Turkish Classical Music Composition with LSTM Self-Attention

Ahmet KAŞİF [1*], Selçuk SEVGEN [2]

[1] *Bursa Technical University, Computer Engineering Department, Bursa, Turkey*
[2] *İstanbul University-Cerrahpaşa, Computer Engineering Department, Istanbul, Turkey*

Cite this paper as: Kasif A. and Sevgen S. (2024).Turkish Classical Music Composition with LSTM Self-Attention, 8(1):25-35.

*Corresponding author: Ahmet KAŞİF
E-mail: ahmet.kasif@btu.edu.tr

Received Date: 19/12/2023
Accepted Date: 01/02/2024

Abstract

Synthetic symbolic music generation, the process of creating new musical pieces using symbolic representations, has gained significant traction in the field of music informatics and computational creativity. It holds immense potential for various applications, ranging from music education and composition assistance to music therapy and personalized music recommendation systems. Classical Turkish music (CTM) exhibits distinct characteristics regarding Western Tonal Classical Music (WCTM) such as melodic organization, formation of rhythmic structure, or melodic expressions. This study tackles the challenge of symbolic music composition, focusing on CTM. Unlike its Western counterpart, CTM incorporates microtonal intervals. These intervals are smaller than the semitones in Western music, allowing for a more nuanced expression of pitch. This leads to a more diverse set of pitch ranges. The proposed method employs a combination of long-short term memory (LSTM) networks and self-attention encoding to capture long-term relational information and generate realistic CTM compositions. LSTMs effectively model sequential dependencies and improve local relations within musical structures, and self-attention improves the context vector, allowing the model to attend to different aspects of the musical context simultaneously. This combination enables the proposed method to generate compositions that are both musically coherent and stylistically consistent with distinct features of CTM. The proposed method was evaluated on two datasets, the SymbTr dataset and Classical Music Piano (CPM) dataset. The assessment of musical contents is evaluated through melodic similarity and stylistic consistency metrics. The results demonstrate that the proposed method is able to generate musical content that is coherent and to produce music that is pleasing-to-hear. Overall, the article presents a novel and effective approach to symbolic music composition, focusing on CTM.

*Keywords: M*usic-Generation, Deep Learning, LSTM, Self-Attention, Music Information Retrieval, Turkish Classical Music.

1. Introduction

The emergence of synthetically generated music has revolutionized the field of music informatics, offering unprecedented opportunities to expand musical diversity, enhance creativity, and personalize music experiences. However, generating music that adheres to the stylistic conventions and expressive nuances of a particular musical tradition remains a significant challenge. This study explores the intricate world of Classical Turkish Music (CTM), a rich musical heritage characterized by its unique microtonal ornamentation, intricate rhythmic patterns, and elaborate ornamental devices. Western classical tonal music (WCTM) gets the majority of attention in terms of generative musical analysis [1]. Yet, regional musical approaches such as CTM pose distinct possibilities and could provide major contributions to the musical world. While there are many similarities between these two musical approaches, some differences also exist. WCTM relies on half-steps as the primary melodic intervals while CTM embraces microtonal intervals, creating subtle pitch variations that add a distinctive flavour to its melodies [2-4]. Rhythmically, CTM showcases complex patterns with uneven subdivisions and syncopation, distinguishing it from the more predictable rhythmic structures of WCTM. Generating symbolic music that faithfully adheres to the rules and expressiveness of CTM presents several formidable challenges. Capturing the intricacies of microtonal intervals, accurately modelling complex rhythmic patterns, and incorporating the nuances of CTM's modal system are just a few of the hurdles that must be overcome. Additionally, ensuring stylistic consistency and achieving originality are crucial aspects of generating compelling CTM compositions. State-of-the-art methods in symbolic music generation have made significant progress, employing techniques such as recurrent neural networks (RNNs) and long short-term memory (LSTM) networks to capture long-term musical relationships and generate coherent compositions. However, these methods often struggle to fully capture the intricate details and expressive qualities of CTM. This study introduces the application of LSTM self-attention to symbolic music generation for CTM. Self-attention, a powerful technique in the field of natural language processing, enables the model to simultaneously attend to multiple aspects of the musical context, facilitating the capture of the subtle nuances and expressive details that characterize CTM. By incorporating self-attention, the proposed method aims to overcome the limitations of traditional symbolic music generation techniques and produce high-quality CTM compositions that accurately reflect the stylistic conventions and expressiveness of the tradition. To evaluate the effectiveness of the proposed method, two datasets are employed: the SymbTr dataset and the Classical Music Piano (CPM) dataset. The SymbTr dataset provides a comprehensive collection of CTM melodies annotated with microtonal intervals and rhythmic information [5]. The CPM dataset offers a diverse range of Western classical music pieces for comparison and evaluation [6]. The proposed algorithm holds significant promise for advancing the field of symbolic music generation and fostering a deeper appreciation for CTM. By generating high-quality CTM compositions, this research can contribute to the preservation and revitalization of this rich musical heritage, expanding the boundaries of musical creativity and enriching the musical landscape. The main contributions of this study are as follows:

