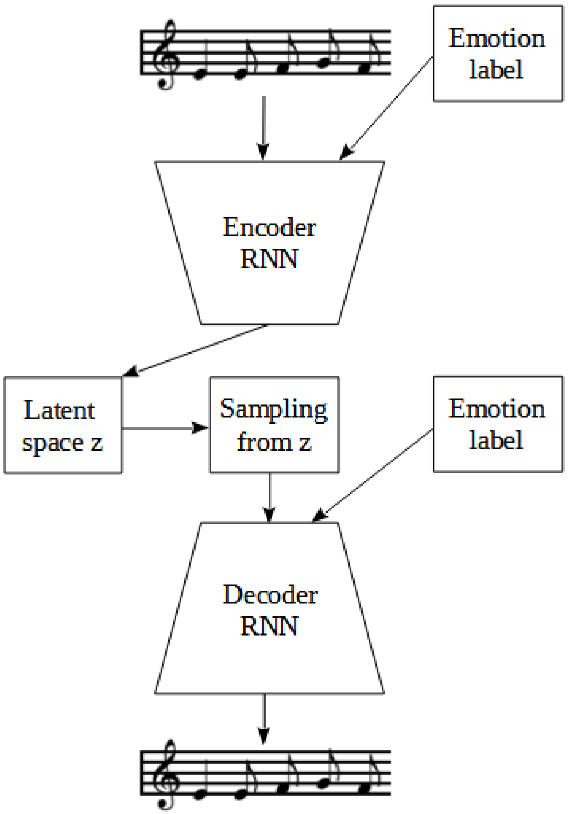


FIGURE 4. CVAE model structure.

A. IMPLEMENTATION OF GENERATIVE MODEL

For building implementation of the CVAE network and conducting the experiments, the Keras[4] deep learning library written in Python with Tensorflow[5] as backend was used. Figs. 5 and 6 show the encoder and decoder of CVAE, which were implemented using the recurrent neural network (RNN). CVAE allows to generate musical sequences with a specific emotion through random sampling from the latent space, which in our case has 20 dimensions.

On the first encoder input (Fig. 5), music sequences with 64 time steps and 31 unique one-hot encoded music pitch values are given. For faster RNN learning, the sequences are normalized (mean: 0.00, std: 1.00). On the second encoder input, one-hot encoded four emotion labels are given. Before concatenating two inputs, the dimension of labels is extended with a Dense layer and reshaped to the same size as the shape of the music sequences. The combined sequences are processed by 512 Gated Recurrent Units (GRU) [49], which make up RNN. The next two Dense layers reduce dimensionality and generate the mean and log variance. The last output layer of the encoder is a sampling of latent vector z.

On the first decoder input (Fig. 6), the samples of latent vector z from the encoder output are given. On the second decoder input, one-hot encoded four emotion labels are given, same as for the encoder. After combining, two inputs are used to layer RepeatVector to prepare the data size for the next layer which is RNN with 512 GRU. The last TimeDistributed layer allows to apply a Dense layer across the time steps of the music sequence.

The CVAE network consist of the encoder and the decoder joined together. The shape of the music sequences on the CVAE input and output is the same (None, 64, 31). The encoder takes input x, and estimates the mean μ, and the standard deviation σ, of the multivariate Gaussian distribution of latent vector z. The decoder takes samples from latent vector z to reconstruct the input on the output as $\tilde{x}$. The loss function is the sum of both the *Reconstruction Loss* ($\mathcal{L}_R$) and *Latent loss* ($\mathcal{L}_L$). *Reconstruction Loss* calculates the difference between input x and output $\tilde{x}$ using cross entropy. *Latent loss* is calculated using the Kullback-Leibler divergence, which calculates the distance between the target distribution (the Gaussian distribution) and the actual distribution in latent vector z:

$$\mathcal{L}_L = -\frac{1}{2}\sum_{i=1}^{K}(1 + \log\sigma_i^2 - \sigma_i^2 - \mu_i^2) \tag{1}$$

where K is the dimensionality of latent vector z, μ_i and σ_i are mean and standard deviation of i dimension of latent vector z.

B. TRAINING OF THE NETWORK

For our classification task, which is the prediction of one category (one pitch of note), the softmax function was used as

