

Graph Convolutional Neural Networks Approaches for Melodic Pattern Analysis in Arab-Andalusian Music

Alessandro Sellani¹[0009-0006-1710-5337], Niccolò Pretto¹[0000-0002-3742-7150],
and Ivan Donadello¹[0000-0002-0701-5729]

Free University of Bozen-Bolzano, Bozen 39100, Italy
asellani@unibz.it, niccolo.pretto@unibz.it, ivan.donadello@unibz.it

Abstract The majority of Music Information Retrieval (MIR) research is Western-centric, and the limited availability of annotated resources poses a challenge for data-intensive approaches. In this work, we implement data-driven models and analyse their classification performance in two fundamental concepts in Arab-Andalusian music: *nawba* and *tābʿ* using symbolic encoding. To address data scarcity, we employ two data augmentation strategies: sliding window segmentation and graph sub-sampling. We process a dataset of Arab-Andalusian digital scores to extract meaningful symbolic features and provide the resulting dataset for experiment reproduction and further research. Our results show that data-driven Machine Learning approaches provide a significant improvement for the aforementioned classification tasks compared to model-based Artificial Intelligence. Moreover, we introduce a method based on a Graph Convolutional Neural Network (GNN) architecture that exploits the relationships between music components. To the best of our knowledge, this is the first application of a GNN to Non-Western MIR. This work has the potential to set a new baseline for state-of-the-art methods which identify *nawba* and *tābʿ*.

Keywords: Graph Convolutional Neural Network · Mode detection · Music Information Retrieval · Non-Western Music · Digital Scores

1 Introduction

Although Western music dominates computational music research, many regions possess extensively studied classical music traditions. However, these rich cultures are as of yet under-represented in music information models. This renders challenging research which employ data-driven models [16]. Recently, international projects have tackled these challenges, e.g., the CompMusic Project [17], which included a comprehensive list of data-driven approaches to culturally diverse repertoires and implemented several information modelling techniques within the scope of the following five music cultures: Hindustani and Carnatic [11], Turkish-makam [21], Arab-Andalusian (*al-Āla*) [15,9] and Beijing Opera [8].

Building upon this effort, our work leverages the Arab-Andalusian corpus curated by the CompMusic Project to classify two fundamental concepts in the *al-Āla* music tradition: *nawba* and *ṭābʿ*. *Nawabāt* (plural of *nawba*) are suites of instrumental and vocal compositions, which describe a particular time of day along with the emotional state it is meant to invoke [9]. Instead, the *ṭubūʿ* (plural of *ṭābʿ*) refer to the melodic mode of the piece. Although Arab-Andalusian canon prescribes a fixed *nawba* and *ṭābʿ* relation [1], due to the fragmentary nature of the tradition, corpus analysis shows this association to be in some occasions functionally fluid. Hence, we treat *nawba* and *ṭābʿ* classification as independent tasks to avoid artificial constraints being placed on the models, and prevent presupposing dependencies.

This work investigates whether Machine Learning (ML) data-driven models can effectively classify *nawba* and *ṭābʿ* in Arab-Andalusian music within the symbolic domain, despite the limited availability of annotated symbolic data. To address this question, we build on prior work by [15] to include *ṭābʿ* detection. Our evaluation compares the classification accuracy of traditional ML models with event-based Deep Learning (DL) architectures and introduces a novel classification methodology using Graph Convolutional Neural Networks (GNNs). However, DL models are particularly data-hungry and the Arab-Andalusian corpus is limited in number of samples. To address data scarcity, we implement two data augmentation methodologies: overlapping sequence segmentation, and graph sub-sampling. Our study is structured as follows: Section 2 introduces symbolic music classification approaches and current research for Non-Western Music Information Retrieval (MIR). Section 3 provides an overview of the dataset and outlines the pre-processing techniques. Section 4 presents the modelling approaches and their implementation details. The results are evaluated and discussed in Section 5.

2 Background

Section 2.1 provides an overview of research in symbolic music, while Section 2.2 briefly states related works in the field of Non-Western MIR.

2.1 Symbolic Music Classification

Symbolic descriptors consist of high-level musical elements that can be categorized into four main types: pitch, harmony, rhythm and timbre [6]. Although audio recordings are more prevalent than digital scores, a significant advantage of symbolic features lies in its higher level of abstraction, as they are able to express nuanced musical ideas shaped by cultural and societal contexts. Something difficult to capture from audio signals [17]. One of the simplest, although widely used symbolic feature representations are Pitch-Class Distributions (PCDs). They provide a compact histogram of pitch occurrences and are highly effective with various models. Several studies have explored this approach extensively [10,3].