- We demonstrate that LSTM-self-attention networks can effectively capture long-term relational information in classical Turkish symbolic musical sequences.
- We show that attentional networks can tackle the musical content generation problem on musical contents other than Western Tonal Music and provide new ways to integrate state-of-art knowledge on local musical approaches.

- We analyse the performance of attentional networks in different architectural configurations and provide a comparative study for future symbolic music generation research on CTM.

The rest of this study is organized as follows: section two provides a detailed analysis of the state of art in musical content generation and summarizes the methods with a comparative manner. Section three covers the presentation of the datasets, conducted input preprocessing steps and the experimental environment followed by the detailed expression of the proposed method and the evaluation metrics. Section four provides the results of assessment analysis. The study is concluded in section five with a discussion of the paper as well as future works.

2. Literature Review

With its distinct melodic structures called maqam as well as monophonic nature and unique rhythmic features, CTM presents a sonic realm distinct from its Western counterpart. This distinctiveness has posed a significant challenge for Music Information Retrieval (MIR) research, particularly in the domain of generative modelling. While numerous musical forms have served as fertile ground for MIR investigations, publicly available datasets dedicated to CTM have remained scarce. However, the emergence of SymbTr, boasting over two thousand MIDI-encoded pieces, marks a pivotal moment in facilitating comprehensive research endeavors [4]. This valuable resource has already fuelled explorations in areas such as music recommendation systems and maqam classification, demonstrating its potential to unlock further insights into the complexities of CTM [7, 8]. Still, the applications on generative domain for CTM is not properly analysed and lacks attention.

The current MIR research is conducted on two major fronts called signal domain and symbolic domain. The symbolic music research requires a high-level representation of musical features such as MIDI encoding to provide a more human-readable format to operate [1]. This representation provides a clear temporal structure. Noteworthy studies for symbolic music generation have highlighted that the use of RNNs and their variations such as LSTMs and Gated Recurrent Unit networks (GRU) produce satisfying results on modelling of short-term sequences but fail to accommodate their performance as the sequence length increases [9-11].

The advent of attention mechanisms marked a game-changer, initially demonstrating remarkable success in applications within Natural Language Processing (NLP) [12, 13]. This success has quickly been translated to other domains, including symbolic music research, where attentional networks have brought forth the possibility to improve modelling of the salient features within musical sequences, leading to significant advancements in tasks like maqam classification and sequence prediction [14, 15].

With the emergence of Deep Learning, significant research has been conducted to explore the characteristics of the Western Tonal Music. However, the distinct nature of CTM made it difficult to apply the gained knowledge to its domain. Thus, there has been only a handful of research in terms of generative modelling in Turkish Music, much less in CTM. Tanberk and Tukel proposed a combined CNN-LSTM network to generate Turkish pop music using a collected dataset which can provide style-specific content [16]. Aydıngun et. al used a collected Classical Turkish Music dataset which again consists of 20 pieces to generate Turkish songs with lyrics [17]. Both studies use small-sized collected datasets and do not offer benchmark results for music generated musical content.

