Identifying Critical Decision Points in Musical Compositions using Machine Learning

Berker Banar
School of EECS
Queen Mary University of London
London, UK
b.banar@qmul.ac.uk

Simon Colton
School of EECS
Queen Mary University of London
London, UK
s.colton@qmul.ac.uk

***Abstract*—In musical compositions, identifying critical points that reveal atypical and unexpected decisions is valuable from a compositional perspective as these points arguably contribute to the enjoyment of listening to music and are useful for applications such as automatic music generation and music understanding. In this study, we suggest a machine learning-based approach for identifying critical decision points, where we utilise two long short-term memory (LSTM) models that originally function as generative networks and are repurposed in our case to identify critical decision points. These models are trained on musical corpora from the classical period and the 20th century providing different angles to the analysis. We demonstrate this approach using two short musical examples and an excerpt from Chopin's Nocturne in E flat major (Op. 9 No. 2). We compare our suggested machine-learning-based approach to two time series analysis methods as the baselines, evaluate the results, and suggest some future directions for this approach.**

***Index Terms*—musical composition analysis, music information retrieval, machine learning, long short-term memory networks, time series analysis**

I. Introduction

Composing music is typically a theory-based practice that has been developed for hundreds of years [1]. Following the rules of music theory provides a basis for composers, however, composers intentionally make some atypical and unexpected decisions in musical pieces, which might not necessarily be aligned with the appropriate music theory [2]. These atypical and unexpected decision points are critical as they arguably contribute to the enjoyment of listening to music, guide audiences' expectations, and give character to the pieces from a compositional point of view [3].

Generating musical compositions using machine learning is an active research field and various neural models have been utilised ranging from long short-term memory networks [4] to transformers [5] [6]. One common practice is to represent music symbolically in the generative music models [7]. Symbolic music representations (e.g. MIDI) consist of musical notes in the form of time series data, where fundamental musical attributes such as note number (pitch), velocity (musical dynamics) and timing are numerically presented [8]. A common paradigm in generative music systems is auto-regressive generation, where the musical properties of the next note are predicted by the model given a musical sequence as the prior and this procedure continues iteratively to generate longer music sequences [5].

Identifying critical decision points that exhibit atypical and unexpected behaviour is useful for generative music studies, where generated pieces can be conditioned on these critical points and generative models can arguably learn musical patterns effectively with a particular emphasis on these points. These critical decision points are also useful for other types of music information retrieval research such as music understanding and music recommendation systems, where we typically need a thorough analysis of the musical pieces [9]. Even though the critical points might be subjective ultimately, having a better understanding of these points through the lenses of neural models and time series analysis methods is insightful for music information retrieval and computational creativity research [10].

In this study, we present a novel machine learning-based method using long short-term memory networks (LSTMs) to identify critical decision points in musical compositions [11]. We repurpose two LSTM-based generative music models for our analysis making use of their auto-regressive behaviour and utilise two distinct training corpora that are curated from the GiantMIDI-Piano dataset [12]. These two musical corpora consist of pieces from the classical period and the 20th century to conduct this analysis from two different era-dependent perspectives. We compare our suggested machine learning-based method to two other traditional time series analysis approaches that we construct as baselines. We demonstrate these methods on two manually designed short musical examples and an excerpt from Frédéric Chopin's Nocturne in E flat major (Op. 9 No.2), evaluate the results of these methods, and suggest some future directions for this approach.

II. Methodology

To identify critical decision points, we employ a machine learning-based method and also two time series analysis methods that are constructed as baselines. These methods use only the pitch information in symbolic music representations of musical pieces, as arguably it is one of the most important and apparent musical features for atypical and unexpected decisions. In all of our methods, we convert the pitch values of musical notes into pitch-classes, which are the octave invariant versions of pitches. There are 128 pitch values (ranging from 0 to 127) in typical symbolic music representation (MIDI) and 12 pitch-classes, namely C, C#, D, D#, E, F, F#, G, G#, A, A# and B. We present the musical pieces to our analysis methods using the pitch-class values of musical notes, as the conversion from pitch to pitch-class simplifies our analysis, where the pitch-class is capable of capturing tonality and pitch-based features.

Our baseline methods are based on two fundamental musical attributes, namely the most common pitch-class and the tonality. These methods rely on a statistical analysis of a given musical piece in terms of these musical attributes to investigate the critical decision points. Since these baseline methods are based on the statistics of the subject musical piece, the analysis is based on the information provided within the piece itself. Therefore, these two baseline methods are called *within piece analysis based on the most frequent pitch-class* and *within piece analysis based on the tonality*.

