DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

TURKISH MAKAM MUSIC COMPOSITION BY USING DEEP LEARNING TECHNIQUES

by
İsmail Hakkı PARLAK

October, 2021
İZMİR

TURKISH MAKAM MUSIC COMPOSITION BY USING DEEP LEARNING TECHNIQUES

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Computer Engineering

by
İsmail Hakkı PARLAK

October, 2021
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"TURKISH MAKAM MUSIC COMPOSITION BY USING DEEP LEARNING TECHNIQUES"** completed by **İSMAİL HAKKI PARLAK** under the supervision of **PROF. DR. YALÇIN ÇEBİ** and we certify that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Yalçın ÇEBİ

Supervisor

Prof. Dr. Cihan IŞIKHAN

Thesis Committee Member

Assoc. Prof. Dr. Derya BİRANT

Thesis Committee Member

Assoc. Prof. Dr. Seda POSTALCIOĞLU

Examining Committee Member

Assoc. Prof. Dr. Ahmet Tuncay ERCAN

Examining Committee Member

Prof. Dr. Okan FISTIKOĞLU

Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENT

I would like to thank my supervisor Prof. Dr. Yalçın ÇEBİ from the bottom of my heart for his supervision, endeavor, and guidance, not just only throughout this study but also in my personal life. I would like to offer my thanks and gratitude to Prof. Dr. Cihan Işıkhan and Assoc. Prof. Dr. Derya Birant for their precious directives and guidance. I was very privileged to have such skillful authorities in their respective fields always guiding me positively throughout this work.

And finally, I would like to express my sincere gratitude to my wife, my son, and my parents for their support, patience, and love. I could not be able to complete this work without their invaluable support. Thank you.

İsmail Hakkı PARLAK

TURKISH MAKAM MUSIC COMPOSITION BY USING DEEP LEARNING TECHNIQUES

ABSTRACT

Although music and other forms of fine arts have been accepted as a part of human existence, people have developed various methods and algorithms throughout history to create new creations in these domains. Especially in the last century, with the invention of the computer and the exponential increase in its processing capacity, new computational methods in artistic creativity have been tried and interesting results have been obtained in related fields. Artificial composers, developed with Deep Learning techniques, which is a sub-field of artificial intelligence, took part in interdisciplinary studies and their artificial compositions began to be followed by those who are interested in the subject with great curiosity and interest. However, the studies of composing music using Deep Learning techniques were mostly performed on Western Music, and Turkish Maqam Music remained untouched in this arena.

In the execution of this thesis, a system that can automatically compose Turkish Makam Music using Deep Learning techniques and an easy-to-use web-browser based graphical interface has been developed. The system, called Automatic Turkish Makam Music Composer (ATMMC), takes 8 starting notes from its user and creates a composition in Aksak or Düyek Usûls in one of Hicaz or Nihâvent Makams, depending on the user's preference. Artificial compositions created by ATMMC can be stored in the user's computer to be opened with Mus2 application. Generated artificial compositions were compared with the source data set according to various metrics and it was seen that there was approximately 84% similarity between the source dataset and artificial compositions. The developed system and its user interface are shared as an open-source project.

Keywords: Deep learning, Turkish Makam Music, artificial intelligence, algorithmic music composition, artificial composition.

DERİN ÖĞRENME TEKNİKLERİ KULLANILARAK TÜRK MAKAM MÜZİĞİ BESTELENMESİ

ÖZ

Müzik ve diğer güzel sanat dalları her ne kadar insan olmanın bir parçası olarak kabul edilmişse de insanlar tarih boyunca bu alanlarda yeni yaratılar ortaya koymak için çeşitli yöntem ve algoritmalar geliştirmişlerdir. Özellikle son yüzyılda bilgisayarın icat edilmesi ve zaman içinde işlem kapasitesinin katlanarak artması ile sanatsal yaratıcılıkta yeni yöntemler denenmiş ve ilgili alanlarda ilginç sonuçlar elde edilmiştir. Yapay zekâ biliminin bir alt alanı olan derin öğrenme teknikleri ile geliştirilen yapay besteciler, disiplinler arası çalışmalarda yer almış ve ortaya çıkardıkları yapay besteler, konu ile ilgilenenler tarafından büyük bir merak ve ilgi ile takip edilmeye başlanmıştır. Ancak derin öğrenme tekniklerini kullanarak müzik besteleme çalışmaları çoğunlukla batı müziği üzerine yapılmış, Türk Makam Müziği bu alanda yalnız kalmıştır.

Bu tez çalışması kapsamında derin öğrenme teknikleri kullanarak otomatik olarak Türk Makam Müziği besteleyebilen bir sistem ve bu sistemin kolay kullanımı için web tarayıcısında çalışan bir grafiksel arayüz geliştirilmiştir. Otomatik Türk Makam Müziği Bestecisi adı verilen sistem, kullanıcısından 8 adet başlangıç notası alıp kullanıcı tercihine göre Hicaz veya Nihâvent Makamlarından birinde Aksak veya Düyek Usûllerde bir yaratı oluşturabilmektedir. Oluşturulan yapay beste Mus2 isimli uygulama ile açılabilecek bir formatta oluşturulup kullanıcının bilgisayarına kaydedilebilmektedir. Oluşturulan yapay besteler, çeşitli ölçütlere göre kaynak veri seti ile karşılaştırılmış ve aralarında yaklaşık %84 benzerlik olduğu görülmüştür. Geliştirilen sistem ve kullanıcı arayüzü açık kaynak olarak paylaşılmaktadır.

Anahtar kelimeler: Derin öğrenme, Türk Makam Müziği, yapay zekâ, algoritmik besteleme, yapay besteleme.

CONTENTS

Page

Ph.D. THESIS EXAMINATION RESULT FORM ii
ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZ v
LIST OF FIGURES.. vi
LIST OF TABLES.. vii

CHAPTER ONE - INTRODUCTION 1

1.1 Overview 1
1.2 Purpose 2
1.3 Contribution to Literature.. 3
1.4 Organization of the Thesis 3

CHAPTER TWO – LITERATURE REVIEW 5

2.1 Evolution of Turkish Music and Its Theory 5
2.2 Turkish Makam Music Concepts and Western Music 8
2.3 Deep Learning and Automatic Music Composition.............................. 11
2.4 Evaluation Methods for Artificial Music Compositions 13

CHAPTER THREE - DATASET 16

3.1 SymbTr Turkish Makam Music Symbolic Data Collection 16
3.1.1 Statistical Review of SymbTr Dataset 17
3.1.2 SymbTr Dataset N-gram Analysis 22
3.2 Dataset Preparation & Representation 25

CHAPTER FOUR – TECHNIQUES AND EVALUATION METHODS 27

4.1 Long Short-Term Memory Networks 27
4.1.1 Activation Functions 28
4.1.2 Long Short-Term Memory Cell Working Principles 29
4.2 Probability Distribution Analysis of Various Features 32
4.2.1 Pairwise Cross Validation 34
4.2.2 Kernel Density Estimation 34
4.2.3 Kullback-Leibler Divergence and Overlapped Area 35
4.2.4 Method Demonstration 36

CHAPTER FIVE – AUTOMATIC COMPOSER 40

5.1 Base Models 40
5.2 Specialist Models 42
5.3 Conductor Models 46
5.4 Full System Overview 48

CHAPTER SIX – GRAPHICAL USER INTERFACE FOR ATMMC 51

6.1 Purpose of the Graphical User Interface 51

6.2 Graphical User Interface Decomposition 51

CHAPTER SEVEN – RESULTS & EVALUATION 58

7.1 Subjective Evaluation 58

7.2 Objective Evaluation 61

7.2.1 Pitch Density Analysis 62

7.2.2 ATMMC Detailed Analysis Results 64

CHAPTER EIGHT – CONCLUSION AND FUTURE WORK.......................... 72

8.1 Conclusion 72

8.2 Future Work 73

REFERENCES 74

LIST OF FIGURES

Page

Figure 2.1 Comparison of TMM and WM divisions per whole tone interval. 10
Figure 3.1 Inner structure of a Mu2 file from SymbTr dataset. 17
Figure 3.2 Percentages of Makams in SymbTr dataset. 18
Figure 3.3 Scale of Hicaz Makam spanning 2 octaves. 19
Figure 3.4 Illustration of note durations. 20
Figure 3.5 Hicaz merged octaves. 20
Figure 3.6 Hicaz separated octaves. 21
Figure 3.7 Nihâvent merged octaves. 22
Figure 3.8 Nihâvent separated octaves. 22
Figure 4.1 Internal structure of an LSTM cell (Olah, 2015). 28
Figure 4.2 Sigmoid (left) and Tanh (right) function plots. 29
Figure 4.3 Depiction of data flow through LSTM cell state. 30
Figure 4.4 Data flow through the LSTM cell's forget gate. 30
Figure 4.5 LSTM cell state calculation. 31
Figure 4.6 LSTM hidden state formation. 32
Figure 4.7 Workflow of Yang and Lerch's method (Yang & Lerch, 2020). 33
Figure 4.8 Kernel density estimation application (Vanderplas, 2007). 35
Figure 4.9 Probability distributions for toy data set 1. 37
Figure 4.10 Probability distributions for toy data set 2. 39
Figure 5.1 Base Model's layer architecture. 41
Figure 5.2 Base model diagram (a) and specialist model diagram (b). 43
Figure 5.3 Operation scheme of Specialist Models. 44
Figure 5.4 Specialist model prediction samples. 45
Figure 5.5 Conductor Model structural decomposition. 47
Figure 5.6 Structural decomposition of ATMMC. 48
Figure 5.7 ATMMC composition process flowchart. 50
Figure 6.1 Overview of ATMMC's graphical user interface. 52
Figure 6.2 ATMMC graphical user interface left-hand side closeup. 53
Figure 6.3 ATMMC graphical user interface right-hand side closeup. 53
Figure 6.4 Before (a) and after (b) of automatic measure creation. 54
Figure 6.5 An erroneous measure. 55

Figure 6.6 Composition progress report pop-up. 56
Figure 6.7 Artificial composition download link. 57
Figure 7.1 Subjective evaluation responses for questions 1 and 2. 59
Figure 7.2 Subjective evaluation responses for questions 3 and 4. 60
Figure 7.3 Subjective evaluation responses for questions 5 and 6. 60
Figure 7.4 Subjective evaluation responses for question 7. 61
Figure 7.5 Hicaz Makam pitch density graph. 63
Figure 7.6 Nihâvent Makam pitch density graph. 64
Figure 7.7 An example Hicaz composition by ATMMC. 65
Figure 7.8 An example Nihâvent composition by ATMMC. 65
Figure 7.9 PDFs of pitch count (PC) feature. 68
Figure 7.10 PDFs of note count (NC) feature. 69
Figure 7.11 PDFs of pitch count histogram (PCH) feature. 69
Figure 7.12 PDFs of note length histogram (NLH) feature. 70
Figure 7.13 PDFs of pitch range (PR) feature. 70
Figure 7.14 PDFs of average pitch interval (PI) feature. 71

LIST OF TABLES

Page

Table 3.1 Tonal properties of Hicaz and Nihâvent Makams. 18

Table 3.2 4-gram frequencies for Hicaz Makam. 23

Table 3.3 4-gram frequencies for Nihâvent Makam. 24

Table 3.4 5-gram frequencies for Hicaz Makam. 24

Table 3.5 5-gram frequencies for Nihâvent Makam. 24

Table 4.1 Absolute measurement results for toy data 1. 36

Table 4.2 Relative measurement results for toy data 1. 37

Table 4.3 Absolute measurement results for toy data 2. 38

Table 4.4 Relative measurement results for toy data 2. 38

Table 7.1 Subjective evaluation responses for question 8. 61

Table 7.2. Absolute metrics for Nihâvent Makam. 66

Table 7.3. Absolute metrics for Hicaz Makam. 66

Table 7.4 Relative metrics for Hicaz and Nihâvent Makams. 67

CHAPTER ONE
INTRODUCTION

1.1 Overview

Human beings have embraced artistic creativity as a faculty of their existence. The human mind, which has evolved through time, has made aesthetic beauty a part of its creations and has turned it into a symbol of advancement. Music, just like architecture, literature, and other forms of fine arts, has become the stamp of nations and reflected their history, culture, sentimental life, and collective experiences.

Throughout the history, great composers of various cultures either exploited conventional methods or invented new techniques to create new music. Canonic composition in the late 15th century (Burkholder et al., 2010) and Mozart's *dice music* (Parlak & Kösemen, 2018) are examples of such inventions in music composition. Especially in the last century, with the invention and rapid development of computers, the number of algorithmic artistic creation experiments exploded. Hiller and Isaacson's string quartet the "Illiac Suite" reserved its place in history books as the first computer composition that utilized musical functions and rules (Sandred et al., 2009). Then between 1967 - 72, comes Xenakis with his stochastic music composition methods (Luque, 2009). As an alternative to stochastic and rule-based systems, around the 90s Cope's EMI (Experiments in Musical Intelligence) software came out (Cope, 1989). EMI was an Artificial Intelligence (AI) composer system that learned the rules of music composition from the dataset it was trained on.

Towards the 2000s, the availability of relatively low-cost general-purpose graphics processing units (GPGPUs) and vast amounts of digital data provided Deep Learning (DL), which is a subset of Artificial Neural Networks (ANN) based AI technology, with a chance to take a huge leap and making its way to being one of the most popular branches of today's computer science research arena. Since then, DL has been continuing to grow rapidly, focusing on automatic classification and prediction as well as artistic computation (Briot et al., 2017).

As an example of artificial artistic creation, Google's "Deep Dream" attracted very broad attention in 2015 by creating psychedelic artistic images. In addition, Aiva Technologies's film score composer (2017) and Google's Bach Doodle (2019) can be pointed out as top-tier examples of recent DL-based artificial music composers (Huang et al., 2019).

Most DL-based artificial composers are built in the domain of western music forms like classical western music, jazz, rock, pop, etc. Unfortunately, studies focusing on composing Turkish Makam Music (TMM) by DL-driven AI are very scarce. Western music and TMM are very different, and they can be very easily distinguished even by an untrained ear. This difference brings about the necessity to apply specialized techniques to DL-based TMM composition, rather than directly copy-pasting DL-based western music composition techniques. In this thesis, a sketch of such TMM specialized DL technique is given, and results are discussed.

1.2 Purpose

The essential incentive of this work is investigating the complex and delicate matter of artificially composing Turkish Makam Music (TMM) and providing a preliminary solution to it by implementing a Deep Learning (DL) based artificial composer system. The described system is called Automatic Turkish Makam Music Composer (ATMMC), and its purpose is composing new TMM songs similar to past compositions in the TMM repertoire. ATMMC operates on the domains of *Hicaz* and *Nihâvent Makam*s and its rhythmic domain comprises *Aksak* (9/8) and *Düyek* (8/8) *Usûl*s.

ATMMC's compositions may induce new ideas for TMM composers, may help conservatory students to quickly create pieces to exercise their instruments, or be a source of art and entertainment. Also, this research may build a foundation for other researchers who are willing to work in this domain.

Another purpose of this study is to provide a graphical user interface (GUI) of ATMMC to non-programmers thus making the system available to anyone interested in it. Through a web application, users can make ATMMC compose new pieces and download the resulting song to their computers in .mu2 format. All source code, results, and training set are shared on GitHub (Parlak, 2020).

1.3 Contribution to Literature

Computational research on TMM is not as abundant as its western music counterpart and is mostly trying to solve classification problems, or more focused on music information retrieval (MIR) aspects. This research contributes to filling the gap of DL-based automatic TMM composition.

Another contribution is the web-based GUI that is developed to utilize ATMMC. There are tens of musical notation desktop and web applications for western music. However, the number of available musical notation applications for TMM is just 2, which are “Nota” and “Mus2” (Yarman, 2010). Moreover, web-based TMM notation applications do not exist yet. This study lays a foundation for a web-based TMM notation application.

ATMMC backend AI system, combined with the web-based GUI, offers a complete solution to the problem of creating artificial compositions for TMM with DL. Before this study, such solutions were completely absent.

1.4 Organization of the Thesis

In this chapter, an overview and purpose of the study, and contribution to the literature are given. The remaining sections of the thesis are given in the following paragraphs.

In Chapter 2, a comprehensive literature review is given covering the topics of evolution of the TMM and its theory, the differences between TMM and western music, usage of DL techniques on artificial music composition, and how artificially composed music can be evaluated.

In Chapter 3, the dataset used in this study is described as well as the methods for preparing it for model training. Also, the expansion of the used dataset for transfer learning is briefly given.

In Chapter 4, the details of ATMMC are given in terms of selection of the type of used neural network, how models are put together and connected for accomplishing different tasks of artificial composition, and graphical representations of the system's operational dynamics.

In Chapter 5, the design and implementation of the GUI is given as well as the screen captures of a use case. Also, a brief user's manual is given in this chapter alongside related graphics.

In Chapter 6, artificial compositions of the system and evaluation of ATMMC's artistic capabilities are given and discussed in detail.

Finally, the summary and conclusion of the thesis are given in Chapter 7. Also, plans, enhancement suggestions for the future, and new ideas are included in this section.

CHAPTER TWO
LITERATURE REVIEW

2.1 Evolution of Turkish Music and Its Theory

A brief history of Turkish music, its origins, evolution, and development as well as the progression of its theory is given in this section for laying the foundation to understand its characteristics. This is also important for understanding today's widespread TMM theories and the reason for the absence of a fully accepted formal theory.

Tıraşçı (Tıraşçı, 2019) gave voice to the history of Turkish music, its cornerstones, and its theoreticians. According to his work, before the Huns, Turks were located in the northern and southern regions of The Tian Shan (Tengri Mountains). Around 2000 BC, the Altai Mountains and Siberia became two significant sites for Turks. At that time, music was performed only by the religious men, who were known as Shamans, for protection, spiritual and healing purposes.

At the age of Huns (3rd century AD), Turks used the pentatonic scale. With one of the oldest Turkish instruments Kopuz, Huns' music traveled to Europe and left its traces, especially in the Balkans and Hungary region (Aydoğan & Özgür, 2015). Also, at the age of Huns, in Chinese sources, it is recorded that Turks used drums to hearten their warriors (Özkan, 2006). Later, music became militarized and military music was institutionalized; thus, the repertoire and the musical activity grew in return. At the age of Göktürks (6th century AD), Turks became neighbors with cultural centers such as China, Persia, Byzantine, and India which led Turkish music to progress in terms of genre and form. Also, at the age of Göktürks, music was a part of Khan's (The leader) assemblies. At these assemblies, musicians paid greater attention to the artistic aspect of the performed music which led to the separation of art and folk music. In that era, Turkish music got rid of being used only for religious purposes and started to appeal to perceptions such as pleasure and aesthetics (Tıraşçı, 2019).

Uygur Turks (8^{th} - 9^{th} century AD) used the 7 tone - diatonic scale and later, they began using the 12 - tone chromatic scale. The oldest Turkish musical note system belongs to Uygur Turks, in which every musical note was represented by a symbol from the Uygur alphabet (Tıraşçı, 2019). By adopting Manichaeism and Buddhism, Uygurs' religious music got richer (Aydoğan & Özgür, 2015). According to Tıraşçı (Tıraşçı, 2019), before adopting Islam, Turkish music genres were:

- Religious music: Shamans used to utter sacred words musically. They used drums and various percussive instruments to accompany their ceremonies.
- Tuğ music: This genre was performed during military and official ceremonies. Various percussion, cymbals, and horn instruments were used. It is believed to be the ancestor of Mehter music.
- Heroic, epic music: This type of music was revolving around epic and heroic events and stories. It was used to increase the mood of the community and soldiers. It also served as transferring historical knowledge to the future generation.
- Toy music: This genre was performed by the palace's musicians at important formal events such as receiving ambassadors or accession to the throne.
- Daily life music: This genre was performed by the folk who expressed their feelings of love, pain, sorrow, or longing.
- Yuğ music: This genre was performed after events of the death of beloved ones to express sorrow and grief.
- Hunting music: When presidents were going out for hunting, Turks used to pitch tents and sing sacred words for the hunt's abundance. This custom continued even after Turks adopted Islam.

After Karakhanids met with Islam, onwards 9th century, Turkish music heavily interacted with Arabic / Islamic music and changed significantly. Arabic quarter-tone ornaments fused with Turkish music and today's Turkish Makam feel began to emerge (Aydoğan & Özgür, 2015). Al-Kindi (9^{th} century) was the first to write on music theory amongst Muslim philosophers. He used Pythagorean ratios in his work (Bozkurt et al., 2009). He related musical notes to celestial bodies and systematized Islamic music. He inspired Al-Farabi and Avicenna (Ibn-Sina) (Tıraşçı, 2019).

Al-Farabi (10th century), studied music through the works of Grecian philosophers and Al-Kindi. He corrected missing and erroneous theoretical information of Greek philosophers and made exceptional studies on the physics of music. Safi al-Din Urmavi (13th century) solved the problem of temporal representation in music with his musical notation system. Before him, there was no representation for temporal information of music. He placed numbers below musical notation and solved the issue of temporal representation. He also invented two musical instruments called *Nüzhe* and *Muğni*. He was first to use the term *Edvar* (cycle) to represent various scales such as Uşşak, Neva, Rast, Hicaz, etc. In addition, he proposed his 17-tone Pythagorean scale by revising Al-Kindi's work (Yarman, 2007). Safi al-Din Urmavi is one of the most remarkable figures in the history of TMM theory. His work was accepted as the fundamental TMM theory until 16th century (Uygun, 2008). Following Safi al-Din Urmavi's work and concept of Edvar, Mahmud Shirazi (14th century) was one of the first who used the term Makam. In his works, he mentioned 17 Makams and their scales (Tıraşçı, 2019).