3. Materials and Methods

3.1. Dataset Description

SymbTr and CPM, two symbolic datasets, are employed in the proposed music synthesis analysis. Both datasets consist of MIDI-encoded files. The SymbTr dataset contains around two thousand pieces for Turkish maqam music, which is the focus of exploration for the study. The CPM dataset contains piano compositions for Western tonal music and contains around 200 musical files. Both datasets are parsed as monophonic music sheets. Chords are parsed as notes, using the base note of the chord as the pitch symbol. The CTM dataset yields less sequences for training than CPM counterpart dataset as the majority of pieces are shorter than pieces in CPM. The summary of features of both datasets are given in Table 1.

Table 1. Data Analysis for SymbTr and CPM datasets

Dataset	SymbTr	CPM
Number of Pieces	1931	221
Number of Unique Pieces	33	85
Number of Unique Durations	48	51
Number of Prepared Input Sequences	48.2k	300k
Average / Maximum Piece Length	325 / 1467	709 / 4312

The MIDI format includes valuable information but cannot be directly supplied to deep learning architecture as an input line. Therefore, the pieces in the datasets are preprocessed using "music21" music processing library into an array-type format [18]. Two features (pitch, duration) are extracted from the pieces. Pitch indicates the frequency class of the played sound and demonstrates a categorical feature. The second feature, duration, is the playing duration for the respective note and is a numerical feature.

3.2. Experimental Environment

The proposed model was developed using Python 3.9.7 and Tensorflow/Keras 2.11.0. A Grid-Search technique was used to optimize the hyperparameters on a computer cluster at the B.T.U High-Performance Clustering Laboratory (HPCLAB). The computers on the BTU-HPCLAB cluster have $Intel® \ Core^{TM}$ i9-10900X CPUs and Nvidia 3090 GPUs. We have employed the CPUs to prepare, preprocess, and analyze the data, and the GPUs to train the model. Two GPUs are utilized simultaneously to accelerate the training process.

3.3. Input Preprocessing

The mapping from MIDI files to a symbolic music dataset requires multi-step preprocessing, which includes the parsing of musical files, decoding temporal information, mapping the note-level data to symbols, and normalization and preparing sequences. We have used Music21 to parse and decode MIDI files. Music21 is a powerful open-source tool which can represent and process various musical formats including MIDI with rich set of musical analysis tools [19]. Pitch and octave values for a single note is then processed together into a pitch symbol, and duration is encoded as a numerical value with a four-digit precision. Temporal input sequences data is constructed from the array data containing symbolic pitch and numerical duration values. The availability of much larger sequences in the CTM dataset gets lower after sequence length threshold of 64, which would hinder the training and also generative performance of the framework. Thus, sequence length of 64 notes has been utilized. The values for the duration feature have been normalized, using min-max normalization [20]. Min-Max normalization is a technique which maps a given array of numerical values. The minimum element is mapped to zero, and the maximum element is mapped to one.

3.4. Proposed Method

The proposed method consists of a recurrent layer to model temporal musical data as well as a self-attention layer to improve the context vector for longer sequences. The recurrent layer accepts the input features and provides an encoded context vector. The context vector is then supplied into a self-attention module where important relations between notes are emphasized. The proposed DL model architecture is depicted in Figure 1. Two input vectors (note, duration) are supplied to the framework which are embedded by using a dedicated embedding layer. Both embeddings are then concatenated into a one single vector. The combined vector is modelled in two consecutive LSTM layers to create a context vector. This vector is then employed in a self-attention layer to improve the range of relations as well as to weight the important relations. The proposed self-attention layer is the standard self-attention configuration.

