Design of Formal Grammar and Model for Composition of Indian Music

A Thesis

Submitted in fulfillment of the requirement
for the award of the degree of

DOCTOR OF PHILOSOPHY

IN

COMPUTER SCIENCE AND ENGINEERING

by
Bhavya
(Registration No. 951603010)

Under the guidance of

Dr. Ajay Kumar
(Professor, CSED)

Dr. Sunita Garhwal
(Associate Professor, CSED)

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
PATIALA – 147004

March 2024

DEDICATED WITH EXTREME AFFECTION AND GRATITUDE TO

My *parents, husband,*

***son* Inay Deswal**

and

My *supervisors,* Dr. Ajay Kumar and Dr. Sunita Garhwal.

CERTIFICATE

I, Bhavya, Regn no 951603010, hereby declare that the thesis entitled "Design of Formal Grammar and Model for Composition of Indian Music" submitted to the Computer Science and Engineering Department at Thapar Institute of Engineering and Technology, Patiala, Punjab, India is an authenticated record of my own work for the award of the degree of "Doctor of Philosophy" under the supervision of Prof. (Dr.) Ajay Kumar and Dr. Sunita Garhwal. This report has not been submitted to any other institution for the award of my other degree.

Bhavya

Bhavya

Regn No. 951603010

Place: Patiala, Punjab (India)

Date: March 15, 2024

This is to certify that the above made statement made by candidate is correct to the best of our knowledge.

Verified by:

A Kumar

Prof. (Dr.) Ajay Kumar

(Supervisor)

Professor, CSED

Thapar Institute of Engg. &Technology

Patiala, Punjab

Sunita

Associate Prof. (Dr.) Sunita Garhwal

(Supervisor)

Associate Professor, CSED

Thapar Institute of Engg. & Technology

Patiala, Punjab

ACKNOWLEDGEMENT

Research is like an ocean; one needs to be guided and supported by several individuals to drive out a handful of pearls from its depth.

I feel privileged to express my profound gratitude and deep regards to my respected Supervisors, *Dr. Ajay Kumar (Professor) and Dr. Sunita Garhwal (Associate professor), Computer Science and Engineering Department (CSED), Thapar Institute of Engineering and Technology, Patiala,* for their erudite guidance, unwavering support, and encouragement throughout the thesis. I feel fortunate to get a lifetime opportunity to work under their considerate and knowledgeable supervision.

I am extremely indebted to the Doctoral committee comprising Dr. Shalini Batra (Head, CSED), Dr. Neeraj Kumar (Professor, CSED), Dr. Ashutosh Mishra (Assistant Professor, CSED), and Dr. Rajesh Khanna (Professor, EIED) for monitoring the progress and providing valuable suggestions for the improvement of this Ph.D. work.

I would also like to take this opportunity to mention my highest regards and love toward Mrs. Sunita Garhwal and her adorable daughters, Dishita and Anaya, who were my constant companions in the personal and professional spheres of my campus life.

I feel immense pleasure to express my unfathomable respect for my parents, Dr. Shamsher Singh Mor and Mrs. Seema Mor, and parents-in- law, Mr. Jaipal Singh and Mrs. Rajesh, for their good wishes, affection, remitting encouragement as well as perpetual inspiration during the entire course. My brother, Mr. Dhruv Mor, has also loved, supported, and encouraged me to do this work. Being my elder brother, he always stood beside me at every point of my life. My heart well up with compassion for my sister-in-law, Mrs. Geetanjali Mor, for her cheerful support. My beloved husband, Mr. Amandeep Deswal, has been invigorating source for my course endeavors. His paramount love and unflinching support paved the way for me to achieve the goals in my work’s final stage. I would also like to acknowledge the contribution and support of my brother-in-law, Mr. Arun Deswal. My love and appreciation for my son Inay Singh Deswal and my niece Ihita Mor, whose smiles were pillars of joy and encouragement. My respected parents and supervisor’s continuous support and motivation ultimately made it possible for me to accomplish this task.

Sincere thanks to my friends, especially, Natasha Singh, Nidhi Kalra, Deepali Bhagat, Sukhandeep Kaur, and Simarpreet Singh, for their kindness and moral support during my study. Thanks for the friendship and memories.

I duly acknowledge the timely and fruitful assistance, moral support, and kind cooperation of all the members of the CSED Department, faculty members, and staff members who were always the need of the hour and provided with the help and facilities I required to complete my thesis.

Above all, I thank to the almighty God for the light he has kindled in me, which has led the path I am treading on.

Bhavya

(Bhavya)

ABSTRACT

Music is considered a universal language that persists everywhere. Indian music is very renowned due to its sophistication and rhythmic diversities. Music composition and mathematical computation are closely related and have similar aesthetics. In the last few years, musicians and researchers have used a computer to design compositional models for musical patterns. Computer-assisted music composition has been an active research area since mid-1900s, and the potential of computer systems in contributing music-related research had been imagined as a future possibility in 1950. Computational musicology is an interdisciplinary area that includes the contribution of music and computer science methods. This thesis explores the intricate world of Indian music composition through formal grammar and modeling techniques. The primary objective is to create a framework that captures the essence of Indian music composition while providing a platform for innovation and automation in music creation.

This thesis proposed a technique for generating musical sequence to help musicians as well as non-musicians compose musical structures. The work aims to generate formal grammar for an Indian musical composition. A musical tool is used to generate a musical sheet for the composition used. We outline musical rules and formal grammar rules used for music composition. The probabilistic context-free grammar is modeled to generate the same progression as the composition's.

Variable Order and Gapped hidden Markov model for unstructured elements can capture variable length dependencies with variable gaps in sequential data. VOGUE uses a Variable-Gap Sequence miner algorithm to extract frequent patterns in a sequence with variable gaps. In this thesis, we applied the VOGUE model to design the musical sequence of notes in bandish of raga Bhairav, a classical Indian music. Furthermore, we analyzed the benefits of VOGUE model over the standard HMM.

Tala of Indian music represents rhythmic aspects in musical compositions. This thesis used formal grammar to examine the patterns in Tala's kaidas and paltas. Results indicate that the patterns in the paltas of Tala's kaida exhibit cross-serial dependencies, which context-free grammar cannot represent. Further, we represented these patterns using the deep pushdown automata mathematical model. The proposed technique can be applied to check the correctness of the patterns used in Tala's Kaidas and Paltas.

The basic proposal of the thesis is to design formal grammar and model for composition of Indian music. Limited work has been done by researchers on the modeling of Indian music using automaton. In this research, we have modeled Indian Tala using deep pushdown automata. Further, formal grammar has been generated for Tabla bols by researchers, but no formal grammar for Indian music composition has been designed previously.

Keywords: Raga, Tala, Musical sheet, State grammar, Deep pushdown Automata, Indian music, Musical sequence, Hidden Markov model, VOGUE, Bhairav Raga, Sequence mining, Kaida, Palta, Raga Classification, Audio Feature Extraction.

PREFACE

This thesis presents my research at the Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, India, from July 2016 to February 2024. This manuscript mainly focuses on designing and development of formal grammar and models for composing Indian music. The work presented in this manuscript is as follows:

1. Survey of various computational musicologies.
2. A novel concept of generating formal grammar for Indian musical composition.
3. Study of various hidden Markov models and their applications.
4. Modeled Indian music using a variable order gapped HMM.
5. India Tala's kaida and palta using formal grammar.

The list of the main research papers accepted and communicated that I have authored or co-authored during my Ph.D. thesis are as follows:

1. **M. Bhavya, S. Garhwal and A. Kumar**, A systematic review of hidden Markov models and their applications, Archives of Computational Methods in Engineering, 28, 1429-1448, 2020. [SCIE Indexed, Impact Factor-9.7]
2. **M. Bhavya, S. Garhwal and A. Kumar**, A systematic literature review on computational musicology, Archives of Computational Methods in Engineering, 27, 923-937, 2020. [SCIE Indexed, Impact Factor-9.7]
3. **M. Bhavya, S. Garhwal and A. Kumar**, MIMVOGUE: modeling Indian music using a variable order gapped HMM, Multimedia Tools and Applications, 80, 14853-14866, 2021. [SCIE Indexed, Impact Factor-3.6]
4. **M. Bhavya, S. Garhwal and A. Kumar**, Mathematical modeling of Indian Tala's Kaidas and Paltas using formal grammar, Journal of Ambient Intelligence and Humanized Computing, 12, 7891-7902, 2021. [SCIE Indexed, Impact Factor-3.662]

TABLE OF CONTENTS

CHAPTERS	TITLE	PAGE NO.
Certificate		
Acknowledgement		
Abstract		
Preface		
Table of Contents		
List of Figures		
List of Tables		
CHAPTER 1:	**INTRODUCTION**	**1-11**
1.1	*Motivation*	1-5
1.2	*Preliminaries Concept*	5-10
1.3	*Organization of the Thesis*	10-11
CHAPTER 2:	**LITERATURE SURVEY**	**12-27**
2.1	*Computational Musicology Methods*	12-13
2.2	*Formal Grammar for Music*	13-14
2.3	*HMM for Musical Composition*	15-18
2.4	*N-Gram for Musical Patterns*	18-21
2.5	*FSM and FST with Music*	22-23
2.6	*AGL for Music*	23-27
2.7	*Gaps in Literature*	27-28
2.8	*Problem Formulation*	28
2.8.1	*Problem Statement*	28
2.8.2	*Objectives*	28
CHAPTER 3:	**GENERATING FORMAL GRAMMAR FOR INDIAN MUSICAL COMPOSITION**	**29-33**

CHAPTERS	TITLE	PAGE NO.
3.1	*Tool Used for Generating Musical Sheet*	29-30
3.2	*Identifying Chords Used in Musical Composition*	30-31
3.3	*Grammar Generation*	30-33
3.3.1	*The First String of Song*	32
3.3.2	*The Second String of Song*	32
3.3.3	*The Third String of Song*	32
3.3.4	*The Fourth String of Song*	32-33
3.3.5	*The Fifth String of Song*	33
3.3.6	*The Sixth String of Song*	33
3.4	*Conclusion*	33
CHAPTER4:	**MIMVOGUE: MODELING INDIAN MUSIC USING A VARIABLE ORDER GAPPED HMM**	**34-43**
4.1	*Introduction*	34
4.2	*Preliminaries*	34-35
4.3	*VOGUE Model for Indian Music*	35-37
4.3.1	*Symbol Emission Probability*	37-38
4.3.2	*The Transition Probability Matrix*	38-41
4.3.3	*State Duration Probability*	42
4.3.4	The Initial State Probability	42-43
4.5	Conclusion	43
CHAPTER 5:	**MATHEMATICAL MODELING OF INDIAN TALA'S KAIDAS AND PALTAS USING FORMAL GRAMMAR**	**44-55**
CHAPTERS	**TITLE**	**PAGE NO.**

5.1	*Introduction*	43
5.2	*Preliminaries Concept*	44-45
5.3	*Representation of Cross-Serial Dependency in Palta*	45-48
5.4	*Representation of Indian Music Using State Grammar*	48-49
5.5	*Mathematical Modeling of Indian Music Palta's Using Deep pushdown Automata*	49-53
5.6	*Conclusion*	53
CHAPTER 6:	**CLASSIFICATION OF RAGA USING MACHINE LEARNING**	**54-63**
6.1	*Introduction*	54
6.2	*Feature Engineering and Data Pre-processing*	55
6.2.1	*Evaluating Feature Importance*	55
6.3	Proposed Model Building	56-57
6.4	*Experiments and Results*	57-60
6.4.1	*Feature Importance Visualization*	57-59
6.4.2	*Training Progress*	59
6.5	*Conclusion*	59-60
CHAPTER 7:	**CONCLUSION AND FUTURE SCOPE**	**61-62**
7.1	*Summary of the Main Contribution*	*61*
7.2	*Future Scope*	61-62
	REFERENCES	**63-80**

LIST OF FIGURES

FIGURE NO.	FIGURE TITLE	PAGE NO.
1.1:	Deep Pushdown Automaton for $a^n b^n a^n b^n$	6
1.2:	Name of notes and octaves	7
1.3:	The C major scale and the degree names	8
2.1:	Representation of various computational methods with Music operations	11
2.2:	HMM papers with Indian and Western music	16
2.3:	N-gram with various musical operations	22
3.1:	Musical sheet for National Anthem of India	29
3.2:	The chord progression in major scale	30
4.1:	VOGUE HMM for Indian music Raga Bhairav	37
5.1:	Example of matra, vibhag, and avartan	44
5.2:	Theka of Teentala	45
5.3:	Kaida 1 and its palta	46
5.4:	Kaida 2 and its palta	47
5.5:	Kaida 3 and its palta	47
5.6:	Cross-serial dependency in palta with 16 matras	47
5.7:	Kaida 4 and its palta	48
5.8:	Kaida 5 and its palta	48
5.9:	Language observed in improvised musical patterns	49
5.10:	Deep pushdown automaton for paltas of kaida 2	49

5.11:	Deep Pushdown Automaton for improvised musical patterns	51
6.1:	Feature importance of Dataset	58
6.2:	Proposed Model for Raga Classification	59
6.3:	Importance of various features	61
6.4:	Features used for raga identification	58
6.5:	Error for training data	59

LIST OF TABLES

TABLE NO.	TABLE TITLE	PAGE NO.
2.1:	Major findings in the field of Formal Grammar and music	13-14
2.2:	Music classification using HMM	18-19
2.3:	Summary of research done for learning music using AGL	25-26
3.1:	Degree Representation	30
4.1:	Frequency of symbols in musical sequence	36
4.2:	Subsequences of length 1 and 2 with gap information	36-37
4.3:	Symbol Emission Probability Matrix of VOGUE model for raga Bhairav	39
4.4:	Transition Probability matrix of VOGUE model for raga Bhairav	41
4.5:	State duration probabilities of VOGUE model for raga Bhairav	42
4.6:	Initial state probabilities of VOGUE model for raga Bhairav	43

CHAPTER 1

INTRODUCTION

This chapter aims to introduce the preliminary concepts, basic notations, motivation, and thesis organization.

1.1 MOTIVATION

Automata theory is a branch of theoretical computer science that deals with the study of abstract machines and the problems they can solve. It plays a crucial role in understanding the fundamental concepts behind computation and forms the theoretical foundation for various aspects of computer science and engineering. An automaton refers to abstract mathematical models of computation. These models are used to study the behaviour and capabilities of computational devices. Finite Automata (FA), Pushdown Automata (PDA), Deep Pushdown Automata (DPDA), and Linear Bounded Automata (LBA) are a few examples of the diverse range of models used in automata theory. Each model has its unique properties, computational power, and applications. The applications of automata theory continue to grow as new challenges and problems arise in various domains like computer science, linguistics, engineering, pattern recognition, and music generation, among others.

Music is an art that combines various sounds to generate a new sound that entertains people. Indian music has evolved through eras. Within our Indian cultural heritage society, Indian music has a crucial part. It plays a pivotal part in our lives by affecting our social, physical, emotional, and cognitive aspects [1]. Little advancements have been done in prior works of formal grammar of various types of Indian music. The specific challenges are the diversity of Indian music rules and the requirement of unique grammar modeling for each type of music composition. Therefore, the field of Indian music is quite untouched by researchers, and a few leads can be found in existing literature.

Perchy and Sarria [2] worked on J.S. Bach’s harmonic rules and designed a compositional music system for assisting musicians. They had considered tonality, measure, progression, and tempo as initializing parameters for composing music. Keller and Morrison [3] used probabilistic context-free grammar for the automatic generation of jazz melodies over chord progressions. They have used Impro-visor to improve the composition of jazz solo music. Quick [4] explored the concept of stochastic grammar in music. Further, he applied the stochastic context-free grammar concept to Kullita for composing music automatically. The context-free grammar (CFG) algorithm was extended to assist different musical CFGs along with another type of grammar called probabilistic temporal graph grammar (PTGG). Groves

[5] proposed a stochastic context-free grammar (SCFG) approach for reducing melodies. The SCFG was induced from a Treebank of solutions for the melody reduction process. The dataset from the generative theory of tonal music was used to train SCFG for melody reduction. Abdallah and Gold [6] compared various models of symbolic music using probabilistic grammar. They explained probabilistic grammar implementation in probabilistic models of symbolic music. Abdallah et al. [7] used computational linguistic approaches for reviewing music analysis using probabilistic grammar. Different probabilistic models of symbolic music were applied to various folk song collections. They explained the use of probabilistic inference and information theory in music. Kitani and Koike [8] proposed an online grammatical induction generator to improve musical expression. ImprovGenerator generates random improvised accompaniment for real-time performances using basic percussion musical patterns. Puente et al. [9] described an automatic method of music composition using grammatical evolution. Roads [10] discussed music representation using different grammars by various authors. The grammar proved to be an efficient way of representing musical structure. Khalifa et al. [11] proposed a music composition tool using a genetic algorithm and integrated it with formal grammar rules. Sidorov et al. [12] proposed an approach for analyzing musical structure. It presented the musical analysis as the smallest grammar problem. This approach automatically generates context-free grammar that represents musical structure. As elaborated above, various researchers have developed many tools and techniques for composing music. Musicians, as well as non-musicians, used different tools to generate musical sequences. Concepts such as formal grammar, musical tools, and sheets are aimed at generating grammar for the musical composition.

Motivated by the work of different authors, we proposed a method for generating formal grammar for Indian music. We used a musical tool, MuseScore, for generating the musical sheet of required musical composition. It has not been previously utilized for modeling and designing formal grammar. The generated grammar satisfies the musical rules and Stochastic Context Free Grammar (SCFG) rules. The SCFG is modeled to generate the same progression as the musical composition.

Wang and Dubnov [13] worked with variable Markov oracle HMM (VMO-HMM) for capturing long-term dependencies in musical sequences. They modeled the harmonic progression of jazz music with VMO-HMM. Kerr [14] modeled hierarchal HMM (HHMM) for mid-level musical patterns. Weiland et al. [15] extracted musical pitch structures from HHMM representing a musical pattern. Hoffman et al. [16] explored the Hierarchical Dirichlet Process HMM (HDP-HMM) application for generating data-driven music. The model trained multiple songs and generated the output of many hybrid inputs. Chen et al. [17] implemented duration-explicit HMM (DHMM). The model considered chord duration for recognition. Benetos and Weyde [18] used pitch-wise DHMM for the transcription of polyphonic music. They modeled the tone duration and temporal evolution. Nakano et al. [19] processed musical signals using Mondrian HMM. The model captured clusters from the hidden states. Nakamura et al. [20] proposed an autoregressive hidden semi-Markov model (HSMM) for the symbolic performance of polyphonic musical scores. Nakamura et al. [21] described an outer-product HMM to capture arbitrary skips and repetitions in polyphonic musical performances. Ozerov et al. [22] represented polyphonic music through factorial-scaled HMM (FS-HMM). The model was a generalization of the Gaussian scaled mixture model. Chordia et al. [23] modeled north Indian tabla sequences with variable length Markov model (VLMM) or variable length HMM (VLHMM). The model determined the next stroke from the audio file of tabla sequences. Senturk [24] performed computational modeling of Turkish folk music with VLMM.

The first-order HMM does not consider the sequential dependencies for longer ranges. For long-range dependencies, higher-order (n-order) HMMs have been proposed. However, due to complex calculations, n-order HMMs are challenging to build, resulting in low prediction accuracy and high state-space complexity [25-26]. The limitations of first-order and higher-order HMM generate the need for a new HMM variant. The proposed approach uses a Variable order and Gapped HMM known as VOGUE (Variable Order and Gapped HMM for Unstructured Elements) for modeling musical sequences.

Bel and Kippen [27] proposed a model for North Indian tabla drumming improvisation and its evaluation with the proficiency of handling complex structures. Bel and Kippen [28] introduced the concept of Bol Processor grammar to represent patterns of repetition and homomorphic transformations. They also proposed the Bol processor (BP2) for the rule-based music composition.

Bel [29] applied the concept of formal grammar in the improvisation of North Indian tabla Drum music and extended the model in terms of substitutions, meta-variables, and remote contexts. Kippen and Bel [30] described the usage of the computer system in the composition of Indian drum music and developed a newer version of the Bol processor. Jurish [31] described the musical structure using formal language theory. Mahmud [32] also modeled and generated the tabla composition based on grammar.

The works of Mahmud [32], Bel and Kippen [27-28], Bel [29], and Kippen and Bel [30] motivated us to design formal grammar and deep pushdown automata for recognizing Tala and its improvised musical patterns in Indian music.

Shinkuma et al. [33] used machine learning for mobile crowd-sensing in spatial information prediction. They applied feature selection techniques to extract input data's contribution from the prediction model. Shetty and Arhary [34] achieved a training accuracy of around 94% and a test accuracy of 75% using Artificial Neural Network on a small sample of 90 songs. Dandawate et al. [35] tried data mining and information extraction approaches to achieve a significant increase in accuracy compared to their predecessors.

Gupta et al. [36] worked on a comprehensive review of scalable machine learning algorithms for big data. Bidkar et al. [37] proposed a raga recognition methodology using pitch, mean, and centroid MFCC variants. The model was tested for ensemble KNN classification algorithms, and ensemble bagged tree with 95.83% and 96.32% accuracy. Joshi et al. [38] used KNN and SVM classifiers on the raga Yaman and Bhairavi datasets to classify and identify raga. Dodia et al. [39] identified Indian raga using chromogram feature of audio files. KNN and SVM were used to identify Bhimpalasi, and Yaman Raga achieved an accuracy of 92% and 91% respectively. Sharma and Bali [40] worked with Hindustani raga identification on the dataset of live performances of ragas. Their work showed an accuracy of 93.38% with the K-star algorithm.

Anand [41] identified raga using a convolution neural network and attained an accuracy of 96.7%. Thakur et al. [42] applied two-fold supervised machine learning to detect rumours on Twitter. They automate filtering by training multiple classification algorithms and use textual characteristics on the filtered data. Kumar et al. [43] proposed raga identification and classification using a non-linear SVM framework with an n-gram distribution of notes and pitch-class profile. They captured temporal information of raga notes by calculating an n-gram histogram. Sharma et al. [44] worked with soft computing for raga identification of Hindustani music. Some ragas feature like dirga swara, vadi swara, distance, and samay of raga to identify raga. John et al. [45] classified Indian classical musical ragas using deep learning. They used the pitch contour of ragas as a key feature to improve the accuracy. Kumar et al. [46] used machine learning techniques to anticipate heart and liver disease. They used a random forest classifier and logistics regression to predict these diseases. Madhusudhan and Chowdhary [47] classified ragas using long short-term memory based on recurrent neural network. They achieved an accuracy of 88.1% on the comp music carnatic dataset. Kumar et al. [48] evaluated machine learning algorithms on academic datasets by applying feature selection techniques. Singha et al. [49] proposed a convolution neural network for raga classification. They used spectrograms of audio note to identify Raga's note. Shah et al. [50] proposed an approach based on deep learning and signal processing. They attained an accuracy of 98.98% on the compMusic dataset. Dighe et al. [51] developed a method using random forest classifier based on swara histograms. They achieved an accuracy of 94% through the experiment. Dodia et al. [52] identified raga using chromogram feature. They attained 91% accuracy with SVM classifier and 92% accuracy with the K-NN classifier. Subramanian et al. [53] proposed a method for modeling role-based access control by applying a three-way formal concept analysis.

Raga identification has been a manual and experiential endeavour; the merging of machine learning and musicology offers a transformative approach to this age-old practice. Motivated by the work of various authors, we designed a model using advanced computer sciences to detect the raga of a given creative piece through an efficient ensemble model in machine learning.

1.2 PRELIMINARIES CONCEPT

1.2.1 Basic Preliminaries of Automata

1.2.1.1 Grammar [54] can be defined by (N, T, S, P) where,

- N is the set of non-terminal symbols,
- T is the set of terminal symbols,
- S is the start symbol, and $S \in N$,
- P is the set of production rules of the form $\alpha \to \beta$, where $\alpha, \beta \in (N \bigcup T)^*$.

A grammar is said to be context-free grammar [55] if its right side consists of only a single non-terminal symbol or the production rule follows the form $\alpha \to \beta$, where $\alpha \in N$,and $\beta \in (N \bigcup T)^*$

1.2.1.2 A grammar is stochastic context-free grammar (SCFG) [55] if for every production rule, $\alpha \in N$ and $\beta \in (N \cup T)^*$. Every production rule $\{A_i \to \beta_j\}$ is assigned with a probability p such that $0 \le p \le 1$ and $\forall_i \sum_j p(A_i \to B_j) = 1$.

1.2.1.3 A HMM [56] is defined by five elements (N, M, A, B, π) where,

- N is the number of states in the model

- M is the number of symbols observed in the alphabet.
- A is the set of transition probabilities defined by $A = \{a_{ij}\} = \mathrm{p}\{\mathrm{q}_{t+1} \mid q_t = i\}, 1 \le i, j \le N$, here q_t is the current state.
- B is the probability distribution of each state, given by $B = \{b_j(k)\} \ge p\{\alpha_t = v_k \mid q_t = i\}, 1 \le \mathrm{k} \le \mathrm{M}$, here v_k denotes the k^{th} observed symbol and α_t denotes the current parameter.
- π is an initial state probability distribution, $\pi = \{\pi_i\} = p\{q_1 = i\}, 1 \le i \le N$

1.2.1.4 A state grammar [57] is a quintuple $\mathrm{G}(V, Q, T, P, S)$, where

- V is a finite non-empty set of symbols. V is known as the total alphabet,
- Q is a finite non-empty set of states, and $V \cap Q = \phi$,
- T is a finite non-empty set of terminals, and $T \subset V$,
- $S \in V - T$ represents the start symbol and
- $P \subseteq (Q \times (V - T)) \times (Q \times V^+)$ is a finite relation.