[4] https://keras.io
[5] https://www.tensorflow.org

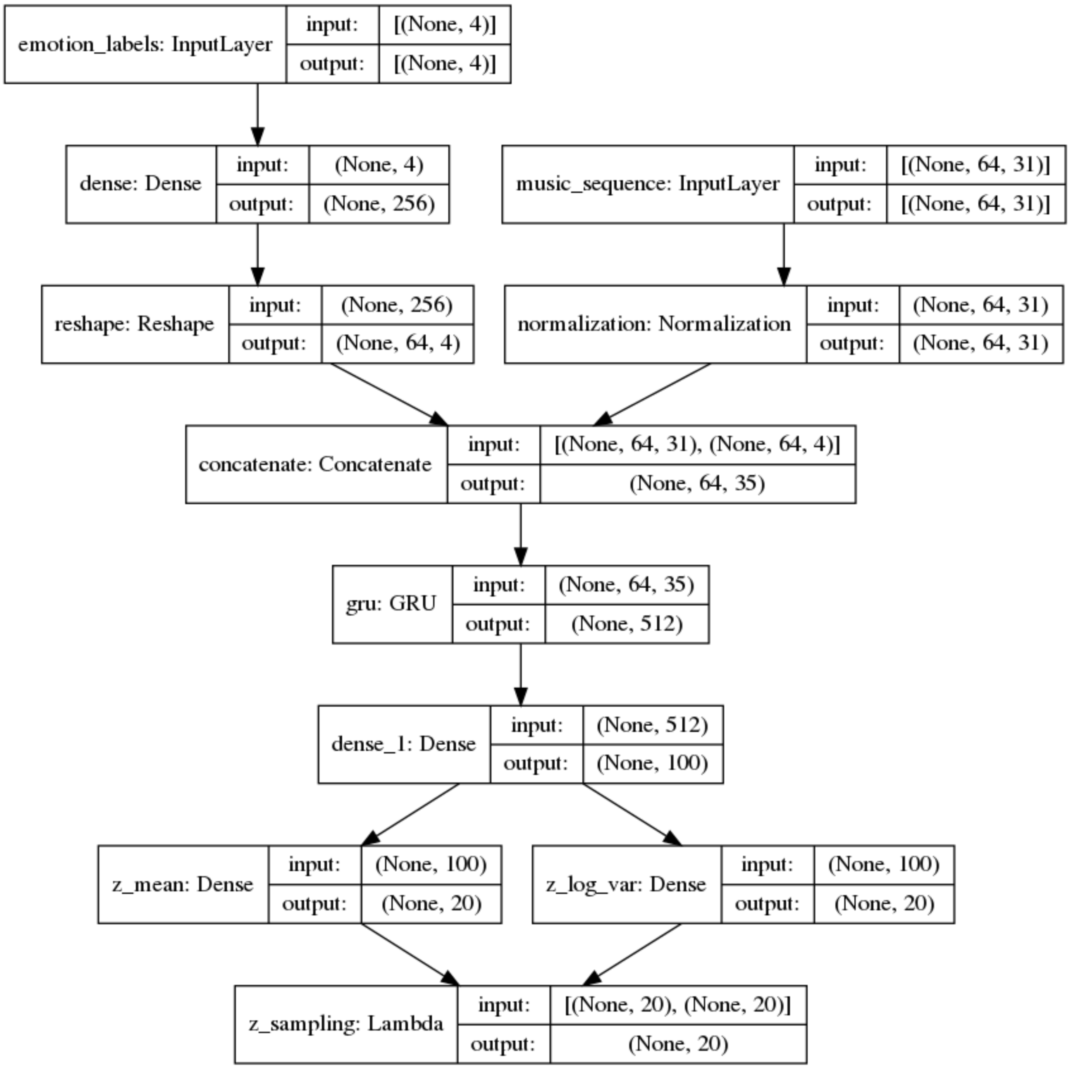


FIGURE 5. CVAE RNN encoder.

the activation for the last decoder layer. As a loss function to train the CVAE network, categorical crossentropy was used, which computes the crossentropy loss between the one-hot pitch values and predictions. A tanh activation function was used for GRU units.

A series of experiments were performed with and without standardization of the input data, the number of GRUs (64, 128, 248, 512), and with varying latent space size. Finally, a combination of 512 GRU and a latent space with dimension 20, and standardization of the input data were selected.

The CVAE was trained with RMSprop optimizer (lr = 0.001). The network was trained with 900 epochs and to avoid overfitting an early stopping strategy was used. The training process was stopped as soon as the loss did not improve any more for 50 epochs. The loss was evaluated on a validation set (20% of the training data).

CVAE+Dense was chosen as a baseline for comparing the results of the obtained models. It differed from CVAE+GRU in that a simple Dense layer in the encoder and decoder was used instead of the recurrent GRU layer. Table 2 presents the validation loss obtained during model building. The number in parentheses next to the model name indicates the number of units used. The best results are marked in bold. From the obtained results, we can see that models CVAE+GRU with more than 64 GRU units are superior to the baseline model (CVAE+Dense). We can see that the recurrent units in CVAE are better suited for encoding and decoding sequential data, which is of course well known. Testing how the use of the baseline model (CVAE+Dense) and the proposed model (CVAE+GRU) affects the obtained metrics for the generated

TABLE 2. Validation loss for the tested models.

Model	Loss
CVAE+Dense(512) - baseline model	0.47
CVAE+GRU(64)	0.50
CVAE+GRU(128)	0.46
CVAE+GRU(256)	0.38
CVAE+GRU(512)	**0.33**

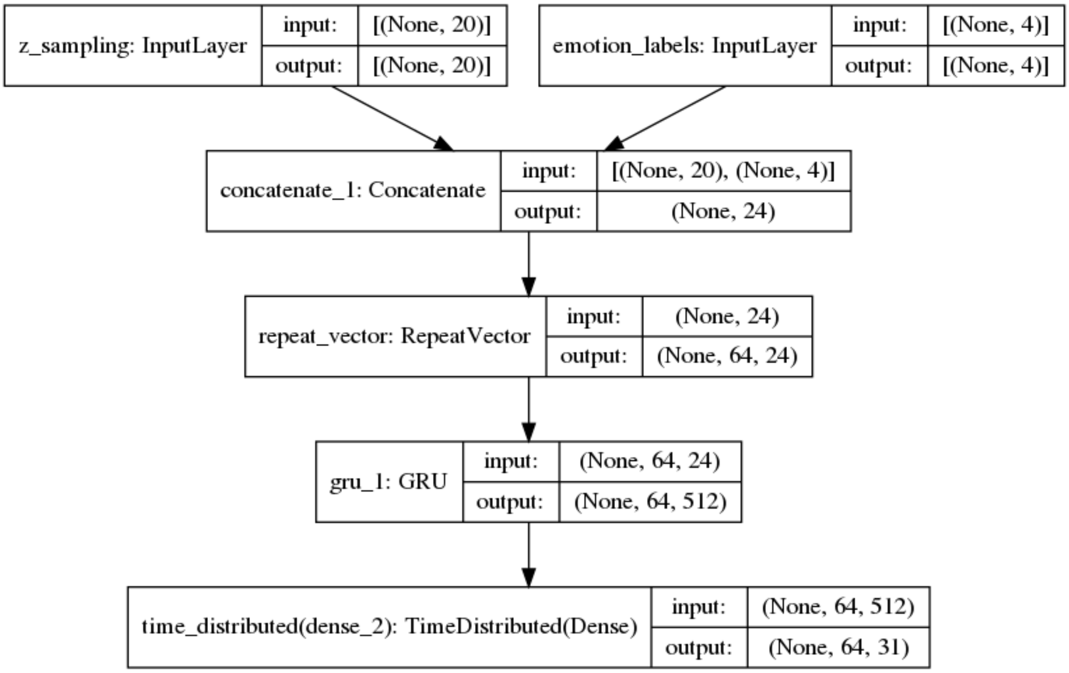


FIGURE 6. CVAE RNN decoder.

music depending on the type of emotion will be presented in Section VI-B.

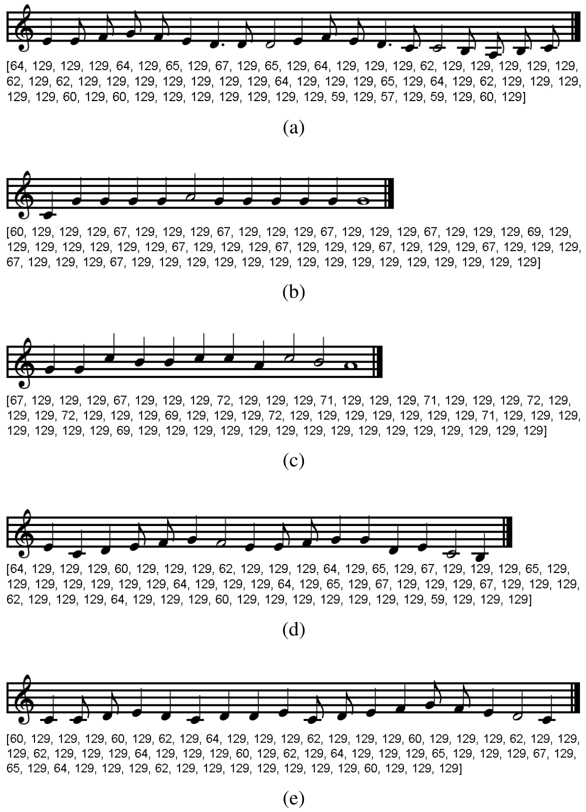
Fig. 7 presents the stages of CVAE training with the use of input and output data visualization. The verification of the degree of training is illustrated by the ability of the decoder to reproduce the input sequence (Fig. 7a) on the output (Figs. 7b, 7c, 7d, 7e). The presented notations were completed with sequences of numbers constituting the pitch representation of a 64-element sequence of a given musical example. It was noticed that in the initial stage of training (Fig. 7b) the sequence was shorter, monotonous, with no clear musical meaning. CVAE is not yet sufficiently trained and is unable to generate a sequence close to the input sequence. The next steps (Figs. 7c, 7d, 7e) show how the sequence obtained at the output of the autoencoder starts to resemble the input sequence.