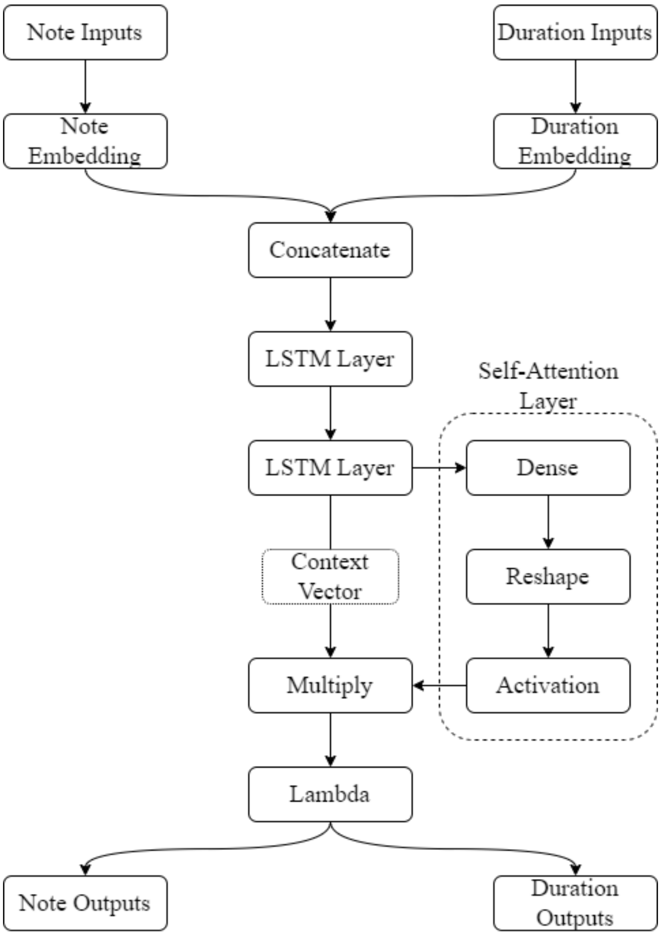


Figure 1. Proposed LSTM Self-Attention Model Architecture

3.5. LSTM Layer

LSTM networks are a type of RNNs with the addition of a memory cell. The most major concern of RNN is that while they are powerful in short sequences, the performance degrades as the length of sequence increases [21]. The cause of this is called vanishing gradients. The gradients fade as the information has to propagate through the temporal network. The LSTM cell improves the information flow through sequences and enables more accurate representation of much longer sequences.

LSTM neurons consist of three gates and a cell state. Cell state is the memory unit of an LSTM neuron which runs information through an entire chain of LSTM layer. LSTM gates are responsible for data manipulation through cell states. The new information enters an LSTM neuron through input gate. The gate selectively adds new information to the cell state. The forget gate decides on the importance of information existing on cell state and discards if the information is found irrelevant. The output gate produces an output for the LSTM neuron based on cell state.

Self-attention networks improve the context vector of RNNs and provide stronger relations between more distant points in a sequence. The context vectors produced by RNN layers provide a relational information between notes, but this relational information weakens due to the mathematical nature of the function and yields a loss of gradients. This loss enlarges as the distance between notes get longer.

The self-attention employs 3 matrices (Q, K, V) to overcome the effect of loss of long-term dependencies and provide more direct relations between distant points in sequences. This mechanism has been one of the most important features in the success of Transformers architectures [22]. The vector Q stands for query vector, K stands for key vector, and V stands for value vector.

Query vectors represent the questions the model asks about each element in the input sequence. Key vectors represent the answers to the questions provided by the query vector and increase weights of attention from more related questions. Value vectors represent the actual information and are used to build the context-aware representation based on the attentive scores calculated by using query and key vectors.

The attention score is calculated with the help of the dot product of the query and key vectors as shown in equation 1. The scores are then normalized in equation 2, using a softmax function to create a probability distribution over the elements of the sequence. Finally, the normalized attention weights are employed to weight the value vectors and provide a weighted sum in equation 3. The produced output represents the context-aware encoding of the element with respect to the query.

$$\text{Attention Scores} = \frac{QK^T}{\sqrt{d_K}} \tag{1}$$

$$\text{Attention Weights} = \text{softmax} * \text{Attention Scores} \tag{2}$$

$$\text{Context} = \text{Attention Weights} * V \tag{3}$$

3.6. Evaluation Metrics

Throughout the deep learning (DL) algorithm's training phase, Root Mean Square Error (RMSE) metric is employed to assess the model's performance. The metric gauge distance-based approximation and has a track record of effectively refining mathematical problems. As shown in Equation 4, a smaller RMSE value denotes a closer match between the original and artificially generated content. Essentially, RMSE offer crucial insights into the DL algorithm's performance, enabling adjustments and enhancements to the model.