Example 1.1: Consider a state grammar $G = (\{S, X, Y, a, b\}, \{P_0, P_1, P_2, P_3, P_4, P_5, P_6\}, \{a, b\}, P, S\}$ for the language $L = \{a^m b^n a^m b^n\}$, with production rules defined by

$(P_0, S) \to (P_0, XY)$ $(P_0, X) \to (P_1, aX)$ $(P_1, Y) \to (P_0, aY)$ $(P_0, X) \to (P_4, a)$

$(P_4, Y) \to (P_6, a)$ $(P_0, X) \to (P_2, bX)$ $(P_2, Y) \to (P_3, bY)$ $(P_3, X) \to (P_5, b)$

$(P_3, X) \to (P_2, bX)$ $(P_0, X) \to (P_5, a)$ $(P_5, Y) \to (P_6, b)$

Derivation for string $aabaab$

$(P_0, S) \Rightarrow (P_0, XY)$ $[(P_0, S) \to (P_0, XY)]$

$\Rightarrow (P_1, aXY)$ $[(P_0, X) \to (P_1, aX)]$

$\Rightarrow (P_0, aXaY)$ $[(P_1, Y) \to (P_0, aY)]$

$\Rightarrow (P_1, aaXaY)$ $[(P_0, X) \to (P_1, aX)]$

$\Rightarrow (P_0, aaXaaY)$ $[(P_1, Y) \to (P_0, aY)]$

$\Rightarrow (P_0, aabaaY)$ $[(P_0, X) \to (P_5, b)]$

$\Rightarrow (P_0, aabaab)$ $[(P_5, Y) \to (P_6, b)]$

1.2.1.5 A deep pushdown automaton (DPDA) [58] is septuple $(Q, \Sigma, \Gamma, R, s, S, F)$, where

- Q is a finite non-empty set of states,
- Σ is a finite non-empty set of input symbols,
- Γ is a finite set of pushdown symbols, $\Sigma \subseteq \Gamma$, $\Gamma - \Sigma$ contains a special symbol $\#$,
- $R \subseteq (I \times Q \times (\Gamma - (\Sigma \bigcup \{\#\})) \times Q \times (\Gamma - \{\#\}^{+})) \bigcup (\mathrm{I} \times \mathrm{Q} \times \{\#\} \times \mathrm{Q} \times (\Gamma - \{\#\})^{*}\{\#\})$ is a finite relation, where I denotes a set of a positive integer,
- $s \in Q$ is the start state,
- $S \in \Gamma$ is the start pushdown symbol, and
- $F \in Q$ is the set of final states.

Example 1.2: Consider a deep pushdown automaton $M = (\{s, q, p, t, u\}, \{a, b\}, \{A, S, \#\}, R, s, S, f)$ for generating language $L = \{a^m b^n a^m b^n\}$,and R can be defined by

$1sS \rightarrow qAA$ $1qA \rightarrow tbA$ $1qA \rightarrow paA$ $1qA \rightarrow bf$ $2pA \rightarrow qaA$ $2tA \rightarrow ubA$

$1qA \rightarrow f_1 a$ $1uA \rightarrow tbA$ $1 f_1 A \rightarrow fa$ $1uA \rightarrow fb$ $1 fA \rightarrow fb$

Derivation of the string $aabaab$

$(s, aabaab, S\#) \Rightarrow_e (q, aabaab, AA\#) \Rightarrow_e (p, aabaab, aAA\#) \Rightarrow_p (p, abaab, AA\#)$
$\Rightarrow_e (q, abaab,$

$AaA\#) \Rightarrow_e (p, abaab, aAaA\#) \Rightarrow_p (p, baab, AaA\#) \Rightarrow_e (q, baab, AaaA\#)$
$\Rightarrow_e (f, baab, baaA\#)$

$\Rightarrow_p (f, aab, aaA\#) \Rightarrow_p (f, ab, aA\#) \Rightarrow_p (f, b, A\#) \Rightarrow_e (f, b, b\#) \Rightarrow_p (f, \wedge, \#)$

Example 1.3: Construct a deep pushdown automaton for language $L = \{a^n b^n a^n b^n\}$, using a state grammar $G_1 = (\{S, X, Y, a, b\}, \{P_0, P_1, P_2, P_3, P_4, P_5\}, \{a, b\}, P, S)$ with production rules

$(P_0, S) \rightarrow (P_0, XY)$ $(P_0, X) \rightarrow (P_1, aX)$ $(P_1, X) \rightarrow (P_2, Xb)$ $(P_2, Y) \rightarrow (P_3, aY)$ $(P_3, Y) \rightarrow (P_0, Yb)$
$(P_1, X) \rightarrow (P_4, b)$ $(P_4, Y) \rightarrow (P_5, ab)$

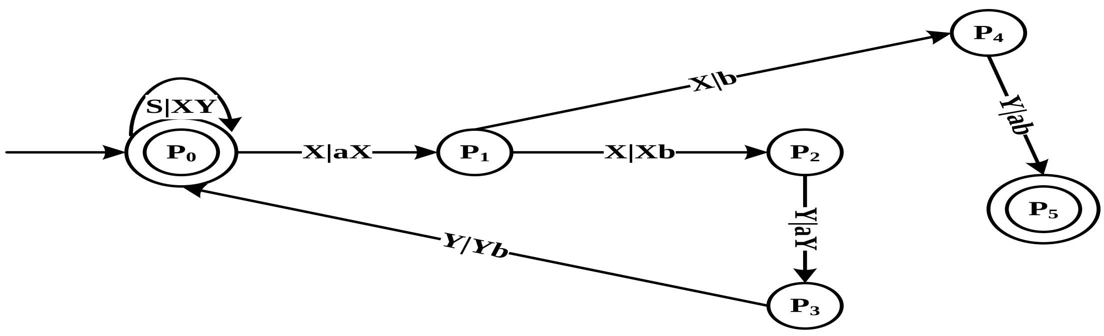

Figure 1.1: Deep Pushdown Automaton for $a^n b^n a^n b^n$

Derivation for string $abab$

$$(P_0, S) \Rightarrow (P_0, XY)\ [(P_0, S) \rightarrow (P_0, XY)]$$

$$\Rightarrow (P_1, aX)\ [(P_0, X) \rightarrow (P_1, aX)]$$

$$\Rightarrow (P_4, abY)\ [(P_1, \mathrm{X}) \rightarrow (P_4, \mathrm{b})]$$

$$\Rightarrow (P_5, abab)\ [(P_4, \mathrm{Y}) \rightarrow (P_5, ab)]$$

1.2.2 Basic preliminaries of music

1.2.2.1 Musical notes

In music, there are a total of seven notes (or swara). Notes are the basic building block of written musical notation. In Indian notation, the notations of notes are S, R, G, M, P, D, and N. All these notes are natural notes (shuddha swara). The natural notes are represented by the white keys, as shown in Figure 1.1. In Western music, notes are A, B, C, D, E, F, and G.

The octave (or saptak) is defined by the set of seven notes. There are three types of octaves, namely lower, middle, and upper. The lower octave notes have a half frequency as compared to middle octave notes, and middle octave notes have a half frequency as higher octave notes [59].

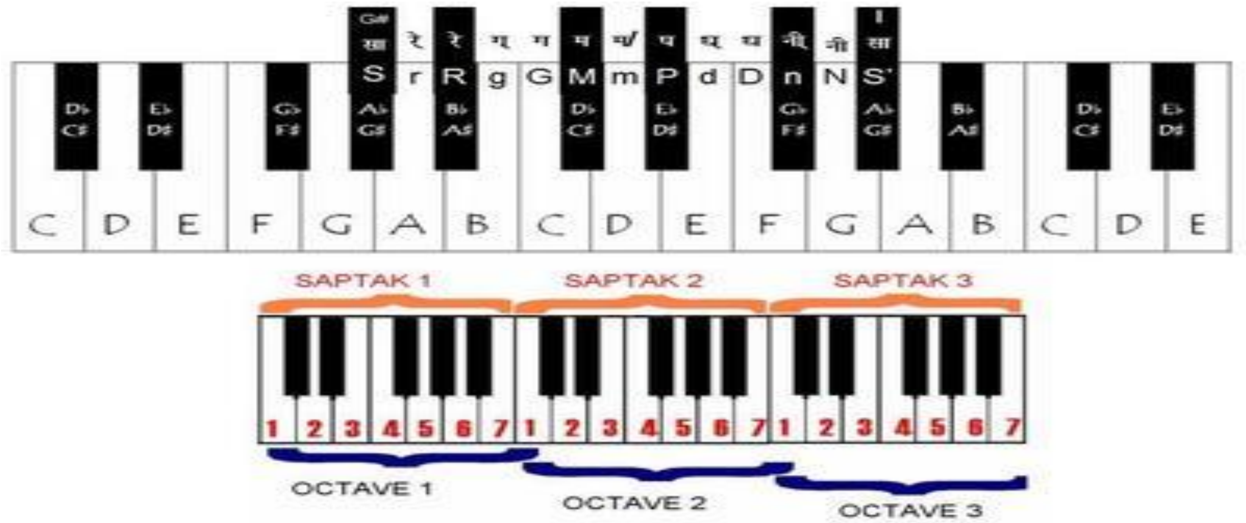

Figure 1.2: Name of notes and octaves [59].

1.2.2.2 Scale

A musical scale is defined as a group of notes ordered by frequency. A scale is the ascending and descending movements from A B C D E F G and G F E D C B A, respectively [60]. In a scale, all the notes must be included in a specified order. Scales are based on patterns of whole steps and half steps. These steps can be explained using the piano for reference (Figure 1.2).

The half step is the key next to any black or white key. Skipping a key by one key is the whole step. For example, E to F is a half step, and C to D is a whole step. Two commonly used scales are the major scale and the minor scale.

Major scale: It is defined as a series of whole steps and half steps in the form –"w w h w w w h" [61].

Minor scale: It is also a sequence of whole and half steps. It is of three types:

- Natural minor: The natural minor scale has a sequence as 'w h w w h w w' [62].
- Melodic minor: In this scale, the sequence is 'w h w www h' [63].
- Harmonic minor: The harmonic minor follows the sequence of steps 'w h w w h wh h' [63].

Where 'w' is the whole step,
'h' is a half step, and
'wh' is the whole half-step.

1.2.2.3 Chords

The set of three or more notes played simultaneously is called a chord. To make a chord, we have to start with the first note and then move up to the next note by skipping one note. For example, to make a C triad in a major scale, the first note is C, skipping D, the next is E, then again skipping F, and the next is G. So, the C triad in a major scale is C E G.

The chord progression is a series of chords that can form musical ideas and phrases [64]. In Western music, chords are used to create harmony. Due to the influence of Western music, Indian music has lately also used chords and chord progressions [65]. Contemporary Indian songs are also based on Western chords. For such Western-motivated Indian music, there is a significant dependency on chords in Indian music.

1.2.2.4 Scale Degree

It is defined as the name for a particular note according to its position on a scale. Degrees are used to specify the chord and interval size. There are a total of seven scale degrees: Tonic, Supertonic, Mediant, Subdominant, Dominant, Submediant, and Leading Tone, as shown in Figure 1.3.

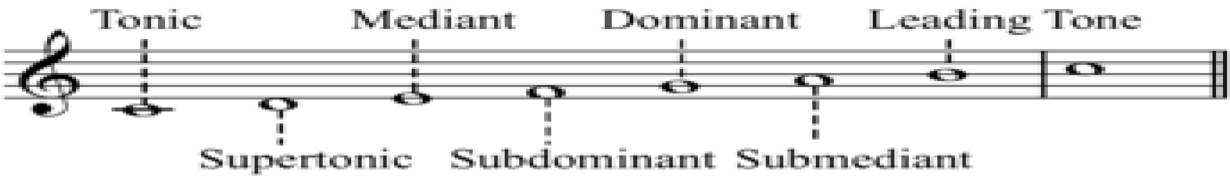

Figure 1.3: The C major scale and the degree names [66].

The names of scale degrees relate to their function and their position on the scale. For example, the Tonic is the basic tone. Subdominant is the degree below the Dominant. Tonic is the first and main note of the scale from which each octave is assumed to begin.

1.3 ORGANIZATION OF THESIS

This thesis is devoted to designing the formal grammar and model for composing Indian music. The results obtained are discussed in chapters 3 to 5. A complete chapter-wise summary of thesis work is given as follows:

- In Chapter 1, gives the motivation for preliminary concepts for research work, which include basic notations, definitions, and the thesis outline.
- In Chapter 2, we discuss different types of computational musicology. We have discussed five methods of computational models for musical patterns using a hidden Markov model (HMM), n-gram, finite-state machine (FSM), finite-state transducer (FST), and artificial grammar learning (AGL). In the last section, gaps in the area of computational musicology, problem statement, and the objectives of the dissertation are explored.
- In Chapter 3, a novel concept of generating formal grammar for Indian musical composition has been proposed. A tool, MuseScore, is used to generate the musical sheet for the audio musical file. From the musical sheet, the musical symbols (notes) are analyzed. A probabilistic context-free grammar is modeled to generate the same progression as composition.
- In Chapter 4, the VOGUE model is applied to designing a musical sequence of notes in bandish of raga Bhairav, classical Indian music. VOGUE integrates two techniques, sequential pattern mining and data modeling to generate the model. In almost every musical form, musical patterns repeat themselves that may or may not be separated by variable gap lengths. VOGUE considers such patterns and, therefore, can model various musical sequences. The musical note sequences in the bandish of a raga are not fixed and may vary according to the performer's choice.
- Chapter 5 examines the patterns in Tala's kaidas and paltas using formal grammar. Results indicate that the patterns in the paltas of Tala's kaida exhibit cross-serial dependencies, which context-free grammar cannot represent. Further, we have represented these patterns using mathematical model deep pushdown automata.
- In Chapter 6, ragas identification has been done using machine learning techniques. A novel approach for raga identification in Indian music using stack ensemble model.

We have ensembled five models, CatBoost, XGBoost, Random Forest, LightGBM, and Decision Tree, based on the prediction accuracy of Indian ragas. Our approach achieves state-of-the-art performance on the task of raga identification and outperforms existing methods. The proposed work can potentially improve the performance of Indian music retrieval systems.

- In Chapter 7, the conclusion and future scope of the thesis are mentioned and outlined to extend the idea of computational musicology.

CHAPTER 2

LITERATURE SURVEY

This chapter discusses the computational aspects of various music operations, such as composition, analysis, retrieval, classification, and implicit learning. We have evaluated the literature based on multiple computational fields like formal grammar, hidden Markov model, n-gram, finite-state machine, finite-state transducer, and artificial grammar learning. The chapter aims to generate a comprehensive description of research on computational musicology. In the last section, gaps in the area of computational musicology, problem statement, and the objectives of this dissertation are explored.

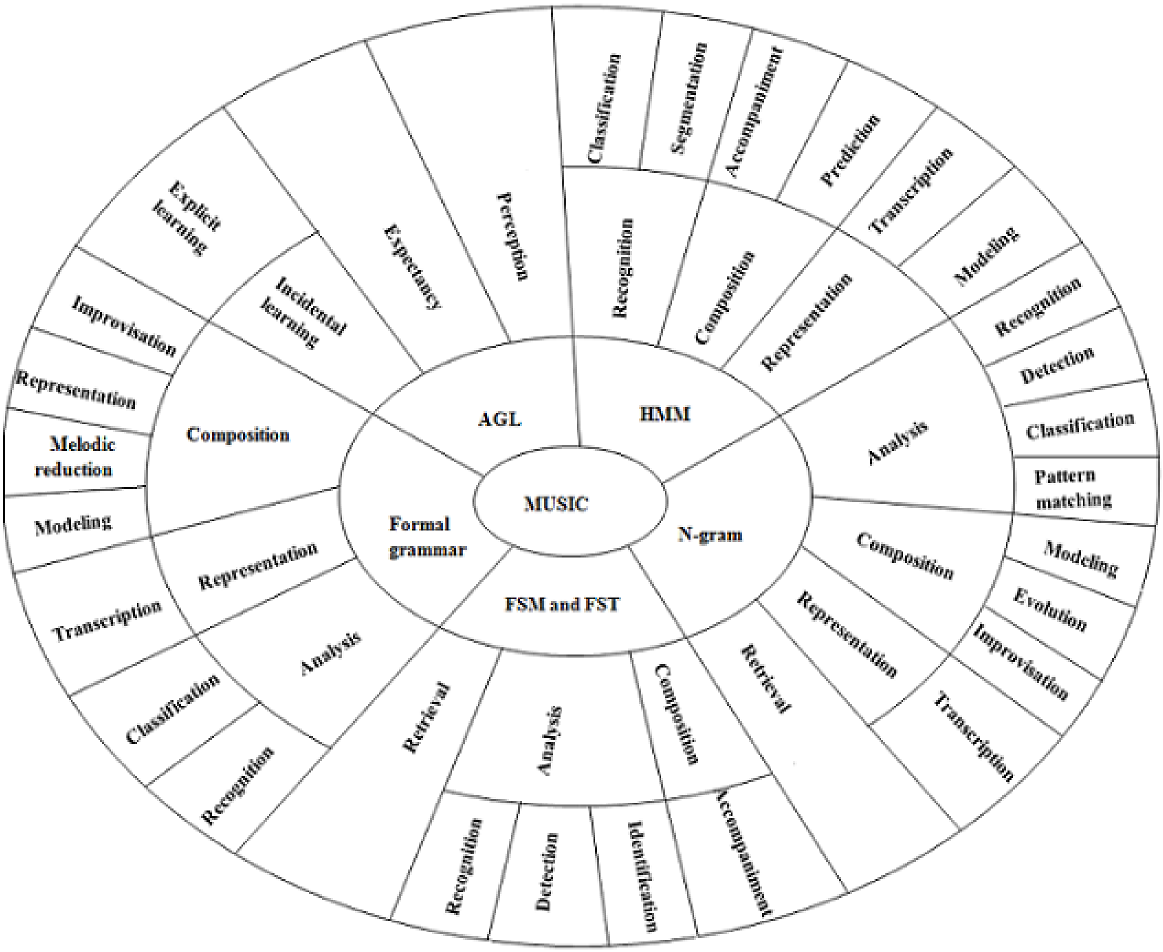

Figure 2.1: Representation of various computational methods with music operations.

2.1 COMPUTATIONAL MUSICOLOGY METHODS

Many researchers developed different computational models for musical patterns using the hidden Markov model (HMM), n-gram, finite-state machine (FSM), finite-state transducer (FST), and artificial grammar learning (AGL). The use of these computational methodologies for building musical models is a significant contribution for musicians as well as for non-musicians. These different methods help in performing different musical operations as shown in Figure 2.1. Computer systems are designed to work on musical sequences due to their practical applications. Several researchers have shown their interest and done research in this area.

2.2 FORMAL GRAMMAR FOR MUSIC

A grammar [67] can be defined by $\mathrm{G(N,T,S,P)}$ where, N is the set of non-terminal symbols, T is the set of terminal symbols, $S \in N$ is the start symbol, and P is the set of production rules of the form $\alpha \to \beta$, where $\alpha, \beta \in (N \cup T)^*$. The work of formal grammar with music is an emerging field. Formal grammar formalizes the rules used to analyze and compose musical sequences. We have explored works by various researchers in composition, analysis, and representation of music with formal grammar.

Gilbert and Conklin [68] performed melodic reduction using probabilistic grammar. Hamanaka et al. [69] proposed a time-span tree generator using SCFG. The generator produced an optimal probabilistic time-span tree.

Bel [70] introduced formal models for representing Indian tabla musical patterns. Pattern and Bol Processor (BP) grammar were used to explain musical structures. Bel and Kippen [27] used BP software to generate musical patterns using formal grammar. The formal models were used to represent improvised Indian tabla music. Bel and Kippen [28] explained different types of BP grammar. This grammar represented transformations of musical structure. Two versions of BP (BP1 and BP2) were used for music improvisation. Bel [29] introduced a new formal grammar for representing Indian tabla music. BP2, with improved features, was used for the composition and improvisation of music.

Jurish [31] represented symbolic music with formal language. The basic musical structures were characterized as formal language. Giraud and Staworko [71] modeled music with parametric grammar. Rohrmeier [72] used phrase-structure grammar rules to represent diatonic harmonic progression. The grammar represented structural dependencies in music sequences. McCormack [73] composed music by using L-system-based grammar. The pitch and duration were used as the grammatical symbols. Puente et al. [74] also composed music using formal grammar. They used grammatical evolution to compose melodies that sound the same as the melodies composed by humans.

Roads [75] performed a literature survey on music representation using different grammar. The grammar was proved to be an efficient way of representing musical structure. Khalifa et al. [76] composed music using a genetic algorithm and formal grammar. They created motifs and applied transformations to the combine motifs to generate musical themes. Sidorov et al.

[77] represented the smallest grammar problem and analyzed the musical structure. The proposed approach generated a context-free grammar for representing musical structures. Table 2.1 depicts the major research work carried out using SCFG for music.

Table 2.1: Major findings in the field of Formal Grammar and music

Year	Author's Name	Software/ Technique	Applications
2007	Gilbert and Conklin [68]	Lilypond software	Melodic reduction
2007	Keller and Morrison [3]	Impro-visor software	Music Composition
2009	Perchy and Sarria [2]	Lilypond software	Music Composition
2010	Kitani and Koike [8]	ImprovGenerator software	Music Composition
2014	Abdallah and Gold [6]	Shortest possible grammar	Generated grammar for musical structures
2015	Hamanaka et al. [69]	Generative theory of tonal music	Generated time span- tree
2016	Abdallah et al. [6]	Programming in statistical modeling	Analyzed symbolic music
2016	Groves [5]	Generative theory of tonal music	Melodic reduction

2.3 HMM FOR MUSICAL COMPOSITION

The hidden Markov model provides a framework for modeling the relationship between a set of notes and acoustic features. HMM is capable of representing flexible musical patterns. HMM is a statistical model that is used for modeling systems. In HMM, the sequence of states depends upon the present state. It consists of emission states (observable) and hidden states. The sequence of states in the model can be identified by analyzing the observed states.

Chai and Vercoe [78] classified different country's folk music with HMM. They represented a significant statistical difference in the music of the different countries. Durey and Clements [79] proposed a model for retrieving the melody of songs. They have used HMM to spot melody in any musical patterns.

Takeda et al. [80] described an HMM for rhythm recognition of Musical Instrument Digital Interface (MIDI) signals. The tempo value, bar locations, and notes were evaluated for recognition. Takeda et al. [81] used a probabilistic approach for recognizing rhythm and tempo. This approach also evaluated temporal factors for a given performance. Pollastri and Simoncelli [82] abstracted musician's style with the trained HMMs. A sequence belongs to a particular composer if the corresponding HMM gives the maximum probability for it. Clement [83] implemented a method using probabilistic grammar to represent and compose music. The model enhanced the learning of other musical features like tempo, rhythm, melody, and duration.

Reddy et al. [84] developed a HMM-based system for identifying Indian raga. The system was based on an automatic notes transcription. HMM was further improved with the pakad matching string algorithm and Mel Frequency Cepstral Coefficients (MFCCs). Bhattacharjee and Srinivasan [85] proposed a transition probability matrix in HMM to represent ragas. This representation helped in the identification and understanding of raga perception.

Krishna [86] proposed an HMM to identify the Carnatic raga. The melakarta ragas were recognized using the proposed model. The notes in the song acted as the states of the model. Sinith and Rajeev [87] designed an HMM system to recognize musical patterns in monophonic South Indian classical music. Each musical pattern was converted into a sequence of frequency jumps. The sequence of frequency jumps was given as input to the system. Mohapatra and Awasthi [88] composed Indian classical music with HMM. They used genetic algorithms with HMM. Pandey et al. [89] designed a system called Tansen for raga identification based on HMM. Yanchenko [90] and Yanchenko and Mukherjee [91] explored HMMs and their variants for composing classical piano musical pieces. The proposed models could compose a new musical structure similar to the musical pieces generated by a human.

Gao [92] segmented and identified music through HMM. They used HMM and feature extraction methods. Shao et al. [93] presented an unsupervised method for music genre classification. The classification was done based on similarity measures provided by HMM. Raphael [94] solved the segmentation problem of acoustic music signals. The musical piece was segmented into different regions for notes and rests in sheet music. The segmentation errors were reduced by using HMM.

Lee [95] introduced an automatic chord transcription system. The symbolic data was used to label musical files with HMM. Orio and Zen [96] proposed an HMM for identifying rock and pop songs based on their melody. HMM was designed from digital musical scores. The states of HMM were represented by the musical notes, and probabilities were evaluated from the data in musical scores.

Cazau and Nuel [97] modeled HMM for automatic music transcription. The proposed model was enhanced by using multi-pitch evaluation. Lee and Slaney [98] used HMM to perform chord transcription and key extraction from raw audio files. The model explained the automatic generation of labeled data for the machine learning models.