Our suggested machine learning-based method repurposes an LSTM network that was originally trained for a music generation setting, where it predicts the pitch-class of the next note given a prior sequence of pitch-classes. The network outputs the predicted pitch-class via a softmax layer, which gives the probabilities of pitch-classes among 12 candidates. To conduct our critical decision points analysis, we run the LSTM network over an already existing musical piece note by note and utilise the softmax layer to compare the predicted note and the actual note in the piece. Since this method

Berker Banar is a research student at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported jointly by UK Research and Innovation [grant number EP/S022694/1] and Queen Mary University of London.

analyses critical decision points based on the characteristics of and the patterns in the provided training corpus to the LSTM network, we call this method *corpus-based analysis using LSTM networks*.

We use a measure of *interestingness* to conduct the critical decision points analysis, where we calculate the interestingness level of each note in the subject piece. In our baseline methods, the interestingness measure is binary, and critical decision points are considered to be the ones with high interestingness. In our machine learning-based method, this measure has twelve discrete levels as per 12 pitch-classes in the softmax layer and higher interestingness levels correspond to more critical notes. The details of the interestingness calculations are given in the next sections.

A. Within Piece Analysis based on the Most Frequent Pitch-Class

We present a musical piece, M, of N notes, as a list of MIDI note numbers, $M = [m_1, \ldots, m_N]$, where $m_i \in \{0, 1, \ldots, 127\}$ representing musical pitches. We convert the list of MIDI note numbers, M, into a list of pitch classes, P, as:

$$P = [p_i = m_i \mod 12 \; : \; m_i \in M]$$

The values of $p_i \in P$ ranging from 0 to 11 represent twelve pitch classes from note C to note B, respectively. To conduct the interestingness analysis, we select a subject note in P, denoted as p_s, and also a window symmetrically centred around p_s with the size of w that is an odd number. Then, we extract the list of windowed samples such that $P_{window} = [p_{s-(w-1)/2}, \ldots, p_{s-1}, p_{s+1}, \ldots, p_{s+(w-1)/2}]$ with $w - 1$ elements, where the first and second halves include samples before and after the subject note, p_s, respectively. Afterwards, to find the most frequent pitch-class in the windowed samples, we check the frequency of occurrence of each pitch-class, pc_k, where $pc_{0,\ldots,11} = 0, \ldots, 11$, as:

$$f_P(pc_k) = |[p_i \in P_{window} : p_i = pc_k]|$$

Then, we create a list of $[f_P(pc_0), \ldots, f_P(pc_{11})]$ and sort it in ascending order to find the most frequently occuring pitch-class, which is the pitch-class of the last item in the list, denoted as pc_{mf}. Afterwards, to determine the interestingness of the subject note, p_s, we use the following function:

$$I(P_{window}, s, pc_{mf}) = \begin{cases} 1, & p_s \neq pc_{mf} \\ 0, & p_s = pc_{mf} \end{cases}$$

Practically, it means if the pitch-class of a note is not the most frequently occurring pitch-class within the subject window, then it's considered to be an interesting note in the sequence. We iteratively repeat this process for the upcoming subject notes to apply this method over a section of the musical piece. This approach is arguably the simplest way of conducting an interestingness analysis. Even though it demonstrates the basics of our analysis approach, it is limited from a musical point of view as it tends to classify all the pitch-classes except the most-frequent one as interesting. Therefore, we construct the second baseline method, which is musically more informative.

B. Within Piece Analysis based on the Tonality

Similar to the most frequent pitch-class method, we obtain $P_{window} = [p_{s-(w-1)/2}, \ldots, p_{s-1}, p_{s+1}, \ldots, p_{s+(w-1)/2}]$, subject note, p_s, and most frequent pitch-class, pc_{mf}. Then, based on the pc_{mf}, we infer the musical tonality of the subject window assuming that pc_{mf} is the pitch-class of tonic note, i.e. the first-degree note of a musical scale (e.g. C in C major scale). To infer the tonality, we generate two template lists for major and minor scales in terms of the pitch-class intervals taking the tonic pitch-class as the reference point. Major scales can be constructed by adding each of the interval values in the set of $\{2, 4, 5, 7, 9, 11\}$ to the tonic. Similarly, the set