Fig. 8 shows one view of the 20-dimensional latent space obtained during model training. New musical examples will be sampled and generated from the latent space. The points in latent space correspond to the training files, and the colors define the emotion assigned to them. We can see that the coordinate values of all points are distributed around mean value equal to 0. Different emotions are not grouped in one place, but spread throughout the entire latent space.

VI. RESULTS AND DISCUSSION

A. EXAMPLES OF GENERATED MUSIC SEQUENCES WITH PROVIDED EMOTION

A trained CVAE model was used to generate new music sequences with a specific emotion. The generation consisted of giving an emotion label and a random sample with a latent space size into the decoder input (Fig. 9).



[64, 129, 129, 129, 64, 129, 65, 129, 67, 129, 65, 129, 64, 129, 129, 129, 62, 129, 129, 129, 129, 129, 62, 129, 62, 129, 129, 129, 129, 129, 129, 129, 64, 129, 129, 129, 65, 129, 64, 129, 62, 129, 129, 129, 129, 129, 60, 129, 60, 129, 129, 129, 129, 129, 129, 129, 59, 129, 57, 129, 59, 129, 60, 129]

(a)

[60, 129, 129, 129, 67, 129, 129, 129, 67, 129, 129, 129, 67, 129, 129, 129, 67, 129, 129, 129, 69, 129, 129, 129, 129, 129, 129, 129, 67, 129, 129, 129, 67, 129, 129, 129, 67, 129, 129, 129, 67, 129, 129, 129, 67, 129, 129, 129, 67, 129, 129, 129, 129, 129, 129, 129, 129, 129, 129, 129, 129, 129, 129, 129]

(b)

[67, 129, 129, 129, 67, 129, 129, 129, 72, 129, 129, 129, 71, 129, 129, 129, 71, 129, 129, 129, 72, 129, 129, 129, 72, 129, 129, 129, 69, 129, 129, 129, 72, 129, 129, 129, 129, 129, 129, 129, 71, 129, 129, 129, 129, 129, 129, 129, 69, 129, 129, 129, 129, 129, 129, 129, 129, 129, 129, 129, 129, 129, 129, 129]

(c)

[64, 129, 129, 129, 60, 129, 129, 129, 62, 129, 129, 129, 64, 129, 65, 129, 67, 129, 129, 129, 65, 129, 129, 129, 129, 129, 129, 129, 64, 129, 129, 129, 64, 129, 65, 129, 67, 129, 129, 129, 67, 129, 129, 129, 62, 129, 129, 129, 64, 129, 129, 129, 60, 129, 129, 129, 129, 129, 129, 129, 59, 129, 129, 129]

(d)

[60, 129, 129, 129, 60, 129, 62, 129, 64, 129, 129, 129, 62, 129, 129, 129, 60, 129, 129, 129, 62, 129, 129, 129, 62, 129, 129, 129, 64, 129, 129, 129, 60, 129, 62, 129, 64, 129, 129, 129, 65, 129, 129, 129, 67, 129, 65, 129, 64, 129, 129, 129, 62, 129, 129, 129, 129, 129, 129, 129, 60, 129, 129, 129]

(e)

FIGURE 7. Stages of CVAE training over epochs, illustrated with (a) input example and output example during training with (b) 50, (c) 100, (d) 500 and (e) 700 epochs.

Fig. 10 shows the generated examples for each emotion. A set of generated MIDI examples can be found

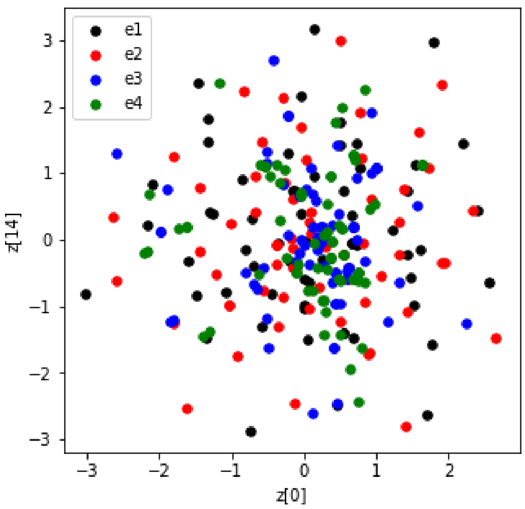


FIGURE 8. Latent space of CVAE.

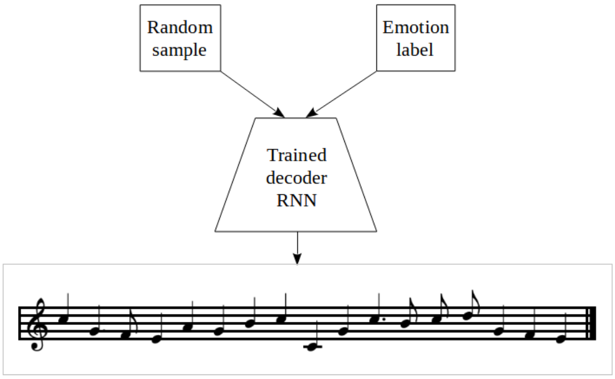


FIGURE 9. Generating new music sequences with a specific emotion.

at link.[6] Figs. 10a and 10b show the notes of examples with the emotions happy (e1) and angry (e2). In both examples, we note an increased amount of notes, indicating greater arousal and locating the emotions in the upper quadrants of Russell's emotion model. A much smaller number of notes is in Figs. 10c and 10d, examples with emotions sad (e3) and relaxed (e4).

We notice the minor scale in the examples of Figs. 10b and 10c, which places them on the negative part of the valence axis of Russell's model – emotions angry (e2) and sad (e3). In Figs. 10a and 10d we notice the sounds of the C major scale, which indicate positive emotions from Russell's model - happy (e1) and relaxed (e4).

B. EVALUATION OF RESULTS USING METRICS

To evaluate the generated music sequences, they were tested using the following metrics [5], [39], [50]:

- *pitch range* - defined as the difference between the highest and the lowest pitch;
- *n pitches used* - defined as the number of unique pitches used in a melody;
- *pitch in scale C major rate* - defined as the ratio of the number of notes in the C major scale to the total number of notes;
- *pitch in scale C minor rate* - defined as the ratio of the number of notes in the C minor scale to the total number of notes;

To test the statistical difference between the training data and the generated samples, a set of 20 musical sequences was generated for each of the four emotions (e1, e2, e3, e4) for a total of 80 examples. Four metrics were calculated for each generated example. The same metrics were also calculated for the training set. Comparing the distributions of the values of these metrics allowed us to assess whether the generated files have the specific emotions.

