Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 3107 – 3118
© TÜBİTAK
doi:10.3906/elk-2101-44

Deep learning for Turkish makam music composition

İsmail Hakkı PARLAK[1,*], **Yalçın ÇEBİ**[1], **Cihan IŞIKHAN**[2], **Derya BİRANT**[1]
[1]Department of Computer Engineering, Faculty of Engineering, Dokuz Eylül University, İzmir, Turkey
[2]Department of Musicology, Faculty of Fine Arts, Dokuz Eylül University, İzmir, Turkey

Received: 11.01.2021 • **Accepted/Published Online:** 20.05.2021 • **Final Version:** 30.11.2021

Abstract: In this paper, we introduce a new deep-learning-based system that can compose structured Turkish makam music (TMM) in the symbolic domain. Presented artificial TMM composer (ATMMC) takes eight initial notes from a human user and completes the rest of the piece. The backbone of the composer system consists of multilayered long short-term memory (LSTM) networks. ATMMC can create pieces in Hicaz and Nihavent makams in Şarkı form, which can be viewed and played with Mus2, a notation software for microtonal music. Statistical analysis shows that pieces composed by ATMMC are approximately 84% similar to training data. ATMMC is an open-source project and can assist Turkish makam music enthusiasts with creating new pieces for professional, educational, or entertainment purposes.

Key words: Turkish makam music, automatic composition, deep learning, machine learning

1. Introduction

By imitating and extending the musical creativity of composers, artificial intelligence (AI) can assist artists to create new musical pieces. Breakthroughs in artificial neural networks (ANNs) and deep learning (DL) techniques have led AI to create impressive results for automatic music composition both in waveform domain [1, 2] and symbolic domain [3–5]. Automatic music generation research in the waveform domain requires much higher computational resources than the symbolic domain. In a standard audio CD, songs are encoded at 44.1 kHz sampling rate [6], which means 1 s of audio consists of 44.1 $\times 10^3$ data points. Symbolic representation of music occupies much fewer data and faster turnaround time; therefore, the majority of automatic music generation research dwells on the symbolic domain [7]. Whether it is pop, rock, jazz, or classical music, the studies on automatic music composition are mostly carried out around western symbolic music. Unfortunately, there is scarcely any work done on the automatic composition of Turkish makam music (TMM), which is nourished by its thousands of years of history and its roots spread to three continents.

TMM differs from western music in many aspects [8]. First of all, TMM is considered a heterophonic/monodic genre; whereas, western music has a polyphonic structure [9]. While performing a piece, TMM musicians play the same melodic idea, but also, they incorporate their virtuosity and decorate the piece by slightly altering melodic phrases or pitches, or they play the same melody in different octaves [10, 11]. On the other hand, western music derives its prosperousness from simultaneous but independent melodies creating harmony.

Secondly, the most obvious difference between TMM and western music is their tuning systems. Western

*Correspondence: ismail@cs.deu.edu.tr

music is formed on 12 equally spaced divisions (semitones) of an octave [12, 13]; whereas, TMM is considered to be based on 24 notes, which are established on 53 equally spaced divisions per octave [14]. Although there are different theories trying to formalize the TMM tuning system, the most widely accepted one is the AEU (Arel–Ezgi–Uzdilek) theory [15]. In AEU theory, a whole step is fractionalized into 9 equal parts each of which is called a "koma" [10]. Although a 1-koma interval may sound minuscule to a westerner or an untrained ear, TMM musicians can proficiently travel between these slight nuances and evoke a wide variety of emotions in listeners.

Last but not least, the two differ in their notions of tone series. In TMM, a Makam, which is an integration of tone series, motion, and ending of the movement, is built on combinations of tetrachords and pentachords [16]. Two Makams having the exact same tone series may have different names and feelings depending on their movements, which is called Seyir [17]. However, in western music, regardless of melodic movement, a tone series is named by its dominant or initial note.

In addition to Makam, two key terms regarding TMM are Usul and Form. Usul describes the temporal properties of the musical piece and can roughly be translated as "meter" [18]. Sequences of percussion strokes with varying accents in a fixed amount of time form Usuls [11]. Western music performances require a conductor who can control the crowd of performers; whereas, in TMM, the internal sense of Usul in musicians and well-played percussion are the sources of correct temporal coherence. Usuls in TMM can vary from a 2-beat usul Nim Sofyan to 124-beat usul Cihar [19]. Form in TMM jargon is used to describe the structural scheme of musical pieces [20]. TMM branches into various forms which are grouped under instrumental, vocal, or religious genres. As an example, Şarkı is a vocal form which consists of Zemin, Nakarat, Meyan, and Nakarat sections. Zemin is the introductory section that displays the characteristics of practiced Makam. Nakarat is the repeating and catchy section. Meyan is the section where a composer is free to stir higher registers and different Makams [21].

Music, like natural languages, can be considered as time-series data, where the ith element of the series is related to a subset of previous elements. Because of this, time series and natural language processing techniques perform successfully also on music [2, 22]. An AI model, which can predict the next note with respect to previous notes is the key element of automatic music composition. Long short-term memory neural networks (LSTM NN) can outperform Markov Chains, "vanilla" recurrent neural networks (RNN), and rule-based systems on learning long-term dependencies in time series data [23]. Internal gates of LSTM cells enable remembering necessary information as well as forgetting the unnecessary ones in the source dataset. Also, LSTM NNs do not suffer from vanishing/exploding gradient problems as vanilla RNNs do, which makes them very popular on time series forecasting tasks [24]. LSTMs are used for both monophonic and polyphonic music generation. Their success in melody generation makes them suitable for automatic TMM composition.