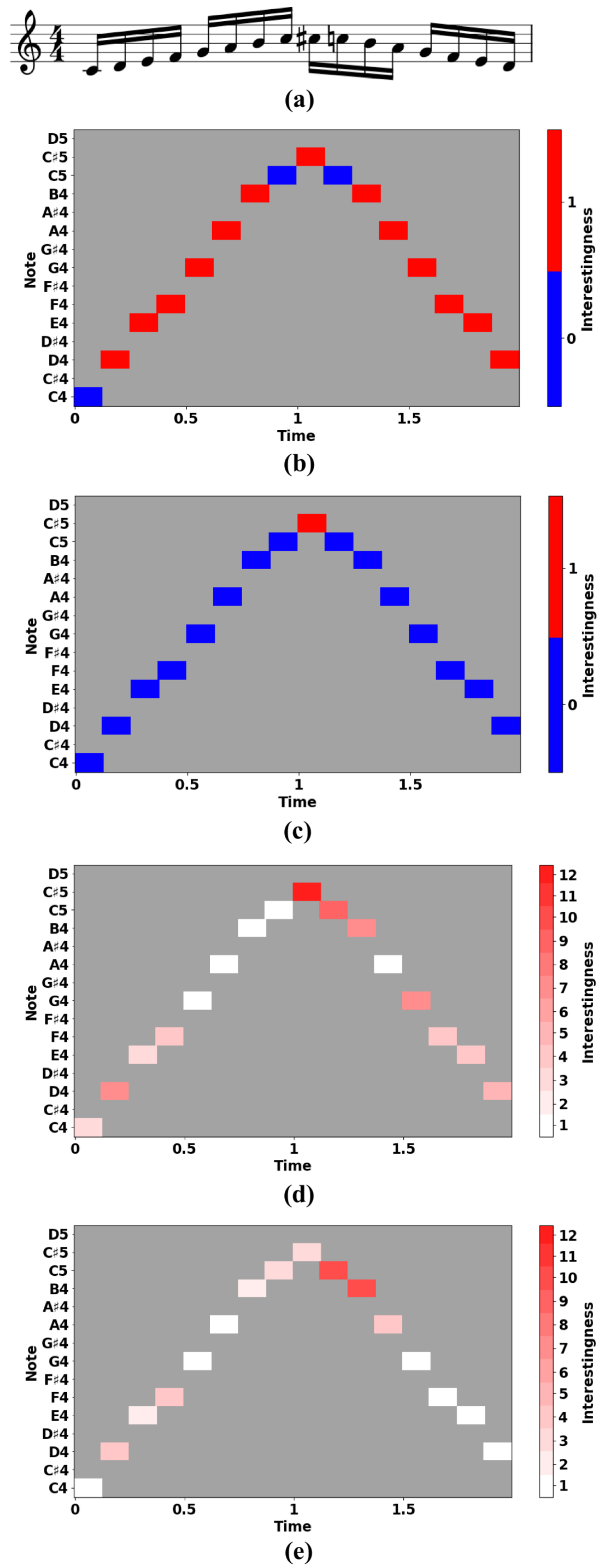


Fig. 1. (a) Manually designed musical excerpt in C major with b9 tension (pitch-class of Db / C#), (b) within piece analysis based on the most frequent pitch-class with w = 51 and w = 101, (c) within piece analysis based on the tonality with w = 51 and w = 101, (d) corpus-based analysis using LSTMs trained on the classical period music, and (e) the 20th century music.