Until the 15th century, there was no distinction between Turkish, Persian, and Arabic music. But after the 15th century, Turkish artistic and cultural thought began to find its place within the new and emerging theoretical studies. Yusuf bin Nazimuddin wrote Risale-i Musiki, which is the first Ottoman musical theoretics. He believed that the movement of the Universe created harmonious sounds which form the basis of music. Inheriting Al-Farabi's thoughts, he defined 12 Makams which relate to 12 zodiacal constellations (Tıraşçı, 2019). At the 18th century, another important figure in Ottoman music scene, Kutbü'n Nâyî (Lead of Ney Players) Osman Dede developed a new musical notation system and created various writings on music theory. He composed pieces in a wide range of structure and form, and he was the first to give titles to pieces in Peşrev form (Erguner, 2007).

In the 20th century, Anatolia was housing three different mindsets of musical groups. The first group was supporting Western music, whereas the second one was standing by traditional Turkish music. And the last group was trying to combine the two. Up until the 20th century, the innovations that emerged in matters such as the sound system, pitches, and Makams could not be based on solid foundations. Rauf Yekta Bey studied the theory of Turkish music and laid solid foundations of the system used today (Tıraşçı, 2019).

In the Turkish Republic's early years, Atatürk attached great importance to music studies and music education, and he made effort to carry Turkish music on a par with the contemporary world's requirements. Hüseyin Sadeddin Arel, who was a student of Rauf Yekta Bey, used the symbols that denote the intervals we use in written music today. With his colleagues Dr. Suphi Ezgi and Prof. Dr. Salih Murat Uzdilek, they created the Arel-Ezgi-Uzdilek (AEU) system which divided an octave into 24 non-equidistant intervals (Aydoğan & Özgür, 2015). AEU system is being used and thought of in today's conservatoires as the official model (Bozkurt et al., 2009). Some may argue that Arel's system depends on Western music theory rather than Turkish music, or it may lack representation of the practical musical performance, but nonetheless, it is the most widely used system in Turkey today (Tıraşçı, 2019).

2.2 Turkish Makam Music Concepts and Western Music

In addition to Classical Indian Music and Chinese Music, two other traditions that can be considered as civilization music today are Western Music (WM), which is the most studied one amongst all computationally, and Turkish Makam Music (TMM) (Barkçin, 2019). Understanding the similarities and differences between TMM and WM is important for deciding how computational composition techniques developed for WM can be applied to TMM. The main elements that distinguish music, which is one of the most important components of the inclusive phenomenon called culture, are the sound system, rhythmic structure, and style used by that culture (Karaosmanoğlu, 2017). Therefore, sound systems, rhythmic structures, and styles of TMM and WM should be studied comparatively.

In almost all fundamental properties of their dynamics, TMM and WM are different than each other (Abidin et al., 2017). Firstly, and most obviously, being a tonal genre, WM revolves around harmony and chord progressions, whereas TMM, which is modal, focuses on melodies (Özkan, 2006). In other words, WM is polyphonic, i.e., it is based on multiple harmonious distinct pitches playing at the same time, whereas in TMM, all orchestral entities play the same pitch in unison or in different octaves with small ornamental differences (Bozkurt et al., 2014; Şentürk & Serra, 2016), which is called heterophony (Yarman, 2008).

TMM is very rich in creating melodies evoking a vast variety of emotions in a wide variety of musical modes which are called Makams (Barkçin, 2019). Very simply put, Makams are modal structures, where melodies begin to form around an initial note and end around a final note (Ederer, 2011). Makams are built on scales and perhaps, the most significant difference between TMM and WM is the method of acquiring the pitches in their various scales. WM divides an octave into 12 equidistant intervals (Şentürk & Chordia, 2011), which are created by dividing a whole step into two equal fractions, i.e., semi-tones (Uyar et al., 2014). But according to Arel-Ezgi-Uzdilek (AEU) theory (Arel, 1968), which is the official TMM theory today (Wright & Turabi, 2001), as illustrated in Figure 2.1, a whole tone is divided into 9 equidistant intervals each of which are called Koma (Şentürk & Serra, 2016). AEU theory divides an octave into 53 equidistant fractions (Karaosmanoğlu, 2017). Within these 53 pitches, 24 are used to describe the practical TMM tuning system (Karaosmanoğlu, 2012). This phenomenon is illustrated in Figure 2.1, where F_0, F_1, ..., F_5 denote the used frequencies within a whole tone interval in AEU theory. Even though a whole step is divided into 9 equidistant Komas, not all of them are used in practice. From Figure 2.1, it can be seen that western semi-tones (W_s) divide a whole tone interval exactly by 2, which corresponds to 4.5 Koma, but even though 4 Koma and 5 Koma intervals correspond to predetermined pitches in AEU theory, 4.5 Koma, which is Western music semi-tone does not have any counterpart in AEU theory.

At this point, it should be noted that a whole tone in AEU theory roughly corresponds to 204 cents and a western whole tone corresponds to 200 cents, where 1 cent is 1/1200th fraction of an octave (Karaosmanoğlu, 2017). To simplify the topic, the 4 cents difference is omitted in the figure. Also, depending on the frequency range, a 4-cent difference is not always discernable by human beings (Zarate et al., 2012).

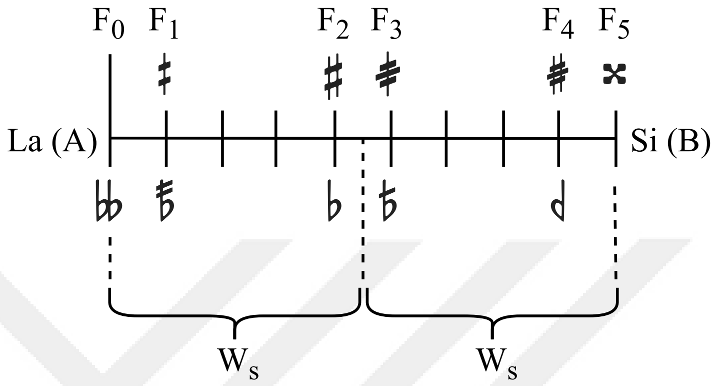


Figure 2.1 Comparison of TMM and WM divisions per whole tone interval.

TMM is taught orally for centuries (Uyar et al., 2014) through a system called Meşk (Tüfekçi, 2014). In Meşk system, master teaches the details of the studied subject to the disciple orally then examines and corrects disciple's abilities closely when needed. As given in Section 2.1, since the efforts for the literalization of TMM theory coincided with the beginning of the republican period, it was late compared to WM in this regard. So even today, there are several different TMM theory suggestions, ranging from 17 to 79 pitches per octave (Yarman, 2007). Even though AEU theory is considered to be imperfect by modern TMM theoreticians, it brought ease to teach and learn TMM formally and thus provided great benefit (Özkan, 2006).

A key term for understanding Makams is Seyir. The term Seyir can be defined as the rules that regulate the circulation of melodies in the Makam (Güvençoğlu & Özgelen, 2020). Two distinct Makams may have different Seyirs while having the exact same tone series. However, regardless of melodic movement, a tone series is named by its dominant or initial note in WM.

Apart from Makam, another important concept of TMM is Usûl. Usûl can superficially be translated to English as the meter. Usûl describes the temporal properties of music in TMM (Şentürk & Chordia, 2011). Usûls are comprised of percussion stroke sequences with assorted velocities in a fixed amount of time (Bozkurt et al., 2014). In TMM, percussions play the composition's Usûl continuously. Whereas, in Classical Western Music, performances require a conductor for organizing the temporal accord (Barkçin, 2019). Usûls can be as simple as a 2-beat Nim Sofyan or a very complex 124-beat Cihar (Gönül, 2015).

The final significant concept in TMM is Form. Form is used to describe the scheme of musical piece's parts and their arrangement (Şenocak, 2012). The two main branches in TMM form scheme are instrumental and vocal music (Özkan, 2006). These branches are further divided into sub-branches. For example, Şarkı is a member of the vocal music form. Şarkı form consists of Zemin, Nakarat, and Meyan sections in Zemin, Nakarat, Meyan, and Nakarat order (Güvençoğlu & Özgelen, 2020). Zemin, which is like an introduction, is the section that shows the characteristics of Şarkı's Makam. Nakarat is the section where melodies are diversified and finalize with Makam's tonic pitch. In the Meyan section, melodies travel between different Makams usually in higher registers (Tüfekçioğlu, 2019).

2.3 Deep Learning and Automatic Music Composition

Definition of music can be reduced to "succession of harmonious frequencies in certain durations". By this definition, it can be deduced that music can be represented as time-series data. Therefore, it would not be unreasonable to expect good results from artificial music composition studies with Deep Learning models that are specialized to work on time series data (Hananoi et al., 2016; Oord et al., 2016).

Recurrent Neural Networks (RNN) are a family of Artificial Neural Networks (ANN) that are used for processing sequential data (Goodfellow et al., 2016). Unfortunately, *vanilla RNN*s may suffer vanishing or exploding gradients when learning long-time dependencies. This brings forth the hindrance of learning features of the training set by deeper layers of the ANN. To overcome the vanishing gradient problem Choi et al. (Choi et al., 2016) propose Long Short-Term Memory (LSTM) systems for music composition. They show that using char-RNN and word-RNNs alongside LSTMs can produce satisfactory results in generating jazz chord progressions.

Oord et al. (Oord et al., 2016) introduce WaveNet in their work, which is a novel study generating raw audio. WaveNets can produce state of the art results when applied to the text-to-speech field and is also able to generate novel and realistic audio waveforms when trained with piano performances. They state that WaveNet is based on PixelCNN and it operates on 16000 samples per second audio files.

In their study of euphonious easy to follow pop music melody generation, Shin et al. (Shin et al., 2017) again deployed LSTM NNs. It is easy to see that most researchers use LSTMs on the task of automatic music generation due to their success in forecasting time series data (Xu, 2020). However, there are various studies that use different types of NN systems such as Generative Adversarial Networks (GANs). In their paper, Shuyu et al. (Li et al., 2019) described such system to compose artificial melodies. Using Hierarchical Recurrent Neural Networks (HRNN) is another modern approach for artificial music composition. Wu et al. (Wu et al., 2020) used three LSTM networks to construct an HRNN for creating symbolic melodies.

There are popular, open-source and well-maintained software and frameworks for Deep Learning and machine learning. One of the most popular libraries used for Deep Learning studies is TensorFlow, which is a Python interface capable of running on both CPU and GPU, as well as on a wide variety of heterogeneous systems (Abadi et al., 2015). Keras is a high-level wrapper Python API for TensorFlow (Chollet, 2015). Keras makes it easier to work on Deep Learning and enables fast experimentation.

2.4 Evaluation Methods for Artificial Music Compositions

It is obvious that artificially composed music should be evaluated to determine the strengths and weaknesses of the proposed systems and to compare the work with other similar studies, thus gaining ideas and insights for future studies with better results. However, this is not a very straightforward task. It is not very easy to formulate and measure what makes a piece of music pleasant. For this reason, researchers use two distinct evaluation strategies: subjective evaluation and objective evaluation. Subjective evaluation is made by exposing the outcomes of composer system to human evaluators and gathering their feedback about the system. Whereas objective evaluation is made by implementing a series of statistical procedures.

An example of subjective evaluation is carried on by Shin et al. (2017). They resorted to human evaluators for evaluation of their artificial composer's outcomes. They obtained information on human evaluators' musical backgrounds and then asked evaluators to choose the most organic and well-structured sounding sample within a set of various pieces. This set contained samples from both authors' and similar studies' results. Then as an additional step, they performed a Turing Test (Turing, 1950) on participants to find out whether they could differentiate authors' machine-made musical compositions from human-made ones or not.

Most research resorting to subjective evaluation strategies conduct their evaluation process through surveys. Liang, Gotham, Johnson, & Shotton (2017) evaluated their study by means of an online survey. In their survey, they collected participants' age and level of musical expertise information. Then they presented various synthetic, and human-made music to participants without telling them their origin and asked them to determine which is synthetic and which is organic. Another example is the study of Chu, Urtasun & Fidler (2016). They conducted a survey amongst 27 participants. They asked participants to compare their results with the results of Google's Magenta (Brain Team, 2020). They also collected participants' commentaries on the reasoning behind their decisions.

For objective evaluation there are different strategies. Marinescu (2019) performed experiments with different types of neural networks and network configurations. To compare and evaluate different generative models, they investigated training loss values and validation accuracy percentages.

Trying to put forward a standard to objective evaluation of melody-based artificial composers, Yang and Lerch (Yang & Lerch, 2020) propose a comprehensive method comprising a collection of metrics. In their paper, they suggest calculating sets of pitch and duration related features within and in-between training dataset and artificially composed songs. They denote that, if a metric is only computed within a single set, it is labeled as "absolute". Absolute metrics deliver information related to the set from which it is computed. They also propose "relative" metrics, which are acquired by comparing training sets and generated sets. They suggest computing several pitch-related and duration-related metrics. They suggest computing the below metrics for extracting pitch-related features of the studied sets:

- Pitch Count (PC): The total number of distinct pitches disregarding duration information per song (sample).
- Pitch Class Histogram (PCH): Histogram of pitches without octave information. For example, Do4#4 and Do5#4 are accepted as equivalent pitches. The computed histogram should be normalized.
- Pitch Class Transition Matrix (PCTM): Octave independent, transition matrix of pitches. Again, duration information is disregarded, and the computed matrix should be normalized.
- Pitch Range (PR): The spread between the highest and the lowest pitches per sample.
- Average Pitch Interval (PI): Total pitch distances between successive notes per sample divided by the total number of pitches.

Yang and Lerch also suggest computing rhythm-based features for obtaining further information about studied datasets. These metrics are described as follows:

- Note Count (NC): Total number of distinct durations disregarding pitch information per sample.
- Average Inter-Onset-Interval (IOI): The average time quanta in-between all sequential notes.
- Note Length Histogram (NLH): Histogram of note durations. NLH should be normalized.
- Note Length Transition Matrix (NLTM): Transition matrix between all durations disregarding pitch information. NLTM should be normalized.

Yang and Lerch instruct that for obtaining absolute metrics, mean and standard deviation of each feature should be computed. In addition, for obtaining relative metrics, first, one should perform pairwise exhaustive cross-validation between features and in result, get a histogram of each feature's distances. Histograms are computed by calculating the Euclidian distances between samples at each cross-validation step (Yang and Lerch use the term "intra-set distances" for histograms computed within a set. For histograms computed between different sets, they use the term "inter-set distances"). After obtaining the histograms, the authors suggest applying kernel density estimation for smoothing the results. Performing kernel density estimation on histograms reveals probability distribution functions (PDF). And finally, the authors suggest computing Kullback-Leibler divergence (KLD) and overlapping area (OA) of inter-set and intra-set PDFs. Yang and Lerch argue that, for having similar intra-set Gaussian distributions, the variance of 2 datasets should be similar. They further state that mean values should be similar for having similar inter-set distributions. Finally, Yang and Lerch point out that for displaying high similarity between source dataset and generated dataset, KLD value between intra-generated dataset and inter-sets should be small, and OA value between intra-generated dataset and inter-sets should be large.

CHAPTER THREE
DATASET

3.1 SymbTr Turkish Makam Music Symbolic Data Collection

Having a well-formed, balanced, and large dataset is a key for success in Deep Learning (DL), but available machine-readable sources for Turkish Makam Music (TMM) are very rare. The largest and the best formatted machine-readable TMM digital data source is SymbTr (Karaosmanoğlu, 2012). SymbTr data set contains 2,200 pieces from 155 distinct Makams encapsulating about 865,000 musical notes. In addition, SymbTr scores are provided in the Text, MusicXML, PDF, MIDI, and Mu2 formats (Şentürk, 2017).

In this thesis, Mu2 format was preferred as the digital representation source. As shown in Figure 3.1, the contents of Mu2 format in SymbTr collection can be viewed by a text editor. Entities in Mu2 files are tab-separated which are distributed into rows. Each row starts with an ID denoting the types of contents of that row. As shown in Figure 3.1, "50" denotes the piece's Makam; "51" denotes the piece's Usûl; "52" denotes the tempo etc. It can also be seen from Figure 3.1 that musical events in Mu2 files start with "9" followed by pitch name, duration's numerator and denominator, and several voicing-related features.

```
9   4   Pay Payda   Legato% Bas Çek Söz-1   Söz-2   0.444444444
50                          Acemaşîrân  B4b5
51      9   4               Ağıraksak
52      1   4   66
57                          Aranağme
58                          ?
59
60                          Acemaşîrân Aranağme (Ağıraksak)
62                          H
63                          TSM
14                          111111111
9                           (
9       1   8   100                 0.055555556
9   Fa5 1   16  95  96  64  ARANAĞME        0.083333333
9   Mi5 1   16  95  96  64  .       0.111111111
9   Fa5 1   8   50  96  64  .       0.166666667
9   Sol5    1   8   50  96  64  .       0.222222222
9   La5 3   16  95  125 64  .       0.305555556
9   Sol5    1   16  95  96  64  .       0.333333333
9   Fa5 1   8   95  96  64  .       0.388888889
9   Mi5 1   8   95  96  2   .       0.444444444
9   Sol5    3   16  95  125 64  .       0.527777778
9   Fa5 1   16  95  96  64  .       0.555555556
9   Mi5 1   8   95  96  64  .       0.611111111
9   Re5 1   8   95  96  2   .       0.666666667
9   Mi5 1   8   95  84  64  .       0.722222222
```

Figure 3.1 Inner structure of a Mu2 file from SymbTr dataset.

3.1.1 Statistical Review of SymbTr Dataset

There are more than 150 distinct Makams in SymbTr but not all of them have as many representatives as others. As shown in Figure 3.2, the two most frequent Makams in the SymbTr dataset are Hicaz (7.1% of the total set with 157 pieces) and Nihâvent (5.9% of the total set with 130 pieces). Since DL models benefit from large datasets in their training processes, target Makams for the thesis scope were chosen to be Hicaz and Nihâvent.

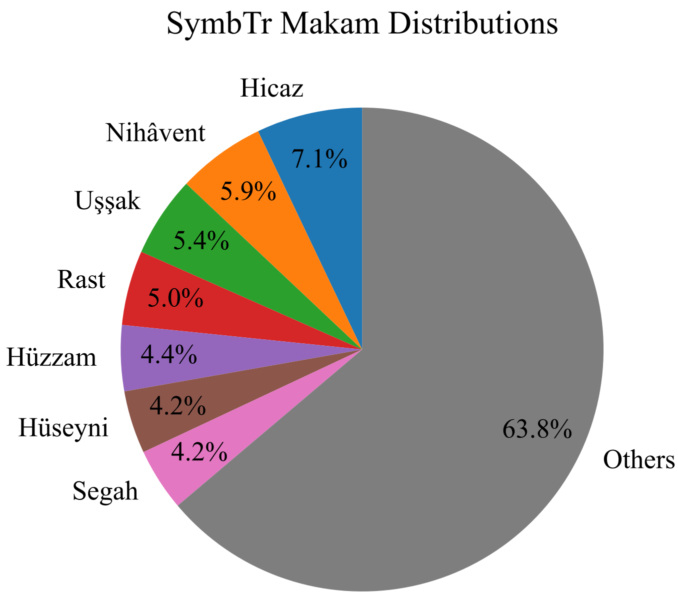


Figure 3.2 Percentages of Makams in SymbTr dataset.

Hicaz and Nihâvent Makams sound very different from each other and naturally are very different in terms of their scales and pitch characteristics. As shown in Table 3.1, their tonic, leading, and dominant pitches are different. In addition, they have completely different scales. As shown in Figure 3.3, Hicaz Makam can be defined as the combination of Hicaz tetrachord in place (Dügâh) and Rast pentachord on Nevâ. Whereas Nihâvent Makam is the combination of Bûselik pentachord on Rast and Kürdî or Hicaz tetrachord on Nevâ (Özkan, 2006).

Table 3.1 Tonal properties of Hicaz and Nihâvent Makams.

Makam	**Tonic**	**Dominant**	**Leading**
Hicaz	A (Dügâh)	D (Nevâ)	G (Rast)
Nihâvent	G (Rast)	D / Bb (Kürdî)	F# (Irak)

In twelve-tone equal temperament Western Music (12-TET WM) theory, the frequency of any pitch is calculated by the Equation (3.1).

$$F_n = F_b \times 2^{N/12} \tag{3.1}$$

where F_n is the frequency of the note to be calculated in Hz; F_b is the base frequency, for example, 440 Hz for A4; and N is the number of semi-tones between F_n and F_b. An octave consists of 12 semi-tones in WM. So, when N reaches 12, F_n becomes 2 x F_b. In other words, the frequency of pitches just doubles when the octave goes higher, and vice versa. The same case applies to TMM, except that the formula used to calculate relative frequencies of pitches is different.

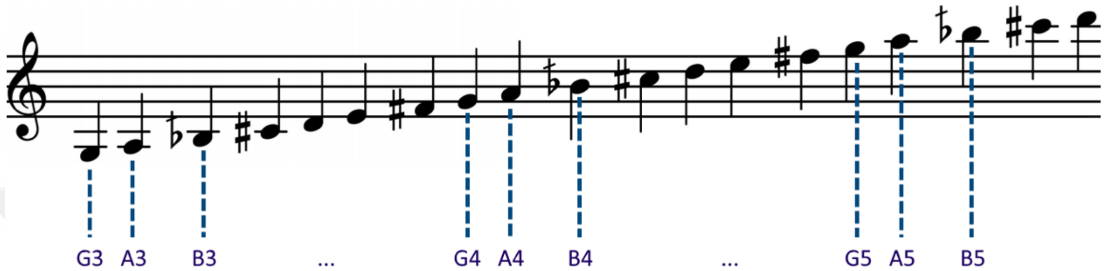


Figure 3.3 Scale of Hicaz Makam spanning 2 octaves.

