A Bassline Generation System Based on Sequence-to-Sequence Learning

Behzad Haki
Music Technology Group
Universitat Pompeu Fabra
Roc Boronat, 138
08018 Barcelona, Spain
behzadhaki88@gmail.com

Sergi Jorda
Music Technology Group
Universitat Pompeu Fabra
Roc Boronat, 138
08018 Barcelona, Spain
sergi.jorda@upf.edu

ABSTRACT

This paper presents a detailed explanation of a system generating basslines that are stylistically and rhythmically interlocked with a provided audio drum loop. The proposed system is based on a natural language processing technique: word-based sequence-to-sequence learning. The word-based sequence-to-sequence learning method proposed in this paper is comprised of recurrent neural networks composed of LSTM units. The novelty of the proposed method lies in the fact that the system is not reliant on a voice-by-voice transcription of drums; instead, in this method, a drum *representation* is used as an input sequence from which a translated bassline is obtained at the output. The drum representation consists of fixed size sequences of onsets detected from a 2-bar audio drum loop in eight different frequency bands. The basslines generated by this method consist of pitched notes with different duration. The proposed system was trained on two distinct datasets compiled for this project by the authors. Each dataset contains a variety of 2-bar drum loops with annotated basslines from two different styles of dance music: House and Soca. A listening experiment designed based on the system revealed that the proposed system is capable of generating basslines that are interesting and are well rhythmically interlocked with the drum loops from which they were generated.

Author Keywords

NIME, Generative music, LSTM, word-based sequence-to-sequence learning, EDM

CCS Concepts

•**Applied computing** → **Sound and music computing;** •**Information systems** → **Natural language generation;** *Music retrieval;*

1. INTRODUCTION

In electronic dance music, many producers do not have a traditional compositional approach to creating musical pieces. For these producers, unpredictivity may be a source of creativity. Moreover, many producers are not trained musicians in the sense that they have limited theoretical music training; hence, compositional tools may allow them to commonly (while not necessarily) go beyond their limited and possibly cliche compositional approaches.

While unpredictivity can improve the aesthetic quality of a piece of work, a great deal of dance music producers may favour creating pieces that are within a specific style. As a result, the motivation behind this work was to provide a method that would enable development of tools assisting users to improve the compositional aspects of their work creatively. Given the prominence of basslines in a variety of dance music styles, we specifically focused our work on the generation of style-specific basslines.

An automatically generated bassline needs to rhythmically work well with the other components of a track. Given that the rhythmic structure of a danceable track is highly correlated with the underlying drum patterns, the bassline needs to be well interlocked with the percussive instruments. In dance music, the percussive patterns are either programmed, performed live or sampled. While in some cases, a symbolic representation of the drum pattern can be available, in most cases only an audio recording of the percussive instruments is available. Hence, practical tools that require rhythmic information should be reliant on audio signals (i.e., they should be capable of extracting rhythmic information from audio recordings).

The primary objective of this work was to develop a system that would generate a style-specific bassline based on a provided "audio" drum loop. As explained in the previous section, the dependency of the system on audio drum loops would make the system more practical. While there are methods to automatically transcribe the drums in a percussive recording, the state-of-the-art methods in this topic still do not yield acceptable results. Also, given the scope of the current work, we could not afford to implement these methods. As a result, another objective of this work was to come up with a meaningful alternative "representation" of the drums (rather than an instrument by instrument transcription) using which the system would generate a bassline. Moreover, to be able to generate basslines within a style, we aimed to focus our work on "popular" tracks that supposedly define a particular style of dance music. Given a lack of such datasets, we aimed to create datasets containing popular tracks within a specific style.

2. RELATED WORK

Algorithmic Composition refers to "using some formal process to make music with minimal human intervention" [1] [13]. With the advent of computers, some composers have used computers to compose music using algorithmic methods. The earliest example of such works is the *Illiac Suite* by Lejaren Hiller and Leonard Isaacson. The entirety of this piece was composed using a high-speed digital computer in 1955-56 at the University of Illinois. To com-

NIME'19, June 3-6, 2019, Federal University of Rio Grande do Sul, Porto Alegre, Brazil.

pose this piece, Hiller and Isaacson created a computer program which would generate musical contents which would be modified and selected based on pre-defined functions and rules [1] [13]. In another approach using computers, Iannis Xenakis, created a program which would compose a piece based on a probability of note distributions provided to the program. In short, earliest Algorithmic Music Composition (AMC) approaches using computers were based on stochastic or rule-based systems [13].