Table 3 presents the mean and standard deviation (σ) of the metrics obtained from the music generated with the proposed and baseline models, and from music used as a training set. Note that for *pitch range* and *n pitches used* the mean values are lower for the baseline model than for the proposed model, especially for emotions e1 and e2. The baseline model produces melodies with less differences between the highest and lowest tones and also with fewer unique pitches used in the melody. The mean and σ values obtained from the music generated with the proposed model are closer to the values obtained from the training set, especially when it comes to the metrics *pitch range* and *n pitches used*.

TABLE 3. Mean and standard deviation (σ) of the metrics obtained from the generated and training sets labeled with emotions e1-e4.

Metric	Emotion	Generated set proposed model Mean (σ)	Training set Mean (σ)	Generated set baseline model Mean (σ)
Pitch range	e1	8.90 (2.55)	9.25 (2.93)	6.90 (1.37)
	e2	9.25 (1.97)	9.01 (2.14)	6.60 (1.66)
	e3	4.95 (1.66)	6.19 (1.68)	5.25 (1.92)
	e4	7.30 (1.71)	7.44 (1.95)	6.05 (2.91)
N pitches used	e1	5.90 (0.99)	6.25 (1.35)	4.95 (1.12)
	e2	6.60 (1.11)	6.28 (1.31)	4.75 (0.89)
	e3	3.85 (0.91)	4.57 (0.85)	3.90 (0.70)
	e4	4.55 (0.86)	4.84 (0.91)	3.65 (0.96)
Pitch in scale C major rate	e1	0.96 (0.10)	0.97 (0.10)	0.99 (0.03
	e2	0.72 (0.16)	0.72 (0.13)	0.73 (0.08)
	e3	0.73 (0.22)	0.77 (0.13)	0.81 (0.08)
	e4	1.00 (0.00)	0.98 (0.10)	1.00 (0.00)
Pitch in scale C minor rate	e1	0.71 (0.12)	0.66 (0.12)	0.70 (0.14)
	e2	0.87 (0.17)	0.91 (0.15)	0.99 (0.03)
	e3	0.90 (0.16)	0.92 (0.15)	0.97 (0.06)
	e4	0.68 (0.14)	0.65 (0.15)	0.71 (0.09)

Distributions of the calculated metrics for the generated (proposed model) and the training set labeled by emotion are shown in Figs. 11, 12, 13 and 14. In Fig. 11 we can see that the *pitch range* is lower for emotions e3 and e4, both in the generated and in the training set. This particularly concerns the emotion sad (e3), which has the lowest values. Similar differences between sets e1, e2 and e3, e4 can be seen in Fig. 12, which presents the number of unique pitches used in a melody. The sequences with emotions happy (e1)

[6] https://github.com/grekowj/musgenvae

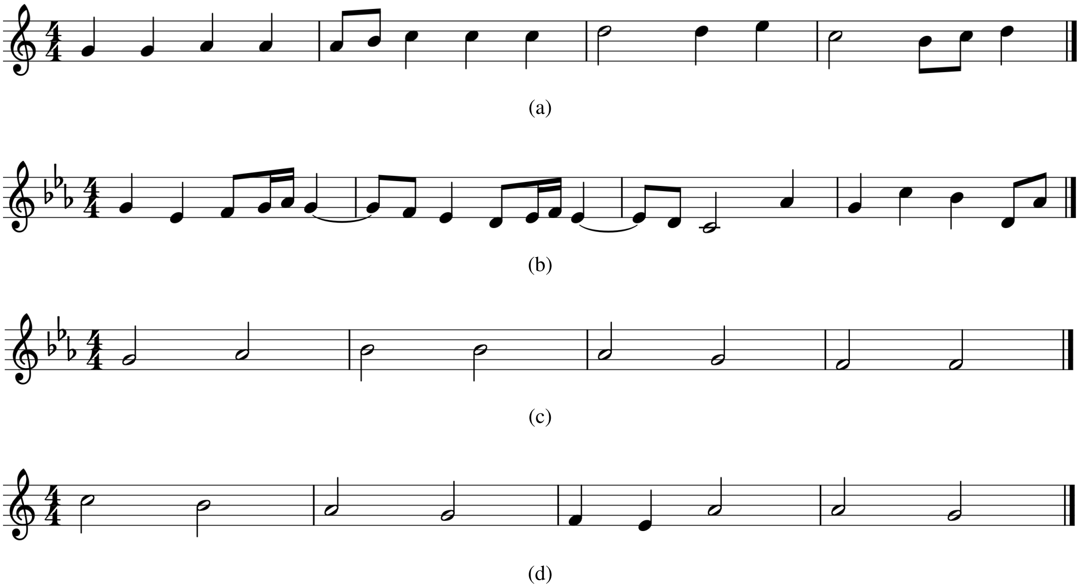


FIGURE 10. Examples of generated music sequences with emotions: (a) e1 - happy, (b) e2 - angry, (c) e3 - sad, and (d) e4 - relaxed.

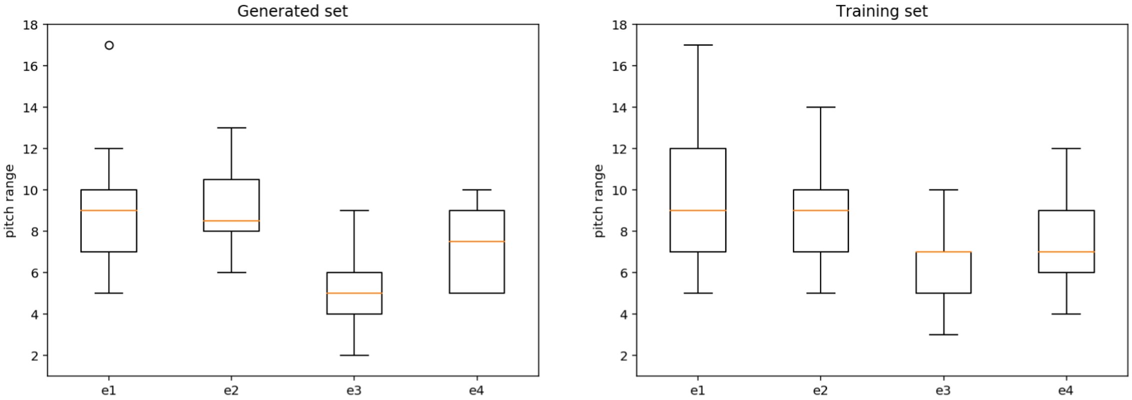


FIGURE 11. Box plots of the metric *pitch range* for the generated and training data sets labeled with emotions e1-e4.

and angry (e2) use more varying sounds than sequences with emotions sad (e3) and relaxed (e4). We could conclude that the *pitch range* and *n pitches used* metrics are suitable for distinguishing emotions on the arousal axis of Russell's emotion model.