In artificial intelligence studies, evaluating the success of the developed model is as important as the creation of it. But this is not a very straightforward task for artistic generation models because measuring the quality of artificially composed music is mostly subjective [25]. However, Yang and Lerch [26] proposes a collection of metrics for objective evaluation of artistic generation systems. They suggest investigating sets of pitch and duration related features within and in-between training and generated data. In this context, if a feature is only measured within a single set, it is called an absolute metric. Absolute metrics deliver information related to the set from which it is extracted. Whereas, relative metrics are obtained by comparing training and generated sets. They suggest computing 5 pitch-related and 4 duration (rhythm) related features. Suggested pitch related features are:

1. Pitch count (PC): Number of unique (distinct) pitches without duration information per song/sample.
2. Pitch class histogram (PCH): Octave independent, normalized histogram of pitches. While computing PCH, La3♯5 and La4♯5 are considered to be equivalent pitches regardless of their octaves.
3. Pitch class transition matrix (PCTM): Octave independent, normalized transition matrix of pitches without rhythm information.
4. Pitch range (PR): The distance between the highest and lowest pitches per sample/song.
5. Average pitch interval (PI): Average pitch distance between successive notes per sample.

Suggested rhythm-based features are listed as follows:

1. Note count (NC): Number of unique (distinct) durations without pitch information per song/sample.
2. Average inter-onset-interval (IOI): The average elapsed time between consecutive notes.
3. Note length histogram (NLH): Normalized histogram of durations.
4. Note length transition matrix (NLTM): Normalized transition matrix between all durations.

To compute absolute metrics, Yang and Lerch [26] specify computing mean and standard deviation of each feature. To compute the relative metrics, first, pairwise exhaustive cross-validation should be performed. This process outputs a histogram of each feature's distances. Histograms are formed by calculating the Euclidian distances between samples at each cross-validation step. If the histogram is computed within a set, Yang and Lerch [26] label it as "intra-set distances"; whereas, if it is computed between different sets, it is labeled as "inter-set distances". Then, for smoothing resulting histograms, they suggest applying kernel density estimation. This process outputs probability distribution functions (PDF). Finally, Kullback–Leibler divergence (KLD) and overlapping area (OA) of inter-set and intra-set PDFs are computed. These metrics show the similarity of two datasets between each other and within themselves. According to the proposed method, to have similar intra-set Gaussian distributions, the variance of 2 sets should be similar. Whereas, to have similar inter-set distributions, mean values should be similar. Finally, Yang and Lerch [26] argue that the smaller the KLD value and the larger the OA value between intra-generated dataset and inter-sets, the more similar the generated dataset to the source dataset.

In light of the information given above, it can be concluded that the artificial composition of TMM requires a specially tailored system that accounts for its characteristic properties. In this paper, we present a novel, open-source, deep learning based artificial Turkish makam music composer (ATMMC) system, which is designed especially for TMM[1]. The presented system can compose structured pieces in Şarkı form for Hicaz and Nihavent Makams. To the best of the authors' knowledge, ATMMC is the first system that can create a complete piece in the symbolic domain having Zemin, Nakarat, and Meyan sections. Also, the composed piece correctly ends in the tonic note of the target Makam. Results of evaluation metrics, which are suggested by Yang and Lerch [26], are given in Section 4.

[1]TMMDLFT (2020). Artificial Turkish makam music composer [online]. Website https://github.com/ihpar/TMMDLFT [accessed 23 Sep 2020]

2. Dataset

Selection and encoding of source data on which the AI model is trained has a significant effect on the accuracy of the resulting model [27]. Unfortunately, compared to western music studies, the quantity of machine-readable public datasets for TMM is very low. SymbTr, which is the largest public dataset created for TMM, was chosen for this study.

2.1. SymbTr

SymbTr, a machine-readable compilation of TMM symbolic scores, is the largest dataset for computational TMM studies [14]. SymbTr houses 2200 pieces in a variety of Makams which are encoded in multiple digital formats such as MIDI, Mu2, MusicXML, Text, and PDF. Mu2 formatted symbolic score files can be displayed, edited, and played with Mus2[2], which is a microtonal notation software developed primarily for TMM. Each line in Mu2 files consists of tab-separated entities. Each line starts with a numeric code, followed by a note name such as Si4b5, Mi5b5, Do6, etc., followed by the duration of the note such as 1/16, 3/8, 1/32, etc., and followed by other information such as legato events, timestamps, etc.

ATMMC is limited to operate only on Hicaz and Nihavent Makams because they have the largest portions in the SymbTr dataset with 157 and 129 total pieces, respectively. Likewise, Şarkı form occupies the largest ratio of forms in Hicaz and Nihavent pieces. For Hicaz Makam, the most frequent Usul is Aksak, which has a 9/8 time signature. For Nihavent Makam, the most frequent Usul in SymbTr is Düyek, which is in 8/8 meter [28]. Thus, the proposed ATMMC is limited to Şarkı form in Aksak Usul for Hicaz Makam and Düyek Usul for Makam Nihavent only.