A relatively recent approach to computer-based AMC is Artifical Intelligence (AI) systems. AI systems may be considered rule-based systems with the distinction that these systems are capable of "defining their own grammar" [13]. AI systems are commonly developed using Machine Learning (ML) techniques. In [7] [3], relevant ML techniques for automatic music composition have been comprehensively reviewed. Based on this review, earlier ML-based techniques were mostly based on Markov chains and genetic algorithms while the most recent ones are based on Neural Networks (NN).

The earliest AMC methods relevant to dance music are based on Markov models. A Markov model is a stochastic model predicting the probability of the occurrence of an event given the previous occurrences leading up to that event. In musical terms, Markov models are quite useful in learning **short-term** temporal structures. Patchet's *Continuator* is an example of a Markov based AMC system [14] [15]. While the above works were not primarily focused on dance music, their capability to generate stylistically consistent content laid the ground for later AMC researches focused solely on dance music such as *"GESMI: Generative Electronica Statistical Modeling Instrument"* [6] and *"GEDMAS: Generative Electronic Dance Music Algorithmic System"* [2], developed by Eigenfeldt et al. [5].

3. TEXT-BASED LSTM NETWORKS

Music can be represented as a sequence of events. Understanding the conditional probabilities between these events allows for generating new musical content. However, fully understanding the conditional probabilities between these events is a complicated task. In the recent years, many machine learning techniques have been employed to understand the relationship between different musical events, and hence, generating new musical content based on the learned information. Specifically, a number of researchers have investigated the applicability of some neural network based natural language processing (NLP) techniques to the field of AMC.

Choi et. al. [4] investigated the effectiveness of character or word based recurrent neural networks (Char-RNN / Word-RNN) in generating chord progressions or multi-voice drum patterns. In their method, the chord progressions and the drum patterns are represented as text. The textual representation is then used to train two proposed Char-RNN and Word-RNN models. The authors realized that both architectures are capable of learning (and hence generating) chord progressions. However, the Char-RNN model, as opposed to the Word-RNN model, failed to learn any meaningful information about drum patterns.

In [12], Makris et. al. extend on the model proposed in [4] by introducing additional features to the model. These features were based on basslines accompanying the drum patterns. In other words, the authors used the basslines accompanying the drum patterns to put additional constraints on the Word-RNN architecture responsible for learning and generating drum patterns.

In addition to word-RNN and conditional word-RNN methods, another NLP technique, Sequence to Sequence (seq2seq) translation, has recently been proposed in [11] for generating drum patterns. In a sequence to sequence translation method, a multilayered LSTM is used to map (encode) an input sequence to a vector of a fixed dimensionality, while another LSTM layer decodes a target sequence from the decoded vector [16] (see Figure 1).

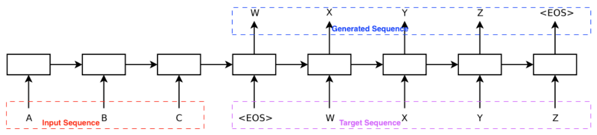


Figure 1: Block diagram of a typical sequence to sequence translation architecture [16]

Hutchings uses this method based on the assumption that in a drum phrase, different instruments speak different languages while saying the same thing [11]. Using seq2seq architecture, the author develops a generative drum machine which is capable of generating a drum groove only using a kick drum pattern and a provided musical style.

Although music is most certainly different from a language, above works illustrate the potentials of using NLP techniques for AMC. The current paper tries to further investigate the applicability of NLP (specifically the method proposed in [11]) for AMC.

4. METHODOLOGY

The objective of this project is to design, implement and test a generative model that is capable of generating a two-bar bassline that is stylistically interlocked with a provided two-bar audio drum loop.

There are numerous resources on how to create "groovy" or "danceable" house music basslines that are well interlocked with a drum pattern; however, these guidelines can be entirely subjective (i.e., they are biased towards the authors' aesthetic preferences), and hence, may not best characterize the style of the music. To avoid these biases, we decided to use machine learning techniques on a corpus consisting of a variety of style-specific music loops compiled from numerous artists/releases. Using the compiled dataset, a generative model was then trained so as to generate a bassline musical score given a provided audio loop.