The human brain perceives different octaves of the same pitch perfectly harmonious. This is due to perfect alignments of pitches' fundamental tones and their harmonics. Thus, same pitches in different octaves can be accepted as equivalent. According to the calculation shown in Figure 3.4, the proportions of pitches of Hicaz Makam in terms of their durations within the SymbTr collection are shown in Figure 3.5 and Figure 3.6. In Figure 3.4, an example of a duration calculation case is given, where the total duration of G notes (blue) is 1/4 + 1/4 + 1/8 +1/2 + 1/8 = 14/8 whereas, B (green) notes have total duration of 1/8 + 1/1 + 1/4 = 11/8. So, the total duration of G is longer than the total duration of B. In Figure 3.5, different octaves of the same pitch are merged into the same space, i.e., A4 and A5 are just summed into A. Whereas, in Figure 3.6, pitches are left as is without any merging or alterations.

Figure 3.4 Illustration of note durations.

As shown in Figure 3.5, where octaves are merged, the longest duration belongs to A (Dügâh) which is the tonic pitch of Hicaz Makam. The second-longest duration belongs to D (Nevâ) which is the dominant pitch of Hicaz.

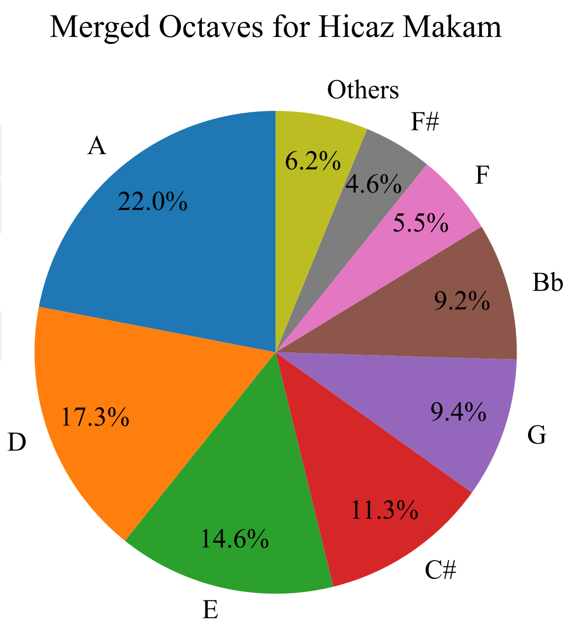


Figure 3.5 Hicaz merged octaves.

As shown in Figure 3.6, where octaves are not merged, i.e., separated, the longest duration belongs to D5 (Nevâ), which is the dominant pitch of Hicaz Makam. And the second-longest duration belongs to A4 (Dügâh), which is the tonic pitch. Both Figure 3.5 and Figure 3.6 show that the total collection of Hicaz pieces in the SymbTr dataset represents Hicaz Makam coherent with its formal definitions. Here it should be noted that pitch names in Figures are given according to WM notation.

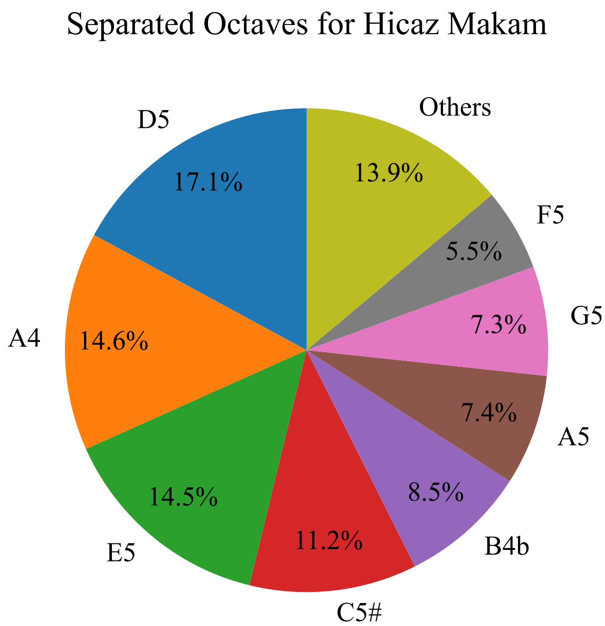


Figure 3.6 Hicaz separated octaves.

Similarly, in Figure 3.7 and Figure 3.8, merged and separated proportions of pitch durations of Nihâvent pieces in the SymbTr collection are given. Again, it can be seen that the longest durations belong to D5 (Nevâ), G4 (Rast), and Bb (Kürdî) pitches, which are dominant and tonic pitches of Nihâvent Makam. This again shows that pieces in Nihâvent Makam of SymbTr collection reflect the official definition of Nihâvent Makam. In conclusion, it can be deduced that SymbTr reflects the tonal characteristics of TMM Makams well, and therefore it is suitable for DL tasks.

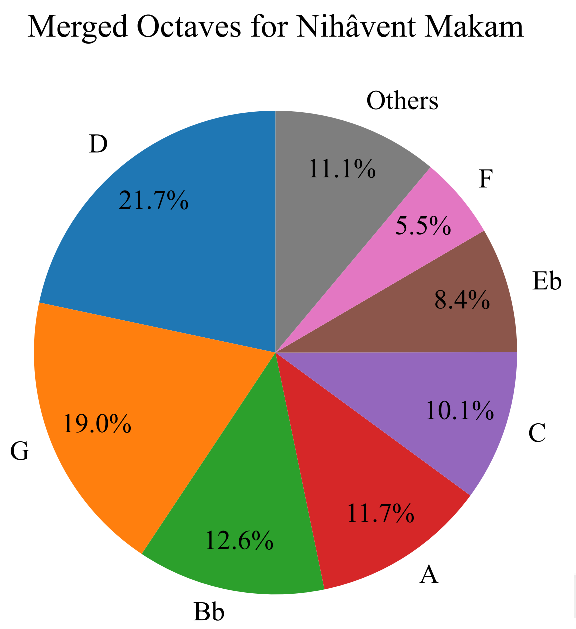


Figure 3.7 Nihâvent merged octaves.

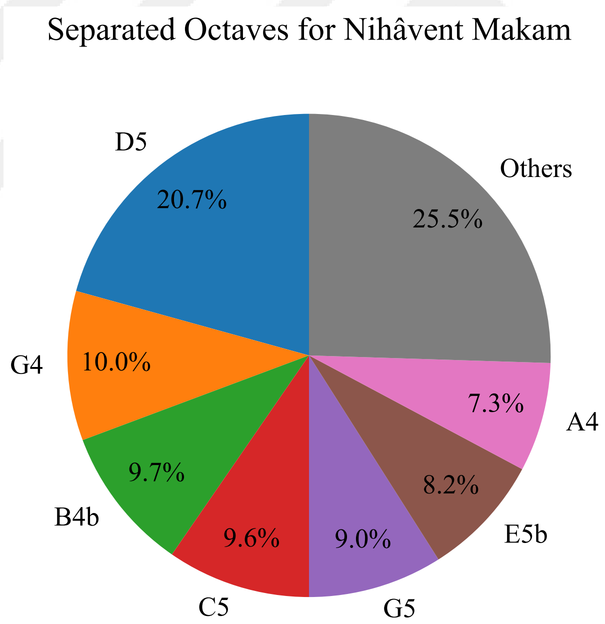


Figure 3.8 Nihâvent separated octaves.

3.1.2 SymbTr Dataset N-gram Analysis

N-grams are continuous sequences of *N* items within a sequential set (Kapadia, 2019). They are useful for information retrieval from the set they are extracted from, as well as calculating probability of future elements of an existing sequence. Relative frequency of a given N-gram$_x$, which indicates the density of a given sequence of length N among all possible sequences with equal lengths within a set, is simply calculated according to Equation (3.2).

$$Relative\ Frequency(Ngram_x) = \frac{Count(Ngram_x)}{Count(All\ Ngrams)} \tag{3.2}$$

In Tables 3.2 thru 3.5, relative frequencies of 3-gram, 4-gram, and 5-gram values for Hicaz and Nihâvent Makams are given. Calculated relative frequencies of 3, 4, and 5-grams are given in the last columns (Rel. Freq.) of each table. Here, note durations are omitted, but only pitches are included in calculations.

The term Çeşni in TMM corresponds to short sequences of 4 or 5 pitches that evoke a feeling of Makam in the listener (Altinköprü, 2018). In other words, they are mini Makam identifiers. Thus, investigating 4-gram and 5-gram frequencies is especially important in TMM analysis because they reveal practical Çeşnis of studied Makam. In all N-gram relative frequency tables, note names are given according to the naming convention of AEU theory. For example, do5#4 represents Do (C) in the 5th octave, which is sharp by 4 Komas, which is the symbolic representation of the pitch "Nîm Hicaz".

Some of very characteristic 4-pitch runs for Hicaz and Nihâvent Makams such as re5-do5#4-si4b4-la4 sequence which is the return-to-tonic motive for Hicaz, and a similar purpose sequence do5-si4b5-la4-sol4 for Nihâvent Makam are listed in Table 3.2 and Table 3.3.

Table 3.2 4-gram frequencies for Hicaz Makam.

1st Pitch	2nd Pitch	3rd Pitch	4th Pitch	Rel. Freq.
fa5	mi5	re5	do5#4	1.69
re5	do5#4	si4b4	la4	1.63
sol5	fa5	mi5	re5	1.28
mi5	re5	do5#4	si4b4	1.21
mi5	fa5	mi5	re5	1.09

Table 3.2 continues

mi5	re5	do5#4	re5	1.05
do5#4	si4b4	si4b4	la4	1.00

Table 3.3 4-gram frequencies for Nihâvent Makam.

1st Pitch	2nd Pitch	3rd Pitch	4th Pitch	Rel. Freq.
re5	do5	si4b5	la4	1.60
mi5b5	re5	do5	si4b5	1.39
do5	si4b5	la4	sol4	1.17
re5	mi5b5	re5	do5	0.99
la4	si4b5	do5	re5	0.92
sol5	fa5	mi5b5	re5	0.82
do5	si4b5	la4	si4b5	0.74

In Table 3.4 and Table 3.5, very characteristic 5-pitch motives can be seen, such as do5#4-si4b4-si4b4-la4-la4 for Hicaz Makam and re5-do5-si4b5-la4-sol4 for Nihâvent Makam.

Table 3.4 5-gram frequencies for Hicaz Makam.

1st Pitch	2nd Pitch	3rd Pitch	4th Pitch	5th Pitch	Rel. Freq.
do5#4	si4b4	si4b4	la4	la4	0.82
fa5	mi5	re5	do5#4	si4b4	0.79
mi5	re5	do5#4	si4b4	la4	0.73
sol5	fa5	mi5	re5	do5#4	0.72
la5	sol5	fa5	mi5	re5	0.65
fa5	mi5	re5	do5#4	re5	0.64
re5	mi5	fa5	mi5	re5	0.58

Table 3.5 5-gram frequencies for Nihâvent Makam.

1st Pitch	2nd Pitch	3rd Pitch	4th Pitch	5th Pitch	Rel. Freq.
mi5b5	re5	do5	si4b5	la4	0.88
re5	do5	si4b5	la4	sol4	0.81
re5	mi5b5	re5	do5	si4b5	0.55
si4b5	la4	sol4	fa4#4	sol4	0.53
re5	do5	si4b5	la4	si4b5	0.51
sol5	fa5	mi5b5	re5	do5	0.45
re5	do5	si4b5	do5	re5	0.44

Tables 3.2 thru 3.5 clearly show that pieces of Hicaz and Nihâvent Makams in the SymbTr dataset represent characteristic motives and Çeşnis of their belonging Makams very well, consequently, SymbTr is proven to be a convenient dataset for computational TMM research.

3.2 Dataset Preparation & Representation

There are various strategies to transform music data into a form suitable for DL models such as timestep sampling, numerical values, binary vectors, textual representation, etc. (Briot et al., 2017; Kumar & Ravindran, 2019; Wu et al., 2020). Different types of music data conversions result in different success rates in DL models' training phases. Usually, the best strategy is chosen through a series of trials and errors. It can be stated that the data preparation phase is the most important and effort-consuming task in DL and data analysis (Barapatre & A, 2017).

A musical note consists of pitch and duration, i.e., vibration of air in a certain frequency for a certain time. To clear up, "A3 ¼" in a 60 beats per minute (bpm) context is a 220Hz oscillation that lasts for 1 second. From this point of view, a monophonic TMM piece can be reduced to oscillations of various successive durations. By this logic, all distinct pitch-duration tuples in the training dataset were converted to distinct integer ids by a dictionary such as "Sol4 1/2" → 1, "Sol4 1/4" → 2, …, "Do6 1/32" → 401, etc. With such a dictionary, it is possible to convert pitch-duration tuples into integers and vice versa.

In SymbTr, there are 405 unique pitch-duration tuples for Hicaz Makam and there are 374 unique pitch-duration tuples for Nihâvent Makam. As the final step of data preparation, each pitch-duration tuple's ids were converted into one-hot encoded vectors v, such that Equation (3.3) holds for Hicaz Makam.

$$v = \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{404} \end{bmatrix} \omega here, v_i \in \{0,1\} \wedge \sum_{i=0}^{404} v_i = 1 \tag{3.3}$$

For Nihâvent Makam, the size of the encoding vector is 374, i.e., the last item in the vector should be v_{373}. By the data representation approach given in this section, representing any pitch-duration as a DL model input or output reduces the artificial music composition task into a one-hot encoded single-label classification problem.

CHAPTER FOUR
TECHNIQUES AND EVALUATION METHODS

Deep Learning is a machine learning area that came up around 2006, which utilizes many layers of non-linear processing techniques for feature extraction and classification tasks over investigated datasets (Deng, 2014). Automatic Turkish Makam Music Composer (ATMMC) is based on a collection of Deep Learning (DL) models which utilize Long Short-Term Memory Networks (LSTM) that interact with each other for accomplishing various tasks of structured artificial TMM composition.

4.1 Long Short-Term Memory Networks

LSTMs emerged for curing vanishing or exploding error signals of the conventional Back-Propagation Through Time (BPTT) algorithms used in Recurrent Neural Networks (RNN), and they succeed in learning very long dependencies exceeding 1000 timesteps (Hochreiter & Schmidhuber, 1997). A closer look at LTSMs' working principles is given in following sections.

An LSTM cell's inner structure and its connections to other cells are illustrated in Figure 4.1. As shown in the figure, LSTM cells consist of pointwise vector addition and multiplication operations, as well as several gates made up of vectors flowing through various activation functions. Definitions of LSTM cells' activation functions, which are shown with red and green circles in Figure 4.1, are given in Section 4.1.1.

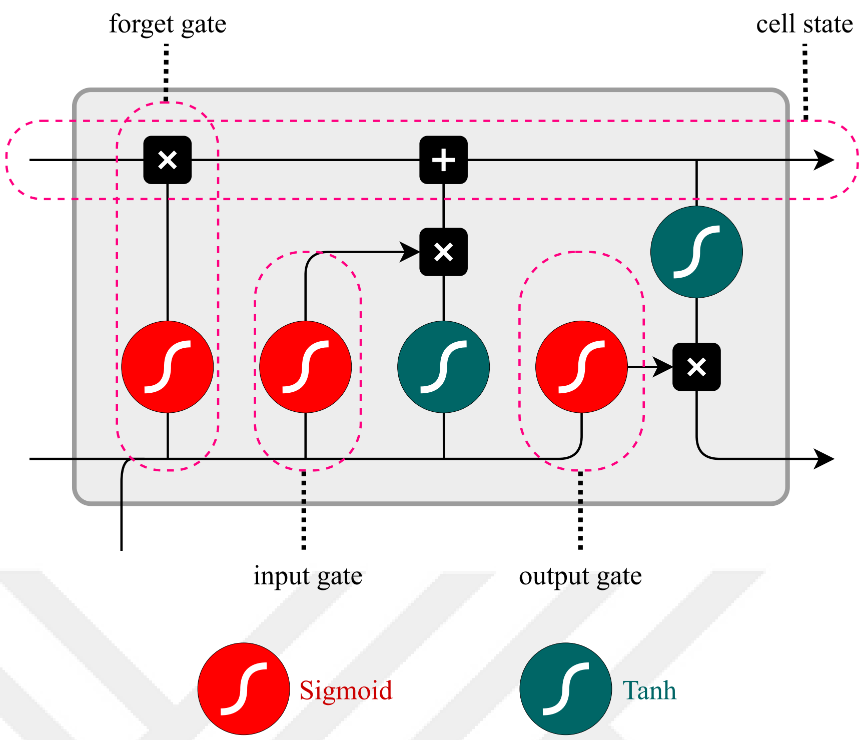


Figure 4.1 Internal structure of an LSTM cell (Olah, 2015).

4.1.1 Activation Functions

Neural networks multiply and add vectors in their layers sequentially. For regulating outputs of consequent multiplications, some forms of activation functions are used. Activation functions should map a vide possible range of inputs into a limited domain, and they should be differentiable for applications of Back-Propagation (Rojas, 1996).

In LSTMs, two different activation functions are utilized which are namely Sigmoid and Tanh. Sigmoid activation function (σ), as given in Equation (4.1) and shown in Figure 4.2, maps its input into (0, 1) interval (Szandała, 2020).

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4.1}$$

Tanh function has a similar shape to the Sigmoid function as displayed in Figure 4.2, but it maps the (-∞, ∞) interval into (-1, 1). The formal definition of the Tanh function is given in Equation (4.2) (Szandała, 2020).

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{4.2}$$

Both Sigmoid and Tanh activation functions are highly utilized in a wide range of neural network types. They both introduce non-linearity to the system and allow for classification in complex spaces. Even though the Sigmoid activation function is computationally less expensive, Tanh can represent mappings in negative domain, hence can save the neural network from getting stuck during the training phase.

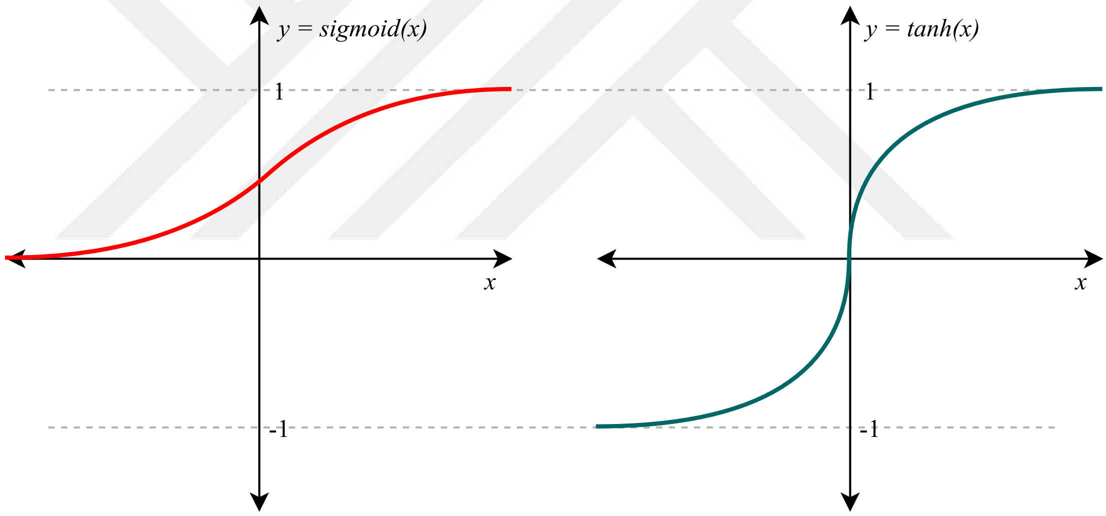


Figure 4.2 Sigmoid (left) and Tanh (right) function plots.

4.1.2 Long Short-Term Memory Cell Working Principles

The most important component of an LSTM cell is the cell carrying the information in the network chain like a conveyor belt (Olah, 2015). As shown with the blue line in Figure 4.3, data flows from an LSTM cell to another through cell state.

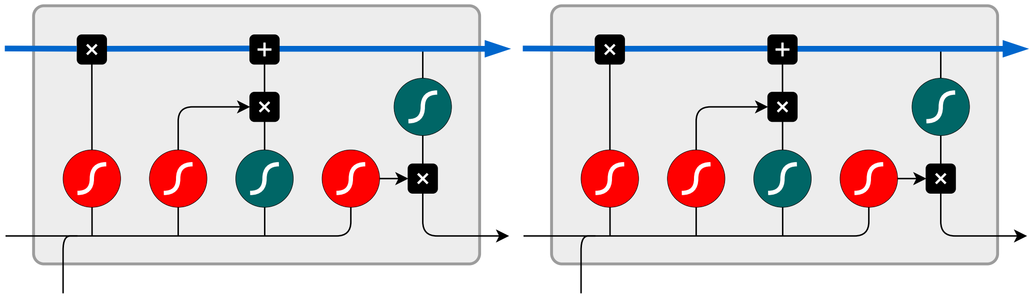


Figure 4.3 Depiction of data flow through LSTM cell state.

At any time step (t), x_t represents the current input, h_{t-1} represents the previous hidden state, and C_{t-1} represents the previous cell state. As shown in Figure 4.4, the previous hidden state and current input are concatenated into a new vector, and this new vector is mapped to the activation value of the forget gate, f_t, as shown in Equation (4.3) (van Houdt et al., 2020).

$$f_t = \sigma(W_f \,.\, [h_{t-1}, x_t] + b_f) \tag{4.3}$$

where W_f is the weight and b_f is the bias associated with the forget gate. At this step, if f_t is calculated to be close to 1, that means the previous cell state will be strongly remembered. Whereas if f_t is close to 0, the previous cell state is going to be forgotten.