2.2. Data representation

There are numerous ways to represent and encode musical data, such as timestep sampling, numerical values, binary vectors, and textual representation to name but a few [25, 27, 29]. However, according to our experiments, for artificial TMM composition, one-hot-encoding of musical data induces better results than other encodings. A musical note is an amalgamation of pitch and duration. For example La4, 1/8 in a song at 60 bpm tempo corresponds to a 440 Hz oscillation, which lasts for 1/2 s. Thus, all notes and rests in any given song can be represented by pitch-duration tuples. In all Hicaz pieces of SymbTr, there are 405 unique pitch-duration tuples, which can be one-hot encoded into vectors v such that $v \in \{0,1\}^{405} : \sum_{i=1}^{405} v_i = 1$. Whereas, pieces in Makam Nihavent of SymbTr sum up to 374 unique pitch-duration tuples. This way, any musical piece in SymbTr can be represented by a series of one-hot encoded pitch-duration tuples.

3. Method

ATMMC consists of two equivalent sub-composers: one trained on Hicaz and the other one is trained on Nihavent Makam. After choosing a target Makam, a user enters 8 pitch-duration tuples to the system, and the system completes the remaining parts of the artificial composition and creates a Mu2 file which can be viewed and played by Mus2 software. Except for the dimensions of training data, the structural compositions of two sub-composers are equivalent. In this section, we describe Nihavent sub-composer. The same principles apply also to Hicaz sub-composer.

[2] Mus2 2.x (2020). Notation software for microtonal works [online]. Website https://mus2.com.tr/en [accessed 25 May 2020]

3.1. Core model

The core model (CM), as represented in Figure 1, is a 3-layer NN, where the first 2 layers consist of 600 LSTM neurons each, followed by a fully connected (dense) layer with a softmax activation function. There is a 50% dropout rate in between all layers. Model is compiled with categorical cross-entropy loss function and root mean square prop (RMSprop) optimizer having a learning rate of 1×10^{-3}. For Nihavent Makam, training data is encoded into a rank-3 input tensor x, where $x \in \{0,1\}^{30845 \times 8 \times 374}$ and rank-2 output tensor y, where $y \in \{0,1\}^{30845 \times 374}$. Here, for each 8 input pitch-duration tuples, there is 1 corresponding output pitch-duration tuple, which represents the next note following the given 8 notes. Nihavent Makam's dataset contains 30853 pitch-duration tuples which yield to $30853 - -8 = 30845$ input-output pairs. For Nihavent Makam, a pitch-duration tuple is represented by a one-hot encoded vector v, where $v \in \{0,1\}^{374}$. When trained on the dataset, CM can learn conditional probability distribution $P(x_t \mid x_1, ..., x_{t-1})$ of next pitch-duration pair at any given timestep t with respect to previous pitch-duration tuples and make predictions for the next pitch-duration tuple.

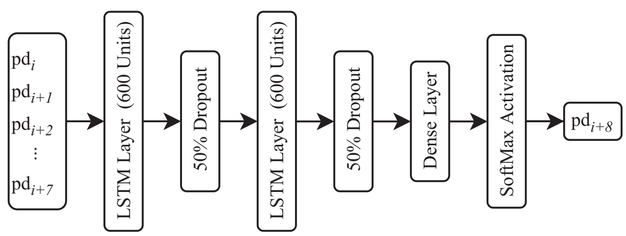

Figure 1. Illustration of core model's architecture. $pd_i, pd_{i+1}, pd_{i+2}, \ldots, pd_{i+7}$ denotes the 8 input pitch-duration tuples; whereas, pd_{i+8} represents the corresponding output pitch-duration tuple.

CMs have been trained on the whole target Makam dataset without explicit Usul or Form information. Also, Zemin, Nakarat, and Meyan sections of the CMs' training dataset were not labeled. This way, they learn general characteristics of targeted Makam over the largest subset of training data available. Later, CMs form the basis of other NNs, which specialize in certain Usul, Form, and sections of the pieces to be artificially composed. We trained different CMs with different hyperparameters over the same data and selected 2 best performing ones for each target Makam in order to provide a basis for specialist models. Best performing CMs were subjectively determined by musicians according to CMs' abilities to output the most suitable pitch-duration tuples for the target Makam they were trained on.

3.2. Specialist model

The specialist model (SM) has the same 3-layer NN structure as CM, but it is specialized in a certain Form, Usul, and section of the musical piece. The number of musical pieces in Şarkı form and Aksak and Düyek Usuls in SymbTr are not sufficient to train a model that can compose well-structured music. To overcome this challenge, we collected 88 additional pieces from various archives in sheet music format and manually translated these new pieces to Mu2 format, and labeled their Zemin, Nakarat, and Meyan sections. Then, SMs were trained on this new dataset, which has its Zemin, Nakarat, and Meyan sections labeled, inheriting their initial layers from already trained CMs. This technique, called transfer learning, is used to increase the accuracy of SMs by transferring the knowledge about the basics of targeted Makams from CMs. Transfer learning is beneficial when the size of the available dataset is insufficient [30].