4.1 Dataset Preparation

One of the objectives of our project was generating *style-specific* basslines. As a result, to study the capabilities of our design in generating *style-specific* basslines, we required at least two different datasets each of which consisted of entries from a very "distinct" style of dance music. The very "distinct" styles of dance music used in this project were House and Soca. This selection was based on the clear difference in the level of rhythmic complexity of drums and basslines in House and Soca music (Soca, or Soul of Calypso, is a style of music based on Calypso, however with soul and funk influences).

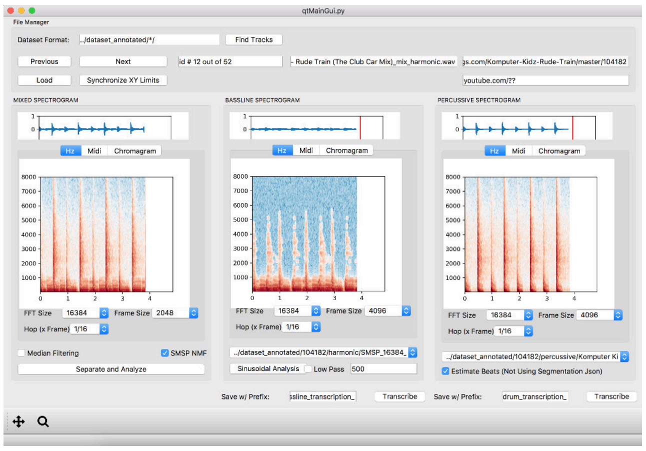
The dataset preparation is based on many drum and bassline loops obtained from popular House or Soca music tracks identified through *Discogs'* catalog [9]. In *Discogs*, the releases contain genre (Electronic, Rock, Funk, ...) and style (sub-genre) tags. Using *Discogs'* API, we collected thousands of releases tagged as Electronic Music (genre) / House Music (style) or Reggae Music (genre) / Soca (style).

One of the advantages of *Discogs* is that for many releases *Youtube* links are available for previewing a release. If a *Youtube* link was available for a relevant Discogs release, we downloaded the audio from the video hosted on the platform using a python script [1]. It should be noted that in order to avoid other biases in the dataset, in the cases where multiple videos were available, the script only downloaded the most popular link where popularity was defined based on views, likes, dislikes and the time elapsed since upload.

About a thousand tracks were collected using the above methodology. From these, we selected and extracted 100 2-bar drum and bassline phrases (50 for House and 50 for Soca), using a semi-automatic process. Initially, the goal was to use a harmonic/percussive separation technique to separate the bassline from the drum loop; however, somewhere along the process, we realized that this separation results in many artifacts in the separated percussive component that may potentially affect the quality of the extracted rhythmic patterns. As a result, in each track, we looked for a 2-bar loop with a bassline and another 2-bar loop containing only a drum loop that was very similar to the actual loop interlocked with the bassline. To be able to perform this process quickly, a script was developed in which beat tracking methods were used to segment the audio recordings in 2-bar Segments. Manually reviewing the segments, for each track (if available), we identified a drums only segment and a very similar segment containing the drums as well as a bassline.

After collecting the loops, a symbolic representation of the audio loops was required so as to be used for designing, training and testing a generative system. The raw audio with only the drum loop could be used for obtaining the symbolic representation. However, the bassline loops were usually mixed with a drum loop. As a result, we needed to decompose the mixed bassline loop into two separate bassline (harmonic) and drums (percussive) components. After the separation, the extracted bassline component could be used for transcription.

An interactive graphical user interface (as shown in Figure 2) was developed so as to facilitate the separation and transcription processes [2].


qtMainGui.py

File Manager

Dataset Format: ../dataset_annotated/*/

id # 12 out of 52

- Rude Train (The Club Car Mix)_mix_harmonic.wav

js.com/Komputer-Kidz-Rude-Train/master/104182

youtube.com/??