Qi et al. [99] analyzed music with an HMM-mixture model based on the Dirichlet process. Sheh and Ellis [100] designed a chord segmentation and recognition system with HMM. The

HMMs were trained with the expectation-maximization algorithm. This system also transcripts chords from unstructured polyphonic audio files.

Ueda [101] discussed a method based on HMM for automatically detecting chords. The HMM used acoustic features for higher recognition rates. Chai and Vercoe [102] segmented musical sequences. They identified a crucial change in piano music. Cano et al. [103] implemented HMMs for matching score performances in music. Batlle et al. [104] designed an HMM to identify songs automatically from noisy audio files. Sertan and Chordia [105] developed a variable-length Markov model for predicting melodies. The model generated melodic improvisation for Turkey's folk music. Mesaros and Tuomas [106] proposed a monophonic model based on HMM.

Miotto et al. [107] modeled music using HMM. The model identified and aligned the music recordings. Zaragoza [108] described an approach for recognizing early handwritten music notations. Jin and Jagadish [109] retrieved music using an HMM indexing mechanism. The musical pieces were represented by their corresponding HMM. Miotto and Orio [110] applied HMM for identifying music. The musical performances were labeled by matching them with pre-labeled recordings in the music database. Blass [111] classified timbre-based drum patterns by HMM. The onset algorithm extracted the timbre features of the data. Chithra et al. [112] presented a method for retrieving musical information for polyphonic Indian music. HMM was developed to extract the required musical information.

A lot of work has been done with Western music and HMM. Figure 2.2 shows the percentage of the papers that used Indian and Western music with HMM. Seven papers used Indian music for HMM, and the rest of the HMM-related papers (forty-three) used Western music.

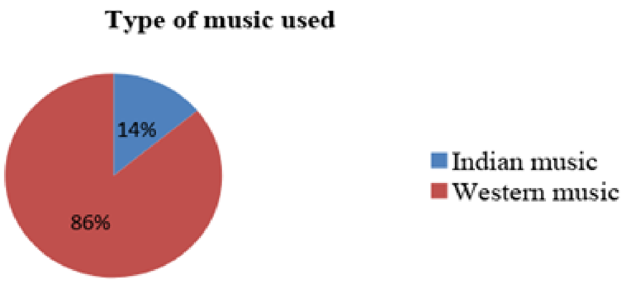

Figure 2.2: HMM papers with Indian and Western music

Flexer [113] used HMM for finding spectral similarity among songs. HMM and Gaussian Mixture Models (GMM) were compared for computing the song's spectral similarity. They shared that HMMs provided better results. Patricio and Chew [114] explored the application of HMM for classifying performances of similar musical pieces. The HMMs were trained on scikit-learning to classify performances automatically. Peeters [115] proposed an automatic estimation system of musical keys from the audio signal. The HMM used chroma vectors as observations. The keys were estimated by finding the likelihood of chroma sequence in the musical corpus.

Chandrashekaran [116] retrieved songs with HMM. HMM provides robustness by handling any noise present in the song. Dhulekar and Shah [117] introduced a method for beat tracking of audio musical signals through HMM. Dalin-Volsing [118] classified musical genres with HMM. Audio data files were classified into four genres: Pop, R&B, classical, and jazz, using proposed HMMs. Kerr [119] designed an HMM to analyze the melodic structure. The analyzed structures were represented in newly generated music compositions using HMM. Berget [120] predicted the musical chords with HMM. Four different HMMs were proposed for predicting chords from the sheet music.

Karpov and Subhramanian [121] classified musical genres by applying HMM and digital signal processing techniques. The model was best suited only for a medium-sized collection of music genres. Petruncio and Johnson [122] evaluated different musical features for classifying musical genres using HMM. Benetos and Weyde [123] used explicit duration HMM to model notes duration for the transcription of polyphonic music from the instrument. Nakamura et al. [124] proposed a model for generating a new score using the symbolic performance algorithm. An autoregressive hidden semi-Markov model was used to follow the score and represent polyphonic music symbolically. Nakano et al. [125] used Mondrian HMM for analyzing musical signals. The model captured the clusters present in the transition of hidden states.

Wang and Dubnov [126] clustered notes based on temporal context similarity. The context-aware HMM captured temporal relationships based on the variable Markov Oracle. Simon et al. [127] introduced MySong to generate musical accompaniment for vocal melodies. MySong used trained HMMs to compose musical chords for different melodies.

A lot of research has been done in the field of HMM for various musical operations like classification, composition, and representation of music. Table 2.2 represents the classification of different kinds of music with HMM and the accuracy achieved in the classification process.

2.4 N-GRAM FOR MUSICAL PATTERNS

N-gram uses pattern matching to retrieve musical information. The increasing musical database is managed and indexed by n-gram. An n-gram is a sequence of n continuous words from a given sequence of text in a corpus. The sentence is split into smaller chunks of consecutive words of n length. An n-gram, having size 1, is termed a unigram.

Similarly, n-grams with size 2 are bigram, and so on. N-gram is an algorithm for identifying the next word in a sequence after analyzing n-1 words. Finding the probability of an upcoming word is related to computing the probability of a sequence of words. We have surveyed the work of different authors to show the efficiency of n-gram for different musical operations.

Scholz [128] used probabilistic n-gram with $n \geq 2$ for modeling chord progressions. It has been observed that n-gram and the smoothing technique give better results. Cheng [129] recognized musical chords by applying the n-gram model and HMM. The musical emotions were classified based on various chord features. The largest common chord subsequence and chord histogram were used for managing, retrieving, and analyzing music.

Doraisamy and Ruger [130] explored n-gram techniques for retrieving full polyphonic music. Two musical factors, pitch, and rhythm, were used for indexing musical data files. The technique improved information using the onset time difference ratio between the pitch events. Doraisamy and Ruger [131] studied the fault tolerance of the n-gram of polyphonic music. The monophonic musical queries were solved using a polyphonic music database. Doraisamy and Ruger [132] examined the retrieval performance for polyphonic music data using concurrent and adjacent n-grams. The proposed method used an n-gram approach with a word position indexer for indexing polyphonic music. Doraisamy and Ruger [133] retrieved monophonic and polyphonic music. An improved n-gram approach was used for indexing monophonic music data to polyphonic music. Query-by-humming was the center point for monophonic queries, and query-by-example was the center point for polyphonic queries. Doraisamy and Ruger [134] worked in the field of polyphonic music. The music information retrieval (MIR) system was designed with n-gram. The pre-processing and indexing of musical documents were performed to retrieve information. A proximity-based operator and the ranking function were used for improved efficiency.

Lo and Lucas [135] composed music using the n-gram system as trainable music evaluators in the evolutionary algorithms. The system had the problem of repetitive musical sequences. N-gram acted as a fitness function in an evolutionary algorithm. Lo and Lucas [136] removed the problem of repetitive sequence using a constraint. Mutation and permutation were applied on a selected note bag to solve the maximum likelihood repetitive sequence problem. Ogihara and Li [137] identified composition styles through n-gram models. N-grams were assigned with a weight proportional to each musical beat count. The different composition styles were generated from chord progression using n-gram profiles.

Table 2.2: Music classification using HMM

Year	Author's Name	Dataset	Accuracy	Sampling rate
2001	Chai and Vercoe [78]	Total 491 folk songs: 187 Irish, 200 German, and 104 Austrian folk songs	75%(for Irish), 77%(for German) , 66% (for Austrian)	---
2001	Pollastri and Simoncelli [82]	Total 605 musical themes: Mozart, Beethoven, Dvorak, Stravinsky, and The Beatles musical themes	42%	---
2002	Karpov and Subramanian [121]	Total 252 songs: 47 Celtic, 69 Western Classical, 70 Techno &Trance, and 66 rock songs	---	11.025 KHz
2004	Shao, Xu and Kankanhalli [93]	Total 200 music pieces: 50 Pop, 50 Country, 50 Jazz, and 50 Classic music	89%	44.1KHz
2013	Blass [101]	Pop, Rock, and Electro solo drum musical pieces	100% (for simple music)	44.1 KHz
2013	Benetos and Weyde [123]	Total 200 music genres: Jazz, Rock, Classical and Dance music	88.3%	---
2017	Volsing [118]	Total 400 music genres: 100 Pop, 100 Classical, 100 Jazz, and 100 R&B music	74.2% (audio data), 54% (theoretical data)	44.1KHz

Takano and Osana [138] designed an automatic music composition system with a genetic algorithm and an n-gram model. Melody blocks were used for music composition. Each melody block was trained with n-grams. Eight different factors calculated the required fitness function to generate melodies. Paulus and Klapuri [139] proposed a system for drum sequence transcription. Conventional and periodic n-grams were used to model the system.

Lap and Kao [140] studied the indexing of musical features using n-gram. Different issues of indexing musical features using simulated queries were considered. Wolkowicz et al. [141] introduced a novel approach for recognizing composers. N-gram was used for indexing and browsing musical libraries. Wang et al. [142] retrieved musical themes by using index structures. The proposed structure was based on the n-gram and inverted files. Indexing and inverted files were used to generate an algorithm for mining musical themes.

Trochidis et al. [143] proposed a new system, CAMEL, for automatic music generation. The system generates new compositions of Carnatic music using n-gram methods. The N-gram method was used to represent short rhythmic patterns in Carnatic music. Unal et al. [144] classified Turkey's makam music using n-gram models with the SRI Language Modeling (SRILM) toolkit. The n-gram-based makam statistical models were used to detect symbolic data of makam music. Frieler [145] generalized n-gram approaches. The approaches were used to measure similarity in a single line of musical patterns. Patel and Mundur [146] extracted recurring patterns in musical scores using n-gram. Kumamoto and Ohta [147] developed an n-gram system for MIR based on impressions.

Doraisamy [148] used the n-gram approach for retrieving polyphonic music. Different n-gram techniques were compared for indexing full polyphonic music. Uitdenbogerd [149] matched musical patterns in symbolic melody by applying n-gram method. The query of indexing the n-gram to search for a melody from the musical collection was solved. Salosaari and Jarvelin [150] designed the MUSIR model for music information retrieval. Time and interval sequences were extracted using a filtering mechanism on MIDI files. The extracted information was changed to an n-gram representation for retrieving different-length musical documents.

Pakhare and Potey [151] identified different composer's styles. Multimodal, pattern recognition and co-updating approaches were studied and compared for music retrieval. Hauser [152] presented a probabilistic model for sequencing musical chords. N-gram was used for analyzing musical chords. Suyoto and Uitdenbogerd [153] proposed n-gram methods for retrieving music. N-gram indexing retrieves symbolic melody from a musical collection. Wang et al. [154] presented a novel method for audio fingerprinting using n-grams. N-grams could identify audio segments even if the musical signal were distorted. N-gram accelerated the searching speed and improved retrieval accuracy.

Hontanilla et al. [155] modeled musical styles using n-grams. The techniques of language modeling captured musician's styles. The musical style models recognized different composers. Zheng et al. [156] classified musical genres with n-gram. The approach used high-level musical features for classification purposes. Felch and Song [157] performed music improvisation using n-gram models. A statistical model was implemented to generate musical sequences automatically that follow musical improvisations. Alpkkocak and Gedik [158] classified Turkish songs based on makams. A two-level hierarchy with a decision tree and linear discriminated classifier was used for classification using an n-gram.

N-gram can relate different musical operations, such as the classification of music, MIR, music type identification, and music modeling. Most researchers worked on the MIR, but less work has been done in music classification as represented in Figure 2.3.

2.5 FSM AND FST WITH MUSIC

An artificial generation or identification of music has many implications for Indian and Western music. The artificial models are based on FSMs and FSTs for analyzing and composing musical sequences. FSM is a computational model based on theoretical machines with finite states. FSM can be modeled for various software and hardware problems in many fields. It changes its state when an external input is given. FSM has only one memory tape. It defines a language by analyzing the set of accepted strings. FST is an FSM with an input and an output tape. Input tape is used to accept the input string, and output tape is used to generate another string.

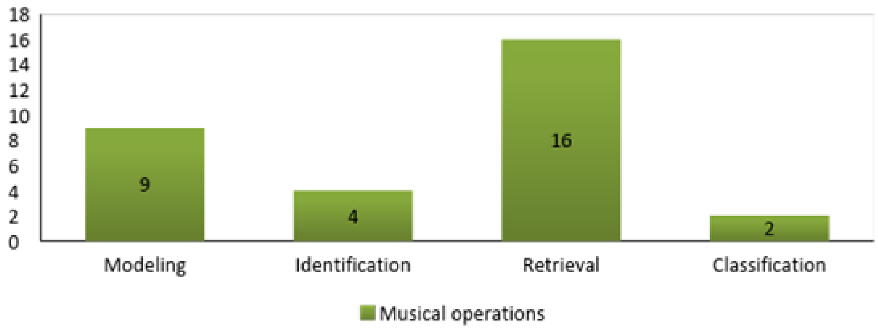

Figure 2.3: N-gram with various musical operations

Das and Choudhury [159] discussed a model for generating Hindustani classical music. The arohana and avarohana of the required raga were used to implement the model. The hand-crafted bigram FSM was used to generate the sequence of notes. The quality of the sequences composed was checked by comparing them with the sequences generated from unigram models. Hosoya et al. [160] described a recognition grammar to recognize lyrics from a singing voice. A finite-state automaton (FSA) accepts only a part of word sequences present in the song database.

Oliwa and Wagner [161] presented a system for music composition using FSM and neural networks. The system reminds all the critical elements used for creating ne1w musical patterns. The inductive learning from neural networks was used to gain musical knowledge. Zaragoza and Oncina [162] recognize pen-based music notations using FSA. The probabilistic FSA was used to identify probabilities of musical sequences. The proposed approach helped in learning handwriting styles from the data. Suzuki et al. [163] used lyrics and melody to design the MIR system. They have used FSA as a recognition grammar in the music retrieval system.

Weinstein and Moreno [164] identified music using weighted FST. Unsupervised training was used to learn musical sequences. The proposed approach matches sequences of music phonemes. Buys and Merwe [165] harmonized four-part musical pieces using weighted FST. The FST was designed for horizontal and vertical structures generated in harmonization. The weights in FST were assigned by estimating the maximum likelihood from the dataset.

Mohri et al. [166] identified and detected music based on weighted FST. The effect of noise and other distortion on the system was analyzed. Mohri et al. [167] identified music based on the Gaussian model and weighted FST. Their model represented songs using FST and detected noises and distortion presented in the song.

Hubler and Hoffmann [168] represented drum patterns using weighted FST. The input to the system was the bar length in the drum patterns. The trained models automatically recognize drum patterns from a given sequence. Forsyth and Bello [169] used FST to compose harmonic accompaniment to a melody. The system was trained on MIDI data. The system was evaluated using computing accuracy and per-symbol distortion. Forsyth et al. [170] composed harmonic accompaniment to the melody using a speech recognition technique for FST.

2.6 AGL FOR MUSIC

AGL is a model used to study implicit learning. Implicit learning is unconscious learning of complicated information. AGL investigates the extent to which any person can detect the given patterns. People can acquire complex musical structures without the intention of learning them. Music can be learned without the awareness of its structure and rules. It can be acquired incidentally only by hearing several musical sequences. Various researchers worked on music and AGL.

Ettlinger et al. [171] reviewed the implicit memory in music. The effect of implicitly acquired knowledge and its associated structures were analyzed. They examined music and language in the context of implicit memory. Rohrmeier and Widdess [172] investigated melodic features in North Indian traditional music through implicit learning. Cross-grammar designs were used on the ragas, which had similar scales. Musicians, as well as non-musicians, learned musical structures with a brief introduction to the musical structures.

Loui [173] studied linking and learning of musical harmony and melody. AGL was used to identify various aspects of music that could enhance learning. Bly et al. [174] learned musical structures and phonological sequences through domain-specific learning. Both the structures were grammatically similar, but the learning was easier for melody-related chord sequences instead of sequences with letter names.

Kuhn and Dienes [175] proved that implicit learning of non-local musical sequences was beyond the learning of musical chunks. Musical sequences from diatonic inversion were used for implicit learning. Desmet et al. [176] commented on the work of Kuhn and Dienes [175] and proved that their result was due to confounds between the by-products of diatonic inversion. Whenever confounds were removed from Kuhn and Deines's dataset, the effect of grammaticality disappeared.

Rohrmeier and Cross [177] studied the implicit learning of harmonically context-free structures. The learning ability of different harmonic structures was analyzed. They concluded that the implicit learning of grammatical structures was different from the learning of ungrammatical structures. Tillmann [178] discussed learning regularities in an artificial tone system. Music cognition including music learning, perception, and expectations were discussed. Jiang et al. [179] investigated whether the rules found in Tang's Chinese poetry could be implicitly learned or not. The results showed that implicit learning is possible in the tonal inversion rules.

Schellenberg [180] examined implicit knowledge in children. A sequence of chords ended by a final chord was used as a dataset. They proved that children had implicit knowledge about harmony in Western music. Selechenkova et al. [181] investigated the effect of temporal regularities on implicit learning. The implicit learning of pitch structure helped listeners to identify the next musical sequences. Tillmann and Poulin-Charronnat [182] studied auditory expectations for newly acquired structures. They identified that newly acquired knowledge could help in developing perceptual expectations for upcoming events. Rohrmeier and

Rebuschat [183] analyzed implicit learning for music acquisition. The validity of cross-culture implicit learning was analyzed. Bigand et al. [184] studied the implicit learning of musical timbres of an artificial grammar. Dienes and Higgins [185] identified the extent to which music can be learned implicitly. They determined the learning ability of musical sequences. Schellenberg [186] examined the implication-realization model for continuous melodies in tone expectancies. The model proved efficient for predicting judgments of different musical styles. Rohrmeier and Cross [187] explored AGL for musical melodies. The effect of Narmour's principle on melodic learning was analyzed. They found that the structures with no basic melodic principles were harder to learn. Rohrmeier et al. [188] worked on incidental and online learning of melodic systems. The study was carried out for musical acquisition using AGL.

The results showed that musicians and non-musicians performed almost similarly. Tillmann et al. [189] demonstrated implicit learning of tonality in a highly structured system using a self-organizing approach. The tones were clustered to form chords in the system.

Bigand and Poulin-Charronnat [190] identified musical capacities that do not depend on musical training. Endress [191] learned the musical melodies from non-adjacent tones. The different constraints which limit the amount of implicit learning of melodies were examined. Tillmann [192] investigated implicit learning of tonal knowledge for analyzing non-musicians. Various researchers conducted experiments to determine the extent of implicit learning of music by children, musicians, and non-musicians, as shown in Table 2.3.

Table 2.3: Summary of research done for learning music using AGL. Here Exp, GS, and VS represent the experiments performed, grammatical, and ungrammatical structures respectively.

Year	Author's Name	Participants	Dataset	Result
1998	Bigand, Perruchet and Boyer [184]	120 Psychology students, University of Bourgogne	20 GS and 25 US	Participants were able to find new sequences by exposing GS to musical timbres.
2005	Kuhn and Dienes [175]	96 Individuals, University of Sussex	120 GS and 120 US	Implicit learning could go beyond the learning of musical chunks.

1996	Schellenberg [186]	Exp 1: 20 Member, Cornell University Exp 2: 26 Member, Cornell University Exp 3: 16 Member, Cornell University	Exp1: 8 melodic fragments (British folk song) Exp 2: 8 melodic fragment (Webern Lieder) Exp 3: 2 melodic fragment (Chinese folk song)	Implication-realization model was over specified. Participants with no musical background acquired musical sequences.
2004	Dienes and Higgins [185]	Exp 1: 4 Groups Exp 2a: 32 students Exp 2b: 14 students	Exp 1: 12 tones Exp 2a: 35 tones Exp 2b:12 tones	Different people with different interests could detect melodies implicitly to different levels.
2005	Schellenberg[180]	Exp 1: 23 Children Exp 2: 36 Children	Exp 1, 2: 24 tones	Children had implicit knowledge of syntactic features that represent Western music.
2005	Tillmann [192]	3 participants	Four categories of words	Implicitly acquired musical knowledge allowed non-musicians to extract relations among musical structures.
2009	Bly, Carrion, Rasch [174]	202 Students, Rutgers Newark University	36 GS and 36 US	Learning was better for participants for harmonically related chord sequences.
2009	Desmet, Poulin-Charronnat, Lalitte, Perruchet [176]	18 Adult musicians	66 GS and 66 US	Participants were able to learn long-distance dependencies.

2010	Tillmann and Poulin-Charronnat [182]	Exp 1: 32 Students, Lyon University Exp 2: 24 Students, University of Burgundy	37 GS and 37 US	Acquiring a new musical structure's knowledge developed a perceptual expectation among participants.
2010	Endress [191]	Exp 1-4: 20, 20, 20, and 20 Italian speakers for each Exp.	Ten different trials	Non-adjacent tone relations were difficult to compute for the participants.
2011	Rohrmer, Rebuscat and Cross [188]	59 Western adults	17 old GS and 16 new GS	Participants acquired musical structures with incidental learning.
2012	Jiang et al. [179]	43 Volunteers	48 GS and 96 US	Participants could detect inversions unconsciously.
2012	Rohrmeier and Widness [172]	65 Western adults	2 Indian ragas: Multani and Todi Raga	Both musicians and non-musicians learned musical patterns.
2012	Loui [173]	24 Undergraduates, University of California	400 Melodic structure	Disrupting melody disabled learning, but disrupting harmony did not affect learning.
2013	Rohrmeier and Cross [188]	31 Adults, University of Cambridge	17 old GS and 16 new GS	Melodic structures that violate melodic principles seemed harder to be learned.
2014	Selchenkova, Jones and Tillmann [181]	35 Students, Lyon University	36 Tuned, mistuned GS, and 36 tuned, mistuned the US	Strong capability to implicitly learn artificial materials with regular exposure to patterns.

Various works on Indian and Western music composition have been surveyed, and key findings have been cited for relating to the objective of the work and their implication research. The relevant available literature has been extensively surveyed and discussed. The work has helped in framing formal grammar and model for Indian music. The work is based on modeling of raga and tala which are the basic rhythmic elements of Indian music. The available literature survey has been concisely utilized to pave the way for guidance and results of work.

2.7 GAPS IN LITERATURE

Based on the literature survey, some of the research gaps were identified for the proposed research work of computational musicology.

1. For many years, work related to formal grammar and music composition has been examined, but there are still various open problems. A literature survey reveals that music highly correlates with formal grammar. Stochastic Context-free grammar (SCFG) or regulated grammar can generate grammar for a musical composition.
2. Many researchers have designed techniques for generating a formal grammar for English music but very little attention has been paid to Indian music grammar generation. Hence, there is a need to introduce a technique for generating a formal grammar for Indian music. The N-gram or HMM approach can be used in conjunction with music to find a composition style from a chord progression, and it will lead to different options for composers and genres for generating new composition styles.
3. Various methods have been developed to generate English music using different finite state methods, but little attention has been given to Indian music. Therefore, ample scope exists for these formal models or probabilistic models to be used with Indian music.

2.7 PROBLEM FORMULATION

In this section, the problem statement and objective of our work have been discussed.

2.7.1 PROBLEM STATEMENT

The research proposed in this work is to design a formal grammar for Indian music. The aim of the research proposal is related to formal grammar and applying it to the composition of Indian music. A song will be given as input in a musical tool to generate the grammar of that song as output. The output can be generated for the song by using the chord progression used in that song. The grammar generated will help to parse the musical composition by providing us with the most probable parse for the given song.

2.7.2 OBJECTIVES

The following are the objectives of this dissertation:

1. Design of a technique for generating a formal grammar for Indian music.

2. To apply any existing tools in aid for the generation of formal grammar.
3. To design a formal model for the generation of Indian music.

CHAPTER 3
GENERATING FORMAL GRAMMAR FOR INDIAN MUSICAL COMPOSITION

We have a set of input symbols and production rules in formal grammar. The required string is generated by using the production rules on input symbols. Similarly, in music, we have a set of input symbols and rules of music composition. So, the concepts of formal grammar and music can be interrelated. The vital part of music generation comprises some musically organized rules. The task of music composition does not have a specific solution. For music composition, there is a need to develop a system that can generate pleasing musical sequences and generate various kinds of music. Musical rules are applied based on scale (major/minor) and key used in the composition.

3.1 TOOL USED FOR GENERATING MUSICAL SHEET

We use a musical tool, MuseScore, to generate a musical sheet for the composition. The musical sheet consists of musical symbols of written musical notation. The symbols on the sheet represent the chords and rhythms of a musical structure.

.

Figure 3.1: Musical sheet for the National Anthem of India

MuseScore is powerful, free, flexible, and open-source software. It is very easy for musicians as well as non-musicians to use. Input can be given using a mouse, keyboard, or virtual piano keyboard. One can also play and listen to the musical sheet generated through MuseScore. The musical sheet for India's National Anthem generated by MuseScore is shown in Figure 3.1

3.2 IDENTIFYING CHORDS USED IN MUSICAL COMPOSITION

As discussed, music and formal grammar can be interrelated; therefore, grammar can be generated for a musical composition. Musical rules are applied based on scale (major/minor) and key used in the composition.

A musical sheet is analyzed for each musical symbol. For notes represented in the sheet, chords, and chord progression are identified. Musical rules for chord progression are shown in Figure 3.2. Using the key used in the composition, each chord is numbered accordingly. For example, in C major, chord CEG and ACE are represented by I and vi.