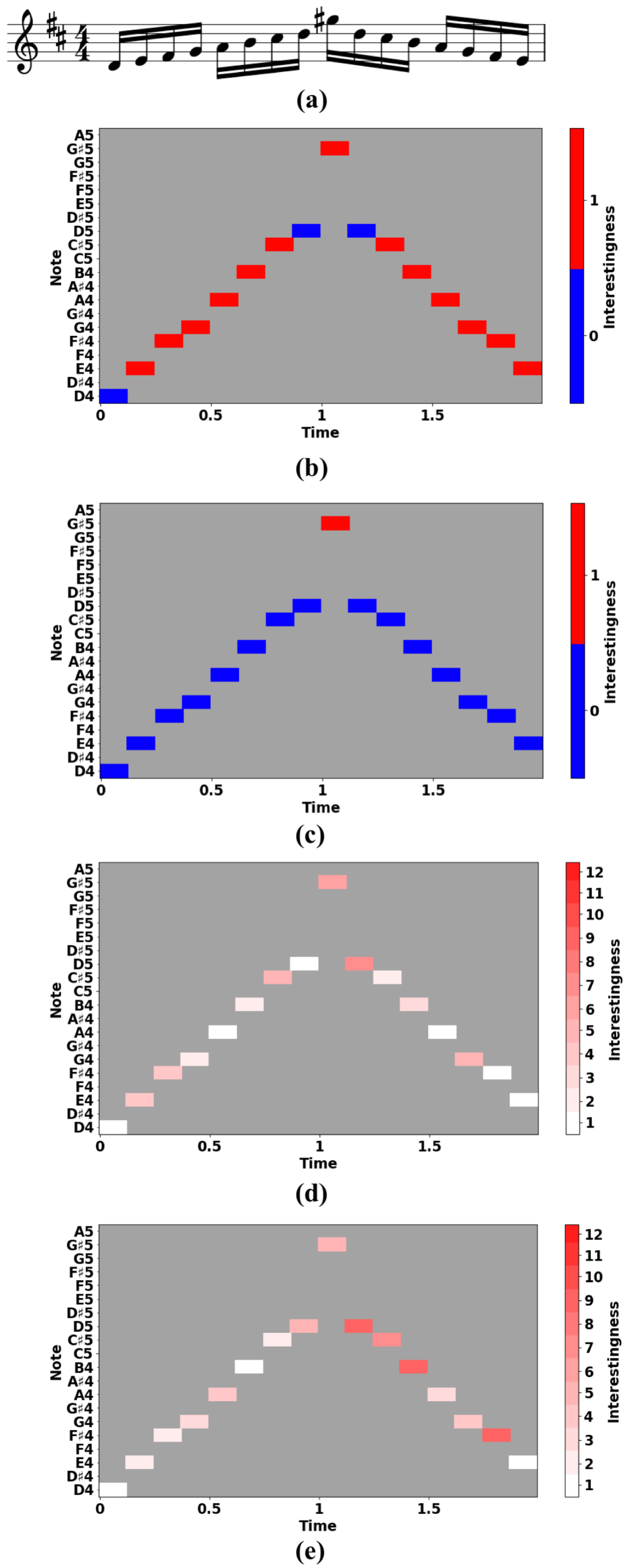


Fig. 2. (a) Manually designed musical excerpt in D major with #11 tension (pitch-class of G#), (b) within piece analysis based on the most frequent pitch-class with w = 51 and w = 101, (c) within piece analysis based on the tonality with w = 51 and w = 101, (d) corpus-based analysis using LSTMs trained on the classical period music, and (e) the 20th century music.

of $\{2, 3, 5, 7, 8, 10\}$ can be used for the minor scales. Using these interval sets, we generate major and minor reference lists such that $R_{major} = [p_{mf}, p_{mf} + 2, p_{mf} + 4, p_{mf} + 5, p_{mf} + 7, p_{mf} + 9, p_{mf}+11]$ and $R_{minor} = [p_{mf}, p_{mf}+2, p_{mf}+3, p_{mf}+5, p_{mf}+7, p_{mf} + 8, p_{mf} + 10]$. Depending on the value of p_{mf}, since the values in the reference lists can be bigger than 11, we process these reference lists with the modulo 12 operator to ensure that they are valid pitch-class values. Then, in our subject window P_{window}, we count the total number of pitch-classes that match with the major and minor reference lists, as:

$$n_{major}(P_{window}) = |[p_i \in T_{major} : p_i \in P_{window}]|$$

$$n_{minor}(P_{window}) = |[p_i \in T_{minor} : p_i \in P_{window}]|$$

Depending on the total number of matching major and minor pitch-classes, whichever is the higher, we infer the tonality as such. Based on the obtained tonality, we select a reference list $R_{selected} = R_{major}$ or $R_{selected} = R_{minor}$. Then, we determine the interestingness of the subject note, p_s, as:

$$I(P_{window}, s, R_{selected}) = \begin{cases} 1, & p_s \notin R_{selected} \\ 0, & p_s \in R_{selected} \end{cases}$$

This interestingness analysis classifies pitch-classes that are not part of the tonality as interesting. Compared to our first baseline method, this method is musically more informative as it considers the notion of tonality / atonality which is related to atypical and unexpected decisions compositionally. However, depending on the window length and the nature of the piece, the tonality-based approach might be limited due to the temporal scope of inferred tonality. Therefore, we propose our machine learning-based method, which is arguably a more capable alternative due to analysing from a corpus-based perspective and its ability to model characteristics and patterns presented in the data.

C. *Corpus-based Analysis using LSTM Networks*

We repurpose a generative LSTM network [11] and utilise it for our critical decision points analysis. The network consists of an embedding layer with the embedding dimension of 256, an LSTM layer with 1024 units, and a softmaxed dense layer for the predictions of 12 pitch-classes, in this respective order from input to output. We use a categorical cross-entropy loss function as per our 12 pitch-classes and an Adam optimiser with the learning rate of 0.005 [13].