MIXED SPECTROGRAM: FFT Size 16384, Frame Size 2048, Hop (x Frame) 1/16; Median Filtering ☐; SMSP NMF ✓

BASSLINE SPECTROGRAM: FFT Size 16384, Frame Size 4096, Hop (x Frame) 1/16; ../dataset_annotated/104182/harmonic/SMSP_16384_; Low Pass ☐ 500

PERCUSSIVE SPECTROGRAM: FFT Size 16384, Frame Size 4096, Hop (x Frame) 1/16; ../dataset_annotated/104182/percussive/Komputer Ki; Estimate Beats (Not Using Segmentation Json) ✓

Save w/ Prefix: ssline_transcription_

Save w/ Prefix: drum_transcription_

Figure 2: The GUI for the developed transcriber software

Processing the obtained loops using the developed software, we were able to come up with a transcription of the basslines in terms of pitch, onset and offset locations (quantized to 16th-note accuracy). For the drums, we did not want to come up with an accurate instrument by instrument transcription. Instead, we wanted to come up with a meaningful representation of the drum activities with which we would be able to train a model. As a result, the drum loop was represented as a 2d matrix of onsets in eight frequency bands at each 16th-note time step (see Figure 3). The eight frequency bands were 40-70Hz, 70-110Hz, 130-145Hz,160-190Hz, 300-400Hz, 5-7kHz, 7-10kHz and 10-15kH [10].

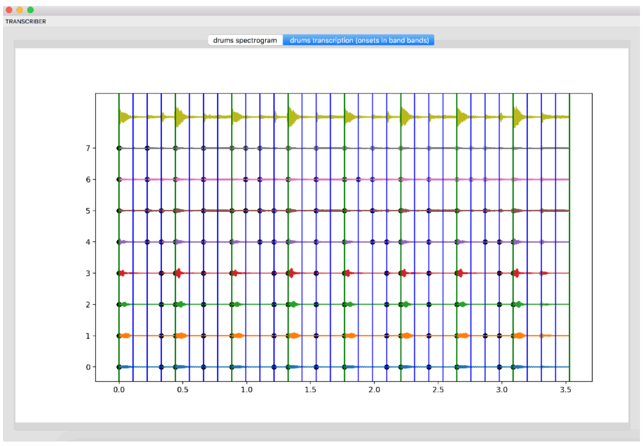


Figure 3: A snapshot of the drum representation

The final datasets[3] used in this project consist of 50 House loops and 50 Soca loops respectively. For each loop in the datasets, we provide a representation of the drum pattern and a bassline transcription obtained using the developed software. Quantized to 16th-note time steps and having a duration of 2 bars, the drums are represented at 32 different time steps. Hence, the drum representations are a 32x8 matrix of onsets. On the other hand, the basslines are vectors of 32x1 where the nth entry identifies whether a note starts (or should be sustained) at the nth time step. Figure 4 shows the overall pitch class distribution of the transcribed basslines in each dataset.

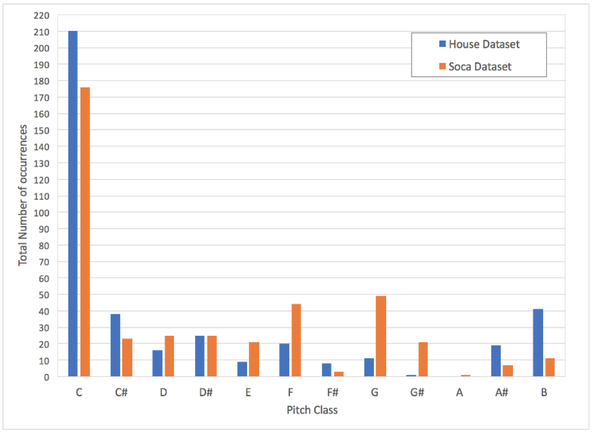


Figure 4: Overall distribution of pitch classes in the datasets

Reviewing the pitch class distributions, we believe that the basslines in the two datasets are harmonically distinct to some extent. It should be noted that the starting pitch of each bassline was transposed to C, hence, there is a predominance of C in the pitch class distribution.