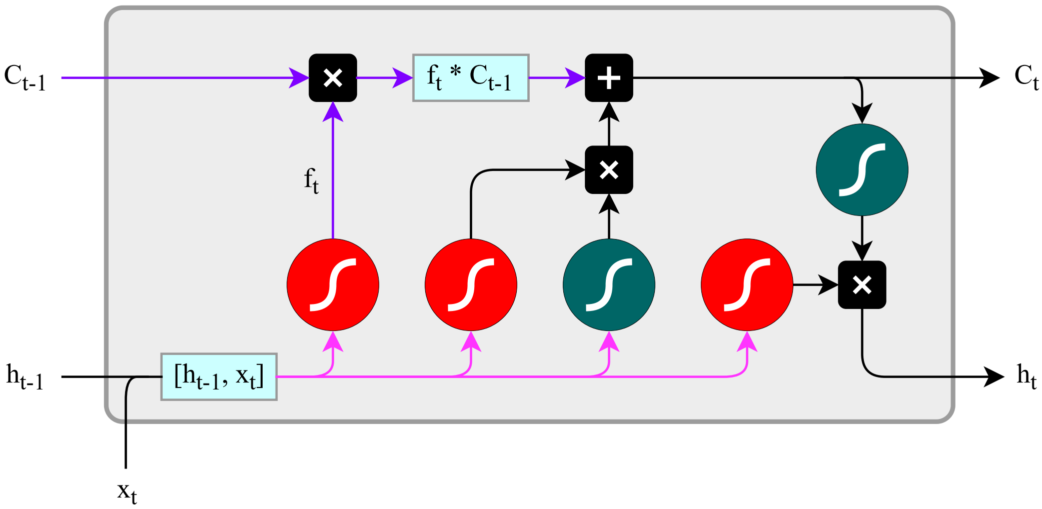


Figure 4.4 Data flow through the LSTM cell's forget gate.

In the next phase, as shown in Figure 4.5, cell state C_t gets updated by the input gate and gets its final value. C_t is calculated according to Equation (4.4), Equation (4.5), and Equation (4.6) (Olah, 2015).

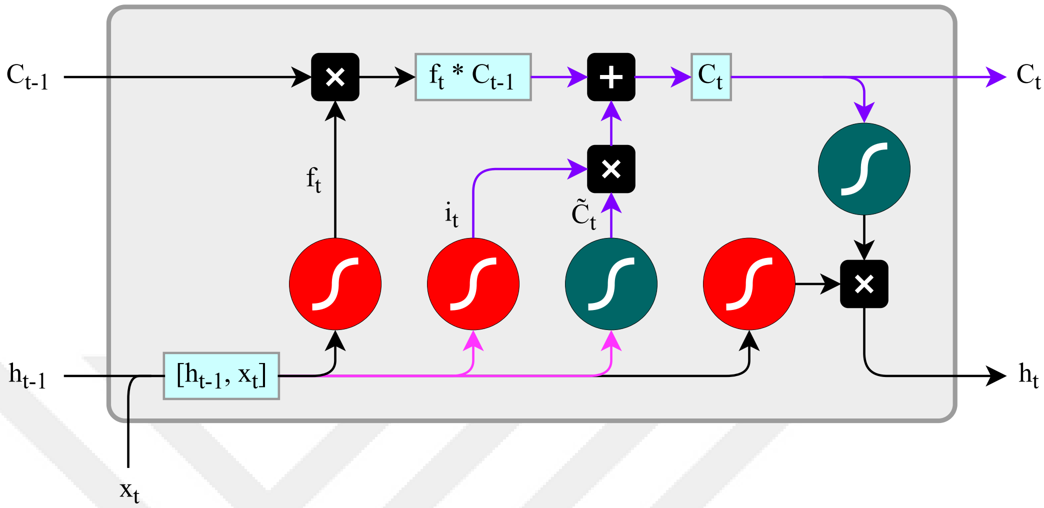


Figure 4.5 LSTM cell state calculation.

$$i_t = \sigma(W_i \, . [h_{t-1}, x_t] + b_i) \tag{4.4}$$

$$\tilde{C}_t = tanh(W_c \, . [h_{t-1}, x_t] + b_c) \tag{4.5}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4.6}$$

Finally, h_t, which is the current hidden state, is calculated according to Equation (4.7) and Equation (4.8) by using the output gate (van Houdt et al., 2020). Output gate decides what the hidden state for the next LSTM cell will be. The hidden state contains information about previous inputs (x) and is used for predictions. An illustration of hidden state calculation is shown in Figure 4.6.

$$o_t = tanh(W_o \, . [h_{t-1}, x_t] + b_o) \tag{4.7}$$

$$h_t = o_t * tanh(C_t) \tag{4.8}$$

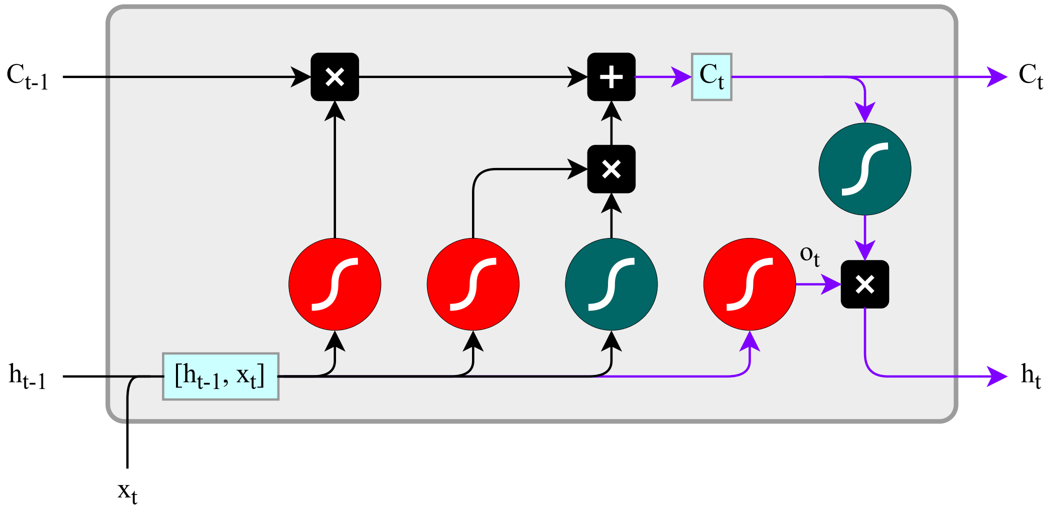


Figure 4.6 LSTM hidden state formation.

4.2 Probability Distribution Analysis of Various Features

Detailed objective evaluations regarding Usûl and form features were performed according to the methods proposed by Yang and Lerch (2020). An overview of Yang and Lerch's methodology is given in Section 2.4. In this section, the details of Yang and Lerch's method's workflow are given.

The general workflow of Yang and Lerch's method is illustrated in Figure 4.7. As described in Section 2.4, Yang and Lerch's method computes both absolute and relative metrics. Absolute metrics are computed within a given set to give insight about its characteristics whereas relative metrics show how related to sets are (Yang & Lerch, 2020).

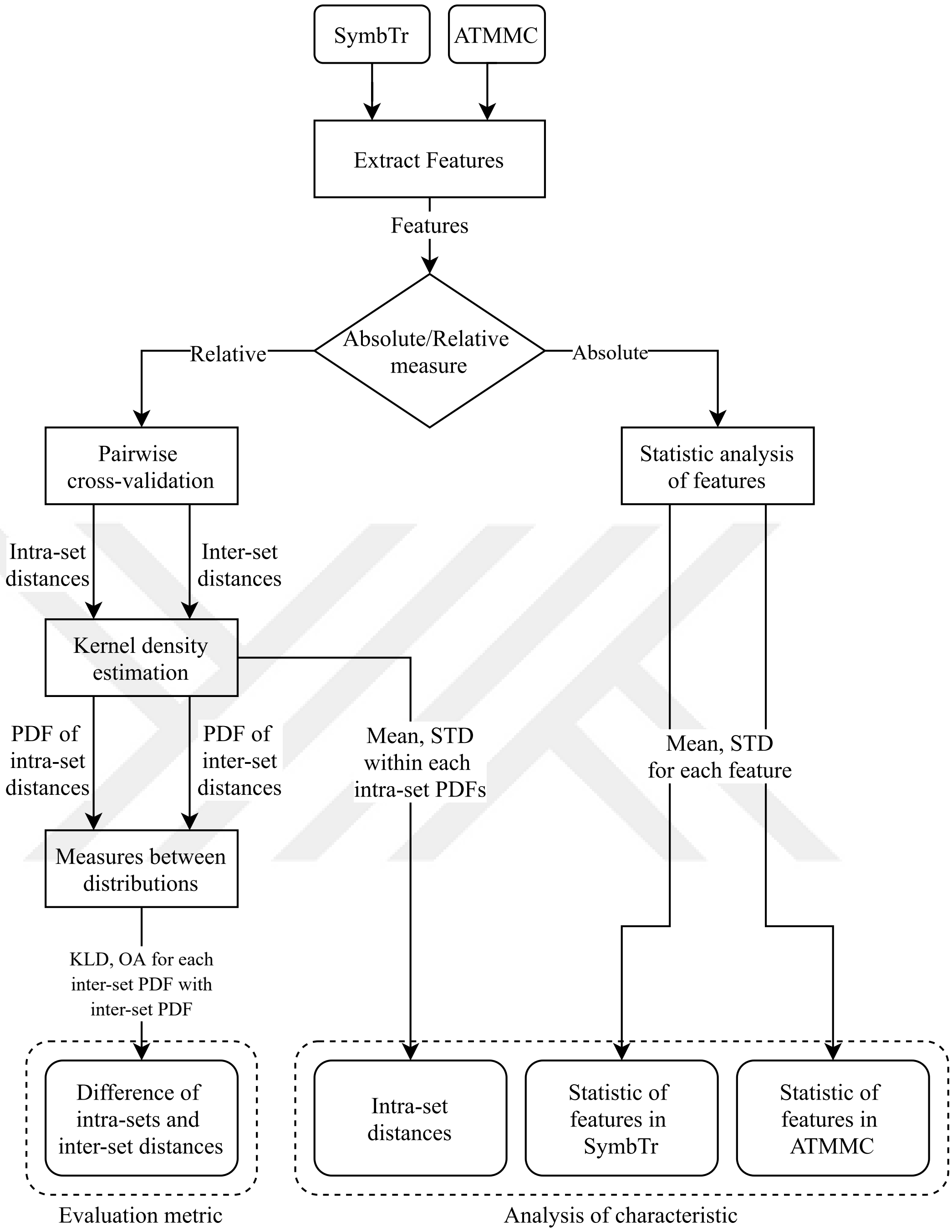


Figure 4.7 Workflow of Yang and Lerch's method (Yang & Lerch, 2020).

Absolute metrics are computed by calculating the mean and standard deviation (STD) of proposed features. However, relative metrics require a number of additional steps. Details of some techniques involved in relative metrics computation are given in the following sub-sections.

4.2.1 Pairwise Cross Validation

Pairwise cross validation is finding the distance of features both within and between sets. If there are two sets such as set_1 = [1, 2, 3] and set_2 = [5, 0, 7] the distance between sets are {*dist*([1], [5, 0, 7]), *dist*([2], [5, 0, 7]), *dist*([3], [5, 0, 7])}, where *dist* denotes Euclidian distance. Since the distance is computed between two sets in this example, it is called inter-set distance (Yang & Lerch, 2020).

When the distance is computer within the same set it is then called intra-set distance. As an example intra-set distance for set_2 would be {*dist*([5], [0, 7]), *dist*([0], [5, 7]), *dist*([7], [5, 0])}.

4.2.2 Kernel Density Estimation

Kernel density estimation is applied to inter-set and intra-set distance histograms for smoothing out the results into probability density functions (PDF). PDF is one of the best probability distribution functions to show how the whole mass is distributed over the x-axis (Węglarczyk, 2018).

An illustration of kernel density application is given in Figure 4.8. It can be seen from the figure that the density estimation with a Gaussian kernel smooths out segmented histograms.

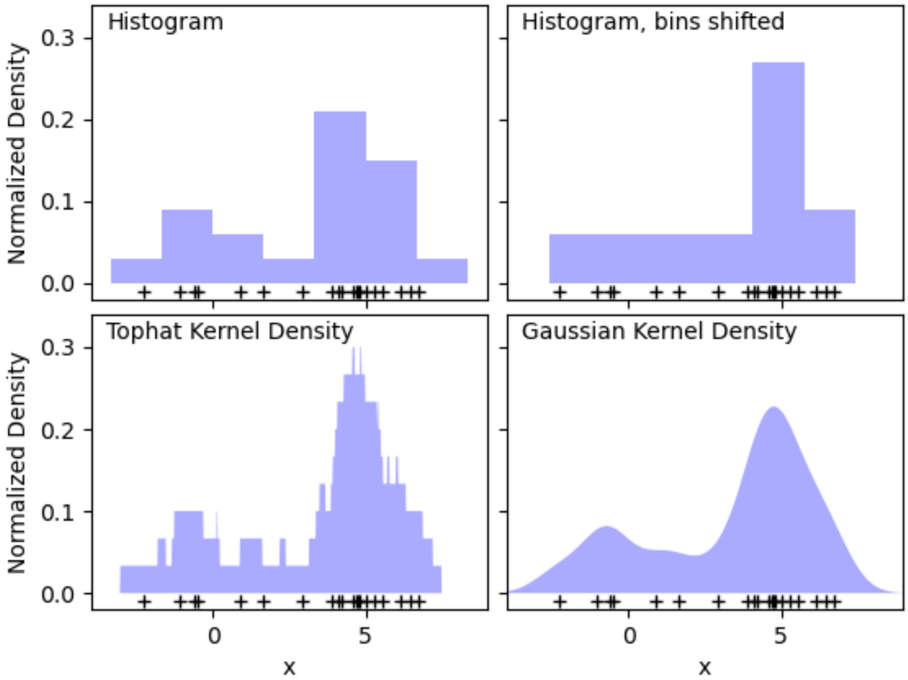


Figure 4.8 Kernel density estimation application (Vanderplas, 2007).

4.2.3 Kullback-Leibler Divergence and Overlapped Area

The final step of relative metrics calculation is finding the Kullback-Leibler Divergence (KLD) and Overlapped Area (OA) values. KLD quantifies the difference between probability distributions (Brownlee, 2019). Definition of KLD between two continuous probability densities *P(x)* and *Q(x)* is given in Equation (4.9) (Zhang et al., 2021).

$$KL(P(x) \mid\mid Q(x)) = \int_{-\infty}^{\infty} P(x) log \frac{P(x)}{Q(x)} dx \tag{4.9}$$

It is easy to see from Equation (4.9) that KL(*P(x)* || *Q(x)*) ≠ KL(*Q(x)* || *P(x)*), i.e., KLD is unbounded. Thus, Yang and Lerch (Yang & Lerch, 2020) suggest further calculating OA, to provide a bounded measure.

In contrast to KLD, OA is a measure of similarity between two PDFs. It is the area of intersection between multiple PDF (Pastore, 2018). Calculating both KLD and OA, hence gives the difference and similarity of the two datasets.

4.2.4 Method Demonstration

To further clarify Yang and Lerch's method (Yang & Lerch, 2020), two toy examples are given in this section. The first set of toy data are:

- base_set = [[18], [15], [18], [17], [19]]
- gen_set_1 = [[19], [18], [14], [18], [17]]
- gen_set_2 = [[1], [2], [1], [0], [2]]

The above sets represent 3 datasets. base_set represents the dataset of 5 pieces (samples) from which gens_set_1 and gen_set_2 are artificially created. The numbers represent pitch count per sample. Computed absolute metrics are given in Table 4.1. From both mean and STD values in Table 4.1, it is obvious that gen_set_1 is much similar to base_set than gen_set_2.

Table 4.1 Absolute measurement results for toy data 1.

	Mean	**STD**
base_set	17.4	1.35
gen_set_1	17.2	1.72
gen_set_2	1.2	0.74

Relative measurement results are shown in Table 4.2 and Figure 4.9. Results in Table 4.2 show that KLD metric between intra-baset_set and inter base_set – gen_set_1 is 0.1 / 0.04 = 2.5 times smaller than KLD metric between intra-baset_set and inter base_set – gen_set_2, which can be interpreted as gen_set_1 is 2.5 times less different to base_set compared to gen_set_2. But the OA difference is much higher. OA of the base_set and gen_set_1 is 0.76 / 2.35e-10 ≈ 3.23e+9 times larger than base_set and gen_set_2's OA.

Table 4.2 Relative measurement results for toy data 1.

	KLD	**OA**
base_set & gen_set_1	0.04	0.76
base_set & gen_set_2	0.10	2.35e-10

A visual representation of base_set, gen_set_1, and gen_set_2 difference PDFs is shown in Figure 4.9. Even though intra-base_set (blue), intra-gen_set_1 (green), and intra-gen_set_2 (purple) distributions have different density peaks, they align on the x-axis closely. But when PDFs of inter-set distances are investigated, base_set & gen_set_2 distance PDF lies on the right-hand side of the graph on the x-axis far away from the intra-set PDFs. This graph proves that PDF plots of inter-set distances can show similar and different datasets clearly.

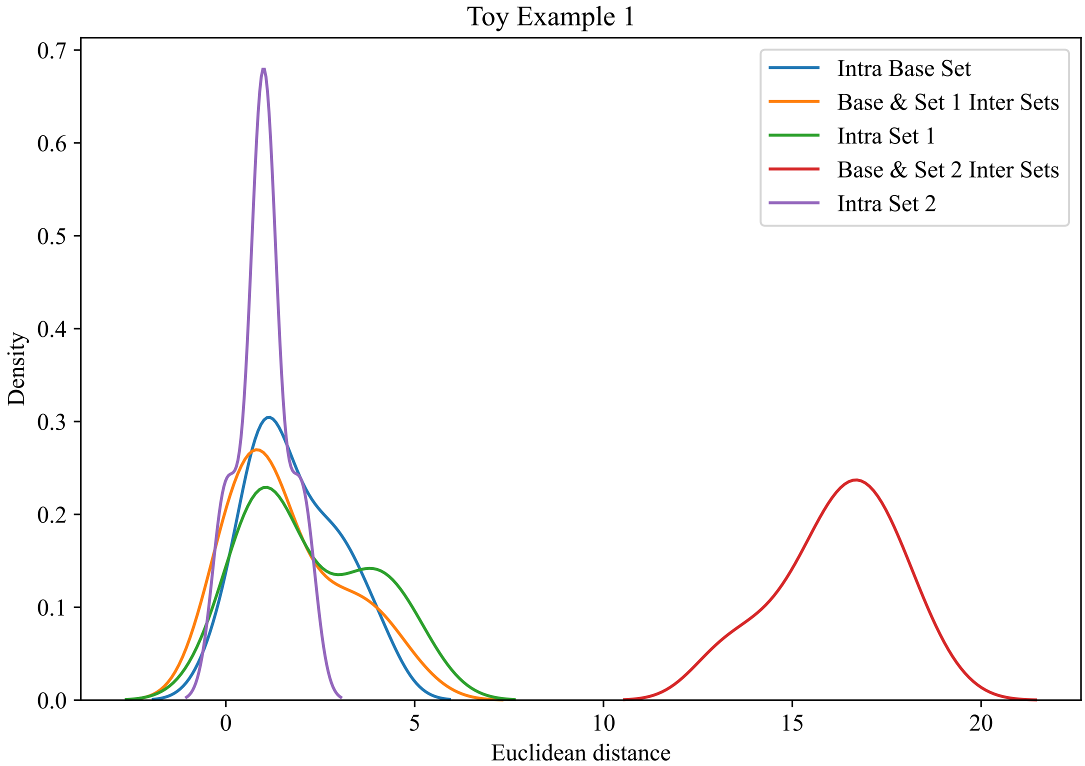


Figure 4.9 Probability distributions for toy data set 1.

The feature metrics in the first toy example had obvious differences. As shown in Table 4.2, both Mean and STD values for gen_set_1 and gen_set_2 were greatly different. For further investigation, another toy example is given below:

- base_set = [[18], [19], [18], [17], [19]]
- gen_set_1 = [[19], [18], [19], [18], [17]]
- gen_set_2 = [[1], [2], [1], [0], [2]]

Computed absolute metrics related to the second toy data are given in Table 4.3. In this example, the mean values are different, but the STDs are the same. Thus, merely looking for absolute metrics may not provide a distinction as clear as the first toy set.

Table 4.3 Absolute measurement results for toy data 2.

	Mean	**STD**
base_set	18.2	0.74
gen_set_1	18.2	0.74
gen_set_2	1.2	0.74

Relative measurement metrics and plots of PDFs for toy data 2 are given in Table 4.4 and Figure 4.10. Table 4.4 shows that just looking at KLD values, gen_set_2 seems to be more similar to base_set than gen_set_1. But when OA values are examined, gen_set_1 is found to have 0.67 / 1.98e-47 ≈ 3.38e+46 times larger OA with base_set, compared to gen_set_2.

Table 4.4 Relative measurement results for toy data 2.

	KLD	**OA**
base_set & gen_set_1	0.02	0.67
base_set & gen_set_2	0.01	1.98e-47

In Figure 4.10, intra-base_set (blue), intra-gen_set_1 (green), and intra_gen_set_2 (purple) PDFs are shown to be aligned nearly perfectly that they are overlapping and covering each other up. However, again examining the inter-set differences show that gen_set_2 is much more similar to base_set than gen_set_2. From both toy set 1 and 2 figures, it can be deduced that placement of inter-set distance PDFs on the x-axis is effective in showing how similar or different the datasets are.