The decisions made by the network over 12 pitch-classes depend on the characteristics of the training corpus. Thus, to conduct this analysis from two different perspectives, we curated two different era-dependent subsets of the GiantMIDI-Piano dataset, which has 10855 classical piano pieces [12]. This dataset also has metadata including the birth years of the composers and using this metadata, we curated two subsets for the pieces whose composers' birth years are in the range of 1700-1720 and 1900-1905. The subset corresponding to 1700-1720 birth years has 114 pieces and 173428 notes, whereas the other subset has 61 pieces and 222182 notes in total. While selecting these birth years, we considered the number of notes they have, which should ideally be around 200000 based on our initial experiments with the LSTM architecture. We also aimed for the subsets to reflect distinct musical periods in terms of their characteristics. The first subset with the birth years from 1700 to 1720 roughly corresponds to the classical period and the second subset roughly corresponds to the 20th century period. As these periods have distinct musical characteristics stylistically, they can provide different angles to our analysis. Using these two subsets, we trained two LSTM models for 3000 epochs each.

We run our trained networks over sequences of pitch-classes and use the softmax predictions to conduct our analysis. We make use of the auto-regressive nature of the networks, where the predictions are made note by note with a moving window for the prior sequence rather than generating many samples in one go. We select a subject note, p_s, within the list of pitch-classes, P, similar to our baseline methods. In this method, the list of $P_{prior} = [p_{s-100}, \ldots, p_{s-1}]$ is obtained from the P, where we have a fixed length of 100 samples for the prior musical sequence to be provided to the network. Using the P_{prior}, the network makes predictions for the subject note, p_s, via the softmaxed probabilities for each of the 12 pitch-classes, each denoted as s_{pc_i}. Then, we have the list of softmaxed probabilities for the subject note, p_s, which is $S_s = [s_{pc_0}, s_{pc_1}, \ldots, s_{pc_{11}}]$. Similar to our baseline methods, we have a measure of interestingness. In this case, it has 12 discrete levels due to twelve pitch-classes rather than being a binary measure. To calculate the interestingness of p_s, we sort the list of S_s in descending order and get the index (from 1 to 12) of s_{pc_i} that corresponds to the pitch-class of the subject note, p_s, in the sorted list. So, this index becomes the interestingness level of the p_s, where lower probability inferred by the model for the existing note means higher interestingness.

III. Experiments and Results

To demonstrate our methods, we manually designed two short musical examples, and also extracted an excerpt from Chopin's Nocturne in E flat major (Op.9 No. 2). The audio files for these musical examples can be found here[1].

A. Manually Designed Musical Excerpts

Our manually designed musical examples exhibit both tonal and atonal features. One example consists of an ascending and descending C major scale with the b9 tension (pitch-class of Db / C#) at the climax of the melody as shown in Figure 1 (a). The other manually designed example consists of ascending and descending D major scale with the #11 tension (pitch-class of G#) similarly at the climax as shown in Figure 2 (a). To conduct our analysis methods over these short musical examples, we looped these melodies nine times to elongate the samples so that our methods can be performed in terms of the window and prior sequence lengths.

For our C major example, we conducted our baseline analysis methods with the window lengths of 51 and 101 for both the most common pitch-class and the tonality cases. In each method, these different window lengths produced the same results. We present the results for our baseline methods in Figure 1 (b) and (c). The method based on the most frequent pitch-class acts as expected as it classifies all the pitches except C as interesting. Even though this is not very informative from a musical point of view, it demonstrates our time series analysis approach. The second baseline based on the tonality classifies only the C# as interesting, which is the expected behaviour as it is not a part of C major scale.

Over the C major example, we conducted our suggested machine learning-based method with the models trained on the classical period and the 20th century music data. These results are shown in Figure 1 (d) and (e), respectively. The model trained on the classical period classifies the pitch of C# as the most interesting note of the melody, which is aligned with our expectations as it is not part of the C major scale. Also, it classifies the C and B notes following the C# as relatively interesting notes, which behaviour still addresses the unexpected and salient part of the melody. The model trained on 20th century music doesn't classify the note C# as one of the highly interesting notes, which is a good indicator of the different perspectives provided by this model since non-tonal notes are more common in the 20th century music compared to the classical period. Instead of the C# note, this model classifies the notes C and B following the C# as the most interesting notes, which is arguably reasonable as it still highlights the relatively unstable part of the melody.

For the example in D major, similarly, we conducted our baseline methods with the window lengths of 51 and 101. The results for different window lengths are the same in each method. Analyses based on the most frequent pitch-class and the tonality are depicted in Figure 2 (b) and (c), respectively. Similar to the C major example, the method based on the most frequent pitch-class classifies any pitch except the tonic (D) as interesting and the method based on the tonality classifies only the non-tonal (G#) note as interesting, which is expected since it is the only note that is not part of D major scale.