[1]Source code is publicly available at `https://github.com/behzadhaki/discogs-youtube-dl`

[2]Source code is publicly available at `https://github.com/behzadhaki/qtHarmonicPercussiveSeparatorTranscriber`

[3]The datasets compiled for this project are publicly available at `https://github.com/behzadhaki/drum_bassline_dataset`

4.2 Automatic Music Composition

In [11], Hutchings uses an LSTM-based Word-RNN seq2seq architecture to generate a style specific drum pattern based on a provided kick drum pattern. Inspired by this approach, we decided to use the same architecture to translate a provided multi-instrument drum pattern to a style-specific bassline.

For every entry in the datasets, a bassline vector and a drum vector were prepared based on the available drum and bassline transcriptions/representations so as to be used for designing and implementing the models. The vectors were obtained in the following manner:

1. Bassline Vector: Using the available transcriptions in the compiled datasets, we constructed a bassline vector of a length of 32 (2 bars) for every entry in the datasets. The i^{th} element in this vector represents the state of the bassline at the i^{th} time step. Moreover, the basslines were all modified such that in all of them, the starting pitch was C3 (MIDI 48). As an example, a typical bassline vector in the dataset looks like: array([48., 49., 1000., ..., 48., 1000., 1000., 0.]). A value of 1000 in this vector identifies that the note at the previous step needs to be sustained. A value of 0 in this vector identifies that there is a silence at the corresponding time step. Any other value in this vector is a MIDI number which identifies whether a note (with the given MIDI number) exists at the corresponding time step (see Figure 5).

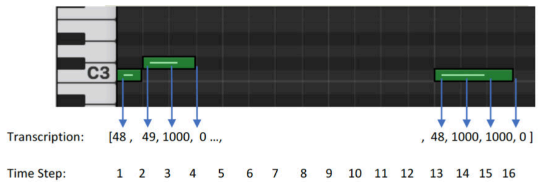


Figure 5: Bassline Coding

2. Drums Vector: Using the available transcriptions in the compiled datasets, we constructed a drums vector of a length of 32 (2 bars) for every entry in the datasets. The i^{th} element in this vector represents the state of the drums at the i^{th} time step. Each i^{th} element of the drums vector is an 8-bit binary encoding in which the k^{th} bit shows whether a drum onset exists at the i^{th} time step. A typical drum vector for a sample loop in the dataset looks like: ['0b11111000', '0b00000000', '..., '0b00000011']. Note that the 0b prefix denotes binary encoding.

In a seq2seq architecture, a one-hot encoding is required for every single input and output node. Input node N represents the drum pattern at the N^{th} time step. For the input of the encoder layer, a one-hot-encoded vector of length 256 is used to denote which of the drum combinations exists at a given time step. For the bassline outputs, a one-hot encoded vector of length M is used where

$M = pitch\ class\ count\ in\ dataset + 1\ (sustain) + 1\ (silence).$

In the compiled dataset for House music, there are 32 distinct notes (i.e., $M = 34$), while in the Soca dataset, there are 34 distinct notes (i.e., $M = 36$).

Two almost identical Word-RNN sequence-to-sequence models were implemented in Keras using a TensorFlow backend[4]. The models consisted of an encoder layer of 128 LSTM units. The decoder layer consisted of the same number of LSTM units. Moreover, the output of the decoder layer was connected to a dense softmax layer consisting of M cells. In the case of House music, the softmax layer consisted of 34 cells ($M = 34$), while in the case of Soca music, the softmax layer consisted of 36 cells ($M = 36$) (See Figures 6).

Test Group	Loop Arrangements	Test Count
House Loops Sub-group A	T_1) Drum + Original Bassline T_2) Drum + Generated Bassline using House model T_{3a}) Drum + Random Bassline from House dataset	5
House Loops Sub-group B	T_1) Drum + Original Bassline T_2) Drum + Generated Bassline using House model T_{3b}) Drum + Random Bassline from Soca dataset	5
Soca Loops Sub-group A	T_1) Drum + Original Bassline T_2) Drum + Generated Bassline using Soca model T_{3a}) Drum + Random Bassline from Soca dataset	5
Soca Loops Sub-group B	T_1) Drum + Original Bassline T_2) Drum + Generated Bassline using Soca model T_{3b}) Drum + Random Bassline from House dataset	5
Overall		20 tests