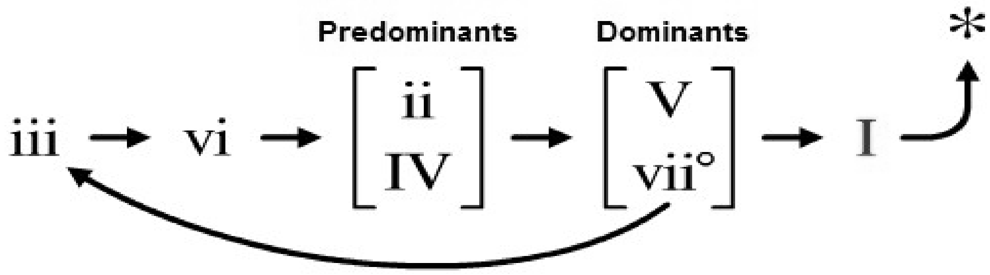

Figure 3.2: The chord progression in major scale [193]

These numbers are then represented with degree notations, as shown in Table 1. The degree notations define the string for a particular phrase.

Table 3.1: Degree Representation

Tonic (T)	Supertonic (S_1)	Mediant (M)	Subdominant	Dominant (D)	Submediant (S_2)	Subtonic
I	Ii	Iii	Iv	V	Vi	vii

For these phrases, we have written the production rules and PCFG. A probabilistic context-free grammar is modeled to generate the same progression as the composition's. Using musical rules and probabilistic context-free grammar rules, the required grammar for the National Anthem is generated.

Degree functions:

$T \rightarrow i$ **[0.53]** **#Tonic**

$\rightarrow iii$ **[0.30]**

$\rightarrow vi$ **[0.17]**

$S_1 \rightarrow iv$ **[0.26]** **#Subdominant**

$\rightarrow ii$ **[0.56]**

$\rightarrow vi$ **[0.18]**

$D \rightarrow v$ **[0.29]** **#Dominant**

$\rightarrow vii$ **[0.29]**

$\rightarrow iii$ **[0.43]**

$M \rightarrow iii$ **[1.0]** **#Mediant**

$S_2 \rightarrow vi$ **[1.0]** **#Submediant**

3.3 GRAMMAR GENERATION

S' is the start symbol, S is used for the phrase, and the end of the phrase is represented by symbol F.

Start:

$S' \rightarrow SF$		**[1.0]**

Phrases:

$S \rightarrow TS_1TS$	**....1**	**[0.07]**
$\rightarrow TS_2S_1TS$	**....2**	**[0.07]**
$\rightarrow S_1DS_2DTF$	**…3**	**[0.07]**
$\rightarrow TMS_1DS$	**….4**	**[0.07]**
$\rightarrow S_1S_1DS_1TS$	**….5**	**[0.07]**
$\rightarrow TS_2DF$	**….6**	**[0.07]**
$\rightarrow TS_1DS_1TS$	**….7**	**[0.16]**
$\rightarrow DTS_2S_1\mathrm{S}$	**….8**	**[0.07]**
$\rightarrow S_1DTS$	**….9**	**[0.07]**
$\rightarrow TDTF$	**….10**	**[0.07]**
$\rightarrow S_1TS_1SDT$	**….11**	**[0.07]**
$\rightarrow DTDTF$	**….12**	**[0.07]**
$\rightarrow S_1TS_1F$	**….13**	**[0.07]**

End:

$F \rightarrow TS_1T$ **….14** **[0.5]**

$\rightarrow TS_1$ **….15** **[0.5]**

A probabilistic context-free grammar is used to generate the grammar of a musical composition. All the production rules are assigned with some probabilities based on Prob$_{(r)}$.

Prob$_{(r)}$$= \frac{\text{number of times r is used in composition}}{\text{total number of rules used}}$

The rules expanded for the same non-terminal symbol should have the sum of their probabilities equal to 1. The probability of the degree (Prob$_{(d)}$) is also calculated in the same manner.

Prob$_{(d)}$$= \frac{\text{number of times d is used in composition}}{\text{total number of degrees used in composition}}$

3.3.1 The first string of song:

$TS_1TTS_2S_1TS_1DS_2DTTS_1T$

Grammar for the string

$S \rightarrow TS_1TS$ **….1**

$S \rightarrow TS_1TTS_2S_1TS$ **….2**

$S \rightarrow TS_1TTS_2S_1TS_1DS_2DTS$ **….3**

$TS_1TTS_2S_1TS_1DS_2DTTS_1T$ **….14**

3.3.2 The second string of song:

$TMS_1DS_1S_1DS_1TTS_2DTS_1$

Grammar for the string

$S \rightarrow TMS_1DS$ **….4**

$S \rightarrow TMS_1DS_1S_1DS_1DS$ **….5**

$S \rightarrow TMS_1DS_1S_1DS_1TTS_2DF$ **….6**

$S \rightarrow TMS_1DS_1S_1DS_1TTS_2DTS_1$ **….15**

3.3.3 The third string of song:

$TS_1DS_1TTS_1DS_1TTS_1T$

Grammar for the string

$S \rightarrow TS_1DS_1TS$ **….7**

$S \rightarrow TS_1DS_1TTS_1DS_1TTS_1T$ **….7**

$S \rightarrow TS_1DS_1TTS_1DS_1TTS_1T$ **….14**

3.3.4 The fourth string of song:

$DTS_2S_1S_1DTTDTTS_1T$

Grammar for the string

$S \rightarrow DTS_2S_1S$ **….8**

$S \rightarrow DTS_2S_1S_1DTS$ **….9**

$S \rightarrow DTS_2S_1S_1DTTDTS$ **….10**

$S \rightarrow DTS_2S_1S_1DTTDTS_1T$ **….14**

3.3.5 The fifth string of song:

$S_1TS_1DTDTDTDTS_1$

Grammar for the string

$S \rightarrow S_1TS_1S$ **….11**

$S \rightarrow S_1TS_1DTS$ **….12**

$S \rightarrow S_1TS_1DTDTS$ **….12**

$S \rightarrow S_1TS_1DTDTDTTS_1$ **….15**

3.3.6 The sixth string of song:

$S_1TS_1TS_1$

Grammar for the string

$S \rightarrow S_1TS_1$ **….11**

$S \rightarrow S_1TS_1TS_1$ **….15**

3.4 CONCLUSION

We proposed a method for generating formal grammar for Indian music. The grammar generated satisfies music rules and SCFG rules. MuseScore is used to generate the musical sheet for the composition. From the sheet, musical symbols are analyzed, and grammar is generated. Our work can help musicians as well as non-musicians in their music composition work. Grammar for different compositions can be generated by using the concept described in this chapter.

CHAPTER 4

MIMVOGUE: MODELING INDIAN MUSIC USING A VARIABLE ORDER GAPPED HMM

4.1 INTRODUCTION

Music composition and mathematical computation are closely related as they have similar aesthetics. In the last few years, musicians and researchers have used computers to design compositional models for musical patterns. The hidden Markov model (HMM) is a statistical method that can be used to analyze and compose musical patterns. Various researchers used HMM to analyze speech recognition, biological sequence, robot control [194], modeling observed real-world data [195], and composing music [196]. In first-order HMM, the present state depends only on its previous state and can emit more than one symbol. Higher-order (n-order) HMM requires a joint evaluation probability for all previous states.

First-order HMM did not consider the sequential dependencies for longer ranges. Higher-order HMM has been proposed to overcome the limitation of first-order HMM, but it results in low accuracy and high state-space complexity. In most raga patterns, a particular VOGUE can handle sub-sequences separated by length gaps. To overcome the limitation of these HMMs, Variable Order and Gapped HMM for Unstructured Elements (VOGUE), Bouqata, Carothers, Szymanski, and Zaki proposed a variant of HMM in 2006. VOGUE integrates two techniques, sequential pattern mining and data modelling, to generate the model. VOGUE is generated by using two steps. The first step uses a Variable-Gap Sequence miner (VGS) algorithm to mine frequent patterns of variable length that have different gap lengths between the elements. The second step uses these mined sequences to build the VOGUE model. VOGUE can model many real-world problems with complex dependencies.

In this chapter, we applied VOGUE to Indian music to model musical sequences. VOGUE can identify both short and long-range dependencies in musical sequences. In almost every musical form, musical patterns repeat themselves that may or may not be separated by variable gap lengths. VOGUE considers such kinds of patterns to model various musical sequences.

4.2 PRELIMINARIES

This section briefly discusses the terminologies of Indian classical music and HMM. The raga is termed the basic unit of Indian classical music [197]. The two phrases, arohana and avarohana represent a raga. The arohana consists of notes in ascending order, and avarohana has notes in descending order that a raga must follow [198]. A bandish is a summary of a raga over which raga is expanded or composed by musicians. The notation of bandish is not fixed and varies according to musicians [199]. The bandish of raga Bhairav is used as a musical sequence to generate the model. The notation of bandish uses arohana (S, Rb, G, M, P, Db, N, S') and avarohana (S', N, Db, P, M, G, Rb, S) [200] of raga Bhairav. Along with the notes, prolongation of a note (−) represents long durations between the notes. HMM represents a probability distribution over observations and can model sequential data efficiently. The HMM has a hidden process and an observable process [201]. Transitions among the HMM states are maintained by a set of transition probabilities.

A HMM is defined by Quintuple $(Q, \sum, A, B, \pi)$ where,

Q is a set of states.

$\sum$ is an alphabet consisting of a set of symbols in the training set.

A represent state transition probabilities and denoted by $A = a_{ij}$, here a_{ij} is the probability from state i to state j.

B denotes the symbol emission probability and $B = b_i(v_k)$ represents the probability emitting v_k at state i.

π represents the initial state.

4.3 VOGUE MODEL FOR INDIAN MUSIC

VOGUE was motivated by variable range dependencies among sequential data. It was introduced to analyze English text and to model real-world problems like protein sequences. The modeling of VOGUE includes two steps:

1. Mining frequent sequences via the Variable-Gap Sequence mining (VGS) algorithm.
2. Modeling mined data. It mines frequently occurring sequences with variable gaps having g symbols between two alphabets.

We model VOGUE for a raga bhairav from Indian classical music. Let Z be the musical sequences in bandish of Bhairav raga. Let F be the frequently mined sub-sequences by VGS and k be the maximum length $(\max gap)$ of the mined sequence. We are considering $k = 2$ for modeling VOGUE with the frequently mined sequences. Let F_1 and F_2, $(F = F_1 \cup F_2)$ be the all frequently mined subsequence with length one and length two, respectively (Table 4.1). Let $\Gamma = \{\mathrm{v}_1 \mid \mathrm{v}_1 v_2 \in \mathrm{F}_2\}$ be the set of all different first position symbols, and $\Theta = \{\mathrm{v}_2 \mid \mathrm{v}_1 v_2 \in \mathrm{F}_2\}$ be the set of all different symbols at second positions of every subsequence in F_2. The mined sequences evaluate the joint probabilities of states for the model.

Table 4.1: Frequency of symbols in musical sequence

Notes	S	R_b	G	M	P	D_b	N	$S^{'}$	Prolongation of note $(-)$
Frequency	23	34	50	59	54	50	46	47	42

Table 4.2 represents the most frequently occurring subsequence in Z with different length symbol gaps. The basic VOGUE model (λ) can be defined by 6-tuples $\{Q, \Sigma, A, B, \pi, \Delta\}$ [182] where $\sum = \{S, R_b, G, M, P, D_b, N, S^{'}, -\}$ and $M = |\sum| = 9$.

The total number of states is calculated by, $|Q| = N = N_1 + N_2 + G + 1$ where

$N_1 = |\Gamma| = 5$ represents the number of different symbols in the first position in F_1.

$N_2 = |\Theta| = 5$ represents the number of different symbols in the second position F_2.

G represents the total number of gap states.

Table 4.2: Sub sequences of length 1 and 2 with gap information

Subsequence	**Frequency**	**g=0**	**g=1**	**g=2**	**Gap symbols (with frequency)**
D_bP	23	19	4	0	$M(2), -(2)$
D_bM	18	2	14	2	$P(15), G(1), D_b(1)$
PP	17	5	10	2	$D_bP(6), M(7), -(1)$
MD_b	12	3	9	0	$P(9)$
MM	18	6	11	1	$P(9), G(4), D_b(1), -(1)$
GR_b	13	9	4	0	$M(3), P(1)$
NN	14	6	3	5	$D_b(2), S(3), S^{'}(8)$
NP	12	0	8	4	$D_b(12), S^{'}(3), G(1)$
GP	14	2	11	1	$M(12), D_b(1)$

Figure 4.1 represents the VOGUE model for the Indian music raga Bhairav (Z). The first symbol states (Q_1^s) are shown as circle-shaped, the second symbol states (Q_2^s) are represented

as double ellipse-shaped, the gap states (Q^g) are shown as ellipse-shaped, and the universal gap (Q^u) is shown as double box-shaped.

The algorithm for implementing the VOGUE model:

1. Define the set of states and the set of symbols. Initialize transition probabilities, emission probabilities, and initial state probabilities.
2. Determine the maximum order of dependencies to consider in the model.
3. For each state, check which state is first symbol state, which one is second symbol state and which is gapped symbol state from the dataset.
4. Calculate various probabilities, i.e.,

- Transition probabilities, A and denoted by $A = a_{ij}$, here a_{ij} is the probability from state i to state j.
- Symbol emission probability B and $B = b_i(v_k)$, represents the probability emitting v_k at state i.
- π, the initial state probability.

5. Draw the model representing various states, probabilities and transitions.

4.3.1 Symbol emission probability (B)

For any symbol state $q_i \in Q_j^s$, emission probability is calculated as [202]:

$$b(q_i, v_m) = \begin{cases} 1, & if\, v_m = v_i \\ 0, & otherwise \end{cases}$$

For instance, the symbol state $q_1(D_b)$ has the probability of emitting a symbol D_b equal to 1.

The emission probability for any gap state $q_{ab} \in Q^g$ is given as [202]:

$$b(q_{ab}, v_m) = \left(\frac{freq_{ab}(v_m)}{\sum_{v_j \epsilon \sum_{ab}} freq_{ab}(v_j)} \right) \times \sigma + \frac{1}{M} \times (1 - \sigma)$$

The emission probability for any gap state $q_{ab} \in Q^g$ is given as [202]:

$$b(q_{ab}, v_m) = \left(\frac{freq_{ab}(v_m)}{\sum_{v_j \epsilon \sum_{ab}} freq_{ab}(v_j)} \right) \times \sigma + \frac{1}{M} \times (1 - \sigma)$$

Here σ is the smoothing parameter for handling the case when a symbol v_m is previously not seen in the training set. For such cases, σ is set to 0.99, which makes the other term $\left(\frac{1-\sigma}{M} = \frac{0.01}{M}\right)$ non-zero for unseen symbols in Z .

For the gap state q_6 $(D_b P)$, the frequency of the gap symbol M is 2, and the total gap symbols are 4. Therefore, the emission probability of M from state q_6 is $\frac{2}{4} \times 0.99 + \frac{0.01}{9} = 0.496.$

For a universal gap q_N , emission probability is calculated by [202]:

$$b(q_N, v_m) = \left(\frac{freq(v_m)}{\sum_{v_m \epsilon \sum} freq(v_m)}\right) \times \sigma + \frac{1}{M} \times (1-\sigma)$$

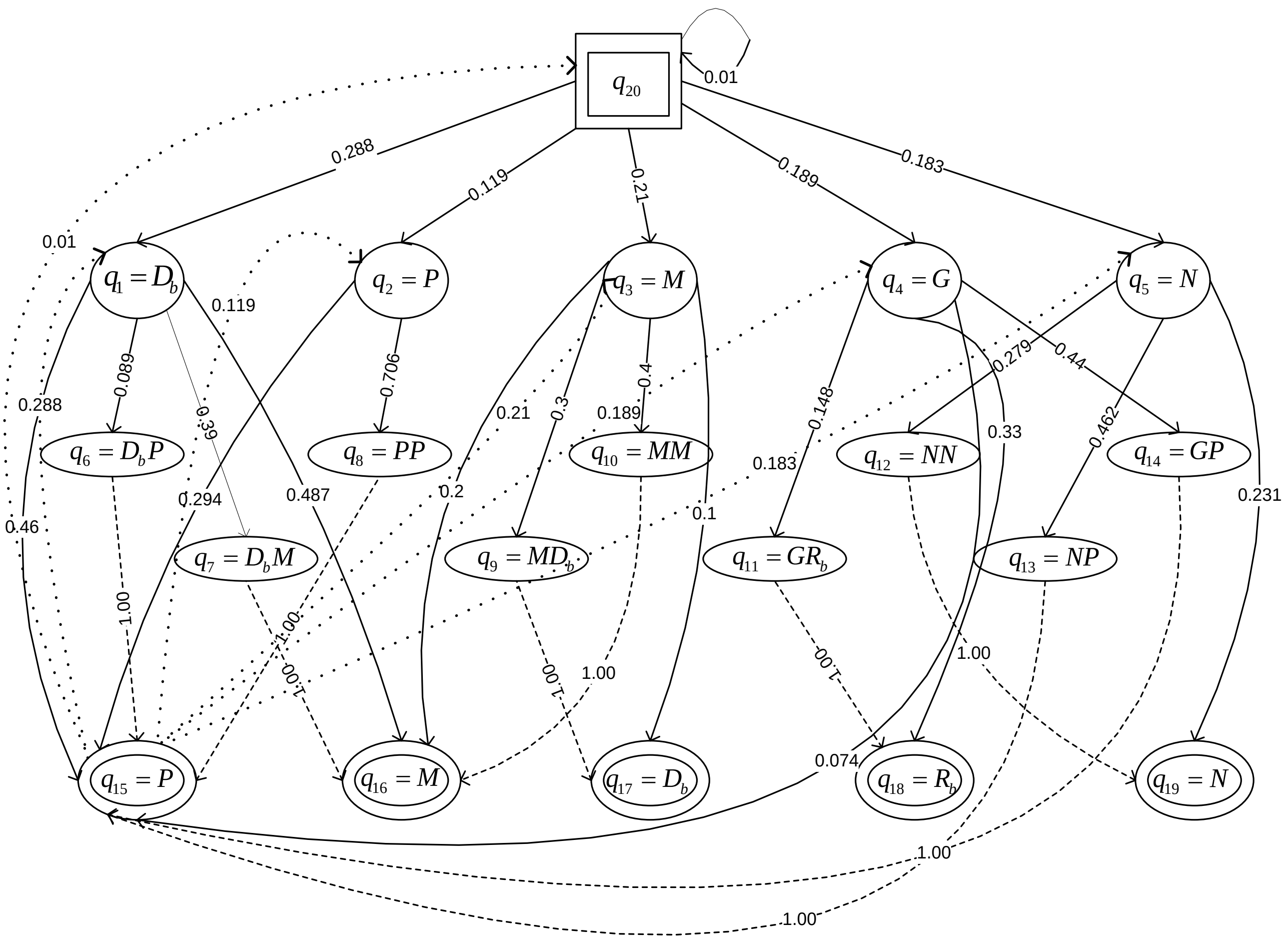

Figure 4.1: VOGUE HMM for Indian music Raga Bhairav

Where $freq(v_m)$ is the frequency of the symbol v_m. The probability of emitting S from the universal state q_{20} is $\frac{23}{405} \times 0.99 + \frac{0.01}{9} = 0.057$. The frequency of S is 23 (Table 4.1), and the total number of symbols is 405. Similarly, other state's emission probabilities are evaluated, as shown in Table 4.3.

4.3.2 The transition probability matrix (A)

The transition probability matrix (A) is calculated as [202]:

$A = \{\alpha(q_i, q_j) \mid 1 \le i, j \le N\}$ Here,

$\alpha(q_i, q_j) = \mathrm{P}(\mathrm{q}^{t+1} = q_j \mid q^t = q_i)$ is the probability of transition from state q_i to q_j. Let $\sum_{ab}$ be the symbols in between the gap of v_a and v_b for any sequence $v_a v_b \in F_2$ and $freq_{ab}(\mathrm{v}_m)$ be the frequency of the symbol v_m between v_a and v_b.

The first symbol state $(q_i \in Q_1^s)$ can take transition either to a gap state $q_j \in Q^g$ (models $g \ge 1$) or to the second symbol state $q_j \in Q_2^s$ (models $g = 0$). Let $freq(\mathrm{v}_a, \mathrm{v}_b)$ be the frequency of $v_a v_b$, and $freq(\mathrm{v}_a, \mathrm{v}_b, g)$ be the frequency of $v_a v_b$ with a gap length g.

The transition probability from the first symbol state $q_i \in Q_1^s$ is calculated by [202]:

$$\alpha(q_i, q_j) = \begin{cases} \dfrac{freq_{(v_i, v_j, 0)}}{\sum_{v_i v_{b \in F_2}} freq_{(v_i, v_b)}}, & if\, q_j \in Q_2^s \\ \dfrac{\sum_{g>0} freq_{(v_i, v_j, g)}}{\sum_{v_i v_b \in F_2} freq_{(v_i, v_b)}}, & if\, q_j \in Q^g \\ 0, & otherwise \end{cases}$$

Table 4.3: Symbol Emission Probability Matrix of VOGUE model for raga Bhairav

	S	R_b	G	M	P	D_b	N	S'	Prolongation of note(-)
q_1						1.000			
q_2					1.000				
q_3				1.000					
q_4			1.000						
q_5							1.000		
q_6	0.001	0.001	0.001	0.496	0.001	0.001	0.001	0.001	0.496
q_7	0.001	0.001	0.059	0.001	0.874	0.059	0.001	0.001	0.001
q_8	0.001	0.001	0.001	0.496	0.001	0.425	0.001	0.001	0.072

q_9	0.001	0.001	0.001	0.001	0.991	0.001	0.001	0.001	0.001
q_{10}	0.001	0.001	0.265	0.001	0.595	0.067	0.001	0.001	0.067
q_{11}	0.001	0.001	0.001	0.892	0.100	0.001	0.001	0.001	0.001
q_{12}	0.199	0.001	0.001	0.001	0.001	0.199	0.001	0.595	0.001
q_{13}	0.001	0.001	0.059	0.001	0.001	0.758	0.001	0.175	0.001
q_{14}	0.001	0.001	0.001	0.914	0.001	0.077	0.001	0.001	0.001
q_{15}					1.000				
q_{16}				1.000					
q_{17}						1.000			
q_{18}		1.000							
q_{19}							1.000		
q_{20}	0.057	0.083	0.123	0.145	0.133	0.123	0.113	0.115	0.103

First symbol states are allowed to transit to the second symbol state or the gap states. Transitions probability from the first symbol state q_1 to the second symbol state q_{15}, i.e., the transition probability from D_b to P, is $\frac{19}{41} = 0.463$. The frequency of D_bP with $g = 0$ is 19, and the frequency of D_b with other frequently occurring symbols having $g \geq 0$ is 41 (see Table 4.2 for details).

Transition probabilities from the first symbol state q_1 to a gap state q_6, i.e., probability from D_b to D_bP is $\frac{4}{41} = 0.097$. The frequency of D_bP with $g \geq 1$ is 4.

The transition probability of the gap state is calculated for $q_i = q_{ab} \in Q^g$ using [202]:

$$\alpha(q_i, q_j) = \begin{cases} 1, & if\, q_j = q_b \in Q_2^s \\ 0, & otherwise \end{cases}$$

The gap state $q_{ab} \in Q^g$ can make transitions to the second symbol state only. If the second symbol of the gap state is the same as the second symbol state, the transition probability is 1. For example, the transition from gap state q_6 $(D_b\, P)$ to q_{15} (P) gives transition probability equal to 1.

Transition probabilities from the second symbol state are computed using [202]:

$$\alpha(q_i, q_j) = \begin{cases} \sigma \times \frac{\sum_{q_b \in Q_2^s} freq_{(v_j, v_b)}}{\sum_{v_a v_{b \in F_2}} freq_{(v_a, v_b)}}, & if q_j \in Q_1^s \\ 1-\sigma, & if q_j = q_N \in Q^u \\ 0, & otherwise \end{cases}$$

Here, σ=0.99 is a smoothing parameter. The state transition to Q^u is done with a small probability of 0.01(1-σ=1-0.99). The second symbol state $q_i \in Q_2^s$ can go to either the first symbol $q_j \in Q_1^s$ (model gap 0) or to the universal gap state q_N.

The transition from the second state q_{15} (P) to the first state q_1 (D_b) is $\frac{0.99 \times 41}{141} = 0.288$. The frequency of all sequences having D_b as the first symbol in any gap state $(0 \le g \le 2)$ is 41 and 141 is the frequency of all nine gap states with $0 \le g \le 2$. The transition probability of $q_i \in Q_2^s$ to the state $q_1, q_2, q_3, q_4, q_5 \in Q_1^s$ and q_{20} is 0.288, 0.119, 0.210, 0.189, 0.183, and 0.1 respectively.

For $q_i = q_N$ (universal state), transition probabilities are estimated by [202]:

$$\alpha(q_i, q_j) = \begin{cases} \sigma \times \frac{\sum_{q_b \in Q_2^s} freq_{(v_j, v_b)}}{\sum_{v_a v_{b \in F_2}} freq_{(v_a, v_b)}}, & if q_j \in Q_1^s \\ 1-\sigma, & if q_j = q_N \\ 0, & otherwise \end{cases}$$

Universal gap state transitions are allowed either to the first symbol states or to itself.