$$RMSE = \sqrt{\frac{1}{N} * \sum_{L=1}^{N} (Y_p - Y_c)^2} \tag{4}$$

4. Result and Discussion

Qualitative experiments are conducted through employment of proposed assessment metrics, and yielding model training scores are depicted in Table 2. The first three columns show the training loss values for CTM analysis, while the last three rows depict the training loss values for WCTM analysis. Hyper-parameters are fixed amongst compared methods for training phase. Both the baseline models and the proposed model were trained for 50 epochs, batch sizes were decided as 256, all methods employed two LSTM layers (RNN layers for the RNN-only architecture) with 128 neurons for each layer, optimizer was selected as RMSProp, and optimizer learning rate was set to 0,001. These hyperparameters were found to be yielding best results with minimal computational complexity. The hyperparameter search was conducted using Grid-Search method, which effectively looks up for all possible combinations of hyperparameters. The hyperparameter search space is given in Table 3.

The training loss analysis yields comparable results in case of both datasets. The duration loss contributes less to total loss compared to pitch loss as the duration loss is encoded using numerical representation. The total loss is calculated by using a weightless sum of pitch loss and duration loss. Evaluation of the proposed LSTM Self-Attention method against baseline methods in training phase results in better loss values in terms of all metrics. The baseline LSTM shows better overall performance against baseline RNN, but both baseline methods (RNN and LSTM) fall back against the proposed method in terms of loss metrics in training performance.

Table 2. Model performance regarding objective metrics, first 3 columns for CTM, last 3 columns for WCTM

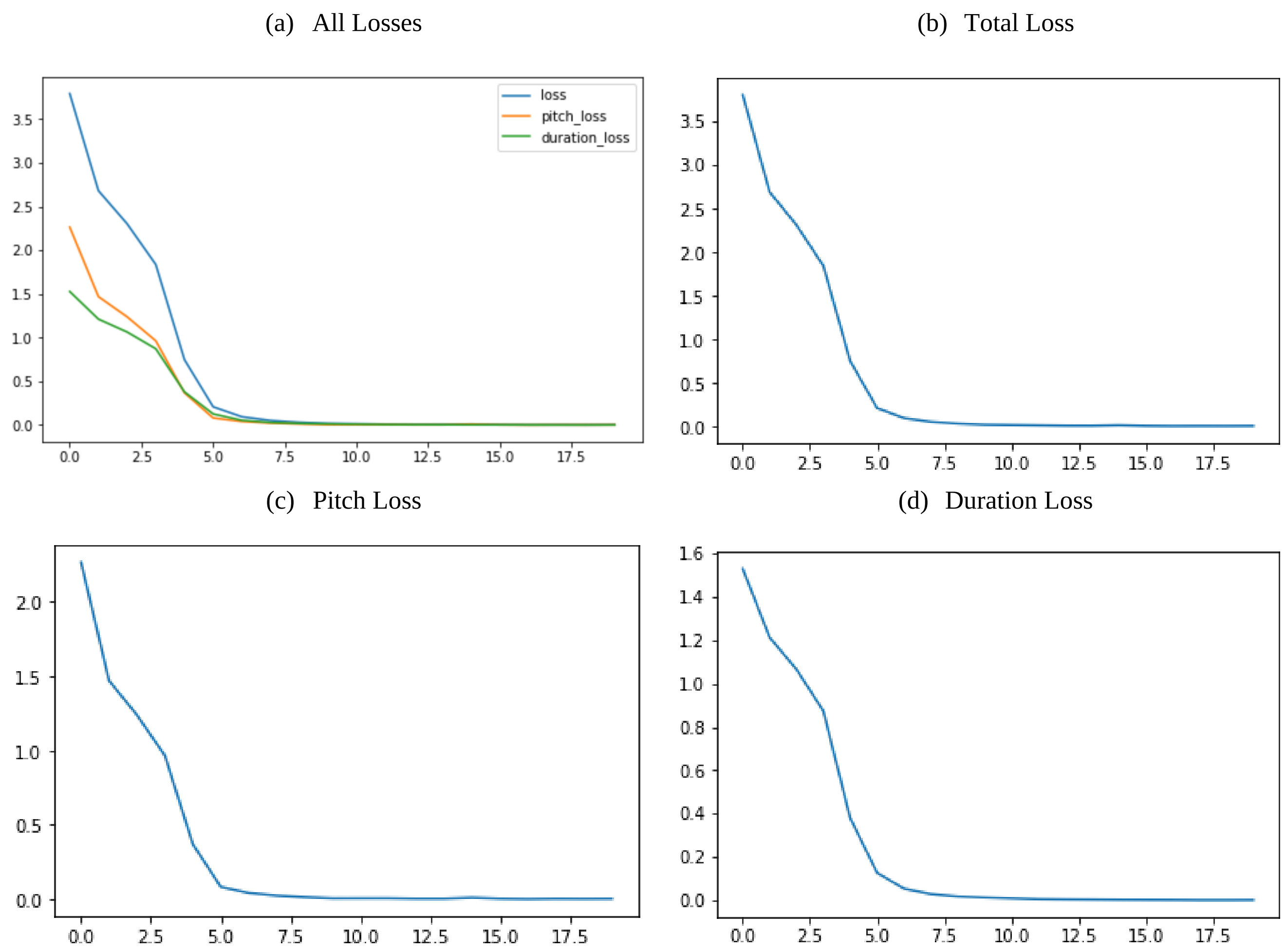
Assessment Type	Methods	RMSE(Pitch)	RMSE(Duration)	RMSE (Total)
CTM Assessment	RNN	0.938	0.004	0.942
	LSTM	0.619	0.002	0.621
	LSTM Self-Attention	0.00022	0.00002	0.000024
WCTM Assessment	RNN	0.816	0.003	0.819
	LSTM	0.524	0.002	0.526
	LSTM Self-Attention	0.00024	0.00001	0.000025