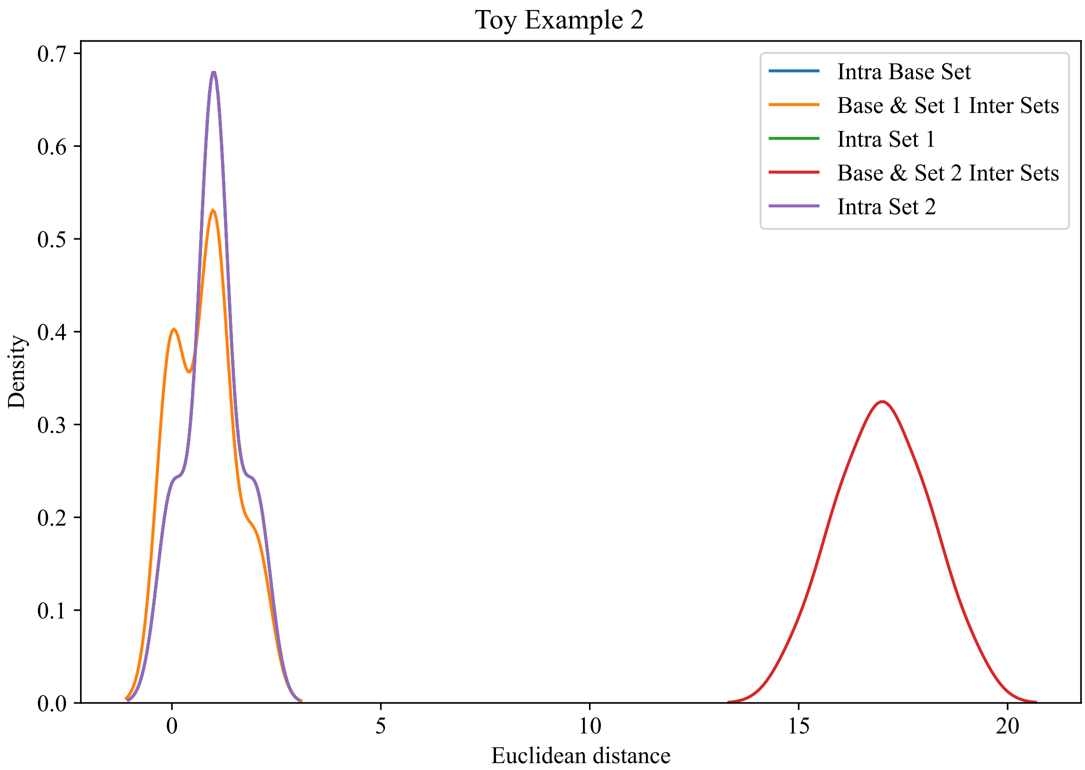


Figure 4.10 Probability distributions for toy data set 2.

CHAPTER FIVE
AUTOMATIC COMPOSER

Automatic Turkish Makam Music Composer (ATMMC) is a collection of Deep Learning (DL) and N-gram probability models. In this chapter, its building blocks and their inner structures are given in detail.

5.1 Base Models

The Base Model (BM) is the most important building block of ATMMC. It is a Deep Neural Network (DNN) from which all other DNNs of ATMMC are derived. As illustrated in Figure 5.1, BM consists of a 600-unit LSTM layer followed by a 50% dropout layer followed by again 600 units LSTM and 50% dropout layers, and finally, a fully connected dense layer and a SoftMax activation layer (Parlak et al., 2021). BM is compiled with categorical cross-entropy loss function and the RMSprop (root mean square prop) optimizer. The learning rate of RMSprop optimizer is 0.001.

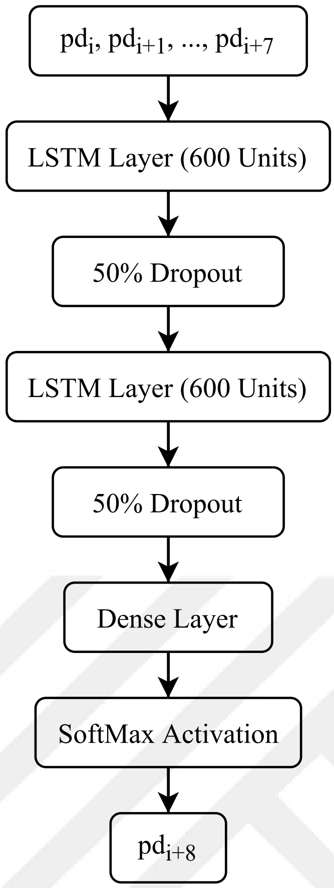


Figure 5.1 Base Model's layer architecture.

As described in Section 3.2, musical notes are represented as pitch-duration tuples encoded into one-hot vectors. BMs' training input and output data are created by matching 8 pitch-duration pairs with 1 following pitch-duration pair. As an example, if the processed piece is $[pd_0, pd_1, pd_2, \ldots, pd_{k+8}]$, then the training input set becomes $[[pd_0, pd_1, \ldots, pd_7], [pd_1, pd_2, \ldots, pd_8], \ldots, [pd_k, pd_{k+1}, \ldots, pd_{k+7}]]$ and training output set becomes $[[pd_8], [pd_9], \ldots, [pd_{k+8}]]$, where each pd_i is a one-hot encoded vector representing a pitch-duration tuple.

Trained BMs learn the conditional probability $P(pd_{t+1} \mid pd_{t-7}, ..., pd_t)$ of the next pitch-duration tuple pd_{t+1} at any given time t with respect to the previous 8 pitch-duration tuples. Thus, in fact, they learn to compose the next note of a musical piece according to the most recent 8 notes.

ATMMC is designed to compose musical pieces with certain Usûl and forms. But BMs are not aware of any compositional structure. Since they are trained on the whole dataset of their target Makams and their datasets consist of mixed Usûl and forms, they learn the general feel of their target Makams. On their own, they output pleasant melodies in their target Makams' scales, but their compositions lack structure.

For each Hicaz and Nihavent Makams, multiple BMs with various hyperparameters were trained and 2 subjectively most musical sounding models were chosen to form the basis for each Makam's Specialist Models.

5.2 Specialist Models

Specialist Models (SM) are the models that specialize in composing sections of Şarkı form in certain Usûls for Hicaz and Nihâvent Makams. An SM can either compose just one of the Zemin, Nakarat, or Meyan sections.

In SymbTr collection, there are 17 pieces within Hicaz Makam having Aksak Usûl and Şarkı form, and the number of Nihâvent pieces in Düyek Usûl and Şarkı form is 21. The number of available pieces in the SymbTr collection is not sufficient for training DL models that are capable of producing structured TMM pieces. To solve this problem, 88 additional pieces were collected from various archives and manually converted to Mu2 formats. Also, as well as the available pieces in SymbTr dataset, their Zemin, Nakarat, and Meyan sections are manually labeled.

SMs are deep LSTM networks that are derived from BMs inheriting their weights. As shown in Figure 5.2, BMs and SMs have the same architecture. Also, as shown in Figure 5.2b, SMs first LSTM layer is frozen, and their other layers are left trainable. The technique of inheriting other models' weights is called Transfer Learning and it is especially useful when the amount of training data is limited (Weiss et al., 2016).

With transfer learning, a DL model inherits its initial weights from an already trained model on a similar domain. During the training phase, having knowledge in a similar domain, the derived model learns new relations about its training data in addition to its past expertise gathered from its base model. This way, knowledge is transferred between models operating on similar domains. The difference between frozen layers and trainable layers shown in Figure 5.2b is weights in trainable layers are updated during the training phase, but weights in the frozen layers are kept unchanging.

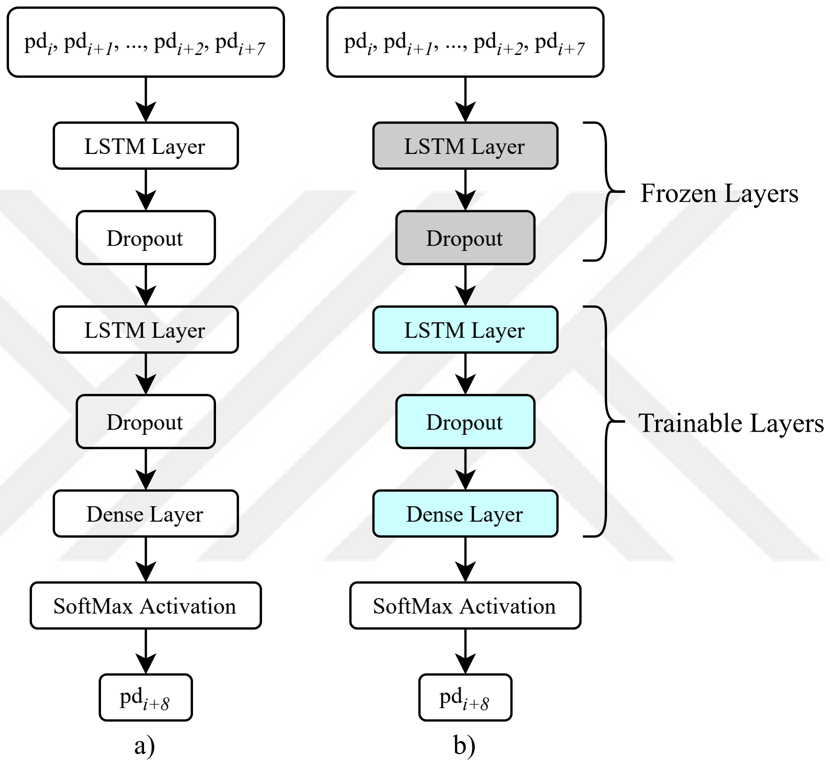


Figure 5.2 Base model diagram (a) and specialist model diagram (b).

For each Zemin, Nakarat, and Meyan sections of target Makam's Şarkı forms, two SMs are trained, adding up to a total of 6 SMs per Makam. As shown in Figure 5.3, Zemin SM is trained for composing Zemin section from 8 initial notes (pitch-duration tuples), Nakarat SM is trained for composing the Nakarat section by using the last 8 notes of the Zemin section, and finally, Meyan SM is trained for composing the Meyan section by using the last 8 notes of Nakarat section. By passing the last 8 notes of each section to the SMs of next section, a connection and harmony between Zemin, Nakarat, and Meyan sections are established.

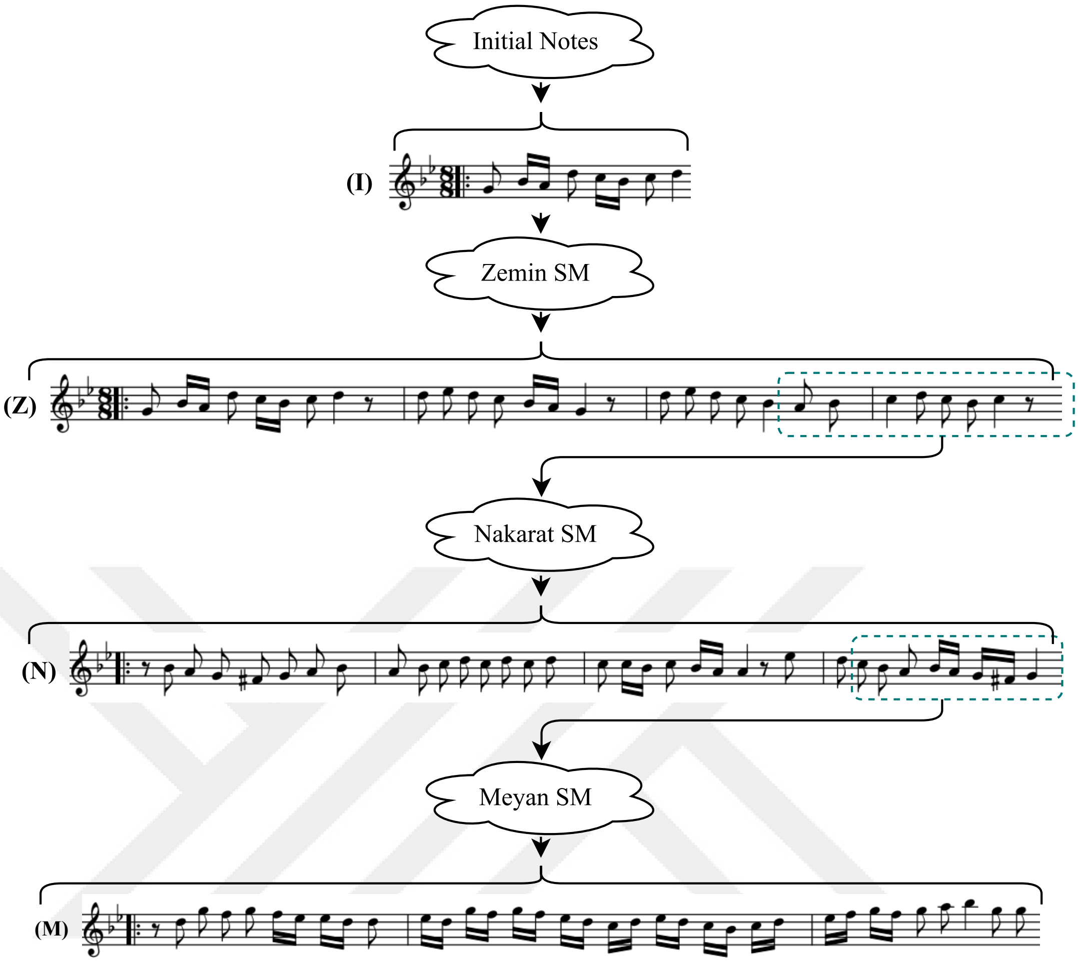


Figure 5.3 Operation scheme of Specialist Models.

The final layer of SMs, the SoftMax layer, outputs a probability distribution as given in Equation (5.1), where z_i represents each element of SM's output vector, and k is the number of unique notes.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{5.1}$$

From SM's output, the prediction for the next note is extracted. Two possible outputs of SMs are shown in Figure 5.4. If SM is confident with its prediction, it will favor a single note, which is a strong candidate, above all other possible notes as shown in Figure 5.4a. But if SM is not confident with its prediction, as shown in Figure 5.4b, it will output several low probability peaks, i.e., multiple weak candidates.

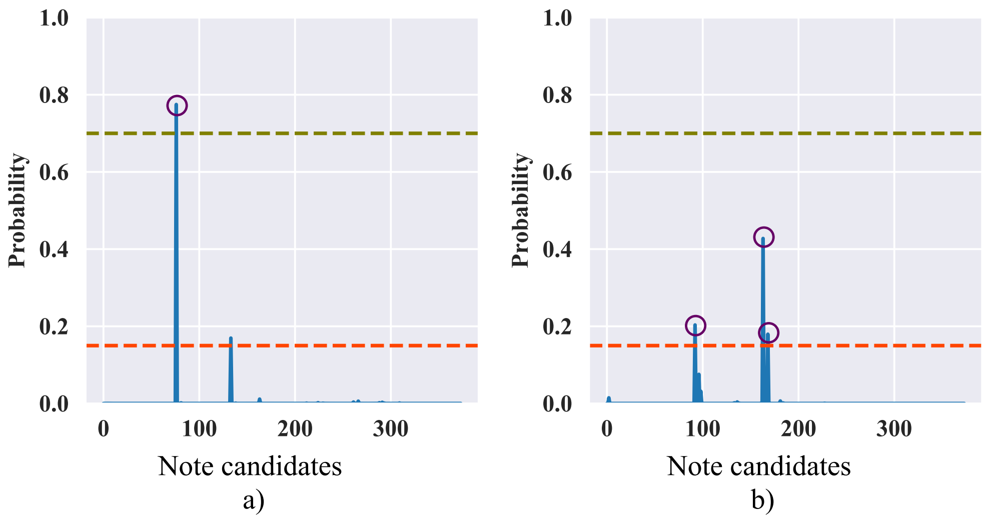


Figure 5.4 Specialist model prediction samples.

Only 1 note should be chosen from SMs candidates. For choosing the candidate from the probability distribution, high and low threshold values are determined. In Figure 4.10, the high threshold is 0.70, and it is shown with a green line. Whereas the low threshold value, which is shown with the red line, is 0.18. In the case of an existing peak above the high threshold, it is chosen to be the next note. But, if there is no peak above the high threshold, peaks above the low threshold are inspected, and the one with the strongest 4-gram probability is chosen to be the next note as given in Equation (5.2). Where pd_t represents 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.

$$P(pd_t \mid pd_{t-3}, pd_{t-2}, pd_{t-1}) = \frac{Count(pd_{t-3}, pd_{t-2}, pd_{t-1}, pd_t)}{Count(pd_{t-3}, pd_{t-2}, pd_{t-1})} \tag{5.2}$$

After 4-gram probability calculation, if a tie occurs between multiple candidates, the next note is chosen randomly from the candidate list.

The artificial compositions are greatly affected by tuning the high and low threshold values. Playing with the threshold values may cause more candidates to pass barriers causing more adventurous or traditional synthetic compositions depending on their values. Moreover, since Zemin, Nakarat, and Meyan sections are composed by separate SMs, conventionality of each section can be separately controlled by related SM's high and low threshold values.

5.3 Conductor Models

As described in Section 5.2, two different SMs are trained per Zemin, Nakarat, and Meyan sections for each Makam. This results in two candidates for the next note to be composed, predicted by each SM. In order to choose one candidate over the other Conductor Models (CoMo) are trained. Again, as illustrated in Figure 5.5, CoMos are deep LSTM networks, which have 3 layers of 100 LSTM units and a fully connected final layer. Between their layers, CoMos have 50% dropout factors. CoMos are compiled with RMSprop (root mean square prop) optimizers and categorical cross-entropy loss functions.

Similar to the training process of SMs, CoMos are first trained over the whole dataset for their target Makam. This way they learn the general characteristics of their Makam without explicit Usûl and form information. Then the trained CoMos are again trained on Zemin, Nakarat, and Meyan sections only in their target Usûls.

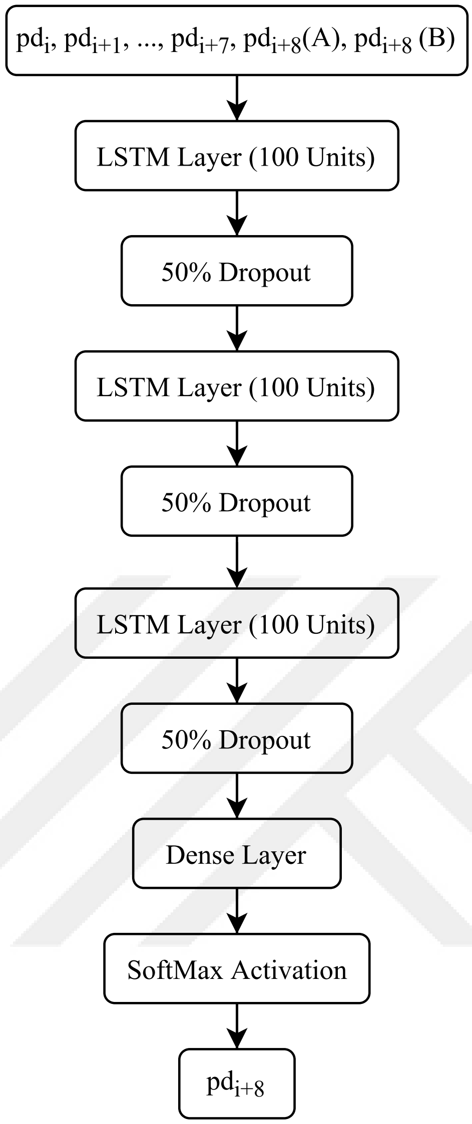


Figure 5.5 Conductor Model structural decomposition.

As shown in Figure 5.6, CoMos receive the 8 previous pitch duration tuples shown as pd$_i$, pd$_{i+1}$, …, pd$_{i+7}$ as well as candidates from two SMs shown as pd$_{i+8}$(A) and pd$_{i+8}$(B). Then, they output a vector $v = [v_a, v_b] : v_a + v_b = 1$ denoting the probability of SM(A)'s and SM(B)'s candidates. If $|v_a - v_b| \geq 0.2$, the candidate with greater probability is determined to be the next note. But, if the probability difference between the candidates is smaller than 0.2, this means CoMo is not very confident with its output, thus, the next note is determined by a random selection between pd$_{i+8}$(A) and pd$_{i+8}$(B).

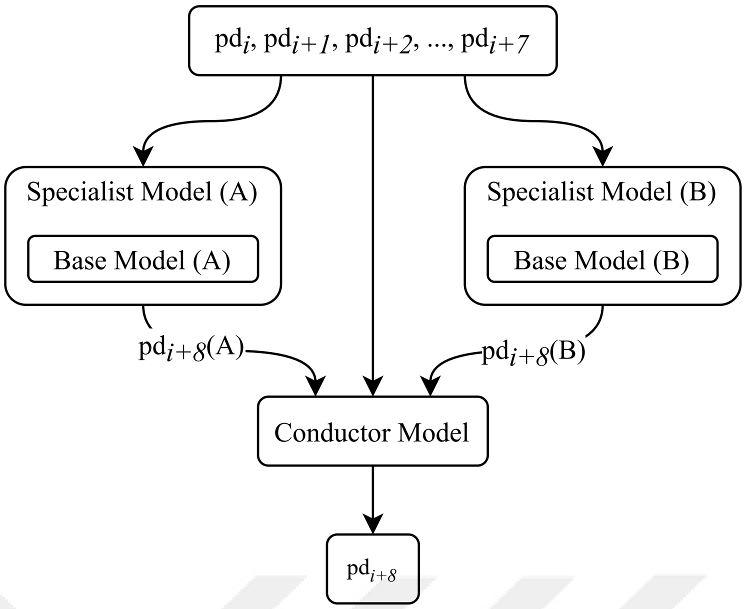


Figure 5.6 Structural decomposition of ATMMC.

5.4 Full System Overview

ATMMC is a collection of SMs and CoMos. There are 6 SMs and 3 CoMos for each of Hicaz and Nihavent Makams, adding up to 12 SMs and 6 CoMos for the whole system.