Transition probability from universal state (q_{20}) to q_1 (D_b) is $\frac{0.99 \times 41}{141} = 0.288$. Here, the frequency of all sequences having D_b as the first note in any gap state having $0 \le g \le 2$ is 41 and 141 is the total number of gap states with $0 \le g \le 2$. Similarly, the transition probabilities for the rest of the states are evaluated (See Table 4.4 for details).

4.3.3 State duration probability (Δ)

The number of symbols emitted from a state is defined as the duration of that state. The first and second symbol states $(q_i \in Q_j^s)$ always emit only one symbol. Thus, the duration is always 1.

The matrix of state duration probability (Δ) is given by [202]: $\Delta = \{\Delta(q_i, d) \mid \mathrm{d} \in [1, \mathrm{maxgap}]\}$.

Let $q_i = q_{ab}$ be the gap state between $q_a \in Q_1^s$ and $q_b \in Q_2^s$, the state duration probabilities are calculated using [202]:

$$\Delta(q_i, d) = \begin{cases} \dfrac{freq_{(v_a, v_b, d)}}{\sum_{g>0} freq_{(v_a, v_b, g)}}, & if q_i = q_{ab} \in Q^g \\ 1, & if q_i \in Q_j^s and d = 1 \\ 0, & otherwise \end{cases}$$

Table 4.4: Transition Probability matrix of VOGUE model for raga Bhairav

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}	q_{12}	q_{13}	q_{14}	q_{15}	q_{16}	q_{17}	q_{18}	q_{19}	q_{20}
q_1						.098	.39								.463	.0487				
q_2								.706							.294					
q_3									0.3	0.4						0.2	0.1			
q_4											.148			.44	.074			.33		
q_5												.279	.462						.231	
q_6															1					
q_7																1				
q_8															1					
q_9																	1			
q_{10}																1				
q_{11}																		1		
q_{12}																			1	
q_{13}															1					
q_{14}															1					
q_{15}	.288	.119	.21	.189	.183															.01
q_{16}	.288	.119	.21	.189	.183															.01
q_{17}	.288	.119	.21	.189	.183															.01
q_{18}	.288	.119	.21	.189	.183															.01
q_{19}	.288	.119	.21	.189	.183															.01
q_{20}	.288	.119	.21	.189	.183															.01

$d = 1$ and $q_6 \in Q^g$ $(D_b P)$, the state duration probability is $\frac{4}{4} = 1$. Here 4 is the frequency of $D_b P$ with $d = 1$ and 4 in the denominator is the frequency of $D_b P$ with $g > 0$.

Table 4.5: State duration probabilities of VOGUE model for raga Bhairav

	q_6	q_7	q_8	q_9	q_{10}	q_{11}	q_{12}	q_{13}	q_{14}	Other q_i
$d = 1$	1	.875	.833	1	.916	1	.43	.667	.916	1.0
$d = 2$		.125	.167		.084		.77	.333	.084	1.0

4.3.4 The initial state probability

The probability of any state q_i at the beginning of the model is computed using: $\pi = \{\pi(i) = P(q_i \mid t = 0), 1 \le i \le N\}$

where π is calculated as [202]:

$$\pi = \begin{cases} \sigma \times \dfrac{\sum_{q_j \in Q_2^s} freq_{(v_i, v_j)}}{\sum_{v_a v_{b \in F_2}} freq_{(v_a, v_b)}}, & if q_j \in Q_1^s \\ 1 - \sigma, & if q_j = q_N \in Q^u \\ 0, & otherwise \end{cases}$$

Initial state probability of initial probability of q_1 (D_b) is $0.99 \times \frac{41}{141} = 0.288$. The frequency of D_b with second symbol states is 41 and the frequency of all subsequences of length 2 is 141.

Table 4.6: Initial state probabilities (π) of VOGUE model for raga Bhairav

States	q_1	q_2	q_3	q_4	q_5	q_{20}
π	.288	.119	.210	.19	.189	.01

4.4 CONCLUSION

Many real-world problems like bioinformatics, text mining, and Web access use data with variable range dependencies. First-order and higher-order HMM cannot handle sequences separated by different length gaps. VOGUE considers all such non-consecutive sequences with gap symbols and durations. The chapter discusses the implementation of VOGUE for modeling Indian music. VOGUE merges two techniques to generate the model: pattern mining and data modeling. In the future, we can apply VOGUE to a larger musical sequence database and increase the maximum length of sub-sequences $(k > 2)$ to model sequences with larger dependencies.

CHAPTER 5

MATHEMATICAL MODELING OF INDIAN TALA'S KAIDAS AND PALTAS USING FORMAL GRAMMAR

5.1 INTRODUCTION

Indian classical music includes two chief elements: Raga and Tala. Raga is used for the melody of music, and Tala is the rhythm in Indian classical music. Rhythm is the fundamental need for creating any musical system. Tala and its improvised patterns hold a critical position in the Indian music system. Tala's improvised patterns are kaidas and paltas generated by applying various permutations and combinations. Cross-serial dependency is analyzed in the paltas of a Tala's kaida. There is a need to analyze the cross-serial dependencies in the patterns of Tala's kaida and the formal modeling of these patterns. The implementation of these mathematical models can be used to check the accuracy of Tala's kaida.

Indian classical music is the most rhythmic and sophisticated music in the world; still, the research on Indian music in recent times is limited and usually superficial. Such factors agitate the need for analytical and computational research on the rhythmic organization of Indian music. Researchers have done a lot of work on Indian ragas, but little research has been carried out with Talas. The work of [29, 32, 31, 71-72] motivated us to design formal grammar and deep pushdown automata for recognizing Tala and its improvised musical patterns in Indian music.

5.2 PRELIMINARIES

Tala: Tala means 'taali' or 'clap'. Talas are recurring fixed-length patterns used in Indian music to control the rhythm with handclaps [32]. Tabla (a percussion musical instrument) is used instead of the handclaps in a musical Tala performance. The whole period of a cycle (avartan) is a collection of the smallest time beat (matras). The beats are arranged into sections (vibhags) of different lengths (Figure 5.1).

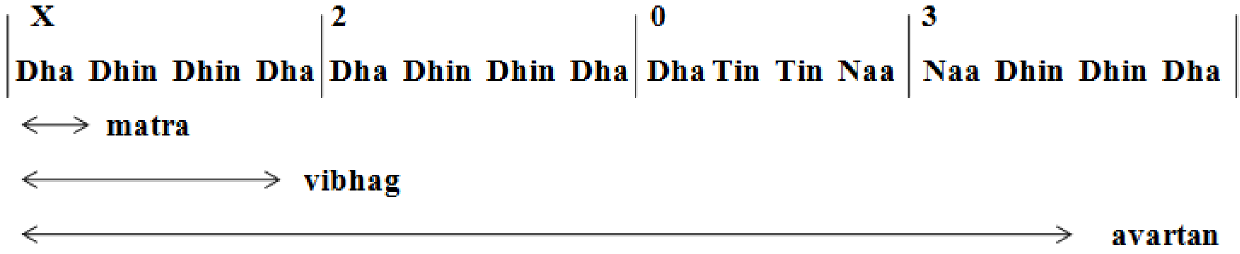

Figure 5.1: Example of matra, vibhag and avartan.

Vibhags are represented through the gesture of hands by a khaali (wave), a taali (clap), and by the movement of fingers [203-204]. The beginning matra of anavartan in Tala is known as sam, which creates a resolution point [205]. Khaali is another important beat in Tala, which means 'empty'. Each Tala has a defined 'theka', a set of onomatopoeic syllables (bols) through which a Tala can be recognized [206]. The bols (dha, dhin, naa, ge, tu, etc.) are used in Indian music to describe Tala. Figure 1, X represents sam, 0 represents khaali and $2, 3,$ are taali.

Kaida and Palta: Kaida is a rhythmic theme with a structured sequence of bols derived from basic Talas. Kaida should maintain the taali and khaali structure similar to the underlying Tala [207]. The fundamentals of musical composition and improvisation in playing tabla sequences are kaida and palta [29]. From the kaidas, paltas (variations) are created. The word palta means to alter or to reverse; they are formed by applying various improvisations through permutation and combinations on a particular kaida. The bols in kaida are substituted by some or many bols of kaida [207].

5.3REPRESENTATION OF CROSS-SERIAL DEPENDENCY IN PALTA

The kaida and palta use various permutations and substitutions as music improvisation tools [29]. Many new musical strings (paltas) can be derived from a given kaida. Our work is to propose a generalized model that expresses fundamental procedures for every possible palta derived from kaida. The research revolves around representing these musical improvisations with state grammar and DPDA. We will focus on Teentala having 16 matras. These sixteen matras are divided into four vibhags, each further having four matras. The claps are on matra 1st (sam), 5th (the beginning of the second vibhag), and 13th (the beginning of the last vibhag). Matra 9th is khaali [208] (Figure 5.2).

Matra	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Taali/Khaali	X				2				0				3			
Bols	Dha	Dhin	Dhin	Dha	Dha	Dhin	Dhin	Dha	Dha	Tin	Tin	Naa	Naa	Dhin	Dhin	Dha

Figure 5.2: Theka of Teentala

Like other diverse art forms, Indian musicians focus on conveying emotions and producing aesthetic aspects [209]. The paltas are composed of musicians that fulfill the musical rules and aesthetic features. Improvisation of Indian tabla sequences tends to obey a defined system. The rules are not explicitly specified as a natural language [29]. Every kaida shows its likelihood of creating paltas. The paltas are composed less systematically as they have no strict rules. The few commonly used flexible approaches for generating paltas [32, 210].

1. Palta should only contain bols from its kaida.

2. Palta should begin with dora (two bols on a single beat like dhadha, tirkit, etc).
3. Palta should contain part of the original kaida at the end of palta.
4. Palta should contain cross-serial dependency in between bols [31-32]. The cross-serial dependency occurs when lines defining the dependency relation between the sequences of words cross over each other [210].
5. Palta consists of permutation, substitution, and arranging of bols from kaida.
6. Palta must exhibit the taali/khaali structure.

We had considered five different kaidas of Teentala and two paltas of each kaida. We analyzed the patterns in paltas to represent improvised tabla sequences with Deep pushdown automata.

Kaida

X	2	0	3
GeGe TeTe GeGe TeTe	**KeKe TeTe KeKe TeTe**	**GeGe TeTe GeGe TeTe**	**KeKe TeTe KeKe TeTe**

Palta 1

X	2	0	3
GeGe TeTe TeTe GeGe	**TeTe TeTe GeGe TeTe**	**KeKe TeTe TeTe KeKe**	**TeTe TeTe GeGe TeTe**

Palta 2

X	2	0	3
GeGe TeTe TeTe TeTe	**GeGe TeTe GeGe TeTe**	**KeKe TeTe TeTe TeTe**	**GeGe TeTe GeGe TeTe**

Figure 5.3: Kaida 1 and its palta

The kaida and paltas of Teentala have taali on 1^{st}, $5^{th}$and 13^{th}matra and khaali on 9^{th}matra. Paltas of kaida 1, kaida 2 and kaida 3 (Figure 5.3, 5.4 and 5.5 respectively) have 16 matras with taali and khaali at the same location as in their respective kaidas. The bols in $2^{nd}$and 4^{th}vibhag of a palta will be the same.

Kaida

X	2	0	3
Dha Dha Te Te	**Dha Dha Tu Na**	**Ta Ta Te Te**	**Dha Dha Tu Na**

Palta 1

X	2	0	3
DhaDha TeTeTeTe DhaDha TeTe	**DhaDha TeTeTeTe DhaDha TuNaTuNa**	**TaTaTaTa TeTe TaTa TeTe**	**DhaDha TeTeTeTe DhaDha TuNaTuNa**

Palta 2

X	2	0	3
DhaDha TeTe TeTe TeTe	**DhaDhaDhaDha TeTe DhaDha TuNa**	**TaTaTaTa TeTe TeTe TeTe**	**DhaDhaDhaDha TeTe DhaDha TuNa**

Figure 5.4: Kaida 2 and its palta

The cross-serial dependency (rule 4) can be seen in bols of palta in between matra 5^{th} to 8^{th} and 13^{th} to 16^{th}, respectively, as shown in Figure 5.6.

Kaida

X	2	0	3
Dha Dha Tir Kit	Dha Dha Tu Na	Ta Ta Tir Kit	Dha Dha Tu Na

Palta 1

X	2	0	3
DhaDha TirKit DhaDha TirKitTirKit	DhaDha TirKit DhaDha TuNaTuNa	TaTa TirKit TaTa TirKit	DhaDha TirKit DhaDha TuNaTuNa

Palta 2

X	2	0	3
TirKit TirKit DhaTir KitDha	DhaDha TirKit DhaDha TuNa	TirKit TirKit TaTir KitTa	DhaDha TirKit DhaDha TuNa

Figure 5.5: Kaida 3 and its palta

X	2	0	3
GeGe TeTe TeTe GeGe	TeTe TeTe GeGe TeTe	KeKe TeTe TeTe KeKe	TeTe TeTe GeGe TeTe

Figure 5.6: Cross-serial dependency in palta with 16 matras

In Figure 5.3, the matra 5^{th} and 13^{th} have the same bol (Te) with equal repetition (2 repetitions $TeTe$). The bols having cross-dependency with each other should be the same in every aspect.

Kaida

X	2	0	3
DhaTi TaaDha TiTaa DhaDha	TiTaa DhaDha TuNa KaTa	TaTi TaaDha TiTaa DhaDha	TiTaa DhaDha TuNa KaTa

Palta 1

X	2	0	3
DhaTi TaaDha TiTaa DhaDha	DhaTi TaaDha TiTaa DhaDha	DhaTi TaaDha TiTaa DhaDha	TiTaaTiTaa DhaDha TuNa KaTa

X	2	0	3
TaTi TaaTa TiTaa TaTa	TaTi TaaTi TiTaa DhaDha	DhaTi TaaDha TiTaa DhaDhaDhaDha	TiTaaTiTaa DhaDha TuNa KaTa

Palta 2

X	2	0	3
DhaTi TaaDha TiTaa DhaDha	TiTaa DhaDha TuNa KaTaKaTa	TiTaa TiTaa DhaDhaDhaDha TiTaa	TiTaa DhaDha TuNaTuNa KaTaKaTa

X	2	0	3
TaTi TaaTa TiTaa DhaDha	TiTaa DhaDha TuNaTuNa KaTa	TiTaa TiTaa DhaDhaDhaDha TiTaa	TiTaa DhaDha TuNaTuNa KaTaKaTa

Figure 5.7: Kaida 4 and its palta

Palta composition entirely depends on the tabla player; it can have 16 or 32 matras (Figure 5.7, 5.8). Nevertheless, the rules designed for creating paltas must be executed. In palta with 32 matras, taali is present on 1^{st}, 5^{th}, 13^{th}, 17^{th}, 21^{st} and 29^{th} matra whereas khaali is on 9^{th} and 25^{th} matras. Palta with 32 matras exhibits cross-serial dependency in between matras of vibhag 4^{th} and 8^{th} respectively.

Kaida

X	2	0	3
DhaTi DhaGe NaDha Tirkit	DhaTi DhaGe DhinNa GinNa	TaTi TaKe NaTa TirKit	DhaTi DhaGe DhinNa GinNa

Palta 1

X	2	0	3
DhaTi DhaGe NaDha Tirkit	DhaTi DhaGe NaDha TirKitTirKit	DhaTi DhaGe NaDha TirKitTirkit	DhaTiDhaTi DhaGeDhaGe DhinNa GinNa

X	2	0	3
TaTi TaKe NaTa TirKitTirKit	TaTi TaKe NaTa TirKit	DhaTiDhaTi DhaGeDhaGe NaDha TirKit	DhaTiDhaTi DhaGeDhaGe DhinNa GinNa

Palta 2

X	2	0	3
DhaTi DhaGe NaDha Tirkit	TiDha Tirkit TiDha TirKit	DhaTi DhaGe NaDha TirKit	DhaTi DhaGe DhinNa GinNa

X	2	0	3
TaTi TaKe NaTa TirKit	TiTa TirKit TiTa TirKit	DhaTi DhaGe NaDha TirKit	DhaTi DhaGe DhinNa GinNa

Figure 5.8: Kaida 5 and its palta

The matra having a cross-serial dependency with the respective matra must have the same bol and should be repeated the same number of times (Figure 5.6). The matras in 1st (y) and 3rd vibhag (z) of palta can have any combination of bols from the kaida, but the bol on matra 5th and 13th; 6th and 14th; 7th and 15th; 8th, and 16th should be the same (Figure 5.9). Paltas can be considered as a language $L = \{ya^m b^n c^o d^p za^m b^n c^o d^p \mid m,n,o,p \geq 1\}$. Palta pattern y, z can be represented by null as they did not represent a fixed pattern. Thus, a non-context-free language $L' = \{a^m b^n c^o d^p a^m b^n c^o d^p \mid m,n,o,p \geq 1\}$ is observed.

X	2	0	3
DhaDha TeTeTeTe DhaDha TeTe	DhaDha TeTeTeTe DhaDha TuNaTuNa	TaTaTaTa TeTe TaTa TeTe	DhaDha TeTeTeTe DhaDha TuNaTuNa
←y→	←a^m→ ←b^n→ ←c^o→ ←d^p→	←z→	←a^m→ ←b^n→ ←c^o→ ←d^p→

Figure 5.9: Language observed in improvised musical patterns (palta)

5.4 REPRESENTATION OF INDIAN MUSIC USING STATE GRAMMAR

We have generated state grammar for all the paltas of Teentala discussed above (Section 5.3).

$G_1 = (\{S, X, Y, Dha, Te, TuNa, Ge, TiTaa, KaTa, DhaTi, DhaGe, DhinNa, GinNa, TirKit\},$

$q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}, q_{16}, q_{17}, q_{18}, q_{19}, q_{20}, q_{21}, q_{22}, q_{23}, q_{24}, q_{25}, q_{26}, q_{27}, q_{28}, q_{29}, q_{30}, q_{31}, q_{32}, q_{33}, q_{34}, q_{35}, q_{36}, q_{37}, q_{38}\},$

$\{Dha, Te, TuNa, Ge, TiTaa, KaTa, DhaTi, DhaGe, DhinNa, GinNa, TirKit\}, P, S)$ be the state grammar for the Teentala paltas with production rules, P as follows:

1. $(q_0, \mathrm{s}) \rightarrow (\mathrm{q}_0, XY)$ 2. $(q_0, X) \rightarrow (q_1, DhaX)$ 3. $(q_1, Y) \rightarrow (q_2, DhaY)$

4. $(q_2, X) \rightarrow (q_1, DhaX)$ 5. $(q_2, X) \rightarrow (q_3, TeX)$ 6. $(q_3, Y) \rightarrow (q_4, TeY)$

7. $(q_4, X) \rightarrow (q_3, TeX)$ 8. $(q_4, X) \rightarrow (q_5, DhaX)$ 9. $(q_5, Y) \rightarrow (q_6, DhaY)$

10. $(q_6, X) \rightarrow (q_5, DhaX)$ 11. $(q_6, X) \rightarrow (q_7, TuNaX)$ 12. $(q_7, Y) \rightarrow (q_8, TuNaY)$

13. $(q_8, X) \rightarrow (q_7, TuNaX)$ 14. $(q_8, X) \rightarrow (q_9, TuNa)$ 15. $(q_6, X) \rightarrow (q_9, TuNa)$

16. $(q_9, Y) \rightarrow (q_9, TuNa)$

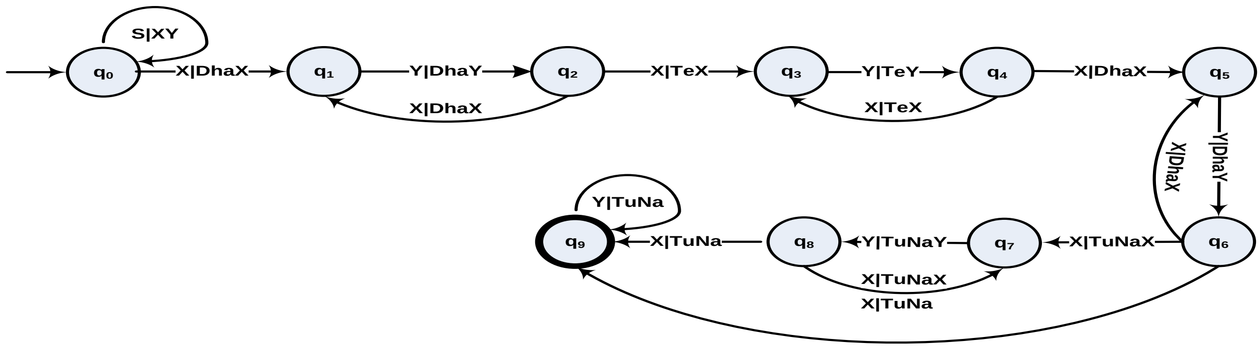

Figure 5.10: Deep pushdown automaton for paltas of kaida 2 (Figure 5.4)

We designed a DPDA (Figure 5.10) for paltas of kaida 2 shown in Figure 5.4. We further added production rules for cascading other values of a, b, c, d in the final DPDA:

$a = TeTe, GeGe, DhaDha, TiTaa, DhaTi$ $b = TeTe, TirKit, DhaDha, DhaGe$

$c = GeGe, DhaDha, TuNa, DhinNa$ $d = TeTe, TuNa, KaTa, GinNa$

The following production rules can generate all the strings of paltas of Teentala discussed.

17. $(q_2, X) \rightarrow (q_{10}, TirKitX)$ 18. $(q_{10}, Y) \rightarrow (q_{11}, TirKitY)$ 19. $(q_{11}, X) \rightarrow (q_{10}, TirKitX)$

20. $(q_{11}, X) \rightarrow (q_5, DhaX)$ 21. $(q_0, X) \rightarrow (q_{12}, GeX)$ 22. $(q_{12}, Y) \rightarrow (q_{13}, GeY)$

23. $(q_{13}, X) \rightarrow (q_{12}, GeX)$ 24. $(q_{13}, X) \rightarrow (q_{12}, TeX)$ 25. $(q_0, X) \rightarrow (q_{14}, TeX)$

26. $(q_{14}, Y) \rightarrow (q_{15}, TeY)$ 27. $(q_{15}, X) \rightarrow (q_{14}, TeX)$ 28. $(q_{15}, X) \rightarrow (q_{16}, GeX)$

29. $(q_{16}, Y) \rightarrow (q_{17}, GeY)$ 30. $(q_{17}, X) \rightarrow (q_{16}, GeX)$ 31. $(q_{17}, X) \rightarrow (q_{18}, TeX)$

32. $(q_{18}, Y) \rightarrow (q_{19}, TeY)$ 33. $(q_{19}, X) \rightarrow (q_{18}, TeX)$ 34. $(q_{19}, X) \rightarrow (q_{20}, Te)$

35. $(q_{17}, X) \rightarrow (q_{20}, Te)$ 36. $(q_{20}, Y) \rightarrow (q_{20}, Te)$ 37. $(q_0, X) \rightarrow (q_{21}, TiTaaX)$

38. $(q_{21}, Y) \rightarrow (q_{22}, TiTaaY)$ 39. $(q_{22}, X) \rightarrow (q_{21}, TiTaaX)$ 40. $(q_{22}, X) \rightarrow (q_{23}, DhaX)$

41. $(q_{23}, Y) \rightarrow (q_{24}, DhaY)$ 42. $(q_{24}, X) \rightarrow (q_{23}, DhaX)$ 43. $(q_{24}, X) \rightarrow (q_{25}, TuNaX)$

44. $(q_{25}, Y) \rightarrow (q_{26}, TuNaY)$ 45. $(q_{26}, X) \rightarrow (q_{25}, TuNaX)$ 46. $(q_{26}, X) \rightarrow (q_{27}, KaTaX)$

47. $(q_{27}, Y) \rightarrow (q_{28}, KaTaY)$ 48. $(q_{28}, X) \rightarrow (q_{27}, KaTaX)$ 49. $(q_{28}, X) \rightarrow (q_{29}, KaTa)$

50. $(q_{29}, Y) \rightarrow (q_{29}, KaTa)$ 51. $(q_{26}, X) \rightarrow (q_{29}, KaTa)$ 52. $(q_0, X) \rightarrow (q_{30}, DhaTiX)$

53. $(q_{30}, Y) \rightarrow (q_{31}, DhaTiY)$ 54. $(q_{31}, X) \rightarrow (q_{30}, DhaTiX)$ 55. $(q_{31}, X) \rightarrow (q_{32}, DhaGeX)$

56. $(q_{32}, Y) \rightarrow (q_{33}, DhaGeY)$ 57. $(q_{33}, X) \rightarrow (q_{32}, DhaGeX)$ 58. $(q_{33}, X) \rightarrow (q_{34}, DhinNaX)$

59. $(q_{34}, Y) \rightarrow (q_{35}, DhinNaY)$ 60. $(q_{35}, X) \rightarrow (q_{34}, DhinNaX)$ 61. $(q_{35}, X) \rightarrow (q_{36}, GinNaX)$

62. $(q_{36}, Y) \rightarrow (q_{37}, GinNaY)$ 63. $(q_{37}, X) \rightarrow (q_{36}, GinNaX)$ 64. $(q_{37}, X) \rightarrow (q_{38}, GinNa)$

65. $(q_{38}, Y) \rightarrow (q_{38}, GinNa)$ 66. $(q_{35}, X) \rightarrow (q_{38}, GinNa)$

5.5 MATHEMATICAL MODELING OF INDIAN MUSIC PALTA'S USING DEEP PUSHDOWN AUTOMATON

Figure 5.11 represents a deep pushdown automaton for the counterpart of state grammar for representing music strings found in paltas.