Analyzing the box plots in Fig. 13, we can see that the musical sequences with emotions e1 and e4 use the C major scale sounds both in the generated and the training set. The use of C major scale sounds in files with emotions e2 and e4 is much smaller. We see an inverse distribution of values using the *pitch in scale C minor rate* metric (Fig. 14), where files with emotions e2 and e3 have greater values than e1 and e4. It could be concluded that the *pitch in scale C major rate* and *pitch in scale C minor rate* metrics are suitable for distinguishing emotions on the valence axis of Russell's emotion model.

To compare the statistics of the obtained value distributions for the individual metrics, the Kolmogorov-Smirnov (KS) statistic [51] was calculated to determine whether two distributions differ (Tables 4, 5, 6 and 7). The smaller the KS value, the more similar both distributions are, the samples are drawn from the same continuous distribution. The lowest values are in bold, which is the greatest similarity between sets.

KS values in one line in the table were computed by selecting a set with a specific emotion (e.g. e1) from the generated sets and compared with each set (e1-e4) from the training sets. This was repeated for the subsequent emotions (e2-e4), obtaining the next lines of the table.

Separate statistics are not always able to identify the most similar sets. To summarize the most similar sets, a win matrix was calculated based on the previous four Tables 4, 5, 6 and 7. Table 8 shows which of the training sets are closest to

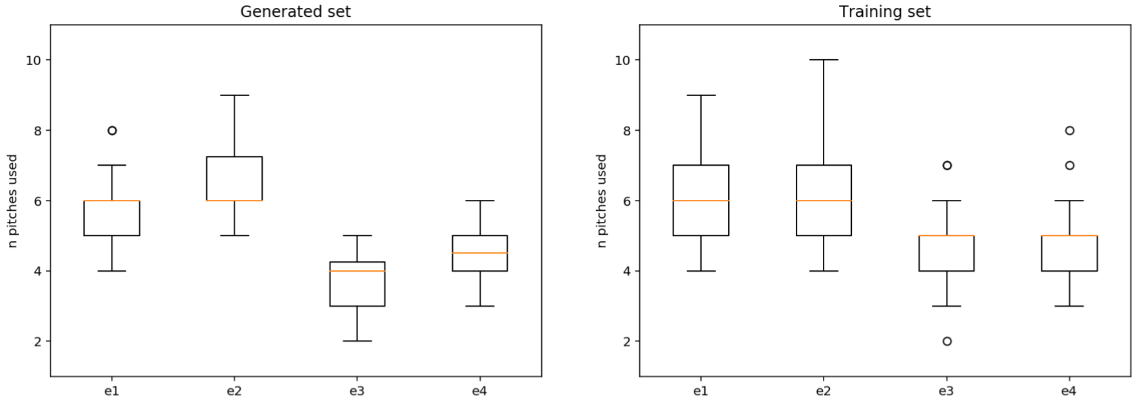


FIGURE 12. Box plots of the metric ***n pitches used*** for the generated and training data sets labeled with emotions e1-e4.

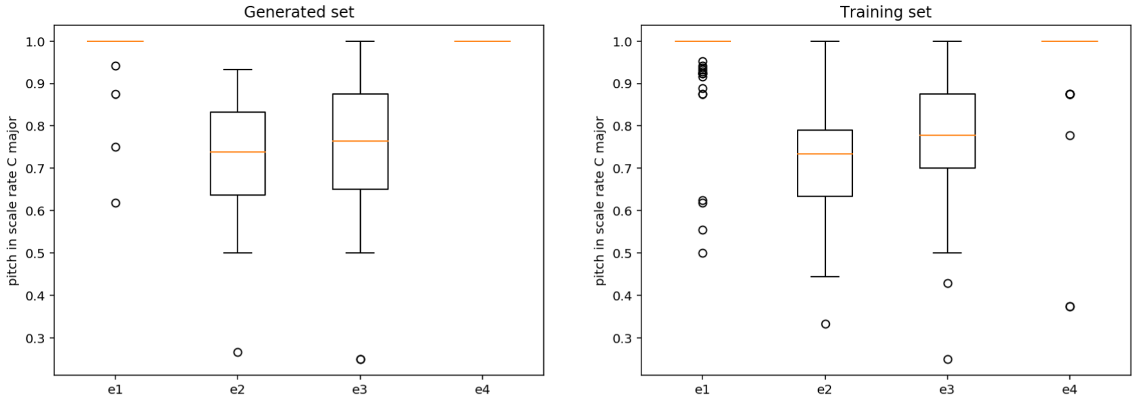


FIGURE 13. Box plots of the metric ***pitch in scale C major rate*** for the generated and training data sets labeled with emotions e1-e4.

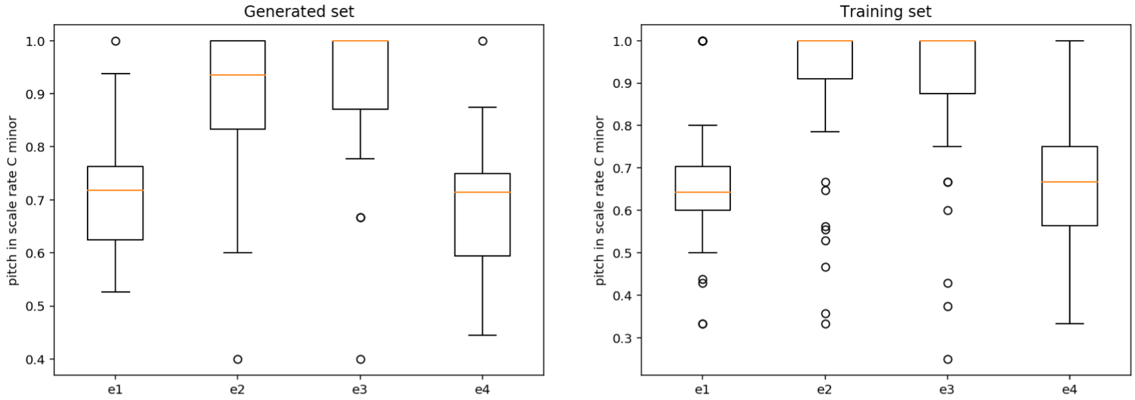


FIGURE 14. Box plots of the metric ***pitch in scale C minor rate*** for the generated and training data sets labeled with emotions e1-e4.

the generated sets with the given emotion. The most similar sets for each metric were recorded with an increment of 1 (or 0.5 in the case of two winners) in the matrix.

The table should be viewed horizontally as it indicates in how many cases the given generated set with a given emotion was similar to the training set with the same emotion. The sum of the horizontal lines in Table 8 is 4.0 as the similarities were counted for four metrics.

From the information generalized in Table 8, we can see that the diagonal values are the largest, i.e. that a given set generated with a given emotion is most similar to its counterpart from the training set. The diagonal values are

TABLE 4. Kolmogorov-Smirnov statistic between distributions of metric *pitch range* obtained from the generated and training sets labeled with emotions e1-e4.