Table 3. Hyperparameter search space

Hyperparameter	Search Space	Best Fit
Layer Neuron Count	[32, 64, 128, 256, 512]	128
Optimizer Selection	[RMSProp, Adam]	RMSProp
Optimizer Learn Rate	[0,01-0,0001]	0,001
Batch Size	[64, 128, 256, 512, 1024]	256

The feature-level loss curves for the total loss, pitch loss, and duration loss as well as all losses for CTM assessment are given in Figure 2. The loss curves show that the sharp decrease on the first epochs is supported with the continuous improvement over the remaining epochs, finally reaching a plateau. This behavior shows that the model has executed a healthy training process. Comparing the feature-wise loss curves, the pitch loss possesses the highest complexity and contributes the most to total loss as it is encoded as a categorical feature.

A musical piece generated by the proposed LSTM Self-Attention is depicted by using Western musical notation in Figure 3. The generated piece is 96 notes long. The piece shows a descending melodic line along with the use of perfect fourth, which is a common approach in many maqams of CTM. Also, the rhythmic development poses characteristic features of CTM.



(a) All Losses

(b) Total Loss

(c) Pitch Loss

(d) Duration Loss

Figure 2. Proposed LSTM Self-Attention Model Training Loss Curves (a) All Losses, (b) Total Loss, (c) Pitch Loss, (d) Duration Loss

The melody starts with a quick flurry of notes in the first measure, followed by slower, sustained notes in the next two measures. This creates a sense of contrast and dynamism, but it is still a simple pattern compared to the intricate rhythmic interplay and layering often found in Turkish music. The representation of rhythmic features using a duration object provides a simple solution, yet it provides a room for improvement for future works. A more complex representation would be addressed to cope with the high-variational tempo characteristic of CTM.