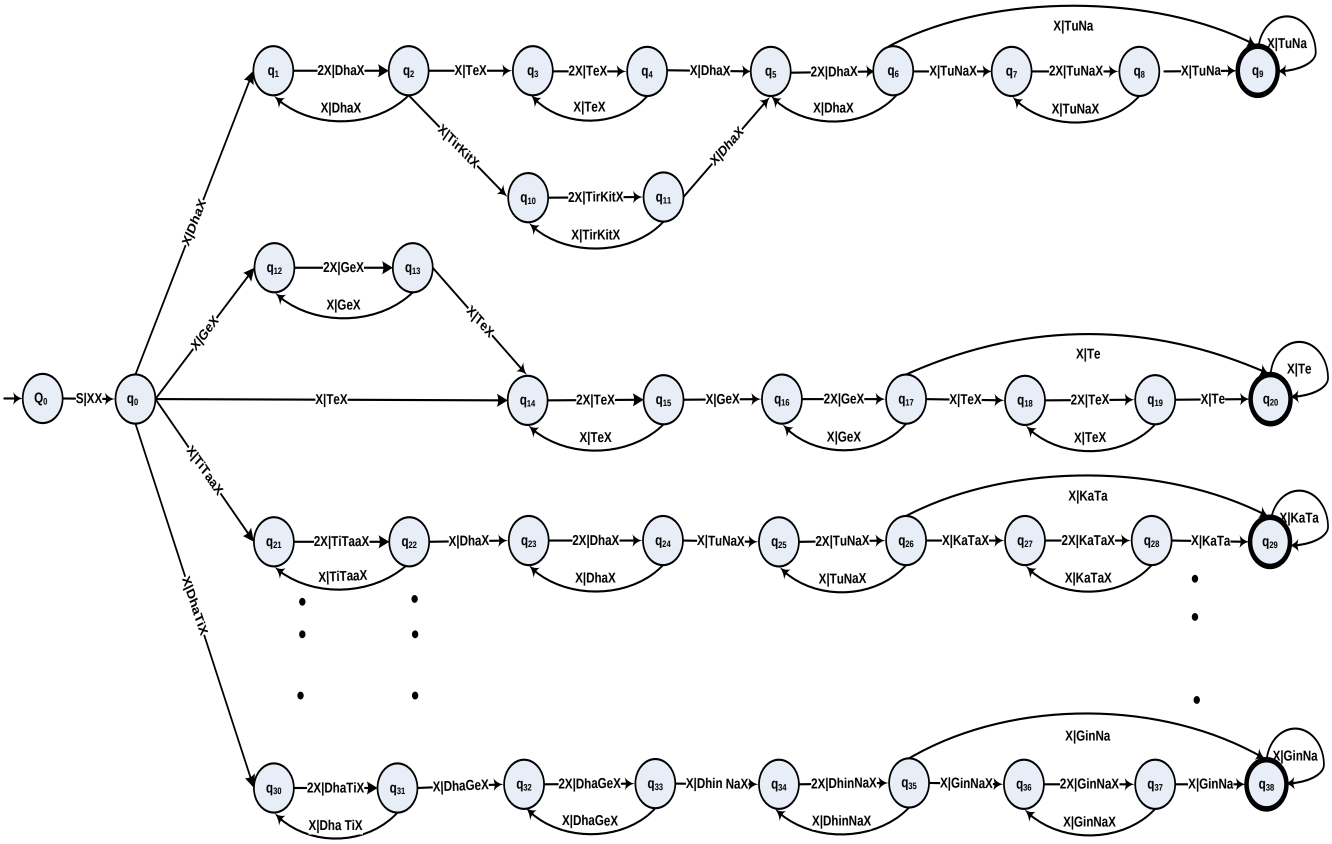

Figure 5.11: Deep Pushdown Automaton for improvised musical patterns

$M = (\{\mathrm{Q}_0, \mathrm{q}_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}, \mathrm{q}_{16}, q_{17}, q_{18}, q_{19}, q_{20}, q_{21}, q_{22}, q_{23}, q_{24}, q_{25},$

$q_{26}, q_{27}, q_{28}, q_{29}, q_{30}, q_{31}, q_{32}, q_{33}, q_{34}, q_{35}, q_{36}, q_{37,} q_{38}\}$, $\{Dha, Te, TuNa, Ge, TiTaa, KaTa, DhaTi, DhaGe,$

$DhinNa, GinNa, TirKit\}, \{X, S, \#\}, R, Q_0, S_{,} \{\mathrm{q}_9, q_{20}, \mathrm{q}_{29}, \mathrm{q}_{38}\})$

For DPDA depicted in Figure 5.11, the production rules (R) are as follows:

$1Q_0S \rightarrow q_0XX$ $1q_0X \rightarrow q_1DhaX$ $2q_1X \rightarrow q_2DhaX$

$1q_2X \rightarrow q_1DhaX$ $1q_2X \rightarrow q_3TeX$ $2q_3X \rightarrow q_4TeX$

$1q_4X \rightarrow q_3TeX$ $1q_4X \rightarrow q_5DhaX$ $2q_5X \rightarrow q_6DhaX$

$1q_6X \rightarrow q_5DhaX$ $1q_6X \rightarrow q_7TuNaX$ $2q_7X \rightarrow q_8TuNaX$

$1q_8X \to q_7TuNaX$ $1q_8X \to q_9TuNa$ $q_9X \to q_9TuNa$

$1q_6X \to q_9TuNa$ $1q_2X \to q_{10}TirKitX$ $2q_{10}X \to q_{11}TirKitX$

$1q_{11}X \to q_{10}TirKitX$ $1q_0X \to q_{12}GeX$ $2q_{12}X \to q_{13}GeX$

$1q_{13}X \to q_{12}GeX$ $1q_{13}X \to q_{14}TeX$ $1q_0X \to q_{14}TeX$

$2q_{14}X \to q_{15}TeX$ $1q_{15}X \to q_{14}TeX$ $1q_{15}X \to q_{16}GeX$

$2q_{16}X \to q_{17}GeX$ $1q_{17}X \to q_{16}GeX$ $1q_{17}X \to q_{18}TeX$

$2q_{18}X \to q_{19}TeX$ $1q_{19}X \to q_{20}Te$ $1q_{17}X \to q_{20}Te$

$q_{20}X \to q_{20}Te$ $1q_0X \to q_{21}TiTaaX$ $2q_{21}X \to q_{22}TiTaaX$

$1q_{22}X \to q_{21}TiTaaX$ $1q_{22}X \to q_{23}DhaX$ $2q_{23}X \to q_{24}DhaX$

$1q_{24}X \to q_{23}DhaX$ $1q_{24}X \to q_{25}TuNaX$ $2q_{25}X \to q_{26}TuNaX$

$1q_{26}X \to q_{25}TuNaX$ $1q_{26}X \to q_{27}KaTaX$ $2q_{27}X \to q_{28}KaTaX$

$1q_{28}X \to q_{27}KaTaX$ $1q_{28}X \to q_{29}KaTa$ $q_{29}X \to q_{29}KaTa$

$1q_{26}X \to q_{29}KaTa$ $1q_0X \to q_{30}DhaTiX$ $2q_{30}X \to q_{31}DhaTiX$

$1q_{31}X \to q_{30}DhaTiX$ $1q_{31}X \to q_{32}DhaGeX$ $2q_{32}X \to q_{33}DhaGeX$

$1q_{33}X \to q_{32}DhaGeX$ $1q_{33}X \to q_{34}DhinNaX$ $2q_{34}X \to q_{35}DhinNaX$

$1q_{35}X \to q_{34}DinNaX$ $1q_{35}X \to q_{36}GinNaX$ $2q_{36}X \to q_{37}GinNaX$

$1q_{37}X \to q_{36}GinNaX$ $1q_{37}X \to q_{38}GinNa$ $q_{38}X \to q_{38}GinNa$

$q_{35}X \to q_{38}GinNa$

The musical string $S^{'} = TiTaaTiTaaDhaDhaTuNaKaTaTiTaaTiTaaDhaDhaTuNaKaTa$ can be seen as $S^{'} = ww$, where $w = \{TiTaa^m Dha^n TuNa^o KaTa^p \mid m,n = 2; o,p = 1\}$. The acceptance of this string with DPDA can be checked by performing pop and expand operations.

The input accepted string $S^{'} = TiTaaTiTaaDhaDhaTuNaKaTaTiTaaTiTaaDhaDhaTuNaKaTa$

$(Q_0, TiTaaTiTaaDhaDhaTuNaKaTaTiTaaTiTaaDhaDhaTuNaKaTa, S\#)$

$\Rightarrow_e (q_0, TiTaaTiTaaDhaDhaTuNaKaTaTiTaaTiTaaDhaDhaTuNaKaTa, XX\#)$
$[1Q_0S \to q_0XX]$

$\Rightarrow_e (q_{21}, TiTaaTiTaaDhaDhaTuNaKaTaTiTaaTiTaaDhaDhaTuNaKaTa, TiTaaXX\#)$
$[1q_0X \to q_{21}$

$TiTaaX]$

$\Rightarrow_p (q_{21}, TiTaaDhaDhaTuNaKaTaTiTaaTiTaaDhaDhaTuNaKaTa, XX\,\#)$

$\Rightarrow_e (q_{22}, TiTaaDhaDhaTuNaKaTaTiTaaTiTaaDhaDhaTuNaKaTa, XTiTaaX\,\#)$
$[2q_{21}X \rightarrow q_{22}Ti$

$TaaX]$

$\Rightarrow_e (q_{21}, TiTaaDhaDhaTuNaKaTaTiTaaTiTaaDhaDhaTuNaKaTa, TiTaaXTiTaaX\,\#)$
$[1q_{22}X \rightarrow q_{21}$

$TiTaaX]$

$\Rightarrow_p (q_{21}, DhaDhaTuNaKaTaTiTaaTiTaaDhaDhaTuNaKaTa, XTiTaaX\,\#)$

$\Rightarrow_e (q_{22}, DhaDhaTuNaKaTaTiTaaTiTaaDhaDhaTuNaKaTa, XTiTaaTiTaaX\,\#)$
$[2q_{21}X \rightarrow q_{22}Ti$

$TaaX]$

$\Rightarrow_e (q_{23}, DhaDhaTuNaKaTaTiTaaTiTaaDhaDhaTuNaKaTa, DhaXTiTaaTiTaaX\,\#)$
$[1q_{22}X \rightarrow q_{23}$

$DhaX]$

$\Rightarrow_p (q_{23}, DhaTuNaKaTaTiTaaTiTaaDhaDhaTuNaKaTa, XTiTaaTiTaaX\,\#)$

$\Rightarrow_e (q_{24}, DhaTuNaKaTaTiTaaTiTaaDhaDhaTuNaKaTa, XTiTaaTiTaaDhaX\,\#)$
$[2\,\mathrm{q}_{23}\,X \rightarrow q_{24}Dha$

$X]$

$\Rightarrow_e (q_{23}, DhaTuNaKaTaTiTaaTiTaaDhaDhaTuNaKaTa, DhaXTiTaaTiTaaDhaX\,\#)$
$[1q_{24}X \rightarrow q_{23}$

$DhaX]$

$\Rightarrow_p (q_{23}, TuNaKaTaTiTaaTiTaaDhaDhaTuNaKaTa, XTiTaaTiTaaDhaX\,\#)$

$\Rightarrow_e (q_{24}, TuNaKaTaTiTaaTiTaaDhaDhaTuNaKaTa, XTiTaaTiTaaDhaDhaX\,\#)$
$[2q_{23}X \rightarrow q_{24}Dha$

$X]$

$\Rightarrow_e (q_{25}, TuNaKaTaTiTaaTiTaaDhaDhaTuNaKaTa, TuNaXTiTaaTiTaaDhaDhaX\,\#)$
$[1q_{24}X \rightarrow q_{25}$

$TuNaX]$

$\Rightarrow_p (q_{25}, KaTaTiTaaTiTaaDhaDhaTuNaKaTa, XTiTaaTiTaaDhaDhaX\,\#)$

$\Rightarrow_e (q_{26}, KaTaTiTaaTiTaaDhaDhaTuNaKaTa, XTiTaaTiTaaDhaDhaTuNaX\,\#)$
$[2q_{25}X \rightarrow q_{26}Tu$

$NaX\,]$

$\Rightarrow_e (q_{29}, KaTaTiTaaTiTaaDhaDhaTuNaKaTa, KaTaTiTaaTiTaaDhaDhaTuNaX\,\#)$
$[1q_{26}X \rightarrow q_{29}$

$KaTa]$

$\Rightarrow_p (q_{29}, TiTaaTiTaaDhaDhaTuNaKaTa, TiTaaTiTaaDhaDhaTuNaX\,\#)$
$\Rightarrow_p (q_{29}, TiTaaDhaDha$

$TuNaKaTa, TiTaaDhaDhaTuNaX\,\#) \Rightarrow_p (q_{29}, DhaDhaTuNaKaTa, DhaDhaTuNaX\,\#)$
$\Rightarrow_p (q_{29},$

$DhaTuNaKaTa, DhaTuNaX\,\#) \Rightarrow_p (q_{29}, TuNaKaTa, TuNaX\,\#) \Rightarrow_p (q_{29}, KaTa, X\,\#) \Rightarrow_e (q_{29},$

$KaTa, KaTa\,\#)\ [q_{29}X \rightarrow q_{29}KaTa] \Rightarrow_p (q_{29}, \wedge, \#)$

5.6 CONCLUSION

In the field of computational musicology, limited work has been dedicated to the analysis of Tala patterns in Indian music. This chapter deals with the structure of Tala and its paltas (variations) for analyzing the patterns in improvised music. The palta of a Tala’s kaida has cross-serial dependencies in its structures, and such patterns cannot be represented using context-free grammar. We have represented these patterns using state grammar. Further, we modeled Teentala’s kaidas and paltas using a deep pushdown automaton. This chapter attempted to show that formal grammar can be exploited to represent highly improvised musical structures in Indian tabla music. This approach can be used to check the accuracy of kaidas and paltas patterns in Indian music.

CHAPTER 6

CLASSIFICATION OF RAGA USING MACHINE LEARNING

6.1 Introduction

Traditional Indian society has always been pivotal to music in its customs, celebrations, and spirituality. The culture has always revolved around music production and other forms of entertainment. Therefore, there has always been an expert class of musicians who have continued to pass the generational heritage of the style and the tone of Indian music throughout the years in its original and modified forms. With a population of about 1.42 billion people, Indian society is diverse in terms of cultural heritage and the type of music that they play. The music and songs in Indian traditional society are often classified based on the raga to which they belong. Raga is an important aspect of a song, and it determines specifically how and when a song is played. After noting the importance of the raga and its great significance in Indian society, it is also interesting to note that traditional Indian society has always embodied the concept of raga specialists. These experts have the knowledge and experience to differentiate between these ragas and then eventually identify one from another based on the context of the song and the play patterns. These expert musicians have the onus to preserve the indigenous traditional knowledge of various forms of music in diverse societies. For many years, people believed that there could never be a system so efficient that it could eventually classify the songs into an appropriate raga based solely on studying the bits and other related patterns in the song.

With little advancement in computer technologies, such as machine learning and neural networks, the ability of a computer or any other machine to predict raga with substantial accuracy was highly improbable. Indeed, highly paid raga experts took advantage of the situation to cash in and enrich them while technologies such as machine learning and neural networks remained elusive. Over the recent years, however, a palpable shift in favor of machine learning and artificial intelligence has suddenly ignited a new debate about whether it can achieve greater precision and accuracy in raga detection through leveraging cutting-edge technologies. We discuss such a possibility and show how it is possible to use advanced computer sciences to detect the raga of a given creative piece efficiently through machine learning.

6.2 Feature Engineering And Data Preprocessing

Within the scope of this experiment, our foremost objective is to identify the most predictive features within the songs, assemble them, and harness their power for raga prediction. The one and most important tool in the entire process was Librosa. Librosa is used for primary data extraction and feature engineering packages. Librosa is a suitable choice for modeling and data preparation due to its versatility and the ability to scale well.

For this experiment, the most important part of the investigation was finding the song's most predictive features. The features selected include chromo-gram power spectrogram (chroma_strf,), spectral centroid (spec_cent), spectral bandwidth (spec_bw), rolloff, zero crossing rate (zcr), mel-spectrogram and root mean square of the spectrogram (rmse). These features are selected as they define the tonal nuances and distinct ragas present in the music. Consequently, our machine learning models' training and the resulting mappings are exclusively built upon these carefully curate features.

6.2.1 Evaluating Feature Importance

A pivotal aspect of the feature engineering process is assessing each feature's relative importance, particularly in terms of its predictive ability. Features that exhibit weak or negligible correlations with the target variable hold minimal sway in shaping the final model's design and development. In light of this, our study strongly emphasizes a thorough investigation into the relative importance of these features within the dataset and their contributions to the overall model-building process.

It is paramount to acknowledge that different features boast varying degrees of predictive potential. As illustrated in Figure 6.1, no feature registers an absolute lack of predictive value. Therefore, no compelling reason exists to eliminate any feature based solely on its predictive capabilities.

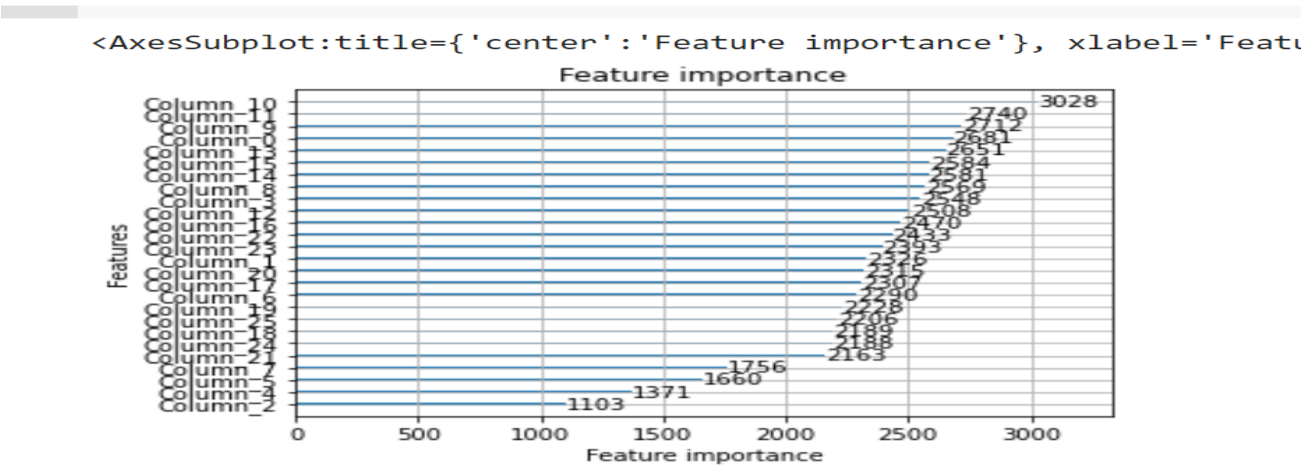

Figure 6.1: Feature importance of dataset

6.3 Proposed Model Building

The core of this research study is the wisdom of the stacked ensemble of the models when it comes to predictions. Based on the features selected above, there were many models selected and tested for prediction, including Logistic Regression, Random Forests, Decision Trees, LightGBM, CatBoost, XGBoost, Naïve Bayes, Support Vector Machine, and even Nearest Neighbour Algorithm. After all these models were fitted individually and their accuracies determined, CatBoost was the best-performing model followed by the XGBoost, Random Forest, and LightGBM, respectively. The decision Tree also had decisively accurate results. Meanwhile, models such as the Support Vector machine had low decisively accurate results in traditional research in this area. Hence, only five classification models were selected for stack ensembling.

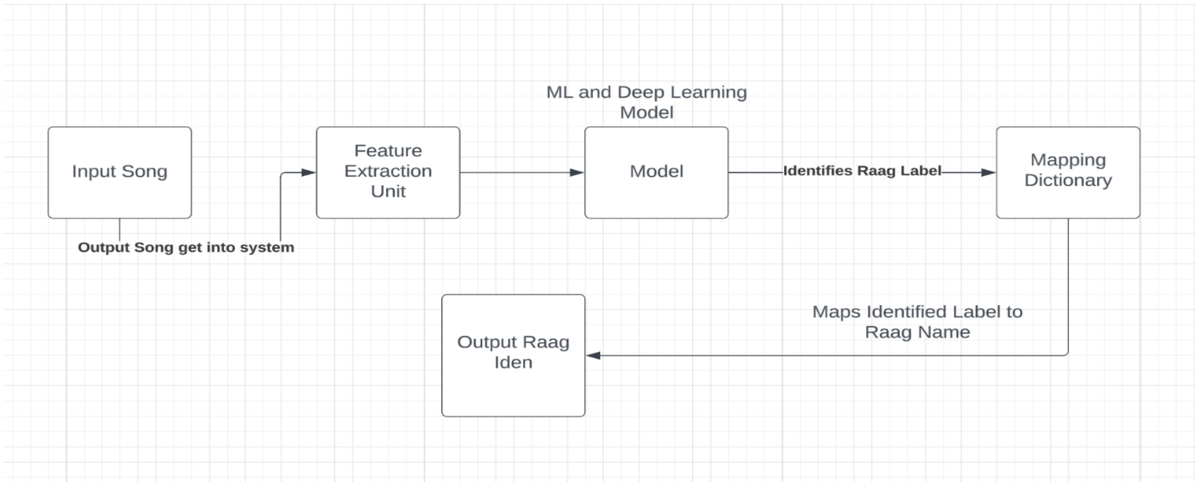

Figure 6.2: Proposed Model for Raga Classification

The entire machine learning system above is a highly complex, resulting from work and prioritization on performance and accuracy. Figure 6.2 explains the proposed model for raga identification. We input the dataset in the model. The dataset have nearly 2000 audio files. The important features from the files are extracted. Features like mfcc, roll off are extracted. The extracted features are used as input to a machine learning model. This model has been trained to take these features and predict the raga associated with the input music. After the machine learning model makes a prediction, the output is in the form of numerical labels. A mapping dictionary is used to convert these numerical labels back into human-readable raga names. The steps included in the proposed model are:

I. User Input: Users upload content, typically songs, for which they want to identify the raga.

II. Feature Extraction: A complex feature extraction system processes the uploaded content to extract relevant features. These features are extracted from the audio data and represent different aspects of the music, which are important for raga identification. These features are saved for later use.

III. Machine Learning Model: The extracted features are used as input to a machine learning model. This model has been trained to take these features and predict the raga associated with the input music. The training process likely involved a dataset of songs with known raga labels, and the model learned to map the extracted features to these labels. It's mentioned that the labels are converted to numerical values for training, which is a common practice in machine learning.

IV. Mapping Dictionary: After the machine learning model makes a prediction, the output is in the form of numerical labels. A mapping dictionary is used to convert these numerical labels back into human-readable raga names. This step is essential for presenting the results to users in a user-friendly way.

V. User Interface: The system likely includes a user interface where users can upload music samples and receive the associated raga predictions in a convenient and understandable format.

VI. Hyperparameter Tuning: Hyperparameter tuning and parameter optimization were performed before arriving at the final ensemble model. This involved techniques like GridsearchCV to search for the best set of hyperparameters for the machine learning models. Tuning hyperparameters is crucial for achieving the best model performance.

VII. Ensemble Model: The final model is a stacked ensemble, a combination of multiple machine-learning models. Stacked ensembles often outperform individual models by leveraging the strengths of each base model. The hyperparameters of these models have been optimized to ensure the best possible performance.

Overall, this system appears to be a well-designed pipeline for raga identification in music, focusing on accuracy and performance. It takes user input, extracts relevant features, uses a machine learning model for prediction, converts the results into a user-friendly format, and incorporates hyperparameter tuning to optimize model performance.

6.4 Experiment and Results

In this section, we delve into the varying predictive power of different features within the context of the raga dataset.