As illustrated in Figure 5.7, ATMMC system works as follows: after choosing the target Makam and Usûl, a user enters 8 initial pitch-duration tuples (pd_0, pd_1, …, pd_7) to the system. Then ATMMC composes the 9^{th} note (next_note) according to user input and appends it to the end of the notes list (Song). Next, ATMMC picks the last 8 notes from the notes list and composes the 10^{th} note (next_note) according to the selected 8-note set. This process of picking the last 8 notes and composing the next note repeats until a 4-bar long Zemin section is completed.

When the Zemin section is completed, ATMMC picks the last 8 notes of the Zemin section and begins composing the Nakarat section. Here, it should be noted that ATMMC changes its SMs and CoMos from ATMMC_Zemin to ATMMC_Nakarat which are specialized in the Nakarat sections. After composing the 4-bar long Nakarat section, ATMMC picks the last 8 notes from the Nakarat section and composes the

Meyan section similarly. Again, while composing the Meyan section, only Meyan section associated SMs and CoMos (ATMMC_Meyan) are utilized.

Once composition processes of Zemin, Nakarat, and Meyan sections are completed, ATMMC merges the composed sections into a single Şarkı (Mus2_File) and writes that Şarkı to the disk in Mu2 format.

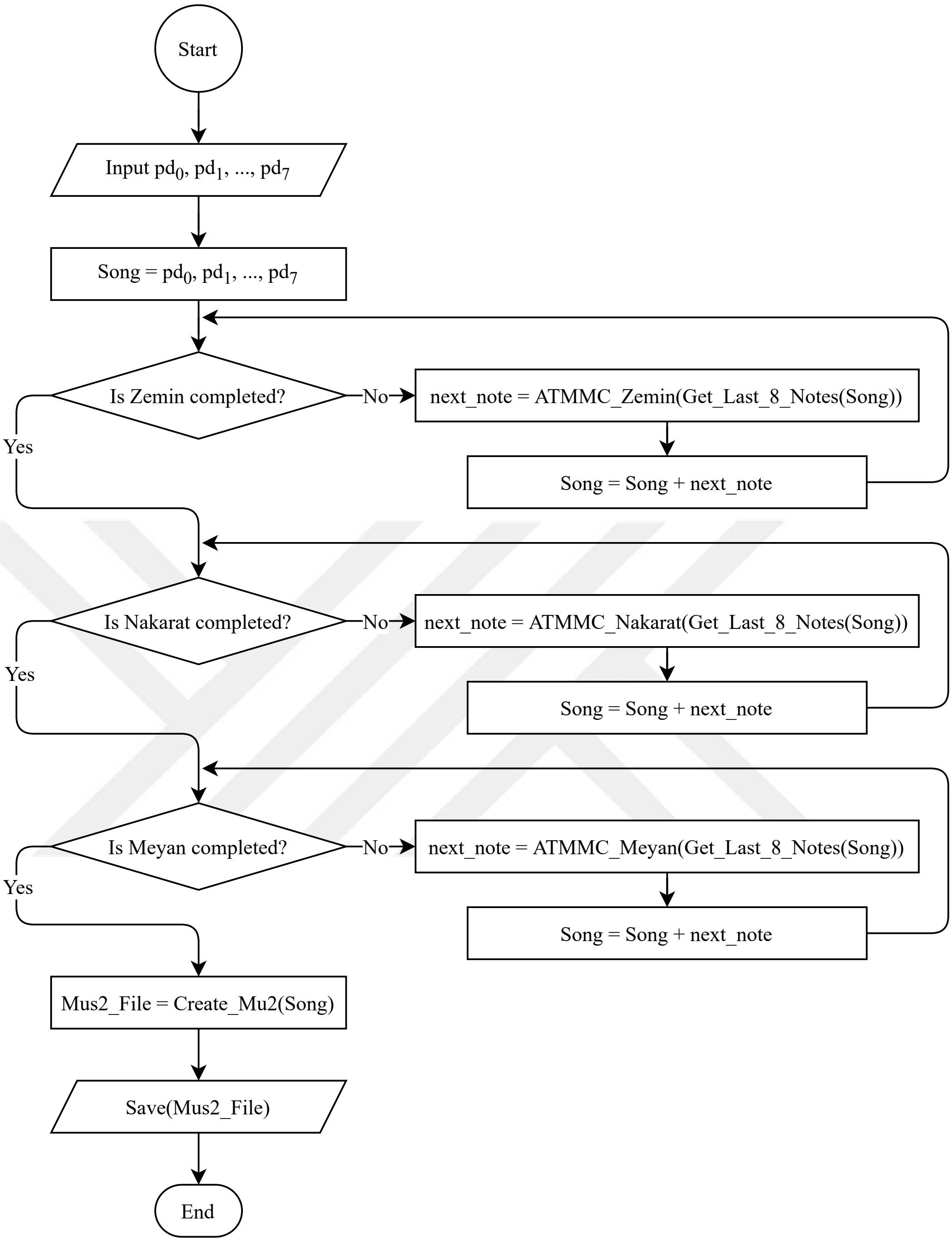


Figure 5.7 ATMMC composition process flowchart.

CHAPTER SIX
GRAPHICAL USER INTERFACE FOR ATMMC

6.1 Purpose of the Graphical User Interface

Automatic Turkish Makam Music Composer (ATMMC) is developed in Python programming language as an open-source project and the source of the application is publicly accessible at https://github.com/ihpar/TMMDLFT. Users can install Python and the necessary libraries on their computers, run the application with the terminal, and create Mus2 files via ATMMC according to the input parameters. However, users who want to use ATMMC via terminal should be familiar with computer programming, albeit at a basic level.

A graphical user interface that can be accessed with an internet browser has been published at http://music.cs.deu.edu.tr/tmmgui to eliminate this programming knowledge requirement and make ATMMC available to users who do not have any computer programming experience (Parlak et al., 2021). With this interface, users can compose artificial creations in ATMMC and save the result file on their computers. The recorded files can then be viewed and played back with the Mus2 application.

6.2 Graphical User Interface Decomposition

The graphical user interface developed for ATMMC is shown in Figure 6.1. Due to the large size of the interface, the picture is split into two, and the details of the left-hand side of the interface, denoted as “L” in Figure 6.1, is given in Figure 6.2. Likewise, the details of the right-hand side denoted as “R” in Figure 6.1, is given in Figure 6.3 for better visibility.

Figure 6.1 Overview of ATMMC's graphical user interface.

As shown in Figure 6.2, the voicing feature of the interface can be turned on or off with the button indicated with "A". While the voicing feature of the graphical interface is active, users can hear the notes they add. In addition, when the space key is pressed, users can listen to all notes they have inserted, one after the other. The frequencies of the pitches used in the vocalization were determined according to the Arel-Ezgi-Uzdilek (AEU) system using the TuneJS library (Bernstein & Taylor, 2003).

Makam and Usûl selection functionalities are shown in Figure 6.2 "B" and "C" respectively. From Figure 6.2 "B", the user can switch between Hicaz and Nihâvent Makams. Likewise, from Figure 6.2 "C", the user can switch between Düyek and Aksak Usûls. When Hicaz Makam is selected, Usûl automatically switches to Aksak. Similarly, when Nihâvent Makam is selected Usûl automatically switches to Düyek. Also changing the target Usûl and Makam updates the time signature and key signature automatically.

From the section shown in Figure 6.2 "D", the user can pick pitch alteration symbols if needed. The pitch alteration symbols given in this section are again from AEU theory.

The section shown in Figure 6.2 "E" is where the user enters and modifies the 8 initial notes required for executing ATMMC.

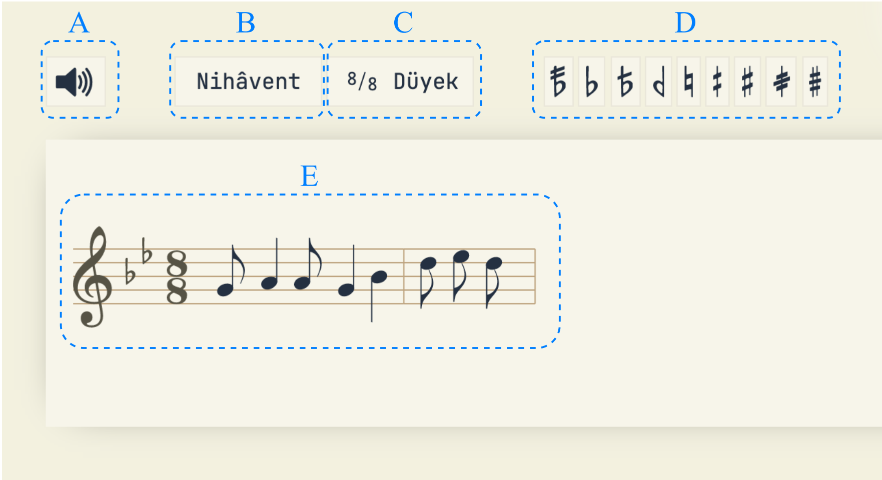


Figure 6.2 ATMMC graphical user interface left-hand side closeup.

In Figure 6.3 “A”, duration symbols are shown. The user can click and select any duration from this section and add the next note accordingly. Silence symbols are shown in Figure 6.3 “B”. Just like inserting notes, the user can enter silence symbols for various durations. And finally, the button for triggering ATMMC’s composition process is shown in Figure 6.3 “C” (Bestele; *Compose*).

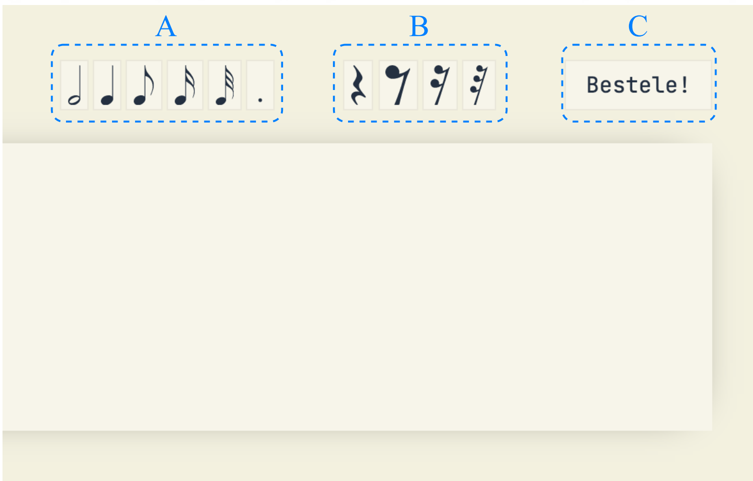
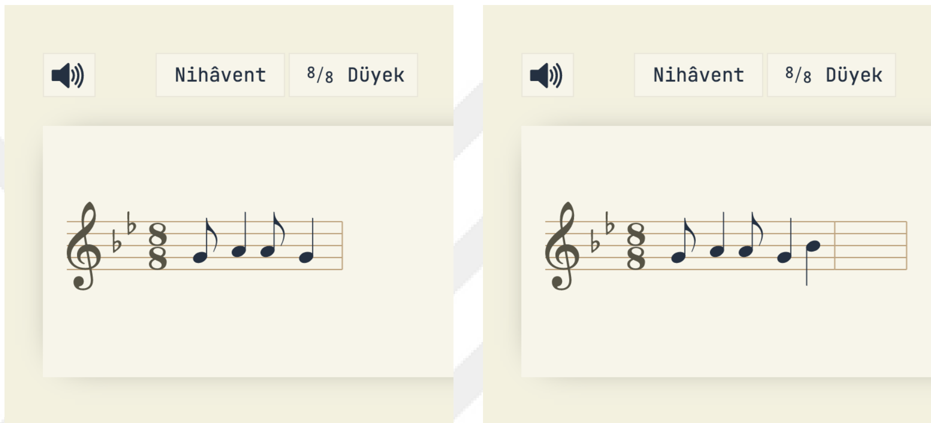


Figure 6.3 ATMMC graphical user interface right-hand side closeup.

As a general guideline for the graphical user interface, the user can select any of the note durations, pitch alteration symbols, or silence symbols from the menu by mouse clicks and then click to the desired area onto the section marked in Figure 6.2 “E” for inserting the selected element. If the user wants to delete the inserted item, highlighting the unwanted item by clicking on it then pressing the delete key on the computer's keyboard will do the task.

As shown in Figure 6.4, a new measure is created automatically when the old measure is fully occupied. In Figure 6.4a, total duration within the measure is 1/8 + 1/4 + 1+8 + 1/4 = 6/8. When a 1/4 note is inserted as shown in Figure 6.4b, the total duration reaches 6/8 + 1/4 = 8/8 and an empty measure is automatically created allowing for new note insertions.



a) b)

Figure 6.4 Before (a) and after (b) of automatic measure creation.

If the total duration in the current measure is incorrect, for example, higher than 8/8 for Düyek Usûl, as shown in Figure 6.5, the notes in the measure turn red, indicating the occurrence of an error in the measure. When the user creates a faulty measure, the button that triggers the composition process gets disabled and the system does not allow the automatic composition process to start.

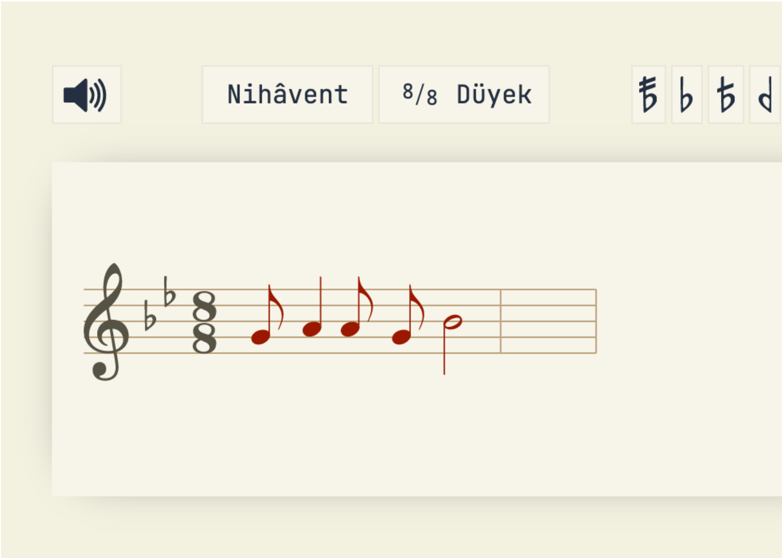


Figure 6.5 An erroneous measure.

After the user enters 8 initial notes correctly, the button indicated with Figure 6.3 “C” becomes active and it becomes possible for the user to send the notes he has written to ATMMC.

When the user initiated the composition process, the status window shown in Figure 6.6 pops up and the progress of the composition processes running in the ATMMC server is displayed to the user in real-time.

Figure 6.6 Composition progress report pop-up.

The status steps in pop-up window shown in Figure 6.6 are in Turkish and they can be translated as:

- Başlangıç: Start
- Besteci başlatıldı: Composer started
- Zemin besteleniyor: Zemin is being composed
- Nakarat bestelenecek: Nakarat is going to be composed
- Meyan bestelenecek: Meyan is going to be composed
- Bitiş: Finish
- İndirme hazırlanacak: Download is going to be prepared

Depending on the ATMMC server load, the composition process can take between 30 seconds and 3 minutes. When the composition process is finished, the composition progress pop-up transitions to its final state as shown in Figure 6.7. Finally, the user can save the ATMMC's artificial composition as a Mus2 compatible file, by clicking the "download ready" link.

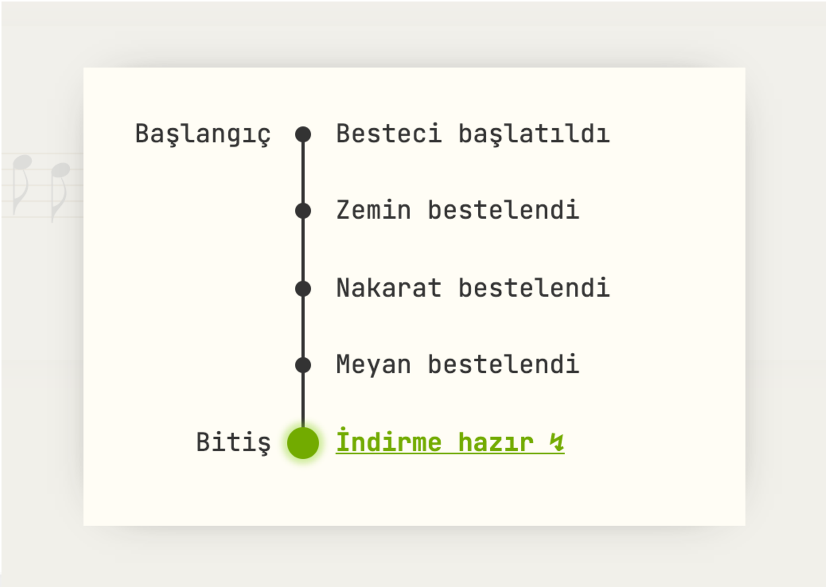


Figure 6.7 Artificial composition download link.

The status steps in pop-up window shown in Figure 6.7 are in Turkish and they can be translated as:

- Başlangıç: Start
- Besteci başlatıldı: Composer started
- Zemin bestelendi: Zemin is composed
- Nakarat bestelendi: Nakarat is composed
- Meyan bestelendi: Meyan is composed
- Bitiş: Finish
- İndirme hazır: Download is ready

CHAPTER SEVEN
RESULTS & EVALUATION

Both subjective and objective evaluation methods were performed for evaluating the ATMMC's effectiveness. For evaluation, 20 Hicaz and 20 Nihâvent pieces were composed by ATMMC, where each piece consisted of 4 bars of Zemin, Nakarat, and Meyan sections. Details and results of the evaluations are given in the following sections.

7.1 Subjective Evaluation

Subjective evaluations were performed by asking a series of questions in the form of a questionnaire to artists and researchers who are accepted as domain experts in Turkish Makam Music. A total of 5 artists participated in the survey and evaluated ATMMC's artificial compositions according to the questions listed below:

- Question 1: To what extent do the ATMMC compositions you have listened to represent the Hicaz Makam?
- Question 2: Do the pieces you have listened to have the characteristics of Aksak Usûl?
- Question 3: Do the pieces you have listened to have the characteristics of Şarkı form?
- Question 4: How original are the pieces you have listened to?
- Question 5: If you hadn't been told that they were composed by a computer, would you think that the songs you've heard were composed by a human being?
- Question 6: Did the songs you have listened to hold artistic beauty?
- Question 7: Would your thoughts change if you listened to the songs you just heard from a real-life professional performer and not from a MIDI instrument?
- Question 8: For what purpose can this artificial composer be used?

The responses of the participants to the first and second questions are given in Figure 7.1. Participants replied that ATMMC's artificial compositions reflect their target Makams perfectly and reflect their target Usûls above average. It should be noted that ATMMC's ability to reflect its targeted Usûls is lower than its ability to reflect its targeted Makams. This is a consequence related to the quantities in training sets. In training sets, the number of pieces in a certain Makam highly surpasses the number of pieces in discrete Usûls. Having much higher amounts of representatives, Makams are better leaned compared to Usûls.

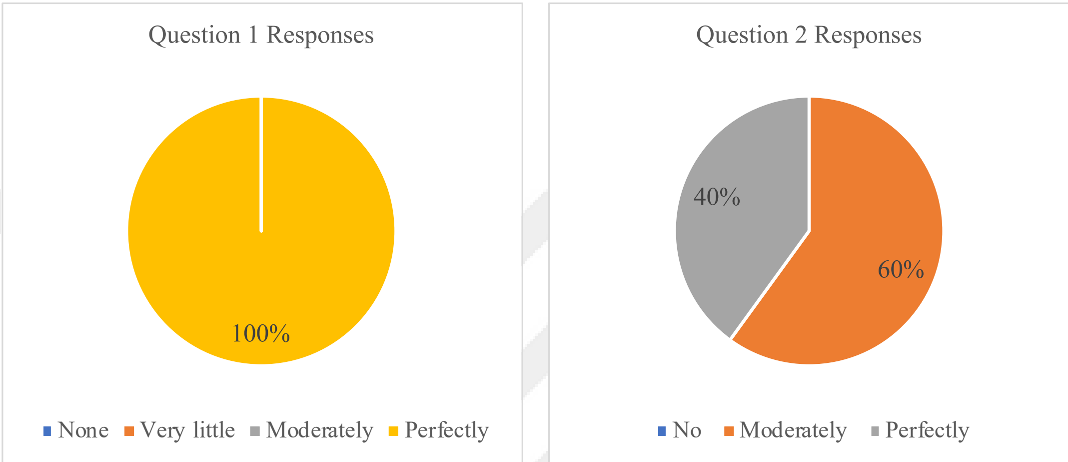


Figure 7.1 Subjective evaluation responses for questions 1 and 2.

Responses to question 3, shown in Figure 7.2, exhibit that the participants think ATMMC's artificial compositions are moderately coherent with the characteristics of Şarkı form. Again, this result can be interpreted similarly to the results of the second question, which is related to the fewer representatives of Şarkı form in the training set.

Responses to question 4, shown in Figure 7.2, reveal that the participants found the ATMMC's compositions moderately original. From the answers given to question 4, it can be concluded that the ATMMC has learned the dataset instead of overfitting it.