We applied our suggested machine learning-based method to the D major example using two different models similarly. As shown in Figure 2 (d), the model trained on the classical period data classifies the D after G# as the most interesting note of the melody, where G# is classified as the second most interesting note. In contrast to the C major example, the model trained on the classical period data doesn't classify the only non-tonal note (G#) as the most interesting one, yet these results show that this model is able to locate the arguably most interesting part of our designed example while gradually paying attention to the other notes as well. The model trained on the 20th century data doesn't classify the G# as the most interesting note, which is reasonable given the characteristics of the musical period. Instead of the G#, this model classifies the notes D, B and F# as the most interesting notes.

Based on these results, the method based on tonality is more informative for the critical decision points analysis than the method based on the most common pitch-class since pitch-classes out of tonality are generally associated with atypical and unexpected decisions. However, one limitation of the method based on tonality is that it only relies on the tonality feature, which is not strictly considered in musical practices necessarily. Also, another limitation of the baseline methods is that they are binary classification methods. Our suggested machine learning-based approach provides us with reasonable and more detailed information, where there is a sensible difference between two different era-specific models.

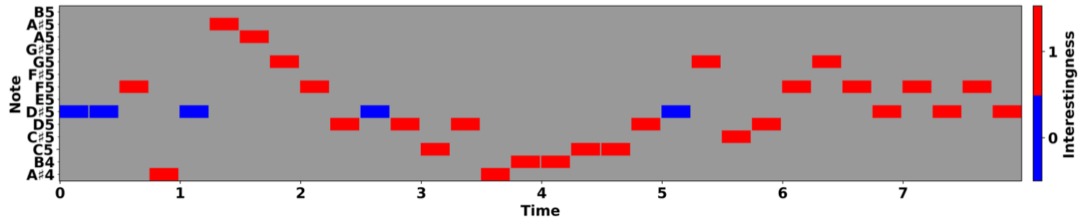
B. Chopin's Nocturne in E Flat Major (Op. 9 No. 2)

In this section, we demonstrate our baseline approaches and the suggested machine learning-based method on an excerpt from a well-known musical composition, Chopin's Nocturne in E flat major (Op. 2 No. 2). We utilised our baseline methods with the window sizes of 51 and 101, whose results are shown in Figure 3 (b), (c), (d) and (e). We also conducted the analysis using our machine learning-based method with two models trained on the classical period and the 20th century dataset, whose results are given in Figure 3 (f) and (g). Since the analysis only focuses on the pitch features, we normalised the note durations to 16th notes to make the effect of pitch-class clearly distinguishable while listening to the piece. Also, to accommodate the window sizes of 51 and 101, and the prior sequence length of 100 for the machine learning-based method, we extracted a longer excerpt than the one in Figure 3. More specifically, we have more than 100 notes before and after the excerpt presented.

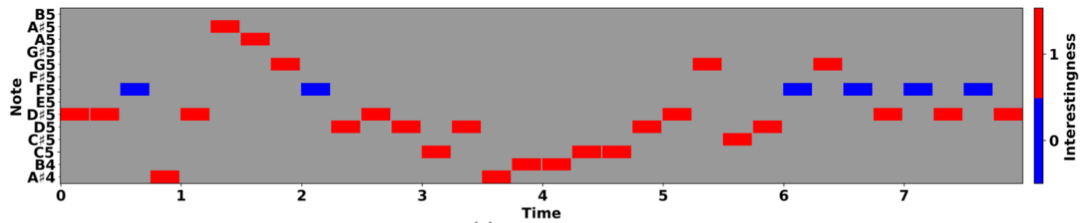
In the most common pitch-class-based analysis, for the window size of 51 as shown in Figure 3 (b), the most common pitch-class is calculated as Eb (D#), which is reasonable as it is the original tonic of the piece. But, apparently, the most common pitch-class has changed for the last quarter of the piece since the last three Eb notes are not considered as 'not interesting'. When we investigated this issue, we found that the most common pitch-class was changed to F or G depending on the subject note. This is one of the limitations of this baseline method, where the calculated specific musical feature

[1] https://soundcloud.com/user-330551093/sets/identifying-critical-points

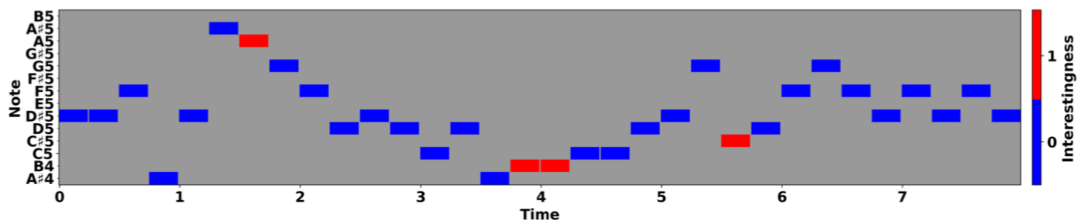
(a)