		Training set			
		e1	e2	e3	e4
	e1	**0.15**	0.18	0.50	0.27
Generated	e2	0.20	**0.17**	0.62	0.43
set	e3	0.74	0.77	**0.47**	0.60
	e4	0.33	0.43	0.30	**0.10**

TABLE 5. Kolmogorov-Smirnov statistic between distributions of metric *n pitches used* obtained from the generated and training sets labeled with emotions e1-e4.

		Training set			
		e1	e2	e3	e4
	e1	0.24	**0.22**	0.56	0.42
Generated	e2	**0.18**	**0.18**	0.76	0.62
set	e3	0.67	0.70	0.37	**0.36**
	e4	0.52	0.52	0.12	**0.11**

TABLE 6. Kolmogorov-Smirnov statistic between distributions of metric *pitch in scale C major rate* obtained from the generated and training sets labeled with emotions e1-e4.

		Training set			
		e1	e2	e3	e4
	e1	**0.07**	0.79	0.79	0.09
Generated	e2	0.84	**0.16**	0.18	0.89
set	e3	0.76	0.32	**0.16**	0.74
	e4	0.20	0.96	0.94	**0.11**

TABLE 7. Kolmogorov-Smirnov statistic between distributions of metric *pitch in scale C minor rate* obtained from the generated and training sets labeled with emotions e1-e4.

		Training set			
		e1	e2	e3	e4
	e1	0.31	0.76	0.74	**0.22**
Generated	e2	0.75	**0.24**	**0.24**	0.64
set	e3	0.75	0.18	**0.07**	0.66
	e4	0.36	0.75	0.75	**0.21**

TABLE 8. Winners between the generated set by the proposed model and the training sets.

		Training set			
		e1	e2	e3	e4
	e1	**2.0**	1.0	0.0	1.0
Generated	e2	0.5	**3.0**	0.5	0.0
set	e3	0.0	0.0	**3.0**	1.0
	e4	0.0	0.0	0.0	**4.0**

also exactly the winners that indicate for the success of this method. All other elements of the table should be 0 in the ideal case. Their values, different from 0, show that the separately used metrics are not as sensitive as they should be and these elements could be interpreted as *errors*. Hence, we can make one more interesting observation, the metric-set used in this evaluation is much more sensitive to arousal than to valence. Such a conclusion can be made by properly summarizing the appropriate elements of the table. Thus we can see that the *errors* between the examined sets are greater between the right and left hemispheres of Russell's model, emotions e1-e2 and e3-e4, i.e. on the valence axis it is 2.5 vs. the 1.5 on the arousal one.

Additionally, KS statistic was calculated between the music generated by the baseline model (CVAE+Dense) and the training sets, which are presented in Table 9. We notice a clear deterioration in the similarities generated to the training sets with emotions from the upper quarters of Russell's model (e1, e2), when arousal is high (diagonal values: 0.0, 2.0). Comparing Table 9 with Table 8, we notice a smaller number of wins on the diagonal, which indicates that the music generated by the baseline model is less similar to the original songs than that generated by the proposed model. This is confirmed by the use of the CVAE+GRU model with recurrent units for sequence processing; it is better suited than CVAE+Dense. GRU provides better possibilities for coding and encoding sequences. The generated music using the proposed model according to the presented metrics is more similar to the music of J.S. Bach, which was used as a training set.

TABLE 9. Winners between the generated set by the baseline model and the training set.

		Training set			
		e1	e2	e3	e4
	e1	0.0	0.0	1.0	**3.0**
Generated	e2	0.0	**2.0**	**2.0**	0.0
set	e3	0.0	0.0	**3.0**	1.0
	e4	0.0	0.0	1.0	**3.0**

The use of CVAE+Dense as the baseline model showed that the non-recursive model is worse at generating music with e1 and e2 emotions than the CVAE+GRU model (Tables 8 and 9). A smaller deterioration in the quality of the generated music was noticed for emotions e3 and e4. The use of a simpler model generates worse music, particularly in the upper quarters of Russell's model. This proves that even the use of a non-recursive model as a baseline in our experiment made sense because it showed changes in the obtained metrics for different emotions, which is a very interesting observation.

C. EVALUATION OF RESULTS USING EXPERT OPINIONS

The same method that was used to label the training dataset (Section III-B), i.e. asking the same three music experts with a university music education to annotate the emotion of the generated music files, was used as a second method of evaluating the generated music sequences.

The evaluation concerned the same files as during the evaluation using the metrics (Section VI-B), i.e. each model was assessed using 80 music sequences, generated 20 for each of the four emotions (e1, e2, e3, e4). Assessment of the generated examples pertained two models: the baseline (CVAE+Dense) and the proposed model (CVAE+GRU). The task of each music expert was to listen and determine the emotions for all the examples generated by a given model, i.e. making 80 annotations for the evaluated model. The annotated examples were mixed up so that their order was not grouped by emotion. The obtained annotations from the music experts were averaged.

Expert annotations of the generated set by the baseline model (CVAE+Dense) and by the proposed model (CVAE+GRU) are presented in Table 10 and 11. The values in the rows refer to the generated files with the given emotion. Due to the fact that 20 files were generated for each emotion, the sum in the rows is also equal to 20. The values in the columns mean the number of files with a given emotion determined by the music experts.

TABLE 10. Expert annotations of the generated set by the baseline model.

		Expert opinions			
		e1	e2	e3	e4
Generated set	e1	**16**	4	0	0
	e2	0	**20**	0	0
	e3	0	0	**20**	0
	e4	0	0	8	**12**

TABLE 11. Expert annotations of the generated set by the proposed model.

		Expert opinions			
		e1	e2	e3	e4
Generated set	e1	**20**	0	0	0
	e2	2	**17**	1	0
	e3	0	0	**14**	6
	e4	0	0	0	**20**

The obtained annotations show that the created models generated music sequences with four categories of emotions. Comparing Table 10 with Table 11, we notice the higher accuracy (89%) of the generated examples by the proposed model relative to the baseline model (85%).

An interesting observation is that the more complex model, which is the proposed model (CVAE+GRU), is better at generating files with positive emotions (e1 - accuracy 100%; e4 - accuracy 100%) and slightly worse at generating files with negative emotions (e2 - accuracy 85%; e3 - accuracy 70%).

In the case of files generated by the baseline model (CVAE+Dense), we notice a deterioration in the generation of files with positive emotions (e1 - 80%, e4 - 60%), i.e. the opposite situation than in the case of the proposed model (greater accuracy for emotions e2 and e3, and worse for e1 and e4). The music experts expressed the opinion that the melodies generated by the simpler model were often underdeveloped, chaotic with high arousal, or monotonous with low arousal, which shifted the emotions towards the negative, the left hemisphere of Russell's model. Also, in the case of the baseline model, the smaller similarities of the generated examples to the original melodies was reflected in the metrics in Section VI-B (Table 9).

We noticed the rule that in both models we have errors mainly between emotions e1 and e2 or between e3 and e4, i.e. on the valence axis of Russell's model. This confirms that assessing and generating music with emotions on the valence axis is more difficult compared with the arousal axis, where there are almost no errors.