For each section of Şarkı form, which are Zemin, Nakarat, and Meyan, a separate SM is trained. As shown in Figure 2, Zemin SM is trained to compose the Zemin section with respect to initial user input, which is 8 pitch-duration tuples. Nakarat SM is trained to compose the Nakarat section with respect to the last 8 pitch-duration tuples of the Zemin section, and finally, Meyan SM is trained to compose the Meyan section with respect to the last 8 pitch-duration tuples of the Nakarat section. This way, a harmony between consecutive sections within an artificial composition is established.

SMs operate on a similar domain to CMs denoted by $x \in \{0,1\}^{N \times 8 \times M} \rightarrow y \in \{0,1\}^{N \times M}$, where N is the number of samples and M is the count of unique pitch-duration tuples. We trained different SMs with different hyperparameters and selected 2 best performing ones for each section per Makam. The best performing SMs were subjectively determined by musicians.

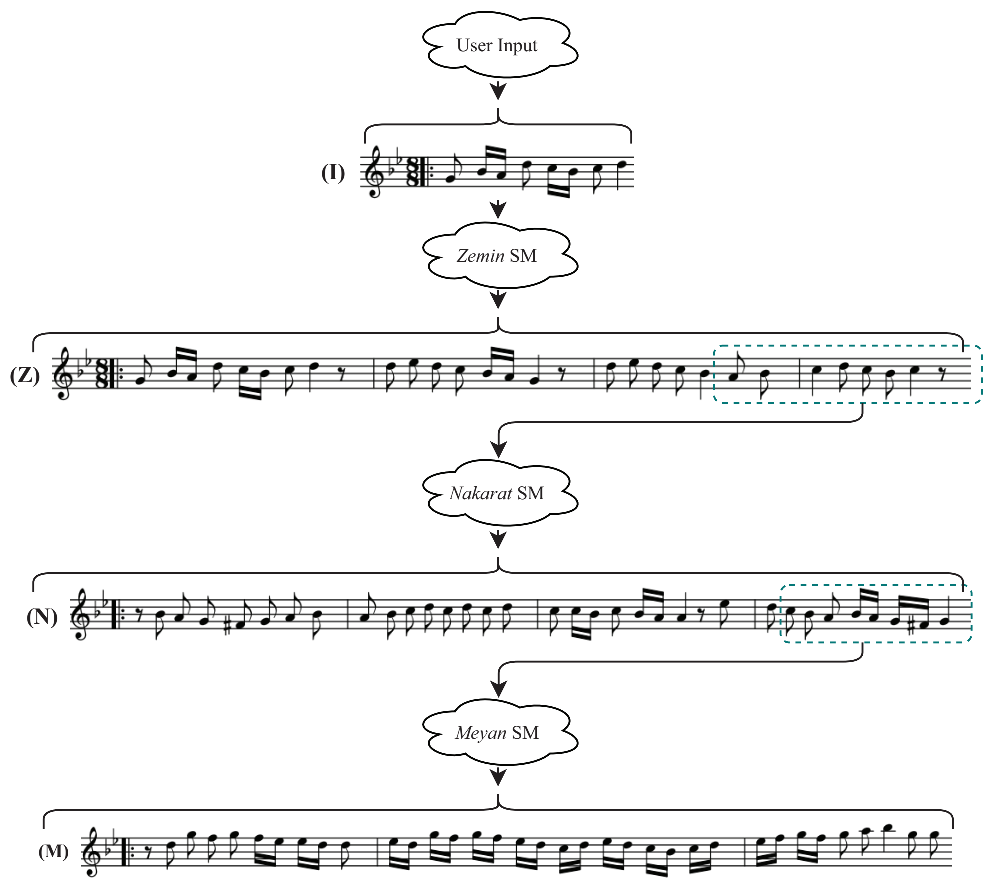

Figure 2. Operation schema of specialist models (SMs). User input is denoted by (I), Zemin section is denoted by (Z), Nakarat section is denoted by (N) and finally, Meyan section is denoted by (M).

The softmax activation function (1), which is the final layer of SMs outputs a probability distribution over pitch-duration tuple classes.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{1}$$

where z_i represents each element of SM's output vector, and k is the number of unique pitch-duration tuples. For choosing the next pitch-duration tuple from the output distribution, we set 2 threshold values: high-threshold, represented as green horizontal dashed line, and low-threshold, represented as the red horizontal dashed line in Figure 3. If the probability of any pitch-duration class exceeds the high-threshold value, as shown in Figure 3a, this means SM is confident with its prediction; thus, the pitch-duration tuple with maximum probability is chosen for the next note. However, as shown in Figure 3b, when SM does not output a prediction favoring a single pitch-duration class, the next note is chosen from a set of weaker candidates, which have a probability value between high-threshold and low-threshold values. In this case, 4-gram probabilities of each candidate are calculated by the formula given in equation (2).

$$P(pd_t \mid pd_{t-3}, pd_{t-2}, pd_{t-1}) = \frac{Count(pd_{t-3}, pd_{t-2}, pd_{t-1}, pd_t)}{Count(pd_{t-3}, pd_{t-2}, pd_{t-1})} \tag{2}$$

where pd_t represents a pitch-duration tuple at timestep t, i.e., the candidate whose 4-gram score is being calculated, and the sequence of $pd_{t-3}, pd_{t-2}, pd_{t-1}$ represents the last 3 pitch-duration tuples fed into the SM. Thus, the 4-gram score of each weak candidate is calculated by dividing the count of $pd_{t-3}, pd_{t-2}, pd_{t-1}, pd_t$ sequence in the whole dataset by the count of $pd_{t-3}, pd_{t-2}, pd_{t-1}$ sequence in the whole dataset. And the candidate with the highest 4-gram score is chosen to form the next note. If a tie occurs amongst weak candidates, the next note is chosen randomly.