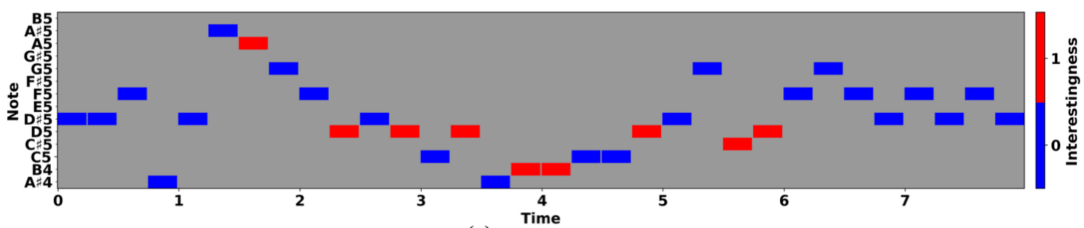
(b)



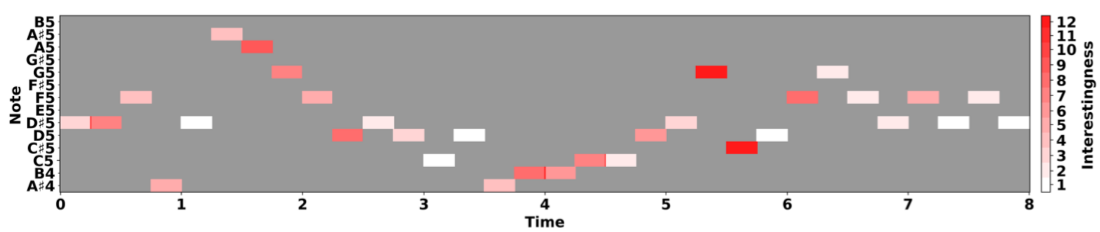
(c)



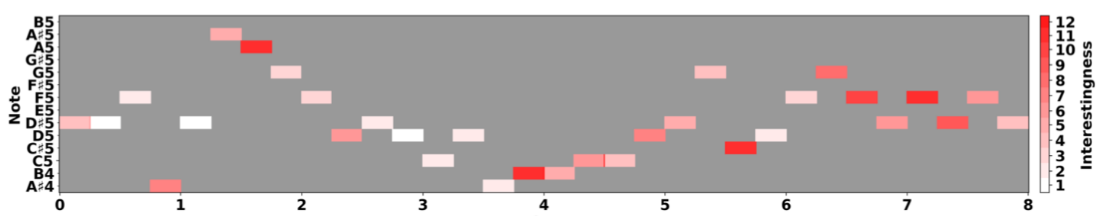
(d)



(e)



(f)



(g)

Fig. 3. (a) A processed excerpt from Chopin's Nocturne in E flat major (Op. 9 No. 2), (b) within piece analysis based on the most frequent pitch-class with w = 51, and (c) with w = 101, (d) within piece analysis based on the tonality with w = 51, and (e) with w = 101, (f) corpus-based analysis using LSTMs trained on the classical period music, and (g) the 20th century music.

that is at the core of the analysis might change temporally. When the window size is 101 (Figure 3 (c)), seemingly the most common pitch-class is calculated as F. We further investigated these results, and found that for the subject note of Db in the third measure, the most common pitch-class is detected as G. This can happen given the nature of algorithm where we have a moving window, but this is a limitation of this approach, where the most common pitch-class changes at a granular level. Also, these results show that the window size affects the calculation of the most common pitch-class.

In the tonality-based analysis, for the window size of 51, whose results are shown in Figure 3 (d), all the notes out of the Eb major scale are classified as interesting, which is reasonable as the tonality of the original piece is Eb major. But, as we know from the most common pitch-class analysis with the window size of 51, for the last quarter of the piece, the calculated most common pitch-class is not Eb, instead, it is F or G. So, this suggests that the inferred tonality for the last quarter of the piece cannot be Eb major and the notes in the last quarter are not classified with respect to Eb major scale. Even though the analysis seems to correctly identify the non-tonal notes considering the original tonality, this example shows that it can be a coincidence, which is a limitation of this method. When we have the window size of 101, F is the most common pitch-class throughout the piece as shown in the analysis above. We investigated the tonality in this case, and F minor was inferred throughout the excerpt except for the Db note in the third measure, where the tonality was inferred as G minor. Given these inferred tonalities, interestingness analysis classifies all the notes correctly, but since F minor is not the original tonality of the piece and it fluctuates for one note (Db), these results show the limitation of this method, where the inferred tonality differs depending on the window size and might deviate at a note level.