In summary, although the annotations showed that the created models generate music sequences with a given emotion with an accuracy above 85%, the mere determination of the emotions of the generated music files by music experts did not give a definite answer which of the tested models is better – the difference in accuracy is only four percentage points. After conducting both evaluations (using metrics and expert opinions), we can see that using additional objective metrics to evaluate the model (Section VI-B) is helpful in this case. In the future, additional parameters for the expert assessment of the generated music sequences could be used, such as rhythm, melody, and musical structure.

VII. CONCLUSION

More and more different kinds of machines and devices are entering our daily life. Robots, machines, and even objects called things offer services and information. In order to improve their interaction and collaboration with the user, in-machine feedback is necessary. The machine perception, concentrated on the analysis of a human's orders but also emotional state, must cause an appropriate quasi-human reaction. Therefore, studies of music generation with a specific emotion are reasonable and current trends.

This article presents the stages of creating a system generating monophonic musical sequences with one of four basic emotions. The generated examples based on random samples from latent space resemble real musical sequences and, additionally, we notice the appropriate emotions in them. A trained model recognizes the patterns influencing emotions in the training set and is able to transfer them to the generated examples.

The evaluation showed that the generated music examples are similar to the training set. Due to the random element, the generated examples are slightly different than in the training set, but their emotional characteristics are similar to the training data.

The limitations of this study include the emotional model we adopt, the musical area used in the training set, and the length of the monophonic pieces. All of these result from the initial stage of our research and were intentionally accepted as a compromise for this pilot study. The emotional model we apply considers just the four main emotional groups from Russell's model.

Thanks to such a system, in any human-machine interaction, a robot would be able to create a varied bunch of suitable and well corresponding to the current human mood melodies. Sensing in meaning ''detecting and tracking'' on the one hand and proper acoustical response of the machine on the other complement the human-machine collaboration making it a bit more human. The system could assist a composer in finding new themes with a specific emotion and could also be used to generate musical sequences in computer games depending on the emotional context, or the background music in shops. Another potential application of the system is music therapy, where the generated melodies with a specific emotion could be used to change or enhance the emotional state of the patient.

In the future, the generating system should be broadened to the possibility of working with polyphonic, four-voice music. Also, the use of emotion descriptions using continuous values, arousal or valence from Russell's model, would be a continuation of this work.

REFERENCES

[1] M. Hägele, K. Nilsson, J. N. Pires, and R. Bischoff, *Industrial Robotics*. Cham, Switzerland: Springer, 2016, pp. 1385–1422, doi: 10.1007/978-3-319-32552-1_54.

[2] S. Ji, J. Luo, and X. Yang, "A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions," 2020, *arXiv:2011.06801*. [Online]. Available: http://arxiv.org/abs/2011.06801

[3] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, *Deep Learning Techniques for Music Generation*. Cham, Switzerland: Springer, 2020.

[4] G. Hadjeres, F. Pachet, and F. Nielsen, "DeepBach: A steerable model for Bach chorales generation," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 1362–1371.

[5] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, Sep. 2018, pp. 1–13.

[6] R. Madhok, S. Goel, and S. Garg, "SentiMozart: Music generation based on emotions," in *Proc. 10th Int. Conf. Agents Artif. Intell.*, 2018, pp. 501–506.

[7] C. C. Pratt, *Music as the Language of Emotion*. Washington, DC, USA: U.S. Government Printing Office: The Library of Congress, 1950.

[8] Y. E. Kim, E. M. Schmidt, R. Migneco, B. G. Morton, P. Richardson, J. J. Scott, J. A. Speck, and D. Turnbull, "State of the art report: Music emotion recognition: A state of the art review," in *Proc. 11th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, Utrecht, The Netherlands, 2010, pp. 255–266.

[9] K. Zhao, S. Li, J. Cai, H. Wang, and J. Wang, "An emotional symbolic music generation system based on LSTM networks," in *Proc. IEEE 3rd Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, Mar. 2019, pp. 2039–2043.

[10] Q. M. Rahman, P. Corke, and F. Dayoub, "Run-time monitoring of machine learning for robotic perception: A survey of emerging trends," *IEEE Access*, vol. 9, pp. 20067–20075, 2021.

[11] E. Coronado, D. Deuff, P. Carreno-Medrano, L. Tian, D. Kulic, S. Sumartojo, F. Mastrogiovanni, and G. Venture, "Towards a modular and distributed end-user development framework for human-robot interaction," *IEEE Access*, vol. 9, pp. 12675–12692, 2021.

[12] A. Kurobe, Y. Nakajima, K. Kitani, and H. Saito, "Audio-visual self-supervised terrain type recognition for ground mobile platforms," *IEEE Access*, vol. 9, pp. 29970–29979, 2021.

[13] M. Gui and X. Xu, "Technology forecasting using deep learning neural network: Taking the case of robotics," *IEEE Access*, vol. 9, pp. 53306–53316, 2021.

[14] M. Khateeb, S. M. Anwar, and M. Alnowami, "Multi-domain feature fusion for emotion classification using DEAP dataset," *IEEE Access*, vol. 9, pp. 12134–12142, 2021.

[15] T. Dimitrova-Grekow and P. Konopko, "New parameters for improving emotion recognition in human voice," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 4205–4210.

[16] D. Williams, B. Fazenda, V. Williamson, and G. Fazekas, "On performance and perceived effort in trail runners using sensor control to generate biosynchronous music," *Sensors*, vol. 20, no. 16, p. 4528, Aug. 2020.

[17] M. Navarro-Cáceres, W. Hashimoto, S. Rodríguez-González, B. Pérez-Lancho, and J. Corchado, "Sensoring a generative system to create user-controlled melodies," *Sensors*, vol. 18, no. 10, p. 3201, Sep. 2018.

[18] Y.-H. Yang and H. Chen, "Machine recognition of music emotion: A review," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, pp. 1–30, May 2012.

[19] L. Lu, D. Liu, and H.-J. Zhang, "Automatic mood detection and tracking of music audio signals," *IEEE Trans. Audio, Speech, Language Process.*, vol. 14, no. 1, pp. 5–18, Jan. 2006.

[20] J. Grekow, "Audio features dedicated to the detection of four basic emotions," in *Proc. IFIP Int. Conf. Comput. Inf. Syst. Ind. Manage.*, Warsaw, Poland, Sep. 2015, pp. 583–591.

[21] B. G. Patra, D. Das, and S. Bandyopadhyay, "Labeling data and developing supervised framework for Hindi music mood analysis," *J. Intell. Inf. Syst.*, vol. 48, no. 3, pp. 633–651, Jun. 2017.

[22] J. A. Russell, "A circumplex model of affect," *J. Personality Social Psychol.*, vol. 39, no. 6, pp. 1161–1178, Dec. 1980.

[23] F. Weninger, F. Eyben, and B. Schuller, "On-line continuous-time music mood regression with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 5412–5416.