6.4.1 Feature Importance Visualization

Understanding the predictive power of features is crucial for our analysis. Figure 6.3 illustrates the average importance of features during model training, shedding light on the differential capabilities of these features.

```
mfcc0           0.469654
spec_bw         0.457474
spec_cent       0.443226
mfcc1           0.439163
rolloff         0.431935
rmse            0.427196
mfcc4           0.424864
mfcc2           0.416729
zcr             0.414261
mfcc7           0.413180
mfcc6           0.401216
mfcc3           0.400495
mfcc12          0.389147
mfcc17          0.387121
mfcc8           0.384854
mfcc10          0.384349
mfcc5           0.381350
mfcc15          0.380746
mfcc11          0.378887
mfcc19          0.377460
mfcc18          0.375290
chroma_stft     0.373438
mfcc16          0.371836
mfcc9           0.370377
mfcc14          0.366787
mfcc13          0.360225
dtype: float64
```

Figure 6.3: Importance of various features

We observe notable differences in the predictive power of features. For instance, features like "mfcc" carry more weight in terms of accuracy than features such as "mfcc13."As seen above, the average importance of features in the model training is different. So, there are ways to determine the type of raga that is more accurate than the others depending on the features under consideration. Here follows a visualization of the feature's importance using a pie chart:

```
<AxesSubplot:ylabel='None'>
```

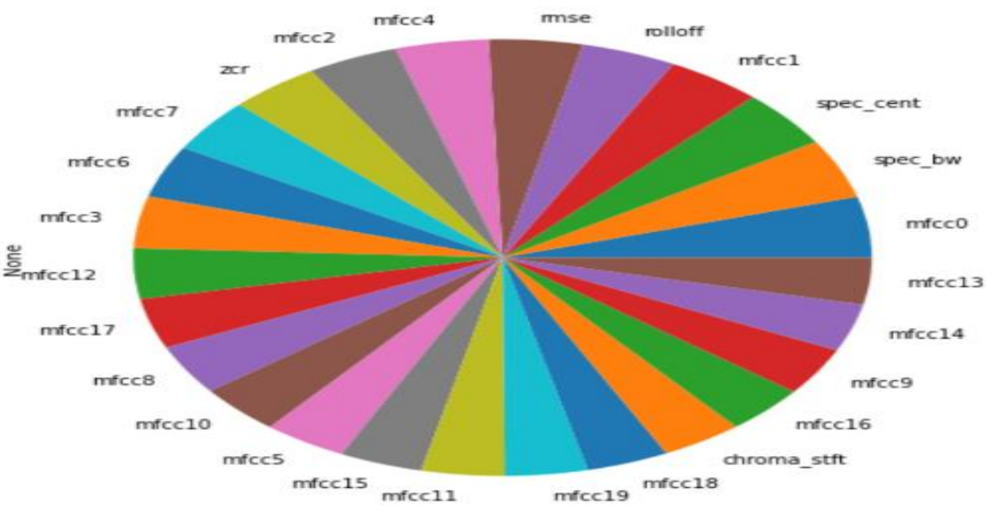

Figure 6.4: Features used for raga identification

From the above, we can see that features such as mfcc have more weight in terms of predictive power than other features like mfcc13.

6.4.2 Training Progress

To gain insight into the model's training progress, we examine the error rate throughout the training process. Figure 6.5 demonstrates the error rate for the training data. The error rate initiates at a relatively high level at the onset of the analysis. However, as the training process unfolds, there is a marked reduction in the error rate. Eventually, as the training nears completion, the error approaches zero.

This systematic reduction in error attests to the effectiveness of our approach and the learning capacity of our model.

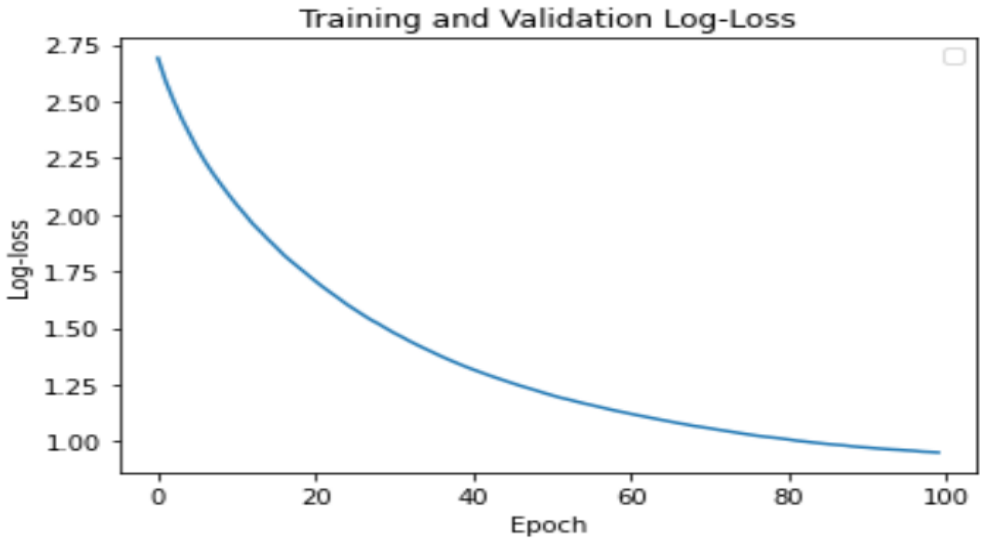

Figure 6.5: Error for training data

6.5 Conclusion

In a nutshell, raga identification is an important area of research in the field of Indian classical music. With the growing availability of digital music recordings and the development of computational methods for music analysis, there is a need for accurate and efficient algorithms for raga identification. We have reviewed some of the existing approaches for raga identification and presented our own method based on a combination of melodic and rhythmic features. Our results indicate that this approach can accurately identify ragas from audio recordings. Our work has focused on developing more sophisticated algorithms that can handle various challenges and contribute to the ongoing efforts to preserve and promote the rich heritage of Indian classical music. We proposed an identification method that employs MFCC variants as features. Our study worked with a stack ensemble of the models. The proposed model was tested, trained, and over 94% testing accuracy was achieved. Hence, this experiment is perhaps the most successful of all others that have been conducted, especially with regards to raga detection from any piece of music.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 SUMMARY OF THE MAIN CONTRIBUTION

In this dissertation, we proposed a technique for generating formal grammar for Indian music. We also applied an existing musical tool to aid in generating formal grammar. An automaton model is also designed for Indian music. The summary of the main contribution is presented as follows:

1. A survey of various existing methods of computational musicology has been carried out. Further, the concept of various HMM's and deep pushdown automata has also been explored.
2. A novel approach to generating probabilistic context-free grammar for Indian music using existing music tools.
3. VOGUE is applied to the musical sequence of notes in the bandish of raga Bhairav, classical Indian music. VOGUE can identify both short-range and long-range dependencies in the musical sequences, i.e., variable-length dependencies, in the musical sequences. Musical patterns repeat themselves and may or may not be separated by variable gap lengths.
4. We represented the palta of a Tala's kaida using state grammar. These structures inhibit cross-serial dependencies that cannot be represented using context-free grammar. Further, we modeled Teentala's kaidas and paltas using a deep pushdown automaton.

The limitation of the proposed work is that we modeled VOGUE for the maximum length of mined sequence of 2. So, there is future scope for applying VOGUE to a larger musical sequence database and increasing the maximum length of sub-sequences $(k > 2)$ to model sequences with larger dependencies.

7.2 FUTURE SCOPE

Following are some stimulating directions for future research building on the proposed work and gleaning knowledge in this dissertation.

1. MIMVOGUE MAXIMUM DEPENDENCIES GAP

   In this dissertation, we have designed a model for Indian music using Variable order gaped HMM for maximum dependencies of gap of 2. In the future, researchers can work on designing VOGUE with larger dependencies to capture higher sequences with the model.

2. MIMVOGUE FOR DIFFERENT RAGAS

   We modeled the model for the Bhairav raga; other ragas can also be modeled with VOGUE.

3. MODELING TALAS WITH DIFFERENT AUTOMATON

   In this dissertation, we have designed a deep pushdown automaton for the paltas of Teentala’s kaida. We can extend our model to accept other talas kaidas, and paltas.

REFERENCES

[1] Why is music important, https://www.theodysseyonline.com / why-is-music-important.(Accessed June 6, 2016)

[2] S. Perchy and G. Sarria, Musical Composition with Stochastic Context-Free Grammars, In Proceedings of Eighth Mexican International Conference on Artificial Intelligence (MICAI),Guanajuato, México, 1-12, 9-13 November 2009.

[3] R. M. Keller and D. R. Morrison, A grammatical approach to automatic improvisation, In Proceedings of Fourth Sound and Music Conference, Lefkada, Greece, 330-337, 11-13 July 2007.

[4] D. Quick, Learning production probabilities for musical grammars, Journal of New Music Research, 45(4), 295-313, 16 Nov 2016.

[5] R. Groves, Automatic melody reduction using a supervised probabilistic Context-free grammar, In Proceedings of the seventh International Society For Music Information Retrieval (ISMIR) Conference, New York City, USA, 775-781, 7-11 August 2016.

[6] S. A. Abdallah and N. E. Gold, Comparing models of symbolic music using probabilistic grammars and probabilistic programming, In proceedings of Joint Sound and Music Computing Conference and International Computer Music Conference (ICMC|SMC), Athens, Greece, 14-20 September 2014.

[7] S. M. Abdallah, N. Gold and A. Marsden, Analysing symbolic music with probabilistic grammars, In D. Meredith (Ed.), Computational music analysis, Switzerland, 157– 189, 2015.

[8] K. M. Kitani and H. Koike, Improvgenerator: Online grammatical induction for on-the-fly improvisation accompaniment, In New Interfaces for Musical Expression (NIME), Sydney, Australia, 469–472, 15-18 June 2010.

[9] A.O. Puente, R.S. Alfonso and M.A. Moreno, Automatic composition of music by means of grammatical evolution, In Proceedings of Conference on APL (ACM Press), Madrid, Spain, 148–155, 22-25 July 2002.

[10] C. Roads, Grammars as representations for music, Computer Music Journal, 3(1), 48-55, March 1979.

[11] Y. Khalifa, B. K. Khan, J. Begovic, A. Wisdom and A. M. Wheeler, Evolutionary music composer integrating formal grammar, In Proceedings of the Ninth Annual Conference

Companion on Genetic and Evolutionary Computation, New York, NY, USA, 2519-2526, 7-11 July 2007.

[12] K. Sidorov, A. Jones and D. Marshall, Music analysis as a smallest grammar problem, In Proceedings of Fifteenth International Society for Music Information Retrieval conference(ISMIR 2014), Taipei, Taiwan, 27-31 October 2014.

[13] C. I. Wang and S. Dubnov, Context-aware Hidden Markov models of Jazz music with Variable Markov oracle, In Proceedings of the International Workshop on Musical Metacreation, Atlanta, USA, 1-8, 19-23 June 2017.

[14] R. Kerr, Melodic analysis using Hierarchical Hidden Markov models, School of Informatics, University of Edinburgh, Scotland, 2011.

[15] M. Weiland, A. Smaill and P. Nelson, Learning musical pitch structures with Hierarchical Hidden Markov model, University of Edinburgh, Scotland, U. K., 2005.

[16] M. D. Hoffman, P. R. Cook and D. M. Blei, Data-Driven Recomposition using the Hierarchical Dirichlet Process hidden Markov model, In Proceedings of Thirty second International Computer Music Conference (ICMC), Belfast, Ireland, UK, 1-7, 24-29 August 2008.

[17] R. Chen, W. Shen, A. Srinivasamurthy and P. Chordia, Chord recognition using Duration-explicit hidden Markov models, In Proceedings of Thirteenth International Society for Music Information Retrieval Conference (ISMIR), Porto, Portugal, 445-450, 8-12 October 2012.

[18] E. Benetos and T. Weyde, Explicit duration hidden Markov Models for multiple-instrument polyphonic music transcription, In Proceedings of Fourteenth International Society for Music Information Retrieval Conference (ISMIR), Curitiba, Brazil, 269-274, 4-8 November 2013.

[19] N. Nakano, Y. Ohishi, H. Kameoka, R. Mukai and K. Kashino, Mondrian hidden Markov model for music signal processing, In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Florence, Italy, 2424-2428, 4-9 May 2014.

[20] E. Nakamura, T. Nakamura, Y. Saito, N. Ono and S. Sagayama, Outer-product hidden Markov model and polyphonic MIDI score following, Journal of New Music Research, 43(2), 183-201, 2014.

[21] E. Nakamura, P. Cuvillier, A. Cont, N. Ono and S. Sagayama, Autoregressive hidden semi-Markov model of symbolic music performance for score following, In Proceedings of Sixth

International Society for Music Information Retrieval Conference (ISMIR), Malaga, Spain, 392-398, 26-30 October 2015.

[22] Ozerov, C. Févotte and M. Charbit, Factorial scaled hidden Markov model for polyphonic audio representation and source separation, In Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'09), New Paltz, New York, 121-124, 18-21 October 2009.

[23] P. Chordia, A. Sastry and S. Şentürk, Predictive tabla modeling using Variable-Length Markov and hidden Markov models, Journal of New Music Research, 40(2), 105-118, 2011.

[24] S. Senturk, Computational modeling of improvisation in Turkish folk music using Variable-Length Markov models, Georgia Institute of Technology, Atlanta, Georgia, 2011.

[25] B. Bouqata, C. D. Carothers, B. K. Szymanski and M. J. Zaki, VOGUE: A novel Variable Order-Gap state machine for modeling sequences, In Proceedings of the European Conference Principles and Practice of Knowledge Discovery in Databases (PKDD'06), Berlin, Germany, 42-54, 18-22 September 2006.

[26] M. J. Zaki, C. D. Carothers and B. K. Szymanski, VOGUE: A variable Order Hidden Markov model with duration based on frequent sequence mining, ACM Transaction Knowledge Discovery from Data, 4(1), 1-31, January 2010.

[27] B. Bel and J. Kippen, Modeling music with grammars: Formal language representation in the Bol processor, In Marsden A, Pople A (ed) Computer Representations and Models in Music, Academic Press, London, 207-238, 1992.

[28] B. Bel and J. Kippen, Bol processor grammars, In Balaban M (ed) Understanding music with AI: Perspectives on Music Cognition, AAAI Press, 366-40, 1992.

[29] B. Bel, Modeling improvisatory and compositional processes, Languages of Design Formalisms for Word, Image and Sound Synthesis 1(1), 11-26, 1992.

[30] J. Kippen and B. Bel, Computers, Composition and the challenge of "New Music" in modern India. Leonardo Music Journal 4(1), 79-84, 1994.

[31] B. Jurish, Music as a formal language, In Zimmer F (ed), Pure data, Wolke Verlag, 2004.

[32] J. Mahmud, Grammar based modeling and generation of tabla compositions. Dissertation, University of Bath, United Kingdom, 2006.

[33] R. Shinkuma, R. Takagi, Y. Inagaki, E. Oki and F. Xhafa, Incentive mechanism for mobile crowdsensing in spatial information prediction using machine learning, Advanced Information Networking and Applications: Proceedings of the thirty fourth International

Conference on Advanced Information Networking and Applications (AINA), Caserta, Italy, 792-803, 15-17 April 2020.

[34] S. Shetty and K. Achary, Raga Mining of Indian Music By Extracting Arohana-Avarohana Pattern, International Jouranal of Recent Trends in Engineering,1, 362-366, 2009.

[35] Y. H. Dandawate, P. Kumari and A. Bidkar, Indian instrumental music: Raga analysis and classification, Proceedings of first International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India, 725-729, 4-5 September 2015.

[36] P. Gupta, A. Sharma and R. Jindal, Scalable machine-learning algorithms for big data analytics: a comprehensive review, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 6(6), 194-214, 2016.

[37] A. B. Anagha, S. Rajkumar, Y. H. Dandawate, North Indian Raga Recognition Using Ensemble Classifier, International Journal of Electrical Engineering and Technology (IJEET), 12(6),251-258, 2021.

[38] D. Joshi, J. Pareek, P. Ambatkar , Indian Classical Raga Identification using Machine Learning, International Semantic Intelligence Conference(ISIC), New Delhi, India, 259-263, 25-27 February 2021.

[39] H. Dodia, S. Pandey, N. P. Chawande, Identification of Raga by Machine Learning with Chromagram, International Research Journal of Engineering and Technology (IRJET) , 07(5), 3220-3225, 2020.

[40] H. Sharma, R. S. Bali, Comparison of ML classifiers for Raga recognition, International Journal of Scientific and Research Publications, 5(10), October 2015.

[41] A. Anand, Raga Identification Using Convolutional Neural Network, Proceedings of Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), Gangtok, India, 25-28 February 2019.

[42] H. K. Thakur, A. Gupta, A. Bhardwaj and D. Verma, Rumor detection on Twitter using a supervised machine learning framework. International Journal of Information Retrieval Research (IJIRR), 8(3), 1-13, 2018.

[43] V. Kumar, H. Pandya, C. V. Jawahar, Identifying Ragas in Indian Music, International Conference on Pattern Recognition (ICPR), Stockholm, Sweden,1-11, 24-28 August 2014.

[44] H. Sharma and R. S.Bali, Raga Identification of Hindustani Music using Soft Computing Techniques, Proceedings of Recent Advances in Engineering and Computational Sciences (RAECS), VIET Punjab University Chandigarh, 2014, Chandigarh, 06 - 08 March.

[45] S. John, M. S. Sinith and S. R. Sudheesh, P. P. Lalu Classification of Indian Classical Carnatic Music Based on Raga Using Deep Learning, Proceedings of IEEE Recent Advances in Intelligent Computational Systems (RAICS), Thiruvananthapuram, India, 110-113, 3-5 December 2020.

[46] D. V. S. Kumar, R. Chaurasia, A. Misra, P. K. Misra and A. Khang, Heart disease and liver disease prediction using machine learning, Data-Centric AI Solutions and Emerging Technologies in the Healthcare Ecosystem, 205-214, 2023.

[47] S. T. Madhusudhan and G. Chowdhary, Deepsrgm-sequence classification and ranking in Indian classical music with deep learning, Proceedings of the twenteth International Society for Music Information Retrieval Conference, 533-540, 2019.

[48] M. Kumar, A. J. Singh, B. Sharma and K. Cengiz, Evaluation of machine learning algorithms on academic big dataset by using feature selection techniques, Intelligent Network Design Driven by Big Data Analytics, IoT, AI and Cloud Computing, 61, 2022.

[49] A. Singha, N. R. Rajalakshmi, J. Arun Pandian and S. Saravanan, Deep Learning-Based Classification of Indian Classical Music Based on Raga, Proceedings of International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 1-7, 03-04 March 2023.

[50] P. D. Shah, N. M. Jagtap, P. T. Talekar and K. Gawande, Raga Recognition in Indian Classical Music Using Deep learning, Proceedings of conference on Artificial Intelligence in Music, Sound, Art and Design, Springer, 248-263, 7-9 April 2021.

[51] P. Dighe, H. Karnick and B.Raj, Swara Histogram Based Structural Analysis And Identification Of Indian Classical Ragas, Proceedings of Conference of the International Society for Music Information Retrieval (ISMIR), Brazil, 35-40, 4-8 November 2013.

[52] H. Dodia, S. Pandey and N. P. Chawande, Identification of Raga by Machine Learning with Chromagram, International Research Journal of Engineering and Technology (IRJET), 7(05), 2020.

[53] C. M. Subramanian, A. K, Cherukuri and C. Chelliah, Role based access control design using three-way formal concept analysis. International Journal of Machine Learning and Cybernetics, 9, 1807-1837, 2018.

[54] D. E. Knuth, Semantics of context-free languages, Mathematical systems theory, 2(2), 127-145, 1968.

[55] R. D. Dowell and S. R. Reddy, Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction, BMC Bioinformatics, 5(1), 71-85, 2004.

[56] C. X. Zhai, A Brief Note on the Hidden Markov Models (HMMs), Department of Computer Science, University of Illinois at Urbana-Champaign, USA, 2003.

[57] A. Meduna and P. Zemek, Regulated grammar and automata. Springer, New York, https://doi.org/10.1007/978-1-4939-0369-6, 2014.

[58] A. Meduna, Deep pushdown automata, Acta Informatica 42(8-9), 541-552, https://doi.org/10.1007/s00236-006-0005-0, 2006.

[59] Octave in music, https://musicterms.artopium.com/o/Octave.html, (Accessed on April 20, 2019).

[60] Scales of music, https://en.wikipedia.org/wiki/Scale, (Accessed on April 20, 2019).

[61] Major scale,https://en.wikipedia.org/wiki/Major_scale, (Accessed on April 20, 2019).

[62] Minor scale, http://musictheoryblog.blogspot.in/2007/01/minor-scales.html, (Accessed on April 20, 2019).

[63] Minor scale, https://en.wikipedia.org/wiki/Minor_scale, (Accessed on April 20, 2019).

[64] Chords in Indian music, http://www.swarmanttra.com/blog/indian-classical-music-and-chords/, (Accessed on April 20, 2019).

[65] Indian Raga, https://indianraga.com/lesson/lesson-2-parallels-between-western-and-carnatic-music/, (Accessed on April 20, 2019).

[66] C major Scale and degree names, https://en.wikipedia.org/wiki/ Degree_(music), (Accessed on April 20, 2019).

[67] A. Meduna, Automata and Languages: Theory and Applications, London: Springer, 2000.

[68] E. Gilbert and D. Conklin, A probabilistic context-free grammar for melodic reduction, Proceedings of the International Workshop on Artificial Intelligence and Music, twentieth International Joint Conference on Artificial Intelligence (IWAI|IJCAI), Hyderabad, India, 83-94, 6-12 January 2007.

[69] M. Hamanaka, K. Hirata and S. Tojo, σGTTM III: Learning-based time-span tree generator based on PCFG, Proceedings of eleventh International Symposium on Computer Music Multidisciplinary Research (ISCMMR), Plymouth, UK, 303-317, 16-19 June 2015.

[70] B. Bel, Pattern Grammar in Formal Representations of Musical Structures. Proceedings of 11th International Joint Conference on Artificial Intelligence (IJCAI), Detroit, USA, 118-145, 20-25 August 1989.

[71] M. Giraud and S. Staworko, Modeling musical structure with parametric grammar, In Proceedings of Fifth International Conference on Mathematics and Computation in Music (MCM), London, UK, 85-96, 22-25 June 2005.

[72] M. Rohrmeier and I. Cross, Tacit tonality: Implicit learning of context-free harmonic structure, In Proceedings of seventh Triennial Conference of European Society for the Cognitive Sciences of Music (ESCOM), Jyvaskyla, Finland, 443-451, 12-16 August 2009.

[73] J. McCormack, Grammar-based music composition, In Stocker (Ed.), Complex Systems 96, IOS Press, Amsterdam, 321-336, 1996.

[74] A. O. Puente, R. S. Alfonso and M. A. Moreno, Automatic composition of music by means of grammatical evolution, In Proceedings of sixteenth Conference on APL (ACM Press), Madrid, Spain, 148-155, 22-25 July 2002.

[75] C. Roads, Grammar as representations for music. Computer Music Journal, 3(1), 48-55, 1979.

[76] Y. Khalifa, B. K. Khan, J. Begovic, A. Wisdom and A. M. Wheeler, Evolutionary music composer integrating formal grammar, In Elleithy K. (Ed.), Advances and Innovations in Systems, Computing Sciences and Software Engineering, 503-508, 2007.

[77] K. Sidorov, A. Jones and D. Marshall, Music analysis as a smallest grammar problem, In Proceedings of fifteenth International Society for Music Information Retrieval Conference (ISMIR), Taipei, Taiwan, 301-306, 27-31 October 2014.

[78] W. Chai and B. Vercoe, Folk music classification using hidden Markov models, In Proceedings of International Conference on Artificial Intelligence, Las Vegas, 25-28 June 2001.

[79] A. Durey and M.A. Clements, Melody spotting using hidden Markov Models, In Proceedings of Second Annual International Symposium on Music Information Retrieval (ISMIR), Bloomington, Indiana, USA, 109-117, 15-17 October 2001.

[80] H. Takeda, N. Saito, T. Otsuki, M. Nakai, H. Shimodaira and S. Sagayama, Hidden Markov model for automatic transcription of MIDI signals, In Proceedings of IEEE Workshop Multimedia Signal Processing, St. Thomas, US Virgin Islands, 428-431, 9-11 December 2002.

[81] H. Takeda, T. Nishimoto and S. Sagayama, Rhythm and tempo recognition of music performance from a probabilistic approach, In Proceedings of the fifth Annual International Symposium on Music Information Retrieval (ISMIR), Barcelona, Spain, 10-14 October 2004.

[82] E. Pollastri and G. Simoncelli, Classification of melodies by composer with hidden Markov models, In Proceedings of first International Conference on Web Delivering of Music (WEDELMUSIC), Florence, Italy, 88-95, 23-24 November 2001.

[83] B. J. Clement, Learning harmonic progression using Markov models, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, 1998.

[84] P. P. Reddy, B. T. Rao, K. R. Sudha and V. M. K. Hari, Automatic raaga identification system for Carnatic music using hidden Markov model, Global Journal of Computer Science and Technology (GJCST), 11(22), 1-9, 2011.

[85] A. Bhattacharjee and N. Srinivasan, Hindustani raga representation and identification: A transition probability based approach, International Journal of Mind, Brain and Cognition, 2 (1-2), 65-94, 2011.

[86] A. S. Krishna, P. V. Rajkumar, K. P. Saishankar and M. John, (2011, January 27-29). Identification of Carnatic ragas using hidden Markov models. IEEE 9th International Symposium on Applied Machine Intelligence and Informatics (SAMI), Smolenice, Slovakia, 107-110, 27-29 January 2011.

[87] M. Sinith and K. Rajeev, Hidden Markov model based recognition of musical pattern in south Indian classical music, IEEE International Conference on Signal and Image Processing. Hubli, India, 2006.

[88] S. Mohapatra, A. Awasthi and A. Mukherjee, Automatic music generation for Indian classical music. (Project report CS365), Department of computer science and engineering, Indian Institute of Technology, Kanpur, 24 April 2010.