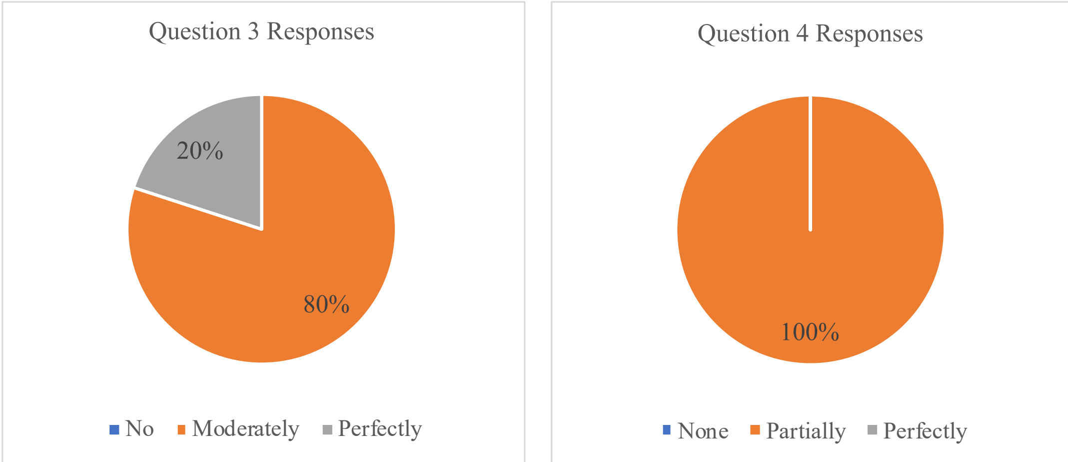


Figure 7.2 Subjective evaluation responses for questions 3 and 4.

It can be seen from Figure 7.3 that the participants would think ATMMC was a human being if they were not told that it was a machine, and they moderately find its compositions artistically pleasing. This result indicates that the system can moderately mimic the composers who composed the dataset it was trained on.

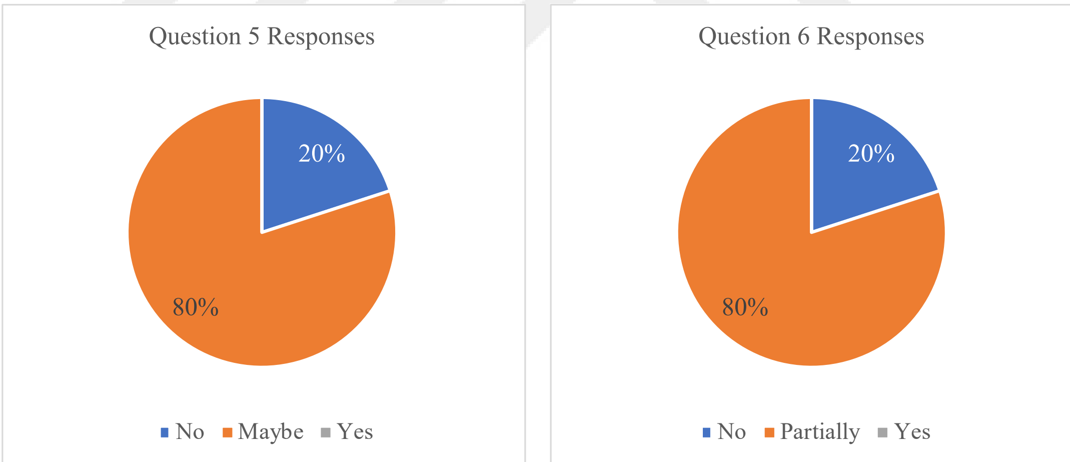


Figure 7.3 Subjective evaluation responses for questions 5 and 6.

The participants would rate ATMMC higher if the synthetic compositions were played back to them from a professional real instrument recording rather than MIDI sounds, as shown in Figure 7.4. This result is particularly important for future studies to record the synthetic compositions with real instruments in a professional environment.

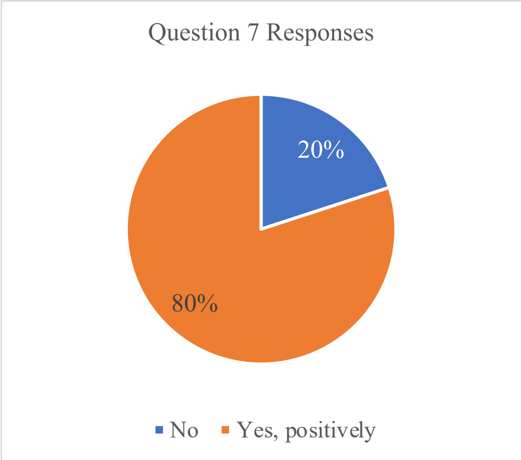


Figure 7.4 Subjective evaluation responses for question 7.

Finally, responses to question 8 are given in Table 7.1. A small percentage of the participants found ATMMC to be useless. Whereas most participants thought that ATMMC could help human composers in composing new musical pieces by introducing new ideas. At the same time, some participants thought that the system can be used to quickly generate etudes for music students in conservatories.

Table 7.1 Subjective evaluation responses for question 8.

Response	Percentage
It is useless.	20%
May assist human composers in composing new pieces.	60%
It can be used to create etudes for students.	20%

In general, when all the subjective evaluations are considered, it can be said that ATMMC can compose above-average pieces in terms of pitch, rhythm, and form representation abilities. Also, it can be deduced that ATMMC cannot be used to replace human composers but to assist them.

7.2 Objective Evaluation

Objective evaluations on ATMMC's compositions were carried out in two branches: pitch density analysis on the overall data set and detailed analysis on song forms with Usûl information. The details of performed analysis and findings are given in following sections.

7.2.1 Pitch Density Analysis

Pitch density analysis was performed by finding the total duration of all pitches in both the SymbTr and ATMMC-compositions datasets. While calculating the total duration, the tempos of the processed pieces were accepted to be the same. In other words, note lengths are not measured with seconds but with note length fractions like 1/2, 1/4, 1/32, etc. Pitches with lower total percentages than %5 were omitted for avoiding the cluttering of results' readability. Finally, the density values were calculated by normalizing the whole sets into [0, 1] intervals.

The pitch density graph of Hicaz Makam is shown in Figure 7.5. It can be seen from the graph that the most emphasized peaks in the Hicaz set are near re5 and la4 pitches, which are the tonic and the dominant of the Hicaz Makam.

The similarity of the pitch densities of SymbTr and ATMMC-compositions sets is calculated as shown in Equation (7.1).

$$Similarity(S, A) = \frac{2}{N} \times \sum_{i=1}^{N} \frac{\min(S_i, A_i)}{S_i + A_i} \tag{7.1}$$

where S and A are pitch histograms of SymbTr, and ATMMC-compositions sets representing pitch densities, and S_i and A_i represent each bin in corresponding histograms. By applying Equation (7.1) to pitch density distribution shown in Figure 7.5, ATMMC-compositions for Hicaz Makam is found to be 67% similar to SymbTr in terms of pitch densities.

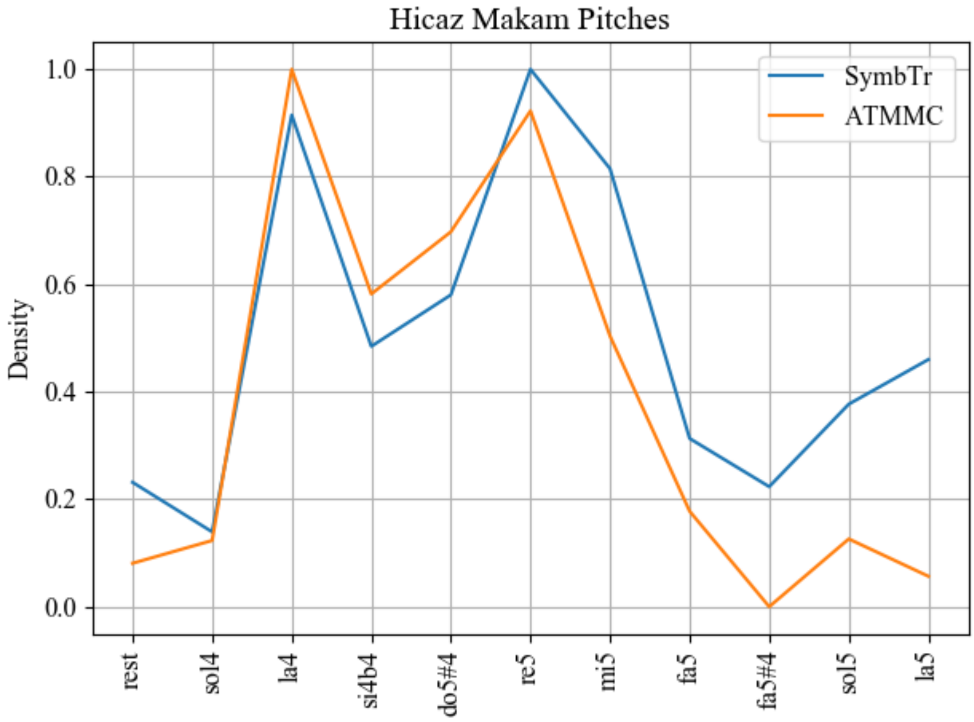


Figure 7.5 Hicaz Makam pitch density graph.

A similar result is shown in Figure 7.6 for Nihâvent Makam. It can be seen that the peaks are located around re5, si4b5, and sol4, which are the tonic and dominant pitches of Nihâvent Makam. Again, it can be seen from the graph that both SymbTr and ATMMC pitch densities are distributed coherently. Applying Equation (7.1), ATMMC-compositions for Nihâvent Makam is found to be 73% similar to SymbTr in terms of pitch densities. From Figure 7.6, it can be deduced that Both SymbTr and ATMMC sets represent Nihâvent Makam in terms of pitch densities very well and ATMMC is efficient in Nihâvent compositions in terms of adequate pitch density distribution.

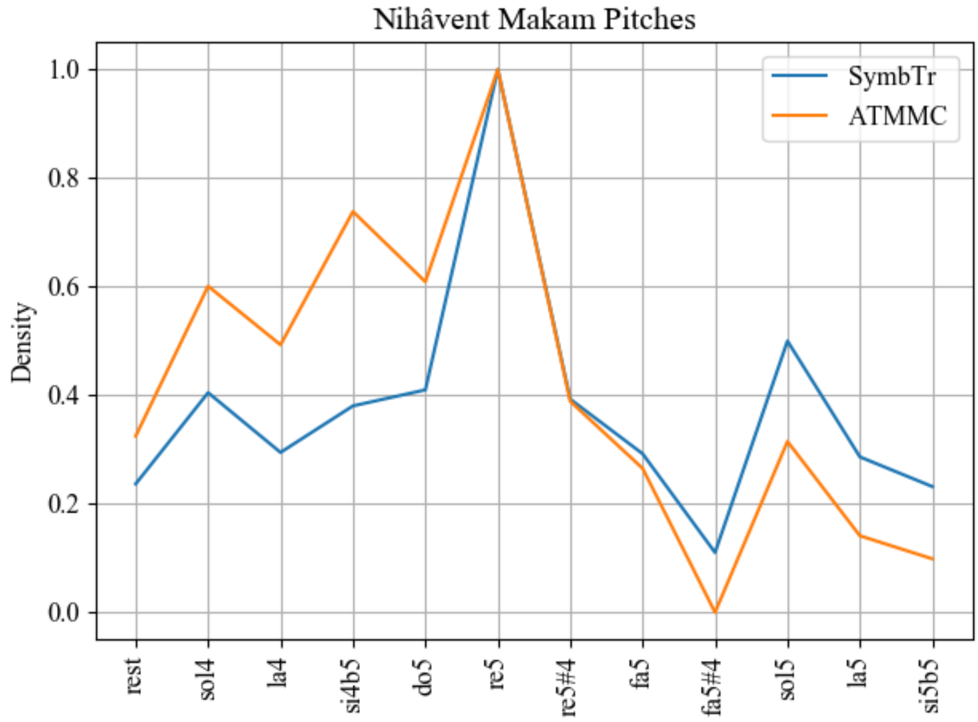


Figure 7.6 Nihâvent Makam pitch density graph.

7.2.2 ATMMC Detailed Analysis Results

For assessing ATMMC's capabilities of reflecting its training set several metrics and their absolute and relative measures are calculated as suggested by Yang and Lerch (Yang & Lerch, 2020). For assessment, 20 pieces for Hicaz Makam and 20 pieces for Nihâvent Makam were artificially composed by ATMMC. Then the composed pieces were compared with the pieces in SymbTr. Comparisons were performed only within Şarkı form and Aksak Usûl for Hicaz Makam and Düyek Usûl for Nihâvent Makam. Two example ATMMC composition samples are shown in Figure 7.7 and Figure 7.8.

Figure 7.7 An example Hicaz composition by ATMMC.

Figure 7.8 An example Nihâvent composition by ATMMC.

Both absolute and relative metrics regarding 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 given in Table 7.2, Table 7.3, and Table 7.4.

The metrics given in Table 7.2 and Table 7.3 show that ATMMC has similar average means both for Nihâvent and Hicaz Makams with SymbTr dataset. When average means are compared, $100 \times (1 - (5.26 - 4.47) / 5.26) \approx 84.98\%$ average similarity for Nihâvent Makam and $100 \times (1 - (8.64 - 7.24) / 8.64) \approx 83.79\%$ average similarity for Hicaz Makam to SymbTr dataset are evaluated. It can also be seen from Table 7.2 and Table 7.3 that Hicaz and Nihâvent sets generated by ATMMC have less diversity in terms of pitch and duration variations. Reduction in diversity for measured features is an expected outcome since ANN models generally exclude outliers and learn the general structure of relations over the set they are trained.

Table 7.2. Absolute metrics for Nihâvent Makam.

	SymbTr		**ATMMC**	
	Mean	**STD**	**Mean**	**STD**
PC	14.05	1.62	11.60	1.20
PC/Bar	4.18	1.25	4.76	1.26
NC	6.60	1.65	4.40	1.01
NC/Bar	2.69	0.75	2.09	0.70
PCH	0.05	0.07	0.05	0.07
PCH/Bar	0.05	0.13	0.05	0.11
NLH	0.07	0.13	0.07	0.17
PCTM	0.00	0.01	0.00	0.01
PR	31.85	2.97	27.35	2.57
PI	3.59	0.47	3.34	0.30
IOI	0.08	0.03	0.04	0.01
NLTM	0.00	0.02	0.00	0.04
Average	5.26	0.75	4.47	0.62

Table 7.3. Absolute metrics for Hicaz Makam.

	SymbTr		**ATMMC**	
	Mean	**STD**	**Mean**	**STD**
PC	14.60	2.95	10.85	1.38
PC/Bar	5.35	1.45	5.29	1.48
NC	6.45	1.82	5.05	1.11

Table 7.3 continues

NC/Bar	3.12	1.03	2.35	0.85
PCH	0.04	0.06	0.04	0.07
PCH/Bar	0.04	0.09	0.04	0.09
NLH	0.06	0.14	0.06	0.15
PCTM	0.00	0.01	0.00	0.01
PR	67.00	8.98	56.50	6.61
PI	7.00	0.52	6.77	0.44
IOI	0.05	0.01	0.03	0.01
NLTM	0.00	0.03	0.00	0.03
Average	8.64	1.42	7.24	1.01

KLD and OA metrics between intra-SymbTr set and inter-ATMMC set for Nihâvent and Hicaz Makams are given in Table 7.4. Both Nihavent and Hicaz sets of ATMMC have the largest OA values for pitch count per bar (PC/Bar) feature. This result can be interpreted as the best feature of ATMMC is its ability in producing diverse pitches per bar.

It can be seen from Table 7.4 that, on average, ATMMC's Hicaz composer slightly outperforms the Nihavent composer both by larger OA and smaller KLD values. This outcome might be related to the larger size of the Hicaz set in SymbTr compared to the Nihâvent collection of SymbTr.

Table 7.4 Relative metrics for Hicaz and Nihâvent Makams.

	Hicaz		**Nihâvent**	
	KLD	**OA**	**KLD**	**OA**
PC	0.243	0.533	0.043	0.604
PC/Bar	0.037	**0.898**	0.029	**0.886**
NC	0.045	0.701	0.115	0.566
NC/Bar	0.065	0.640	**0.010**	0.618
PCH	0.056	0.638	0.250	0.811
PCH/Bar	0.051	0.511	0.165	0.595
NLH	**0.012**	0.814	0.165	0.603

Table 7.4 continues

PCTM	0.194	0.573	0.114	0.630
PR	0.154	0.669	0.249	0.551
PI	0.023	0.855	0.102	0.820
IOI	0.166	0.358	0.340	0.548
NLTM	0.025	0.732	0.153	0.541
Average	0.089	0.660	0.144	0.647

For further demonstration of ATMMC's effectiveness, PDFs (Probability Distribution Functions) of various features of Hicaz and Nihavent Makam for both inter and intra sets are given in Figures 7.9 thru 7.14. Intra-SymbTr sets are represented with blue lines, whereas intra-ATMMC sets are represented with green lines. Distances between SymbTr and ATMMC sets' intra-distances are represented by the orange lines (inter sets).

It can be seen from Figure 7.9 that inter-set distributions of pitch count (PC) feature for both Hicaz and Nihâvent Makams align with intra-SymbTr and intra-ATMMC sets on the x-axis. As demonstrated in Section 4.2.4, this alignment shows that the two datasets are harmonious.

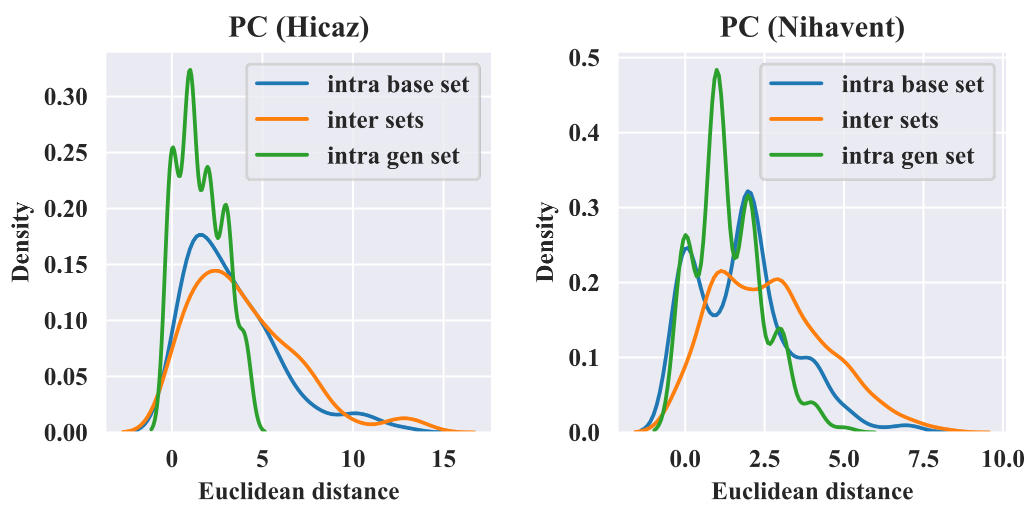


Figure 7.9 PDFs of pitch count (PC) feature.

It can be seen from Figure 7.10 and 7.11 that inter-set distributions of note count (NC) and pitch class histogram (PCH) features for both Hicaz and Nihâvent Makams align with intra-SymbTr and intra-ATMMC sets on the x-axis. Again, this alignment shows that the two datasets are coherent with each other.

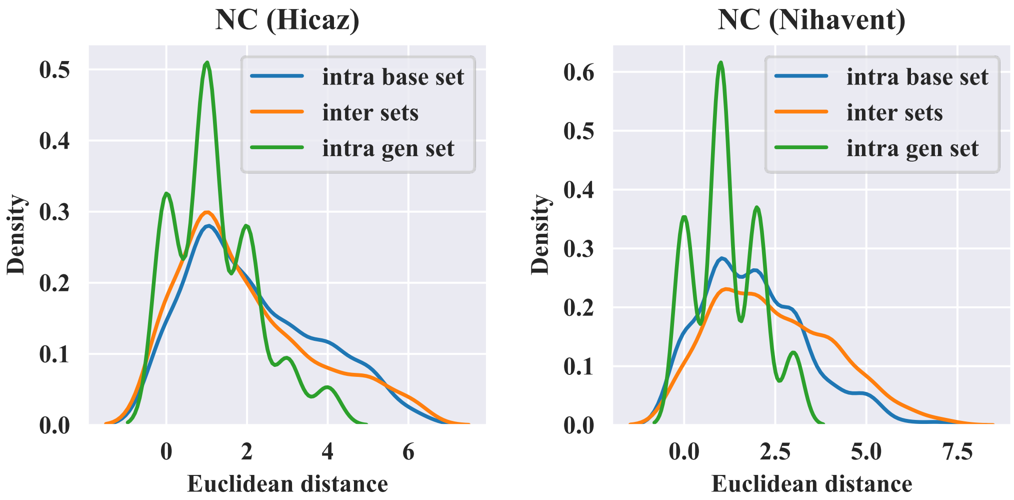


Figure 7.10 PDFs of note count (NC) feature.

Especially in Figure 7.11 Nihâvent part, it is visible that the three distributions align both in the x-axis and y-axis very well. This well-established alignment is also supported by the data from Table 7.4.

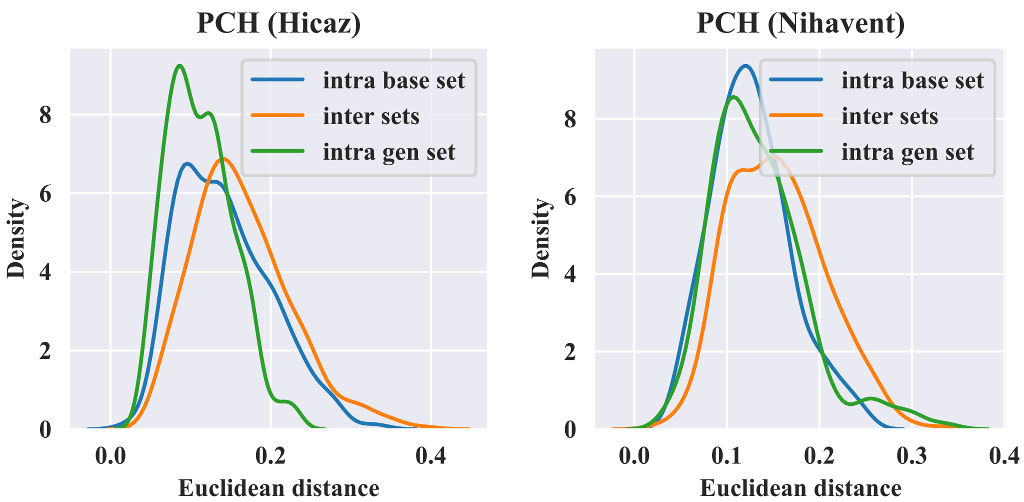


Figure 7.11 PDFs of pitch count histogram (PCH) feature.