These results from the baseline methods show the limitations of these techniques. Typical time series analysis approaches might not be reliable and informative, and they rely on a single musical metric (such as the most common pitch-class or the tonality) that depends on the window size. Also, they might fluctuate at a granular level for a subject note. So, these limitations encourage us to use a machine learning-based approach that does not rely on a specific musical metric and is able to model complex interdependencies and patterns in the data.

We analyse the same excerpt with our suggested machine learning-based approach using the models trained with the classical period and the 20th century data, whose results are shown in Figure 3 (f) and (g), respectively. The classical period model identifies all of the non-tonal pitches as highly interesting reasonably. These pitches are A natural in the first measure, B natural in the second measure, and B natural and Db in the third measure. Also, it classifies the G before the Db in the third measure as one of the most interesting notes. Arguably, this might be due to having a relatively high interval between the subject note and the previous note, where the subject note corresponds to the beginning of the musical theme. The last seven notes are classified as some of the least interesting ones and this makes sense as this figure is very common in the classical period. When we analyse the excerpt with our 20th century model, all the non-tonal pitches are classified as the most interesting notes except the B natural at the beginning of the third measure as shown in Figure 3 (g). This might be due to having a repetition of the same pitch. Also, in contrast to the classical period model, the last seven notes are classified as some of the highly interesting notes. This is reasonable in the context of 20th century music as these kinds of musical figures are much rarer in the 20th century compared to the classical period.

Compared to the baseline methods, our suggested machine learning-based approach is more capable of capturing tonality-related interestingness and also referring to other musical features as we've pointed out above. Also, our machine learning-based approach provides more information about interestingness on a 12-level scale compared to the baseline methods that have a binary scale. The results of the machine learning-based approach need further investigation from a deeper compositional point of view, which we plan to perform as part of our future work including some human-subject evaluations as well.

IV. Conclusions

In this study, we present a novel machine learning-based approach for identifying critical points in musical compositions, which reveal atypical and unexpected decisions. We compare our suggested method to our baseline methods using various musical examples. We evaluate the results, addressing the limitations of the baseline methods and point out the advantages and the potential of our suggested machine learning-based approach. Identifying critical decision points in musical compositions can be useful for automatic music generation models that can be conditioned on the interestingness and also music understanding studies.

In our future work, we plan to conduct human-subject studies with composer and non-composer participants to evaluate the interestingness levels of the notes in various musical pieces, which would allow us to further analyse the performance of our suggested methodology. Also, we are interested in introducing other musical attributes such as dynamics and timing to the analysis, experimenting with other machine learning architectures as different inference mechanisms, and investigating their behaviour in the context of identifying critical decision points. Moreover, we will use the interestingness levels to condition generative music models, where musical pieces with continuous control of interestingness can be composed.

References

[1] T. Christensen, *The Cambridge history of Western music theory*. Cambridge University Press, 2006.

[2] C.-E. Cella, "Music information retrieval and contemporary classical music: A successful failure," *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, 2020.

[3] M. T. Pearce, M. H. Ruiz, S. Kapasi, G. A. Wiggins, and J. Bhattacharya, "Unsupervised statistical learning underpins computational, behavioural, and neural manifestations of musical expectation," *NeuroImage*, vol. 50, no. 1, pp. 302–313, 2010.

[4] D. Eck and J. Schmidhuber, "A First Look at Music Composition using LSTM Recurrent Neural Networks," *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale*, vol. 103, p. 48, 2002.

[5] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. M. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer: Generating music with long-term structure," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2019.

[6] B. Banar and S. Colton, "A systematic evaluation of GPT-2-based music generation," in *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*. Springer, 2022, pp. 19–35.

[7] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, "Midinet: A convolutional generative adversarial network for symbolic-domain music generation," *ArXiv*, vol. abs/1703.10847, 2017.

[8] G. Loy, "Musicians make a standard: the midi phenomenon," *Computer Music Journal*, vol. 9, no. 4, pp. 8–26, 1985.

[9] M. Schedl, "Deep learning in music recommendation systems," *Frontiers in Applied Mathematics and Statistics*, vol. 5, p. 44, 2019.

[10] S. Colton and G. Wiggins, "Computational creativity: The final frontier?" *Frontiers in Artificial Intelligence and Applications*, vol. 242, pp. 21–26, 01 2012.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] Q. Kong, B. Li, J. Chen, and Y. Wang, "GiantMIDI-piano: A large-scale midi dataset for classical piano music," *arXiv preprint arXiv:2010.07061*, 2020.

[13] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2014.