[89] G. Pandey, C. Mishra and P. Ipe, Tansen: A system for automatic raga identification, In Proceedings of the first Indian International Conference on Artificial Intelligence (IICAI) ,Hyderabad, India, 1350-1363, 18-20 December 2003.

[90] A. Yanchenko, Classical music composition using hidden Markov model (Doctoral dissertation), Department of Statistical Science, Duke University, USA, 2017.

[91] A. K. Yanchenko and S. Mukherjee, Classical music composition using state space models, (Inar Xiv: 1708.03822 preprint), 12 August 2017.

[92] S. Gao, N. C. Maddage, C.H. Lee, A hidden Markov model based approach to music segmentation and identification, In Information, Communications and Signal Processing, and In Proceedings of fourth Joint Conference of the 4th International Conference on Pacific Rim Conference on Multimedia (ICICS-PCM), Singapore, 1576-1580,15-18 December 2003.

[93] X. Shao, C. Xu and M. S. Kankanhalli, Unsupervised classification of music genre using hidden Markov model, In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, 2023-2026, 27-30 June 2004.

[94] C. Raphael, Automatic segmentation of acoustic musical signals using hidden Markov models, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 21(4), Washington, USA,360-370, 1999.

[95] K. Lee, A system for automatic chord transcription from audio using genre-specific hidden Markov models, In International Workshop on Adaptive Multimedia Retrieval, Berlin, Heidelberg, 134-136, 5 July 2007.

[96] N. Orio and C. Zen, Song identification through hmm-based modeling of the main melody, In Proceedings of thirty third International Computer Music Conference (ICMC), Copenhagen, Denmark, 248-251, 27-31 August, 2007.

[97] D. Cazau and G. Nuel, Investigation on the use of hidden-Markov models in automatic transcription of music (InarXiv: 1704.03711 preprint), 12 April 2017.

[98] K. Lee and M. Slaney, A unified system for chord transcription and key extraction using hidden Markov models, In Proceedings of eighth International Society for Music Information Retrieval Conference (ISMIR), Vienna, Austria, 245-250, 23-30 September 2007.

[99] Y. Qi, J. Paisley and L. Carin, Music analysis using hidden Markov mixture models. IEEE Transactions on Signal Processing, 55(11), 5209-5224, 2007.

[100] A. Sheh and D. P. W. Ellis, Chord segmentation and recognition using EM-trained hidden Markov model, In Proceedings of fourth International Society for Music Information Retrieval Conference (ISMIR), Baltimore, Maryland, 183-189, 26-30 October 2003.

[101] Y. Ueda, Y. Uchiyama, T. Nishimoto, N. Ono and S. Sagayama, HMM based approach for automatic chord detection using refined acoustic features, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Dallas, USA, 5518-5521, 14-19 March 2010.

[102] W. Chai and B. Vercoe, Detection of key change in classical piano music, In Proceedings of the sixth International Symposium on Music Information Retrieval (ISMIR), London, UK, 468-473, 11-15 September 2005.

[103] P. Cano, P. Loscos and J. Bonada, Score-performance matching using HMMs, In Proceedings of twenty third International Computer Music Conference (ICMC), Beijing, China, 441-444, 22-28 October 1999.

[104] E. Batlle, J. Masip and E. Guaus, Automatic song identification in noisy broadcast audio, In Proceedings of thirteenth International Conference on Signal and Image Processing, Kauai, Hawaii, USA, 101-111, 14-16 August 2002.

[105] S. S1ertan and P. Chordia, Modeling melodic improvisation in Turkish folk music using variable-length Markov models, In Proceedings of twelfth International Society for Music Information Retrieval Conference (ISMIR), Miami, Florida, USA, 269-274, 24-28 October 2011.

[106] A. Mesaros and V. Tuomas, Automatic recognition of music audio and lyrics, In Proceedings of eleventh International Conference on Digital Audio Effects (DAFx), 321-324, Espoo, Finland, 321-324, 1-4 September 2008.

[107] R. Miotto and N. Orio, Automatic identification of music works through audio matching, In Proceedings of the eleventh International Conference Theory and Practice of Digital Libraries, Budapest, Hungary, 124-135, 16-21 September 2007.

[108] J. C. Zaragoza, A. H. Toselli and E. Vidal, Early handwritten music recognition with hidden Markov models, In Proceedings of fifteenth International Conference on Frontiers in Handwriting Recognition (ICFHR), Shenzhen, China, 319-324, 23-26 October 2016.

[109] H. Jin and H. V. Jagadish, Indexing Hidden Markov Models for Music Retrieval, In Proceedings of Third International Society for Music Information Retrieval, Paris, France, 20-24, 13-17 October 2002.

[110] R. Miotto and N. Orio, Automatic identification of music works through audio matching, In Proceedings of Eleventh International Conference Theory and Practice of Digital Libraries, Budapest, Hungary, 124-135, 16-21 September 2007.

[111] M. Blass, Timbre-based drum pattern classification using hidden Markov models, In Proceedings of Sixth International Workshop on Machine Learning and Music (MML), Prague, 13 September 2013.

[112] S. Chithra, M. S. Sinith and A. Gayathri, Music information retrieval for polyphonic signals using hidden Markov model, Procedia Computer Science, 46, 381-387, April 2015.

[113] A. Flexer, E. Pampalk and G. Widmer, Hidden Markov models for spectral similarity of songs, In Proceedings of Eight International Conference on Digital Audio Effects (DAFx), 131-136, 20-22 September 2005.

[114] C. V. Patricio and E. Chew, Application of hidden Markov model to music style classification via timing and loudness features, In Proceedings of Operations Research Belgium Conference (ORBEL), Louvain-la-Neuve, Belgium, 1-2, 28-29 January 2016.

[115] G. Peeters, Musical key estimation of audio signal based on HMM modeling of chroma vectors, In Proceeding of Ninth International Conference on Digital Audio Effects (DAFx), Montreal, Canada, 121-131, 18-20 September 2006

[116] A. Chandrashekaran, A. Ramakrishna, A. Jain, G. Yang, N. Kohli, R. Swaminathan, T. Chopra and Y. Xie, Song retrieval system using hidden Markov models, http://mlsp.cs.cmu.edu/courses/fall2010/projects/reports/Anoop, (Accessed on 20 march 2010).

[117] L. Dhulekar and S. K. Shah, Musical audio beat tracking using hidden Markov model, International Journal of Science and Research (IJSR), 3(5), 1054-1058, 2014.

[118] S. Dalin-Volsing, Classification of musical genres using hidden Markov models (Master's thesis), Centre of mathematical sciences, Lund University, Sweden, 2017.

[119] R. Kerr, Melodic analysis using hierarchical hidden Markov models (Master's thesis), School of Informatics, University of Edinburgh, Scotland, 2011.

[120] G. E. Berget, Using hidden Markov models for musical chord prediction (Master's thesis), Department of Mathematical Sciences, Norwegian University of Science and Technology, Norway, July 2017.

[121] I. Karpov and D. Subramanian, Hidden Markov classification for musical genres (Course Project), Rice University, Texas, United States, 2002.

[122] D. Petruncio and M. A. H. Johnson, Evaluation of various features for music genre classification with hidden Markov models (Master's thesis), University of Illinois, United States, March 2002.

[123] E. Benetos and T. Weyde, Explicit duration hidden Markov models for multiple-instrument polyphonic music transcription, In Proceedings of fourteenth International Society for Music Information Retrieval Conference (ISMIR), Curitiba, Brazil, 269-274, 4-8 November 2013.

[124] E. Nakamura, P. Cuvillier, A. Cont, N. Ono and S. Sagayama, Autoregressive hidden semi-Markov model of symbolic music performance for score following, In Proceedings of Sixteenth International Society for Music Information Retrieval Conference (ISMIR), Malaga, Spain, 392-398, 26-30 October 2015.

[125] N. Nakano, Y. Ohishi, H. Kameoka, R. Mukai and K. Kashino, Mondrian hidden Markov model for music signal processing, In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Florence, Italy, 2424-2428, 4-9 May 2014.

[126] C. I. Wang and S. Dubnov, Context-aware hidden Markov models of Jazz music with variable Markov oracle, International Workshop on Musical Metacreation, Atlanta, USA, 1-8, 19-23 June 2017.

[127] I. Simon, D. Morris and S. Basu, MySong: automatic accompaniment generation for vocal melodies, In Proceedings of twenty-sixth Annual Conference on Human Factors in Computing Systems (CHI), Florence, Italy, 725-734, 5-10 April 2008.

[128] R. Scholz, E. Vincent and F. Bimbot, Robust modeling of musical chord sequences using probabilistic N-grams, In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Taipei, Taiwan, 53-56, 19-24 April 2009.

[129] H. T. Cheng, Y. H. Yang, Y. C. Lin, I. B. Liao and H. H. Chen, Automatic chord recognition for music classification and retrieval, In Proceedings of IEEE International Conference Multimedia and Expo (ICME), Hannover, Germany, 1505-1508, 23-26 June 2008.

[130] S. Doraisamy and S. M. Ruger, An Approach Towards A Polyphonic Music Retrieval System, In Proceedings of second Annual International Symposium on Music Information Retrieval (ISMIR), Bloomington, Indiana, USA, 187-193, 15-17 October 2001.

[131] S. Doraisamy and S. Ruger, A comparative and fault-tolerance study of the use of N-grams with polyphonic music, In Proceedings of third of International Conference on Music Information Retrieval (ISMIR), Paris, France, 101-106, 13-17 October 2002.

[132] S. Doraisamy and S. Ruger, Robust polyphonic music retrieval with n-grams, Journal of Intelligent Information Systems, 21(1), 53-70, 2003.

[133] S. Doraisamy and S. Ruger, Position indexing of adjacent and concurrent n-grams for polyphonic music retrieval, In Proceedings of fourth International Conference on Music Information Retrieval (ISMIR), Baltimore, USA, 227-238, 27-30 October 2003.

[134] S. Doraisamy and S. M. Ruger, A polyphonic music retrieval system using n-grams, In Proceedings of Fifth International Conference on Music Information Retrieval (ISMIR), Barcelona, Spain, 204-209, 10-14 October 2004.

[135] M. Lo and S. M. Lucas, Evolving musical sequences with n-gram based trainable fitness functions, In Proceedings of IEEE Congress on Evolutionary Computation, Vancouver, Canada, 601-608, 16-21 July 2006.

[136] M. Y. Lo and S. M. Lucas, N-gram fitness function with a constraint in a musical evolutionary system, In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Singapore, 4246-4251, 25-28 September 2007,

[137] M. Ogihara and T. Li, N-Gram Chord Profiles for Composer Style Representation, In Proceedings of nineteenth International Conference on Music Information Retrieval (ISMIR), Philadelphia, USA, 671-676, 14-18 September 20008.

[138] M. Takano and Y. Osana, Automatic composition system using genetic algorithm and N-gram model considering melody blocks, In Proceedings of IEEE Congress on Evolutionary Computation (CEC), 1-8, Brisbane, 10-15 June 2012.

[139] J. K. Paulus and A. P. Klapuri, Conventional and periodic N-grams in the transcription of drum sequences, In Proceedings of IEEE International Conference on Multimedia and Expo (ICME), Baltimore, Maryland, July 2003.

[140] Y. C. Lap and B. Kao, A study on n-gram indexing of musical features, In Proceedings of IEEE International Conference on Multimedia and Expo (ICME), Hangzhou, China, 869-872, 10-14 May 2000.

[141] J. Wolkowicz, Z. Kulka and V. Keselj, N-gram based approach to composer recognition, Archives of Acoustics, 33(1), 43–55, 2008.

[142] C. Wang, J. Li and S. Shi, N-gram inverted index structures on music data for theme mining and content-based information retrieval, Pattern Recognition Letters, 27(5), 492–503, 2006.

[143] K. Trochidis, C. Guedes, A. Anantapadmanabhan and A. Klaric, CAMeL: Carnatic percussion music generation using n-gram models, In Proceedings of thirteenth Sound and Music Computing Conference (SMC), Hamburg, Germany, 31 August -3 September 2016.

[144] E. Unal, B. Bozkurt and M. K. Karaosmanoglu, N-gram based statistical makam detection on Makam music in Turkey using symbolic data, In Proceedings of thirteenth International Society of Music Information Retrieval Conference (ISMIR), Porto, Portugal, 43-48, 8-12 October 2012.

[145] K. Frieler, Generalized n-gram measures for melodic similarity, Data Science and Classification, In V. Batagelj, H. H. Bock, A. Ferligoj, A. Ziberna (Eds.),Berlin, Heidelberg, 289-298, 2006.

[146] N. Patel and P. Mundur, An n-gram based approach for finding the repeating patterns in musical data, In Proceedings of European Internet and Multimedia Systems and Applications (IMSA), Hawaii, USA, 407-412, 15-17 August 2005.

[147] T. Kumamoto and K. Ohta, A query by musical impression system using n-gram based features, In Proceedings of IEEE Conference on Cybernetics and Intelligent Systems, 992-998, 1-3 December 2004.

[148] S. Doraisamy, Polyphonic music retrieval: The n-gram approach (PhD dissertation), Department of Computing, Imperial College London, September 2004.

[149] A. L. Uitdenbogerd, N-gram pattern matching and dynamic programming for symbolic melody search, In Proceedings of third Annual Music Information Retrieval Evaluation eXchange (MIREX), Vienna, 23-27 September 2007.

[150] P. Salosaari and K. Jarvelin, MUSIR- A retrieval model for music (Technical report), Department of Information Studies, University of Tampere, Finland, July 1998.

[151] S. Pakhare and M. M. Potey, A survey of style identification approaches in music information retrieval, International Journal of Computer Trends and Technology (IJCTT), 36(1), 17-21, 2016.

[152] C. Hauser, Probabilistic modeling of musical chord sequences for music analysis. Metiss team, Research Institute of Computer Science and Random systems (IRISA), France, 4 June 2009.

[153] I. Suyoto and A. L. Uitdenbogerd, Simple efficient n-gram indexing for effective melody retrieval, In first Annual Music Information Retrieval Evaluation eXchange (MIREX), London, UK, 11-15 September 2005.

[154] Q. Wang, Z. Guo, G. Liu and J. Guo, Audio fingerprinting based on n-grams, International Journal of Digital Content Technology and its Applications (JDCTA), 6(10), 361-368, 2012.

[155] M. Hontanilla, C. Perez-Sancho and J. M. Inesta, Modeling musical style with language models for composer recognition. Pattern Recognition and Image Analysis, In J. M. Sanches, L. Mico and J.S. Cardoso (Eds.). Berlin Heidelberg, 740-748, 2013.

[156] E. Zheng, M. Moh and T. S. Moh, Music genre classification: A n-gram based musicological approach, In Proceedings of IEEE Seventh International Advance Computing Conference (IACC), 671-677, 5-7 January 2017.

[157] K. Felch and Y. Song, Implementing a music improviser using n-gram models-Natural Language Processing (Final Project), Massachusetts Institute of Technology, Cambridge, USA, 2012.

[158] A. Alpkocak and A. Gedik, Classification of Turkish songs according to makams by using n-grams, In Proceedings of fifteenth Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN), Mugla, Turkey, 21-23 June 2006.

[159] D. Das and M. Choudhury, Finite state models for generation of Hindustani classical music, In Proceedings of eleventh International Symposium on Frontiers of Research in Speech and music (FRSM), Bhubaneswar, Orissa, India, 59-64, 6-7 January 2005.

[160] T. Hosoya, M. Suzuki, A. Ito, S. Makino, L. A. Smith, D. Bainbridge and I. H. Witten, Lyrics recognition from a singing voice based on finite state automaton for music information retrieval, In Proceedings of Sixth International Conference on Music Information Retrieval (ISMIR), London, UK, 532-535, 11-15 September 2005.

[161] T. Oliwa and M. Wagner, Composing music with neural networks and probabilistic finite-state machines, In Workshops on Applications of Evolutionary Computing (EvoMUSART), In M. Giacobini et al. (Eds.), Berlin, Heidelberg, 503-508, 26 March 2008.

[162] J. C. Zaragoza and J. Oncina, Recognition of pen-based music notation with finite-state machines, Expert Systems with Applications, 72, 395-406, 2017.

[163] M. Suzuki, T. Hosoya, A. Ito and S. Makino, Music information retrieval from a singing voice using lyrics and melody information, EURASIP Journal on Advances in Signal Processing, 1-8, 2007.

[164] E. Weinstein and P. Moreno, Music identification with weighted finite-state transducers, In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 689-692, 15-20 April 2007.

[165] J. Buys and B. Merwe, Chorale harmonization with weighted finite-state transducers, In Proceedings of the twenty third Annual Symposium of the Pattern Recognition Association of South Africa (PRASA), Pretoria, South Africa, 95-101, 29-30 November 2012.

[166] M. Mohri, P. Moreno and E. Weinstein, Robust music identification, detection, and analysis, In Proceedings of eigth International Conference on Music Information Retrieval (ISMIR) (pp. 135-139). Vienna, Austria, 135-139, 23-30 September 2007.

[167] M. Mohri, P. Moreno and E. Weinstein, Efficient and robust music identification with weighted finite-state transducers, IEEE Transactions on Audio, Speech and Language Processing, 18(1), 197-207, 2010.

[168] S. Hubler and R. Hoffmann, Modeling drum patterns with weighted finite-state transducers, In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, Canada, 719-723, 26-31 May 2013.

[169] J. P. Forsyth and J. P. Bello, Generating musical accompaniment using finite-state transducers, In Proceedings of sixteenth International Conference on Digital Audio Effects (DAFx), Maynooth, Ireland, 1-7, 2-5 September 2013.

[170] J. P. Forsyth, R. M. Bittner, M. Musick and J. P. Bello, Improving and Adapting Finite State Transducer Methods for Musical Accompaniment, In Proceedings of forty-first International Computer Music Conference (ICMC), Denton, Texas, USA, 290-297, 25 September-1 October 2015.

[171] M. Ettlinger, E. H. Margulis and P. C. Wong, Implicit memory in music and language, Frontiers in Psychology, 2, 1-10, 2011.

[172] M. Rohrmeier and R. Widdess, Incidental learning of modal features of north Indian music, In Proceedings of Twelfth International Conference on Music Perception and Cognition and

Eight Triennial Conference of the European Society for the Cognitive Sciences of Music, Thessaloniki, Greece, 857-866, 23-28 July 2012.

[173] P. Loui, Learning and liking of melody and harmony: Further studies in artificial grammar learning, Topics in Cognitive Science, 4, 554-567, 2012.

[174] B. Bly, R. E. Carrion and B. Rasch, Domain-specific learning of grammatical structure in musical and phonological sequences, Memory & Cognition, 37(1), 10-20, 2009.

[175] G. Kuhn and Z. Dienes, Implicit learning of nonlocal musical rules: implicitly learning more than chunks, Journal Experiment Psychology: Learning, Memory, Cognition, 31(6), 1417-1432, 2005.

[176] C. Desmet, B. Poulin-Charronnat, P. Lalitte and P. Perruchet, Implicit learning of nonlocal musical rules: A comment on Kuhn and Dienes, Journal of Experimental Psychology: Learning, Memory, and Cognition, 35(1), 299-305, 2009.

[177] M. Rohrmeier and I. Cross, Tacit tonality: Implicit learning of context-free harmonic structure, In Proceedings of seventh Triennial Conference of European Society for the Cognitive Sciences of Music (ESCOM), Jyvaskyla, Finland, 443-451, 12-16 August 2009.

[178] B. Tillmann, Music cognition: Learning, perception, expectations. Computer Music Modeling and Retrieval (CMMR), In R. Kronland-Martinet, S. Ystad, and K. Jensen (Eds.), Berlin, Heidelberg, 11-33, 2008

[179] S. Jiang, L. Zhu, X. Guo, W. Ma, Z. Yang and Z. Dienes, Unconscious structural knowledge of tonal symmetry: Tang poetry redefines limits of implicit learning. Consciousness & Cognition, 21(1), 476-486, 2012.

[180] E. G. Schellenberg, Expectancy in melody: Tests of the implication-realization model. Cognition, 58(1), 75-125, 1996.

[181] T. Selchenkova, M. R. Jones and B. Tillmann, The influence of temporal regularities on the implicit learning of pitch structures, The Quarterly Journal of Experimental Psychology, 67(12), 2360-2380, 2014.

[182] B. Tillmann and B. Poulin-Charronnat, Auditory expectations for newly acquired structures, The Quarterly Journal of Experimental Psychology, 63(8), 1646-1664, 2010.

[183] M. Rohrmeier and P. Rebuschat, Implicit learning and acquisition of music. Topics in Cognitive Science, 4(4), 525-553, 2012.

[184] E. Bigand and P. Perruchet and M. Boyer, Implicit learning of an artificial grammar of musical timbres, Current Psychology of Cognition, 17(3), 577-600, 1998.

[185] Z. Dienes and C. L. Higgins, Can musical transformations be implicitly learned? Cognitive Science, 28(4), 531-558, 2004.

[186] E. G. Schellenberg, E. G. (1996). Expectancy in melody: Tests of the implication-realization model. Cognition, 58(1), 75-125, 1996.

[187] M. Rohrmeier and I. Cross, Artificial grammar learning of melody is constrained by melodic inconsistency: Narmour's principles affect melodic learning, PloS one, 8(7), 1-9, 2013.

[188] M. Rohrmeier, R. Rebuschat and I. Cross, I. (2011). Incidental and online learning of melodic structure. Consciousness and Cognition, 20(2), 214-222, 2011.

[189] B. Tillmann, J. J. Bharucha and E. Bigand, Implicit learning of tonality: a self-organizing approach, Psychological Review, 107(4), 885-913, 2000.

[190] E. Bigand and B. Poulin-Charronnat, Are we "experienced listeners"? A review of the musical capacities that do not depend on formal musical training, Cognition, 100(1): 100-130, 2006.

[191] A. D. Endress, Learning melodies from non-adjacent tones, ActaPsychologica, 135(2), 182–190, 2010.

[192] B. Tillmann, Implicit investigations of tonal knowledge in non-musician listeners, Annals of the New York Academy of Science, 1060(1), 100–110, 2005.

[193] The chord progression in major scale, http://www.electricchili.com/wp-content/uploads/2010/06/ScreenHunter_13-Jun.-20-08.47.jpg, (Accessed on 25 May 2019).

[194] C. X. Zhai, A Brief Note on the Hidden Markov Models (HMMs), Department of Computer Science, University of Illinois, USA, March 2003.

[195] K. Smith, Hidden Markov Models in Bioinformatics with Application to Gene Finding in Human DNA, Machine Learning Project, 308-761, 17 January 2002.

[196] J. McCormack, (1996). Grammar-based music composition, Complex Systems, 3, 321-336, IOS Press, 321-336, 1996.

[197] G. Pandey, C. Mishra and P. Ipe, Tansen: A system for automatic raga identification, In Proceedings of 1st Indian International Conference on Artificial Intelligence (IICAI), Hyderabad, India, 1350-1363, 18-20 December 2003.

[198] D. Das and M. Choudhury, Finite state models for generation of Hindustani classical music, In Proceedings of eleventh International Symposium on Frontiers of Research in Speech and music (FRSM), Bhubaneswar, Orissa, India, 59-64, 6-7 January 2005.

[199] What is bandish in music, https://www.thehindubusinessline.com/blink/watch/What-isbandish/article20904915.ece., (Accessed on 20 May 2019).

[200] S. Chithra, M. S. Sinith and A. Gayathri, Music information retrieval for polyphonic signals using hidden Markov model, Procedia Computer Science, 46, 381-387, 2015.

[201] C. X. Zhai: A Brief Note on the Hidden Markov Models (HMMs), Department of Computer Science, University of Illinois at Urbana-Champaign, USA, 2003.

[202] M. J. Zaki, C. D. Carothers and B. K. Szymanski, VOGUE: A variable Order Hidden Markov model with duration based on frequent sequence mining, ACM Transaction Knowledge Discovery from Data, 4(1), 1-31, January 2010.

[203] M. Clayton, The rhythmic organization of north Indian classical music: Tal, lay and laykari, Dissertation, University of London, 1992.

[204] S. Bhaduri, O. Das, S. K. Saha and C. Mazumdar, Rhythm analysis of tablā signal by detecting the cyclic pattern, Innovations Syst Softw Eng 11(3):187-195, https://doi.org/10.1007/s11334-015-0247-5, 2015.

[205] M. Wright and D. Wessel, An improvisation environment for generating rhythmic structures based on north Indian "Tal" patterns, In: Proceedings of the International computer music conference (ICMC), Ann Arbor, Michigan, USA, 1-4, 1998.

[206] S. K. Pudaruth, A reflection on the aesthetics of Indian music. In: With Special Reference to Hindustani Raga-Sangita, SAGE Open 6(4):1-13. https://doi.org/10.1177/2158244016674512, 2016.

[207] Kaida in Indian Music, https://swadiksha.com/lessons/view?lesson_id=38. (Accessed 16 July 2019).

[208] Palta in Indian music, https://swadiksha.com/lessons/view?lesson_id=39, (Accessed 16 July 2019).

[209] N. A. Jairazbhoy, The Rāgs of north Indian music: Their structure and evolution, 2nd edn. Popular Prakashan, Bombay, 1995.

[210] Cross Serial Dependencies, https://en.wikipedia.org/wiki/Cross-serialdependencies, (Accessed 16 July 2019).