Table 1: Listening experiment tests

To fulfill the requirements of the seq2seq LSTM architecture, a reverse and shifted version of the bassline was also used as an input during the training process. A sequence to sequence model tries to translate an input phrase to an output phrase. Unlike language in which for an input sentence there can be only one translation, in our case, for a given input (drum pattern), more than one output (bassline) can be valid. Hence, our assumption was that if we train the model on the entire dataset, a generated bassline should be different from the bassline used for training. Moreover, if a model is over-fitted to the training set, the output can still be quite usable for practical applications. For these reasons, we decided to train our models on the entirety of the datasets and use randomly selected entries from the same training sets to evaluate the trained models. Moreover, the training was done for 4000 epochs and a batch size of 50. These values were selected by trial and error such that reasonably varying basslines were generated for randomly tested inputs.

5. SYSTEM EVALUATION

A subjective evaluation of the generated basslines shows that the density of the events in the generated basslines are usually higher than the original basslines. Moreover, while the generated basslines were never identical to the original bassline associated with a drum loop, some generated basslines shared rhythmic and melodic similarities [5].

To better evaluate the developed bassline generation system, we prepared a listening experiment. The listening test aimed to study whether the generated basslines were comparable to real basslines existing in the datasets. The experiment comprised 20 tests each of which consisted of 3 different loops. The tests were prepared as detailed in Table 1.

For each test, three different arrangements of a single drum loop combined with three different basslines were created. The first bassline was the actual bassline accompanying the drum loop in the dataset. The second bassline was generated based on the model developed for the style of the drum loop. Lastly, a random bassline either from the same style (Sub-group A) or the other style (Sub-group

[4]The source codes used for preparing the data as well as constructing, training and predicting the model are publicly available at `https://github.com/behzadhaki/bassline_seq2seq`

[5]Some examples of the generated basslines can be found in `https://drive.google.com/open?id=1Tpl5aHKRC8FF48u7HhprT2phTZpM_gd0`

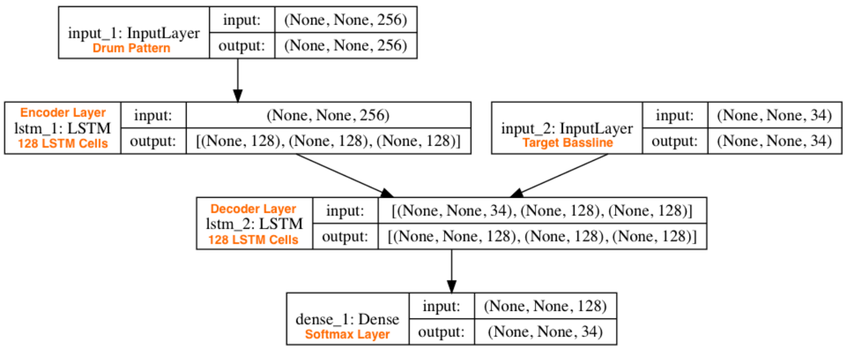


Figure 6: The architecture of the proposed model for House Music

B) was selected to accompany the drum loop. The drum loops and the non-generated basslines were selected randomly from the training dataset. The three arrangements were randomly presented to the participants without mentioning the conditions under which they were created.

The participants were asked to go through the tests and rank the three different loop arrangements presented in each test based on their perceptual interests. Before submitting the results, participants were asked to identify the prominent factors which played a role in making their judgments. Moreover, the questionnaire started with a description of the experiment and a number of demographic questions.

23 people participated in the experiment. The majority of the participants were 20 to 40 years old. The demographic questions showed that the majority of the participants were musically experienced. Moreover, we were interested as to what factors constitute the likeability of a bassline. Hence, we asked the users to select from a number of factors which may have had affected their judgments. The results show that for 78% of the participants, the melodic structure of a bassline contributed to the likeability of the bassline. Moreover, 70% and 65% of the users reported that respectively the danceability of the loops and the drum/bassline rhythmic consistency contributed to the likeability of the bassline.

Prior to performing statistical analysis on the responses, an overall grade was generated for every audio loop in the experiment using the ordinal rankings provided by participants [6]. Analysis of Variance (ANOVA) was performed to investigate whether the mean of the responses for different loops were different. If the ANOVA analysis indicated that some means are different in an analysis subset, a post-hoc analysis was performed to identify the different means [7].