Tuning up high and low threshold values for each section (Zemin, Nakarat, and Meyan) of artificial composition enables the end-user to control the resulting piece. Low threshold values may elevate randomness and originality for the resulting piece, where higher threshold values may result in more traditional outcomes.

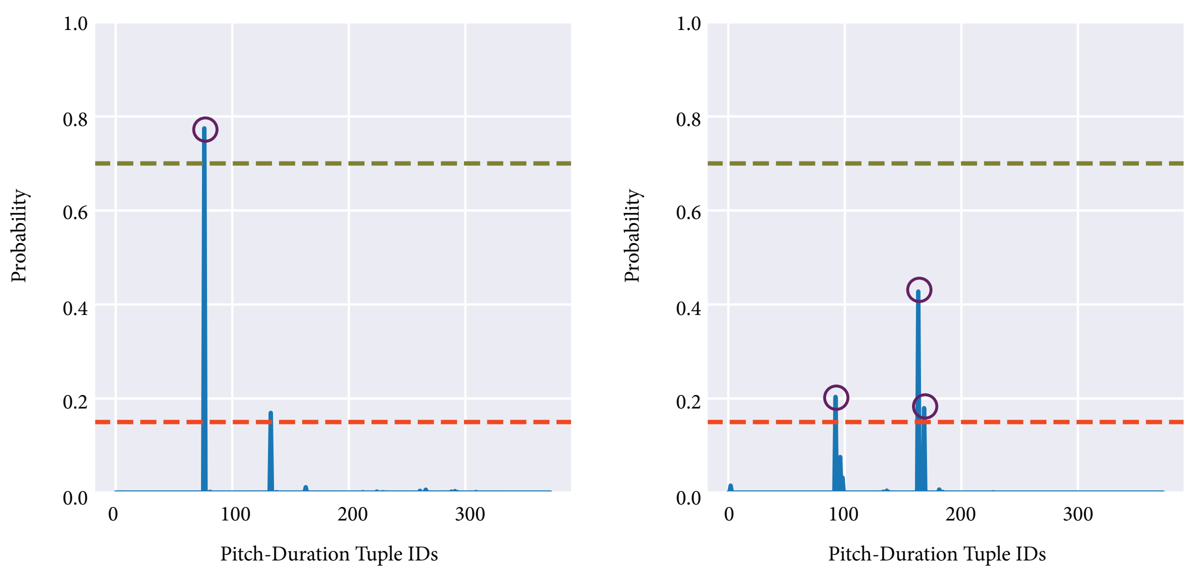

Figure 3. Two predictions made by a specialist model (SM) are shown side by side. The high threshold value (green dashed line) is set to be 0.7, and the low threshold value (red dashed line) is set to be 0.15. At the left-hand side (a), there is a pitch-duration entity that has a higher probability than 0.7, so, it is selected to be the next pitch-duration tuple. Whereas, on the right-hand side (b), there aren't any strong candidates. In this case, a selection will be performed amongst these 3 weaker candidates which have probabilities between 0.15 and 0.7.

3.3. Conductor model

For each section of the artificial composition, two SMs are trained with slightly different hyperparameters. We trained the final types of models, Conductor Models (CoMo), for choosing between 2 predictions made by 2 SMs. CoMos are deep neural networks that have three 100-neuron LSTM layers followed by a fully connected layer. Each layer has a 50% dropout factor. CoMos are compiled with RMSprop optimizers and categorical cross-entropy loss functions.

Just like SMs, CoMos are first trained on the target makam of SymbTr without explicit form and usul information. Then, they inherit this knowledge and specialize in sections of target makam's Şarkı forms. For each section of Şarkı form (Zemin, Nakarat, and Meyan), there is 1 CoMo, which evaluates predictions of 2 SMs and decides the outcome, hence determines the next note. The relationship between CMs, SMs, and CoMos is represented in Figure 4.

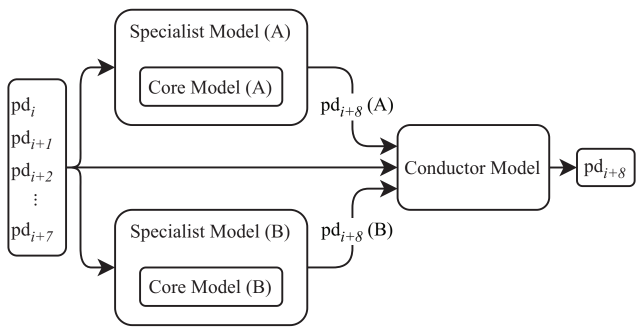

Figure 4. Representation of the artificial composer system for any of Zemin, Nakarat, or Meyan sections. Input pitch-duration tuples $pd_i, pd_{i+1}, pd_{i+2}, \ldots, pd_{i+7}$ are fed into both specialist model (A) and specialist model (B). Specialist models are compiled from core models using their initial layers. Finally, the conductor model picks one of the specialist models (A)'s output pd_{i+8} (A) or specialist models (B)'s output pd_{i+8} (B) as the next pitch-duration tuple pd_{i+8} according to the 8 input pitch-duration tuples.