Again, in Figure 7.12 it is obviously visible that the three distributions align both in the x-axis and y-axis very well, especially for Hicaz Makam. This alignment can be interpreted as ATMMC's ability to mimic note lengths (NLH) in SymbTr is sufficient.

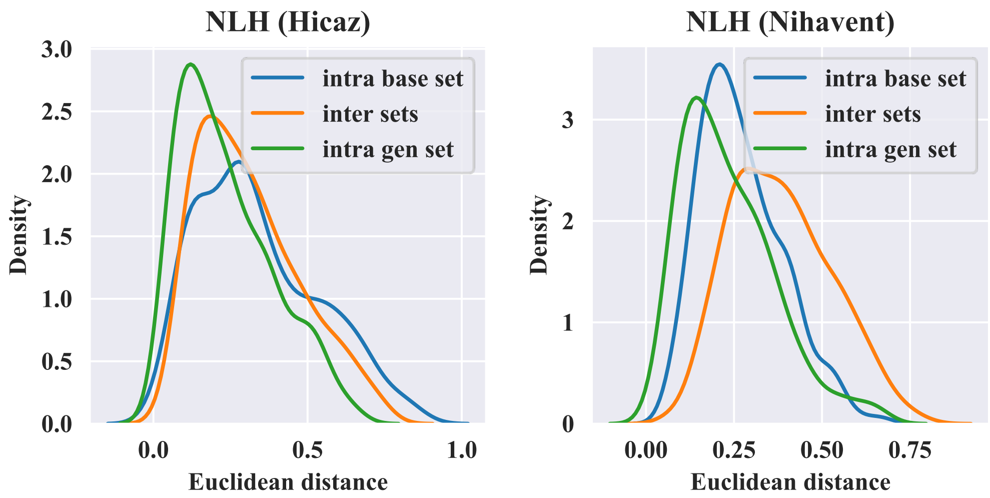


Figure 7.12 PDFs of note length histogram (NLH) feature.

However, in Figure 7.13 pitch range (PR) features are not as well-aligned as other features. This means ATMMC squeezes the highest and lowest notes into a smaller interval than the samples in the SymbTr dataset. Since the highest and the lowest pitches may be considered as outliers, this result can be interpreted as ATMMC's lacking in representing the source dataset's outliers.

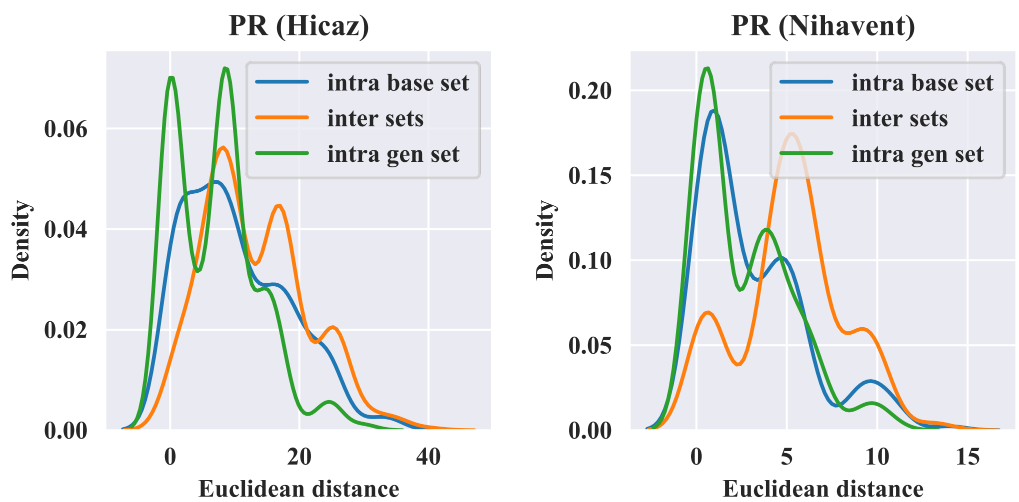


Figure 7.13 PDFs of pitch range (PR) feature.

Finally, it can be seen from Figure 7.14 that the distributions for average pitch interval (PI) are aligning very well for both Makams. As also supported by the data in Table 7.4, this alignment can be interpreted as ATMMC's ability to manage the distance of jumps between consecutive pitches very well.

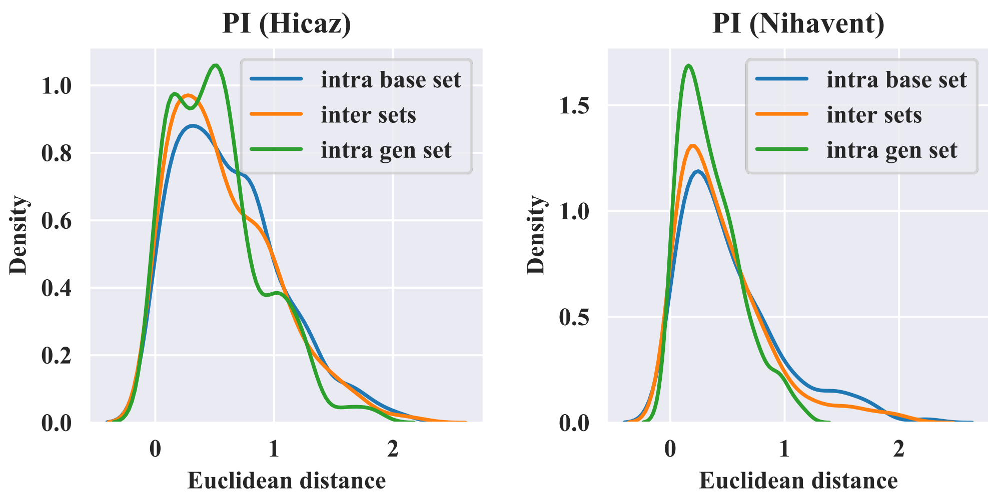


Figure 7.14 PDFs of average pitch interval (PI) feature.

CHAPTER EIGHT
CONCLUSION AND FUTURE WORK

8.1 Conclusion

Deep Learning (DL) based music composition has become a popular field of research due to the availability of large digital music datasets, state-of-the-art DL technologies, and high-power computer systems. Although the majority of DL-based music research is carried on Western Music, there are also few studies on Turkish Music. However, the majority of these DL-based Turkish Music research is in the field of music information retrieval (MIR), and the number of studies on automatic music composition is negligible.

In the scope of this thesis, a DL-based symbolic music generation system, Artificial Turkish Makam Music Composer (ATMMC) was developed by training Long Short-Term Memory (LSTM) networks on SymbTr, which is the most comprehensive open-source dataset, compiled for computational Turkish Music research.

As the first of its kind and novel system, ATMMC can create artificial compositions in Hicaz and Nihâvent Makams in Şarkı Forms. Also, for the system to be used by users without programming knowledge, a graphical user interface that can be accessed via the internet browser at http://music.cs.deu.edu.tr/tmmgui/ has been developed. Via the graphical user interface, users can enter 8 initial notes and command ATMMC to complete their compositions. When the automatic composition process ends, users can download the resulting composition to their computers in Mu2 format, to be viewed and played by Mus2 software later.

The effectiveness of the ATMMC system has been evaluated both subjectively through a survey and objectively with various metrics. As the result of subjective evaluations, the system was found to be in between above-average and good segments. Objective evaluations show that ATMMC is around 84% similar to the dataset on which it was trained. In addition, due to the availability of a slightly larger dataset, artificial compositions in Hicaz Makam are found to be slightly better than the artificial compositions in Nihavent Makam.

When subjective and objective evaluation results are addressed as a whole, it can be concluded that ATMMC can be used to assist human composers in creating new TMM pieces. It also can help composers to overcome writer's block by quickly offering composition drafts. Finally, ATMMC has been found to be useful in quickly creating copyright-free etudes for conservatory students.

8.2 Future Work

Automatic TMM composition is mostly an untouched research field and is widely open to experimentation. Plans for future studies can be listed as follows, but not limited to:

1. Training different DL models on TMM data and compare their results to find the most suitable one for TMM.
2. Widening the scope of ATMMC by training it on other Makams.
3. Creating a fully functional web application for TMM notation.
4. Creating a new open-source digital TMM dataset compiled from the saved pieces revealed by the use of the to-be-developed web application.
5. Optimizing the future ATMMC models into JavaScript for front-end web.
6. To provide suggestions to users through DL models in the web application to be developed.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2016). TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *Arxiv Preprint Arxiv*, 1603.04467.

Abidin, D., Öztürk, Ö., & Özacar Öztürk, T. (2017). Klasik Türk müziğinde makam tanıma için veri madenciliği kullanımı. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, *32*(4), 1221–1232. https://doi.org/10.17341/gazimmfd.369557

Altinköprü, H. (2018). Maqam modulations in traditional Turkish art music. *Eurasian Journal of Music and Dance*, *12*, 1–11.

Arel, H. S. (1968). *Türk musikisi nazariyatı dersleri*. İstanbul: Hüsnütabiat Matbaası.

Aydoğan, S., & Özgür, Ü. (2015). *Gelenekten geleceğe makamsal Türk müziği*. Ankara: Arkadaş Yayıncılık.

Barapatre, D., & A, V. (2017). Data preparation on large datasets for data science. *Asian Journal of Pharmaceutical and Clinical Research*, *10*(13), 485. https://doi.org/10.22159/ajpcr.2017.v10s1.20526

Barkçin, S. (2019). *40 Makam 40 anlam* (D. Yabul, Ed.; 1st ed.). İstanbul: Ketebe.

Bernstein, A., & Taylor, B. (2003). *TuneJS*. Retrieved September 16, 2020, from https://github.com/abbernie/tune

Bozkurt, B., Ayangil, R., & Holzapfel, A. (2014). Computational analysis of Turkish makam music: Review of state-of-the-art and challenges. *Journal of New Music Research*, *43*(1), 3–23. https://doi.org/10.1080/09298215.2013.865760

Bozkurt, B., Yarman, O., Karaosmanoğlu, M. K., & Akkoç, C. (2009). Weighing diverse theoretical models on Turkish maqam music against pitch measurements: A comparison of peaks automatically derived from frequency histograms with proposed scale tones. *Journal of New Music Research*, *38*(1), 45–70. https://doi.org/10.1080/09298210903147673

Brain Team, G. (2020). *Magenta*. Retrieved January 12, 2019, from https://magenta.tensorflow.org

Briot, J.-P., Hadjeres, G., & Pachet, F. (2017). Deep learning techniques for music generation - a survey. *ArXiv Preprint ArXiv: 1709.01620.*

Brownlee, J. (2019). *How to calculate the KL divergence for machine learning*. Retrieved February 8, 2021, from https://machinelearningmastery.com/divergence-between-probability-distributions/

Burkholder, J. P., Grout, D. J., & Palisca, C. (2010). *A history of Western music* (10th ed.). New York City: W. W. Norton.

Choi, K., Fazekas, G., & Sandler, M. (2016). Text-based LSTM networks for automatic music composition. *ArXiv Preprint ArXiv:1604.05358*

Chollet, F. (2015). *Keras: The Python deep learning library*. Retrieved June 11, 2019, from https://keras.io/

Chu, H., Urtasun, R., & Fidler, S. (2016). Song from PI: A musically plausible network for pop music generation. *5th International Conference on Learning Representations, ICLR 2017 - Workshop Track Proceedings*, 1–9.

Cope, D. (1989). Experiments in musical intelligence (EMI): Non-linear linguistic-based composition. *Interface*, *18*(1–2). https://doi.org/10.1080/09298218908570541

Deng, L., & Yu, D. (2014). Deep learning: Methods and applications. *Foundations and Trends in Signal Processing* 7(3–4) 197-387.

Ederer, E. B. (2011). *The theory and praxis of makam in classical Turkish music 1910-2010*. Santa Barbara: University of California.

Erguner, S. (2007). Osman Dede, nayi. *TDV İslâm Ansiklopedisi*, 33, 461–462.

Gönül, M. (2015). Türk mûsikîsi usûllerinin gösterimi, ifadesi ve tasnifine bir bakış. *İSTEM*, *13*(25), 31–46.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. In *MIT Press*. Cambridge: MIT Press.

Güvençoğlu, Ş., & Özgelen, O. Z. (2020). *Türk makam müzikçisinin seyir defteri - 1*. İstanbul: Pan Yayıncılık.

Hananoi, S., Muraoka, K., & Kiyoki, Y. (2016). A music composition system with time-series data for sound design in next-generation sonification environment. *2016 International Electronics Symposium (IES)*, 380–384.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Huang, C.-Z. A., Hawthorne, C., Roberts, A., Dinculescu, M., Wexler, J., Hong, L., & Howcroft, J. (2019). The Bach doodle: Approachable music composition with machine learning at scale. *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019*, 793–800. http://arxiv.org/abs/1907.06637

Kapadia, S. (2019). *Language models: N-gram. A step into statistical language*. Retrieved March 22, 2020, from https://towardsdatascience.com/introduction-to-language-models-n-gram-e323081503d9

Karaosmanoğlu, M. K. (2012). A Turkish makam music symbolic database for music information retrieval: Symbtr. *Proceedings of the 13th ISMIR Conference, Porto, Portugal*, *ISMIR*, 223–228. http://compmusic.upf.edu/system/files/static_files/KemalKaraosmanoglu_Ismir2012.pdf

Karaosmanoğlu, M. K. (2017). *Müzik aritmetiği ve ses sistemleri* (1st ed.). İstanbul: İTÜ Vakfı.

Kumar, H., & Ravindran, B. (2019). Polyphonic music composition with LSTM neural networks and reinforcement learning. *ArXiv Preprint ArXiv: 1902.01973.*

Li, S., Jang, S., & Sung, Y. (2019). Automatic melody composition using enhanced GAN. *Mathematics*, *7*(10), 883. https://doi.org/10.3390/math7100883

Liang, F., Gotham, M., Johnson, M., & Shotton, J. (2017). Automatic stylistic composition of Bach chorales with deep LSTM. *Ismir 2017*.

Luque, S. (2009). The stochastic Synthesis of Iannis Xenakis. *Leonardo Music Journal*, *19*, 77–84.

Marinescu, A.-I. (2019). Bach 2.0 - generating classical music using recurrent neural networks. *Procedia Computer Science*, *159*, 117–124. https://doi.org/10.1016/j.procs.2019.09.166

Olah, C. (2015). *Understanding LSTM networks*. Retrieved March 11, 2021, from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Oord, A. van den, Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. (2016). WaveNet: A generative model for raw audio. *ArXiv*, *21*(3), 793–830. https://doi.org/10.1162/neco.2008.04-08-771

Özkan, İ. H. (2006). *Türk musikisi nazariyatı ve usulleri kudüm velveleleri* (18th ed.). İstanbul: Ötüken Neşriyat.

Parlak, İ. H. (2020). *TMMDL*. Retrieved April 5, 2021, from https://github.com/ihpar/TMMDLFT

Parlak, İ. H., & Kösemen, C. (2018). Automatic music generation by true random numbers for Turkish makams. *2018 4th International Conference on Computer and Technology Applications (ICCTA)*, 64–68. https://doi.org/10.1109/CATA.2018.8398657

Parlak, İ. H., Çebi, Y., & Işıkhan, C. (2021). A Graphical User Interface for Deep Learning-Based Automatic Turkish Makam Music Composer. *2021 11th International Hisarli Ahmet Symposium*, 125–126.

Parlak, İ. H., Çebi, Y., Işıkhan, C., & Birant, D. (2021). Deep learning for Turkish makam music composition. *Turkish Journal of Electrical Engineering & Computer Sciences*. https://doi.org/10.3906/elk-2101-44

Pastore, M. (2018). Overlapping: A R package for estimating overlapping in empirical distributions. *Journal of Open Source Software*, *3*(32). https://doi.org/10.21105/joss.01023

Rojas, R. (1996). The backpropagation algorithm. In *Neural Networks* (149–182). Berlin: Springer.

Sandred, Ö., Laurson, M., & Kuuskankare, M. (2009). Revisiting the Illiac suite - a rule-based approach to stochastic processes. *Sonic Ideas/Ideas Sonicas*, *2*, 42–46. https://www.researchgate.net/publication/260791942

Şenocak, E. (2012). *Tarihi süreç içinde Türk müziğinde şarki formu*. Retrieved April 4, 2020, from http://earsiv.halic.edu.tr/xmlui/bitstream/handle/20.500.12473/1658/342441.pdf?sequence=1&isAllowed=y

Şentürk, S. (2017). *SymbTr*. Retrieved February 7, 2018, from https://github.com/MTG/SymbTr

Şentürk, S., & Chordia, P. (2011). Modeling melodic improvisation in Turkish folk music using variable-length Markov models. *12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, 269–274.

Şentürk, S., & Serra, X. (2016). Composition identification in Ottoman-Turkish makam music using transposition-invariant partial audio-score alignment. *13th Sound and Music Computing Conference (SMC 2016)*, 434–441.

Shin, A., Crestel, L., Kato, H., Saito, K., Ohnishi, K., Yamaguchi, M., Nakawaki, M., Ushiku, Y., & Harada, T. (2017). Melody generation for pop music via word representation of musical properties. *ArXiv*, 1–9. http://arxiv.org/abs/1710.11549

Szandała, T. (2020). Review and comparison of commonly used activation functions for deep neural networks. *In Bio-inspired Neurocomputing* (203-224). Singapore: Springer.

Tıraşçı, M. (2019). *Türk musikisi nazariyatı tarihi*. İstanbul: Kayıhan.

Tüfekçi, A. (2014). *Exploring Ney Techniques* (1st ed.). İstanbul: Pan Yayıncılık.

Tüfekçioğlu, S. (2019). Osmanlıda yenilikçi hareketlerle birlikte Türk mûsikîsine eklenen yeni türler ve bu türlerin teori-pratik ikilemi. *Güzel Sanatlar Enstitüsü Dergisi*, 173–182. https://doi.org/10.32547/ataunigsed.523556

Turing, A. (1950). Computing machinery and intelligence. *Mind*, *59*(236), 433–460. https://doi.org/10.1093/mind/LIX.236.433

Uyar, B., Atlı, H. S., Şentürk, S., Bozkurt, B., & Serra, X. (2014). A Corpus for computational research of Turkish makam music. *Proceedings of the 1st International Workshop on Digital Libraries for Musicology - DLfM '14*, 1–7. https://doi.org/10.1145/2660168.2660174

Uygun, M. N. (2008). Safiyüddin el-Urmevî. *TDV İslâm Ansiklopedisi* 35, 479–480.

Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, *53*(8), 5929–5955. https://doi.org/10.1007/s10462-020-09838-1

Vanderplas, J. (2007). *Density estimation — scikit-learn 0.24.2 documentation*. Retrieved February 21, 2021, from https://scikit-learn.org/stable/modules/density.html

Węglarczyk, S. (2018). Kernel density estimation and its application. *ITM Web of Conferences*, *23*, 00037. https://doi.org/10.1051/itmconf/20182300037

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, *3*(1), 9. https://doi.org/10.1186/s40537-016-0043-6

Wright, O., & Turabi, A. H. (2001). Klasik Türk mûsikîsi'nde Çârgâh: Tarih-teori çelişkisi. *M.Ü. İlahiyat Fakültesi Dergisi*, *21*(1977), 81–104.

Wu, J., Hu, C., Wang, Y., Hu, X., & Zhu, J. (2020). A hierarchical recurrent neural network for symbolic melody generation. *IEEE Transactions on Cybernetics*, *50*(6), 2749–2757. https://doi.org/10.1109/TCYB.2019.2953194

Xu, X. (2020). *LSTM networks for music generation*. 1–7. http://arxiv.org/abs/2006.09838

Yang, L.-C., & Lerch, A. (2020). On the evaluation of generative models in music. *Neural Computing and Applications*, *32*(9), 4773–4784. https://doi.org/10.1007/s00521-018-3849-7

Yarman, O. (2007). A comparative evaluation of pitch notations in Turkish makam music: Abjad scale & 24-tone Pythagorean tuning – 53 equal division of the octave as a common grid. *Journal of Interdisciplinary Music Studies*, *1*(1), 51–62. https://doi.org/10.13140/RG.2.2.14971.72483

Yarman, O. (2008). *79-tone tuning & theory for Turkish maqam music as a solution to the non-conformance between current model and practice.* Doktora Tezi, İstanbul Teknik Üniversitesi, İstanbul.

Yarman, O. (2010). Türk makam müziğini bilgisayarda temsil etmeye yönelik başlıca yazılımlar. *Müzikte Temsil Müziksel Temsil Sempozyumu II*, 320–327. https://doi.org/10.13140/RG.2.2.24566.19526

Zarate, J. M., Ritson, C. R., & Poeppel, D. (2012). Pitch-interval discrimination and musical expertise: Is the semitone a perceptual boundary? *The Journal of the Acoustical Society of America*, *132*(2), 984–993. https://doi.org/10.1121/1.4733535

Zhang, Y., Liu, W., Chen, Z., Li, K., & Wang, J. (2021). On the properties of Kullback-Leibler divergence between gaussians. http://arxiv.org/abs/2102.05485