Figure 7 illustrates the summary of means and standard deviations of the responses for Subsets 1, 2, 3 and 4. The ANOVA analysis on results showed that:

1. For House Loops (Sub-group A), on average, the generated bassline (T_2) is more interesting than the original bassline (T_1) and the bassline selected from the same style (T_{3a}). Moreover, the random basslines are as interesting as the original ones.

2. For House Loops (Sub-group B), on average, the generated bassline (T_2) is as interesting as the original bassline (T_1) and the bassline selected from the other style (T_{3b}). Moreover, the random basslines are significantly less interesting than the original and generated ones.

3. For Soca Loops (Sub-groups A and B), on average, all three basslines are equally interesting.

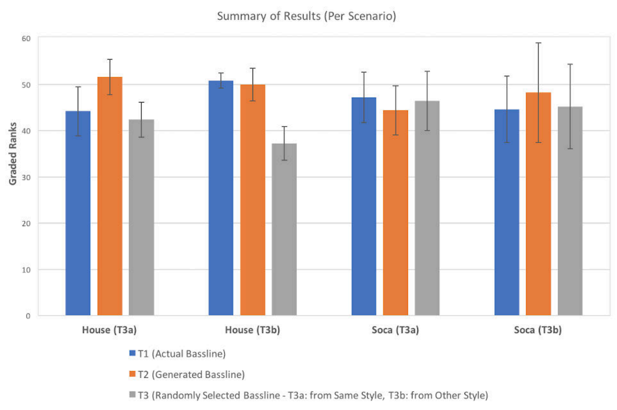


Figure 7: Summary of Results (Group by Group)

Finally, all of the collected data was analyzed irrespective of the style of the loop or the style of the bassline. Figure 8 illustrates the summary of means and standard deviations of the responses irrespective of the style.

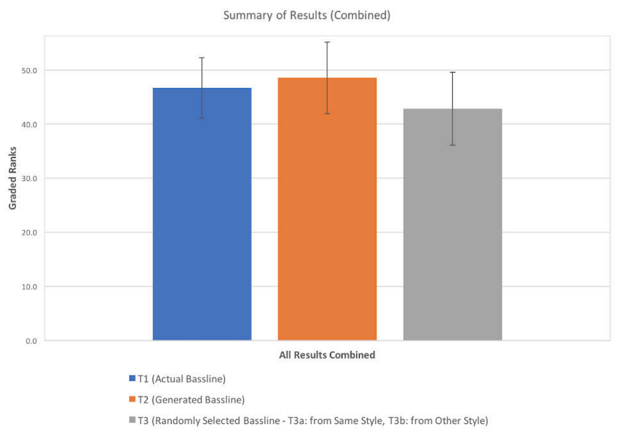


Figure 8: Summary of Combined Results

The above ANOVA analysis shows that, generally, the generated basslines are as interesting as the original basslines, while the random basslines are significantly less interesting.

6. CONCLUSIONS

Given the small size of the datasets (50 loops for each style), we were limited to the most possible compact architecture.

[6] A grade of 3, 2, or 1 was assigned to "Most Interesting", "Neutral" or "Least Interesting" loops respectively so as to generate an overall grade.

[7] A detailed analysis of the results is available in Appendix B of the thesis published for this project [8]

As a result, we limited the encoder/decoder networks to a single LSTM layer. However, we tried different number of LSTM units in the encoder/decoder network. Our subjective evaluation (based on the rhythmic and melodic structure of the generated basslines) showed that the most interesting basslines were obtained using 128 LSTM units in each of the encoder/decoder networks. Moreover, we trained the selected model using different number of epochs, and based on our subjective evaluation, the most interesting basslines were generated using 4000 epochs. The resulting models were capable of generating basslines that were majorly melodically coherent and rhythmically complex. We also believe that not only the generated basslines were interesting but also they were well interlocked with the input drum loop.

To better investigate the quality of the generated basslines, we conducted a listening experiment. In the case of House music, the results of the experiment showed that the model generated basslines which were as interesting as the original bassline while they were significantly more interesting than the randomly selected basslines. In the case of Soca music, the evaluation showed the same pattern with the exception that the random basslines were also as likeable as the original and generated versions.