Putting it all together, as shown in Figure 5, a user enters 8 pitch-duration tuples (initial notes) into the ATMMC system. ATMMC outputs the next note with respect to the given user input and appends it to the end of the 8 initial notes resulting in a set of 9 notes. Then ATMMC picks the last 8 notes of the 9-notes set and outputs the 10th note and appends it to the end of the notes set. This process of picking the last 8 notes from the notes-set and outputting and appending the next note continues until the 4 bars of the Zemin section are completed. When the Zemin section is completed, ATMMC picks the last 8 notes of the Zemin section and composes the Nakarat section. Likewise, when 4 bars of the Nakarat section are completed, ATMMC picks the last 8 notes of the Nakarat section and composes the 4-bar long Meyan section. And finally, ATMMC merges its compositions of Zemin, Nakarat, and Meyan sections sequentially into a Mu2 file and writes it to disk.

4. Results

To demonstrate the effectiveness of the proposed system, we generated a total of 40 pieces: 20 for Nihavent Makam and 20 for Hicaz. All pieces in Hicaz Makam are in Aksak Usul, whereas pieces in Nihavent Makam are in Düyek Usul. All artificially composed 40 pieces are in Şarkı form and start with four bars of Zemin section, followed by four bars of Nakarat section, then a four-bar long Meyan section comes and finally, pieces return to Nakarat and end.

The evaluation process was carried out by using the methods proposed by Yang and Lerch [26], which are

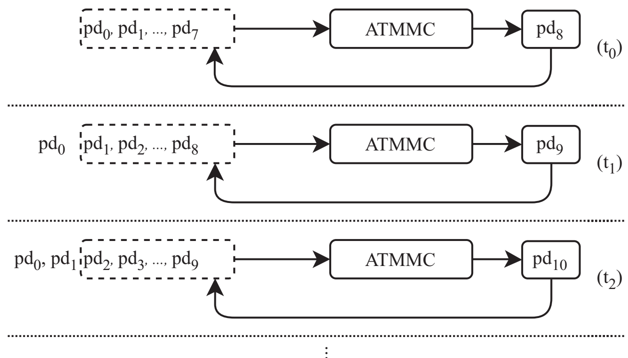

Figure 5. Illustration of Automatic Turkish Makam Music Composer's (ATMMC) composition process. In the beginning (t_0) user enters 8 pitch-duration tuples ($pd_0, pd_1, \ldots$) and starts automatic composition. At each iteration t_i, ATMMC outputs a single pitch-duration tuple pd_{i+8}, appends it to the end of the input list and selects the last 8 pitch-duration tuples from the input list at next iteration.

given in Section 1. Absolute metrics of both training and generated datasets of Nihavent and Hicaz Makams are given in Table 1. Mean and standard deviation values for pitch count (PC), note count (NC), and pitch class histogram (PCH) are also calculated per bar (shown as Feature/Bar). For calculating given metrics, the bar count of each sample must be equal. For satisfying this requirement, we picked 4 bars of Zemin, 4 bars of Nakarat, and 2 bars of Meyan sections from each sample, adding up to 10 bars per sample.

Table 1. Absolute (characteristic) metrics for Nihavent and Hicaz training datasets and generated datasets. Mean and standard deviation (STD) of pitch count (PC), pitch count per bar (PC/Bar), note count (NC), note count per bar (NC/Bar), pitch class histogram (PCH), pitch class histogram per bar (PCH/Bar), note length histogram (NLH), pitch class transition matrix (PCTM), pitch range (PR), average pitch interval (PI), average inter-onset-interval (IOI), and note length transition matrix (NLTM) features are shown as well as the average values. Definitions of displayed features are given in Section 1.

	Nihavent				**Hicaz**			
	Training dataset		**Generated dataset**		**Training dataset**		**Generated dataset**	
	mean	**STD**	**mean**	**STD**	**mean**	**STD**	**mean**	**STD**
PC	14.05	1.62	11.60	1.20	14.60	2.95	10.85	1.38
PC/Bar	4.18	1.25	4.76	1.26	5.34	1.45	5.29	1.48
NC	6.60	1.65	4.40	1.01	6.45	1.82	5.05	1.11
NC/Bar	2.69	0.75	2.09	0.70	3.12	1.03	2.35	0.85
PCH	0.05	0.07	0.05	0.07	0.04	0.06	0.04	0.07
PCH/Bar	0.05	0.13	0.05	0.11	0.04	0.09	0.04	0.09
NLH	0.07	0.13	0.07	0.17	0.06	0.14	0.06	0.15
PCTM	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.01
PR	31.85	2.97	27.35	2.57	67.00	8.98	56.50	6.61
PI	3.59	0.47	3.34	0.30	7.00	0.52	6.77	0.44
IOI	0.08	0.03	0.04	0.01	0.05	0.01	0.03	0.01
NLTM	0.00	0.02	0.00	0.04	0.00	0.03	0.00	0.03
Average	5.26	0.75	4.47	0.62	8.64	1.42	7.24	1.01