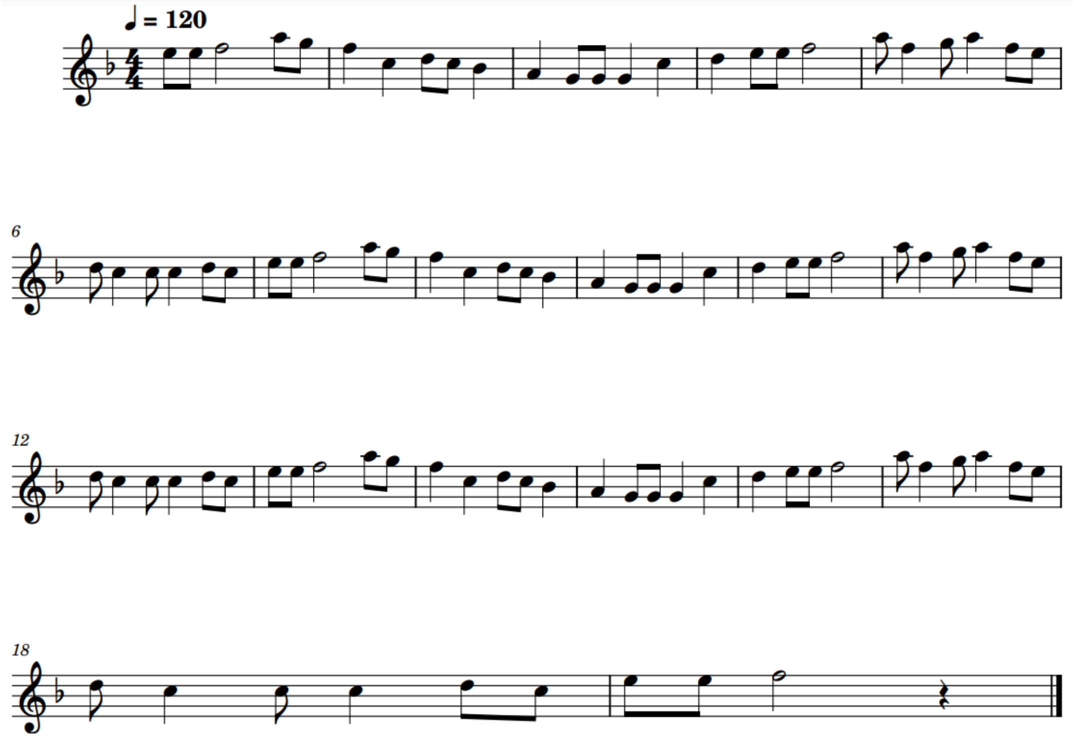


Figure 3. Synthetic CTM piece generated with the proposed method of LSTM Self-Attention

5. Conclusion

Synthetic symbolic music generation has emerged as a promising avenue within music informatics and computational creativity, offering the promising prospect of emulating and potentially even surpassing human compositional capabilities. This study specifically delves into the realm of Classical Turkish Music (CTM), woven from intricate melodic structures, dynamic rhythmic tapestry, and a microtonal palette that expands the expressive boundaries of pitch. Unlike its Western Tonal Classical Music (WCTM) counterpart, CTM bends the rigidity of major and minor scales, instead embracing the fluidity of maqams with their unique modal configurations and expressive microtonal nuances. These maqams, with their intricate ornamentation and characteristic melodic contours, paint a sonic landscape with emotional depth and increased expressive potential. This interplay of maqams and rhythms creates a rich set of musical possibilities, demanding a model capable of capturing both the long-term melodic coherence and the nuanced rhythmic details that define CTM. To meet this challenge, we propose a novel approach that leverages the combined strengths of LSTM and self-attention networks. Self-attention mechanisms, inspired by recent advances in natural language processing, empower the model to "pay attention" to various musical elements within the sequence simultaneously, allowing it to grasp the subtle interplay between maqams, rhythms, and individual notes. LSTM networks, on the other hand, excel at capturing long-term dependencies within the musical sequence, ensuring the generated compositions exhibit a smooth flow and melodic coherence. The efficacy of this approach is evident in the generated CTM pieces, which showcase a remarkable level of musical coherence and stylistic consistency. Evaluated against established SymbTr and CPM datasets, our method not only demonstrates its ability to produce musically pleasing compositions, but also sticks to the stylistic nuances that define CTM. Quantitative metrics confirm that our generated compositions successfully capture the essence of CTM and resonate with the listener's expectations of this genre. The proposed method not only contributes to the field of music informatics by advancing the state of the art in symbolic music generation, but also opens new doors for exploring the diversity and complexity of CTM. Future research can leverage this work as a springboard to delve deeper into the theoretical underpinnings of CTM and develop even more sophisticated models capable of capturing the full spectrum of its expressive potential. The effect of using musical features such as volume and tempo can be investigated to obtain richer representations. By doing so, we can not only preserve and celebrate this treasured musical heritage but also pave the way for exciting new artistic creations that draw inspiration from its rich and timeless beauty.