One interesting observation was that in the case of Soca loops the basslines randomly selected from the other style were as likeable as the original and generated basslines. We suspect that the higher rhythmic complexity of the Soca loops allow for more variations in the basslines, and as a result, there are more basslines deemed acceptable and likeable. Moreover, in the case of Soca music, the compositions are more diverse in terms of instrumentation. In fact, based on our observations, numerous brass, synthesizer and vocal components existed in every track. The instrumentation in Soca tracks may imply that the basslines need not only be interlocked with the drums but also match other instruments in the track (i.e., the quality of a bassline can only be evaluated in the context of a multi-instrument loop). Hence, with a lack of instruments other than drums and bass, in the listening experiment, basslines do not convey enough information to establish a clear trend in the likeability of the loops. Moreover, the unfamiliarity of the users with this style might have had contributed to this trend.

In conclusion, the method proposed in this report is capable of generating basslines within a specific style. To generate a bassline, an audio drum loop can be fed into the system, from which a symbolic representation will be derived in terms of percussive onsets in 8 different frequency bands. The symbolic representation is fed into a seq-2-seq model to generate a melodically and rhythmically interesting bassline.

7. FUTURE WORKS

In future iterations, a dataset using a producer's tracks can be created to investigate whether the model is capable of replicating aesthetic preferences of a producer. Moreover, an alternative method for possible drum representations is to use Variational Auto-Encoders (VAEs). The benefit of this method may be that a large dataset containing of only drum loops can be used for training the VAE. Subsequently, the latent space of the VAE can be used as a representation of the drum loops in a given dataset (containing both drums and basslines). This alternative representation can then be used for training the model proposed in this paper. Finally, an interactive software tool can be developed for musicians/producers. This software can be trained either on a provided corpus or using a dataset compiled from the musicians'/producers' compositions. This tool can be developed such that it would provide real-time recommendations. The effectiveness of this tool should be fully evaluated for both production and performance purposes.

8. REFERENCES

[1] A. Alpern. Techniques for algorithmic composition of music. 95:120, 1995.

[2] C. Anderson, A. Eigenfeldt, and P. Pasquier. The generative electronic dance music algorithmic system (gedmas). In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment (AIIDE 13) Conference*, 2013.

[3] J.-P. Briot. *Deep Learning Techniques for Music Generation*. Springer, 2018.

[4] K. Choi, G. Fazekas, and M. Sandler. Text-based lstm networks for automatic music composition. *arXiv preprint arXiv:1604.05358*, 2016.

[5] A. Eigenfeldt. Towards a generative electronica, Mar 2013.

[6] A. Eigenfeldt and P. Pasquier. Considering vertical and horizontal context in corpus-based generative electronic dance music. In *ICCC*, pages 72–78, 2013.

[7] J. D. Fernández and F. Vico. Ai methods in algorithmic composition: A comprehensive survey. *Journal of Artificial Intelligence Research*, 48:513–582, 2013.

[8] B. Haki. Drums and Bass Interlocking: A Bassline Generation System Based on Sequence-to-Sequence Learning, Apr. 2019. Available at `https://doi.org/10.5281/zenodo.2631874`.

[9] J. Hartnett. Discogs. com. *The Charleston Advisor*, 16(4):26–33, 2015.

[10] P. Herrera, A. Yeterian, and F. Gouyon. Automatic classification of drum sounds: a comparison of feature selection methods and classification techniques. In *Music and Artificial Intelligence*, pages 69–80. Springer, 2002.

[11] P. Hutchings. Talking drums: Generating drum grooves with neural networks. *arXiv preprint arXiv:1706.09558*, 2017.

[12] D. Makris, M. Kaliakatsos-Papakostas, I. Karydis, and K. L. Kermanidis. Combining lstm and feed forward neural networks for conditional rhythm composition. In *International Conference on Engineering Applications of Neural Networks*, pages 570–582. Springer, 2017.

[13] J. A. Maurer. The history of algorithmic composition, Mar 1999.

[14] F. Pachet. The continuator: Musical interaction with style. *Journal of New Music Research*, 32(3):333–341, 2003.

[15] F. Pachet, P. Roy, G. Barbieri, and S. C. Paris. Finite-length markov processes with constraints. *transition*, 6(1/3), 2001.

[16] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.