Data in Table 1 can be interpreted as for both Makams, generated sets are similar to training sets, but they have less diversity in terms of pitch and duration variations. Since ANN models learn the general structural relations of the system they are trained over and exclude outliers, reduction in diversity is an expected outcome. Otherwise, it is called memorization, which is something undesired for such systems. For Nihavent makam, the average of the generated set's mean values resembles the average of the training set's mean values by $100 \times (1 - (5.26 - 4.47)/5.26) \approx \mathbf{84.98\%}$. Similarly, the resemblance of mean values' averages of training and generated sets for Hicaz makam is $100 \times (1 - (8.64 - 7.24)/8.64) \approx \mathbf{83.79\%}$. Overall, it can be concluded that the absolute metrics of ATMMC's generated sets resemble the training sets around **84%**.

Another result is that, training samples in Hicaz makam are slightly more diverse than samples in Nihavent makam. Hicaz Makam has a larger family than Nihavent Makam; encapsulating variations such as Hicaz Humayun, Hicaz Uzzal, and Hicaz Zirgule Makams. A larger family leads to wider variation.

Computed Kullback–Leibler divergence (KLD) and overlapping area (OA) values of relative metrics for Nihavent and Hicaz generated datasets are given in Table 2. Both Nihavent and Hicaz generated sets have the largest OA values for pitch count per bar (PC/Bar) feature, which can be interpreted as among all relative metrics, ATMMC best performs in producing diverse pitches per bar. On average, the Hicaz dataset slightly outperforms the Nihavent dataset both by larger OA and smaller KLD values.

Table 2. Kullback–Leibler divergence (KLD) and overlapping area (OA) values for pitch count (PC), pitch count per bar (PC/Bar), note count (NC), note count per bar (NC/Bar), pitch class histogram (PCH), pitch class histogram per bar (PCH/Bar), note length histogram (NLH), pitch class transition matrix (PCTM), pitch range (PR), average pitch interval (PI), average inter-onset-interval (IOI), and note length transition matrix (NLTM) metrics between intra-generated set and inter-set probability distribution functions for Nihavent and Hicaz Makams. Definitions of displayed features are given in Section 1.

	Nihavent		**Hicaz**	
	KLD	**OA**	**KLD**	**OA**
PC	0.043	0.604	0.243	0.533
PC/Bar	0.029	**0.886**	0.037	**0.898**
NC	0.115	0.566	0.045	0.701
NC/Bar	**0.010**	0.618	0.065	0.640
PCH	0.250	0.811	0.056	0.638
PCH/Bar	0.165	0.595	0.051	0.511
NLH	0.165	0.603	**0.012**	0.814
PCTM	0.114	0.630	0.194	0.573
PR	0.249	0.551	0.154	0.669
PI	0.102	0.820	0.023	0.855
IOI	0.340	0.548	0.166	0.358
NLTM	0.153	0.541	0.025	0.732
Average	0.144	0.647	**0.089**	**0.660**

5. Conclusion and future work

This paper presents the first deep learning-based symbolic music generation system, artificial Turkish makam music composer (ATMMC), for pieces in Şarkı form in Hicaz and Nihavent Makams. By altering the high and low threshold values, users can control the diversity and conventionalism of resulting songs. According to Yang and Lerch's [26] evaluation method which is given in Section 1, generated datasets are found to be similar to source dataset approximately by 84%. Due to the availability of a larger dataset, results in Hicaz Makam are found to be slightly better than the resulting samples in Nihavent Makam.

ATMMC can be used to assist human composers in creating new pieces for professional, educational, or entertainment purposes. It also can help composers to overcome writer's block. We plan to investigate different AI models and algorithms for achieving even better results. We also plan to expand the scope of ATMMC by training it on other Makams. Finally, for easier usability of ATMMC by non-programmers, we plan to create a graphical user interface.

References

[1] Kumar NH, Ashwin PS, Ananthakrishnan H. MellisAI - an AI generated music composer using RNN-LSTMs. International Journal of Machine Learning and Computing 2020; 10 (2): 247-252. doi: 10.18178/ijmlc.2020.10.2.927

[2] Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O et al. WaveNet: a generative model for raw audio. ArXiv 2016; arXiv:1609.03499 [cs.SD].

[3] Roberts A, Engel J, Mann Y, Gillick J, Kayacik C et al. Magenta sudio: augmenting creativity with deep learning in Ableton Live. In: International Workshop on Musical Metacreation (MUME); Charlotte, USA; 2019. pp. 1-7.

[4] Huang C, Hawthorne C, Roberts A, Dinculescu M, Wexler J et al. The Bach doodle: approachable music composition with machine learning at scale. ArXiv 2019; arXiv:1907.06637 [cs.SD].

[5] Hadjeres G, Pachet F, Nielsen F. DeepBach: a steerable model for Bach chorales generation. In: Thirty-fourth International Conference on Machine Learning; Sydney, Australia; 2017. pp. 1362-1371.