References

[1] Briot, J. P., & Pachet, F. (2020). Deep learning for music generation: challenges and directions. Neural Computing and Applications, 32(4), 981-993.

[2] Bozkurt, B., Gedik, A. C., & Karaosmanoglu, M. K. (2009, April). Music information retrieval for Turkish music: problems, solutions and tools. In 2009 IEEE 17th Signal Processing and Communications Applications Conference (pp. 804-807). IEEE.

[3] Kızrak, M. A., & Bolat, B. (2017). A musical information retrieval system for Classical Turkish Music makams. Simulation, 93(9), 749-757.

[4] Karaosmanoğlu, M. K. (2012). A Turkish makam music symbolic database for music information retrieval: SymbTr. In Proceedings of 13th International Society for Music Information Retrieval Conference; 2012 October 8-12; Porto, Portugal. Porto: ISMIR, 2012. p. 223–228. International Society for Music Information Retrieval (ISMIR).

[5] Krueger, B. Classical piano midi page (2016). URl: http://www.piano-midi.de/(Last accessed 28/11/2023).

[6] Ay, G., & Akkal, L. B. (2009). İTÜ Türk Musikisi Devlet Konservatuarı Türk müziğinde uygulama-Kuram sorunları ve çözümleri. Uluslararası çağrılı kongre bildiriler kitabı.

[7] Öztürk, Ö., Özacar, T., & Abidin, D. (2018, September). KORAL: Türk Müziği için Makam Tabanlı Öneri Motoru Tasarımı. In 2018 International Conference on Artificial Intelligence and Data Processing (IDAP) (pp. 1-4). IEEE.

[8] Abidin, D., Öztürk, Ö., & Öztürk, T. Ö. (2017). Klasik Türk müziğinde makam tanıma için veri madenciliği kullanımı. Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 32(4), 1221-1232.

[9] Mangal, S., Modak, R., & Joshi, P. (2019). LSTM based music generation system. arXiv preprint arXiv:1908.01080.

[10] Shah, F., Naik, T., & Vyas, N. (2019, December). LSTM based music generation. In 2019 International Conference on Machine Learning and Data Engineering (iCMLDE) (pp. 48-53). IEEE.

[11] Wu, J., Hu, C., Wang, Y., Hu, X., & Zhu, J. (2019). A hierarchical recurrent neural network for symbolic melody generation. IEEE transactions on cybernetics, 50(6), 2749-2757.

[12] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[13] Niu, Z., Zhong, G., & Yu, H. (2021). A review on the attention mechanism of deep learning. Neurocomputing, 452, 48-62.

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

[15] Muhamed, A., Li, L., Shi, X., Yaddanapudi, S., Chi, W., Jackson, D., ... & Smola, A. J. (2021, May). Symbolic

music generation with transformer-gans. In Proceedings of the AAAI conference on artificial intelligence (Vol. 35, No. 1, pp. 408-417).

[16] Tanberk, S., & Tükel, D. B. (2021, January). Style-specific Turkish pop music composition with CNN and LSTM network. In 2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI) (pp. 000181-000185). IEEE.

[17] Aydıngün, A., Baglu, D., Canbaz, B., & Kökbıyık, A. Derin Ögrenme ile Türkçe Sarkı Besteleme Turkish Music Generation using Deep Learning.

[18] Cuthbert MS, Ariza C (2010) music21: A toolkit for computer-aided musicology and symbolic music data. Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010).

[19] Cuthbert, M. S., & Ariza, C. (2010). music21: A toolkit for computer-aided musicology and symbolic music data.

[20] Patro, S.G.K, & Sahu, K. K. (2015). Normalization: A preprocessing stage. arXiv preprint arXiv:1503.06462.

[21] Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6(02), 107-116.

[22] Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. AI Open.