[24] E. Coutinho, G. Trigeorgis, S. Zafeiriou, and B. Schuller, "Automatically estimating emotion in music with deep long-short term memory recurrent neural networks," in *Proc. MediaEval Workshop*, Wurzen, Germany, Sep. 2015, pp. 1–3.

[25] J. Grekow, "Music emotion maps in arousal-valence space," in *Proc. 15th IFIP Int. Conf. Comput. Inf. Syst. Ind. Manage., (CISIM)*, Vilnius, Lithuania, Sep. 2016, pp. 697–706.

[26] R. Delbouys, R. Hennequin, F. Piccoli, J. Royo-Letelier, and M. Moussallam, "Music mood detection based on audio and lyrics with deep neural net," in *Proc. 19th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, Paris, France, Sep. 2018, pp. 370–375.

[27] J. Grekow, "Musical performance analysis in terms of emotions it evokes," *J. Intell. Inf. Syst.*, vol. 51, no. 2, pp. 415–437, Oct. 2018.

[28] D. Herremans, C.-H. Chuan, and E. Chew, "A functional taxonomy of music generation systems," *ACM Comput. Surv.*, vol. 50, no. 5, pp. 1–30, Nov. 2017.

[29] F. Carnovalini and A. Rodà, "Computational creativity and music generation systems: An introduction to the state of the art," *Frontiers Artif. Intell.*, vol. 3, p. 14, Apr. 2020.

[30] D. Williams, A. Kirke, E. R. Miranda, E. Roesch, I. Daly, and S. Nasuto, "Investigating affect in algorithmic composition systems," *Psychol. Music*, vol. 43, no. 6, pp. 831–854, Nov. 2015.

[31] M. Scirea, P. Eklund, J. Togelius, and S. Risi, "Can you feel it?: Evaluation of affective expression in music generated by MetaCompose," in *Proc. Genetic Evol. Comput. Conf.*, Jul. 2017, pp. 211–218.

[32] L. Ferreira and J. Whitehead, "Learning to generate music with sentiment," in *Proc. 20th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, Delft, The Netherlands, 2019, pp. 384–390.

[33] A. Radford, R. Jozefowicz, and I. Sutskever, "Learning to generate reviews and discovering sentiment," 2017, *arXiv:1704.01444*. [Online]. Available: http://arxiv.org/abs/1704.01444

[34] G. Hadjeres, F. Nielsen, and F. Pachet, "GLSR-VAE: Geodesic latent space regularization for variational autoencoder architectures," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2017, pp. 1–7.

[35] A. Valenti, A. Carta, and D. Bacciu, "Learning style-aware symbolic music representations by adversarial autoencoders," in *Proc. 24th Eur. Conf. Artif. Intell. (ECAI)*, Santiago de Compostela, Spain, Sep. 2020, pp. 1563–1570.

[36] R. Guo, I. Simpson, T. Magnusson, C. Kiefer, and D. Herremans, "A variational autoencoder for music generation controlled by tonal tension," in *Proc. Joint Conf. AI Music Creativ. (CSMC+MuMe)*, Oct. 2020, pp. 1–12. [Online]. Available: https://arxiv.org/abs/2010.06230

[37] M. Cuthbert and C. Ariza, "Music21: A toolkit for computer-aided musicology and symbolic music data," in *Proc. 11th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, Utrecht, The Netherlands, Aug. 2010, pp. 637–642.

[38] *List of Works in the Music21 Corpus*. Accessed: Jun. 30, 2021. [Online]. Available: https://web.mit.edu/music21/doc/about/referenceCorpus.html

[39] H.-W. Dong, K. Chen, J. McAuley, and T. Berg-Kirkpatrick, "MusPY: A toolkit for symbolic music generation," in *Proc. 21st Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, Oct. 2020, pp. 1–8.

[40] X. Wang, C. Xiaoou, D. Yang, and Y. Wu, "Music emotion classification of Chinese songs based on lyrics using TF*IDF and rhyme," in *Proc. 12th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, Jan. 2011, pp. 765–770.

[41] J. Grekow, "Human annotation," in *From Content-based Music Emotion Recognition to Emotion Maps of Musical Pieces*. Cham, Switzerland: Springer, 2018, pp. 13–24.

[42] A. P. Oliveira and A. Cardoso, "Towards affective-psychophysiological foundations for music production," in *Affective Computing and Intelligent Interaction*. Berlin, Germany: Springer, 2007, pp. 511–522.

[43] J. Grekow, "Audio features dedicated to the detection of arousal and valence in music recordings," in *Proc. IEEE Int. Conf. Innov. Intell. Syst. Appl. (INISTA)*, Jul. 2017, pp. 40–44.

[44] A. Gabrielsson, "Emotion perceived and emotion felt: Same or different?" *Musicae Scientiae*, vol. 5, no. 1, pp. 123–147, Sep. 2001.

[45] A. Aljanaki, Y. H. Yang, and M. Soleymani, "Developing a benchmark for emotional analysis of music," *PLoS ONE*, vol. 12, no. 3, pp. 1–22, 2017.

[46] L. J. Cronbach, "Coefficient alpha and the internal structure of tests," *Psychometrika*, vol. 16, no. 3, pp. 297–334, Sep. 1951.

[47] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. 2nd Int. Conf. Learn. Represent. (ICLR)*, Banff, AB, Canada, Apr. 2014, pp. 1–15.

[48] K. Sohn, X. Yan, and H. Lee, "Learning structured output representation using deep conditional generative models," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2015, pp. 3483–3491.

[49] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1–15.

[50] H.-W. Dong, W.-Y. Hsiao, and Y.-H. Yang, "Pypianoroll: Open source Python package for handling multitrack pianorolls," in *Proc. 19th Int. Soc. Music Inf. Retr. Conf. Late-Breaking Demos (ISMIR)*, Paris, France, Sep. 2018, pp. 101–108.

[51] J. L. Hodges, "The significance probability of the Smirnov two-sample test," *Arkiv Matematik*, vol. 3, no. 5, pp. 469–486, Jan. 1958.

JACEK GREKOW received the M.S. degree in computer systems from the Technical University, Sofia, Bulgaria, in 1994, the B.S. degree in music from Vienna Conservatoire (Konservatorium der Stadt Wien), Austria, in 1996, the M.S. degree in music from the Department of Instrumental and Educational Studies, Fryderyk Chopin University of Music, Warsaw, Poland, in 2006, and the Ph.D. degree in computer sciences from the Faculty of Information Technology, Polish-Japanese Academy of Information Technology, Warsaw, in 2009. He is currently an Associate Professor with the Faculty of Computer Science, Bialystok University of Technology, Poland. His research interests include the data mining, music emotion recognition, music generation, and deep learning.

TEODORA DIMITROVA-GREKOW received the Ph.D. degree in electronics and automation from Vienna University of Technology, Austria, in 1997. She is currently an Assistant Professor with the Department of Computer Science, Bialystok University of Technology, Poland. Her research interests include robotic navigation and sensoric, analysis and signal processing, and speech emotion recognition.