[6] Deng J, Schuller B, Eyben F, Schuller D, Zhang Z et al. Exploiting time-frequency patterns with LSTM-RNNs for low-bitrate audio restoration. Neural Computing and Applications 2020; 32 (4): 1095-1107. doi: 10.1007/s00521-019-04158-0

[7] Dieleman S, Oord A, Simonyan K. The challenge of realistic music generation: modelling raw audio at scale. In: 32nd International Conference on Neural Information Processing Systems; Montreal, Canada; 2018. pp. 8000-8010.

[8] Abidin D, Öztürk Ö, Öztürk T. Using data mining for makam recognition in Turkish traditional art music. Journal of the Faculty of Engineering and Architecture of Gazi University 2017; 32 (4): 1221–1232 (in Turkish with an abstract in English). doi: 10.17341/gazimmfd.369557

[9] Yarman O. 79-tone tuning & theory for Turkish maqam music. PhD, İstanbul Technical University, İstanbul, Turkey, 2008.

[10] Şentürk S, Serra X. Composition identification in Ottoman-Turkish makam music using transposition-invariant partial audio-score alignment. In: 13th Sound and Music Computing Conference (SMC 2016); Hamburg, Germany; 2016. pp. 434-441.

[11] Bozkurt B, Ayangil R, Holzapfel A. Computational analysis of Turkish makam music: review of state-of-the-art and challenges. Journal of New Music Research 2014; 43 (1): 3-23. doi: 10.1080/09298215.2013.865760

[12] Parncutt R, Reisinger D, Fuchs A, Kaiser F. Consonance and prevalence of sonorities in western polyphony: roughness, harmonicity, familiarity, evenness, diatonicity. Journal of New Music Research 2018; 48 (1): 1–20. doi: 10.1080/09298215.2018.1477804

[13] Kızrak MA, Bolat B. A musical information retrieval system for classical Turkish music makams. Simulation 2017; 93 (9): 749-757. doi: 10.1177/0037549717708615

[14] Karaosmanoğlu MK. A Turkish makam music symbolic database for music information retrieval: Symbtr. In: 13th International Society for Music Information Retrieval Conference (ISMIR); Porto, Portugal; 2012. pp. 223–228.

[15] Wright O, Turabi AH. Klasik Türk musikisi'nde çargah: tarih-teori çelişkisi. Marmara Üniversitesi İlahiyat Fakültesi Dergisi 2001; 21 (2): 81-104 (in Turkish).

[16] Öztürk O. How was the traditional makam theory westernized for the sake of modernization. Rast Müzikoloji Dergisi 2018; 6 (1): 1769-1787. doi: 10.12975/pp1769-1787

[17] Aydemir M. Türk Müziği Makam Rehberi. İstanbul, Turkey: Pan Yayıncılık, 2010 (in Turkish).

[18] Şentürk S, Chordia P. Modeling melodic improvisation in Turkish folk music using variable-length markov models. In: 12th International Society for Music Information Retrieval Conference (ISMIR 2011); Miami, FL, USA; 2011. pp. 269–274.

[19] Gönül M. A Look at the representation, statement and classification of Turkish music rhythms. İslam, Sanat, Tarih, Edebiyat ve Musikisi Dergisi (İSTEM) 2015; 25 (1): 31-46 (in Turkish with an abstract in English).

[20] Şenocak E. Tarihi süreç içinde Türk müziğinde şarkı formu. MSc, Haliç University, İstanbul, Turkey, 2012 (in Turkish).

[21] Tüfekçioğlu S. The new music forms added to Turkish music with Tanzimat and the theory-practice dilemma of these forms. Atatürk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi 2019; 42 (1): 173–182 (in Turkish with an abstract in English). doi: 10.32547/ataunigsed.523556

[22] Hananoi S, Muraoka K, Kiyoki Y. A music composition system with time-series data for sound design in next-generation sonification environment. In: International Electronics Symposium (IES); Denpasar, Indonesia; 2016. pp. 380-384.

[23] Briot JP, Hadjeres G, Pachet F. Deep Learning Techniques for Music Generation. New York, NY, USA: Springer International Publishing, 2020.

[24] Xu X. LSTM networks for music generation. ArXiv 2020; arXiv:2006.09838 [eess.AS].

[25] Wu J, Hu C, Wang Y, Hu X, Zhu J. A hierarchical recurrent neural network for symbolic melody generation. IEEE Transactions on Cybernetics 2020; 50 (6): 2749-2757. doi: 10.1109/TCYB.2019.2953194

[26] Yang LC, Lerch A. On the evaluation of generative models in music. Neural Computing and Applications 2020; 32 (9): 4773-4784. doi: 10.1007/s00521-018-3849-7

[27] Briot JP, Hadjeres G, Pachet FD. Deep learning techniques for music generation - a survey. ArXiv 2019; arXiv:1709.01620 [cs.SD].

[28] Karaman S. Problems in performing and naming 9 beats tempos in classical Turkish music. Türkiyat Araştırmaları Dergisi 2014; 35 (1): 461-478 (in Turkish with an abstract in English). doi: 10.21563/sutad.187109

[29] Kumar H, Ravindran B. Polyphonic music composition with LSTM neural networks and reinforcement learning. ArXiv 2019; arXiv:1902.01973 [cs.SD].

[30] Weiss K, Khoshgoftaar TM, Wang D. A survey of transfer learning. Journal of Big Data 2016; 3 (1): 1-40. doi: 10.1186/s40537-016-0043-6