On the other hand, Neural Networks (NNs) possess the ability to reveal complex, hidden structures in data and as such sequence-based NNs. For instance, Recurrent Neural Networks (RNNs) [4] and Convolutional Neural Networks (CNNs) [12] have also been applied in the symbolic domain. Recently, graph-based representations have emerged as a powerful tool for a range of MIR tasks, from music generation [7] to cadence detection [14]. In this work we also test the hypothesis that graph-representations based on popularity measures and note relationships can express musical structures for the Arab-Andalusian music better than statistical aggregates and sequential DL models.

2.2 Non-Western MIR

Most research is still conducted on commercial Western metadata. There is a compelling need to diversify datasets and re-orient MIR. Towards this goal, the CompMusic Project has generated a variety of diverse collections, reinforcing the consensus among ethnomusicologists that analytical approaches for Western music are not universally applicable [17]. In addition to providing the Arab-Andalusian corpus, it also established baseline benchmarks for *nawba* classification which we expand upon in this work.

Classical music traditions vary in their theoretical foundations. For instance, Turkish Makam music employs more elaborate systems, like the 53-Tone Equal Temperament (TET) derived via Ariel theory [10,19], which is distinct from the standard 12-TET Western framework. However, 12-TET, although not universal, finds use in some Non-Western music traditions [5,15].

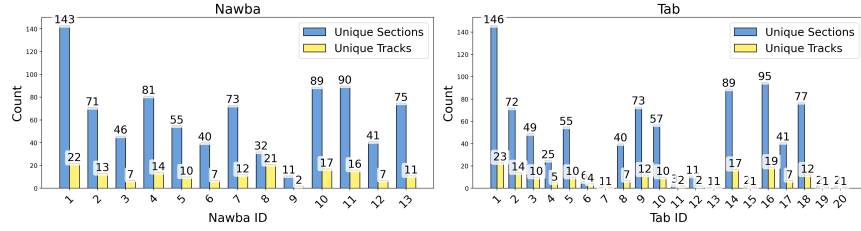
Music is composed of repeated motifs, and pattern detection plays a crucial role in MIR. Automatic pattern retrieval for the *al-Āla*, *Hindustani* and *Carnatic* music traditions has been performed showing promising results [9,11]. For DL models, CNNs have been applied to detect string-playing techniques of Korean *Gugŭn* performances [13], and to classify *tori* by analysing pitch contour with CNNs [12]. Notably, the use of GNN in MIR is relatively new, and no graph-based approaches have been applied to Non-Western music in the literature.

3 Data

Section 3.1 briefly describes the Arab-Andalusian dataset, while Section 3.2 goes over the construction of the data employed for training. Finally, Sections 3.3 and 3.4 explain the preparation and augmentation of the samples, respectively.

3.1 The Arab-Andalusian Dataset

The Arab-Andalusian dataset is a collection of 164 long recordings (56 hours) belonging to the *al-Āla* music tradition, which have been manually annotated and selected by musicologist Amin Chachoo [2]. It was gathered as part of the CompMusic Project and is available on Zenodo (<https://zenodo.org/records/1291776>).

Figure 1: Class distribution of *nawba* and *tab* tracks against annotated sections.

The dataset is divided into three collections: MusicXML scores, audio recordings and lyrics, along with metadata annotations and pitch distribution analysis. However, as this study focuses on symbolic features, only the digital scores are relevant for our task. Out of 164 tracks, 158 are available as MusicXML. Each piece is divided into temporal sections, each assigned with the following labels: *nawba*, *tab*, *mizān* and *form*. The *mizān* refers to the rhythmic mode, which is structurally divided into *forms* (tempi). However, these are not taken into consideration in the analysis. The distribution of tracks and sections (847 in total) per *nawabāt* and *tubū* are presented in Figure 1. For sake of clarity we maintained the IDs from [15]. For further details, see Appendix A, Tables 1-2.

3.2 Data Pre-processing

The dataset described in the previous Section needs to be pre-processed to set up our experiment and train the models. Each track’s XML is parsed with the *music21* library (<https://www.music21.org>). This parsing returns a stream of notes, among other measure features such as tempo, key or presence of chords. Every note is assigned the track and its corresponding section identifier along with their relative labels. Table 1 illustrates the dataset obtained after the extraction of symbolic values from the digital scores.

The pre-processing revealed some discrepancies between the symbolic and audio representations. Each notes’ temporal timestamps in the scores was computed and an average mismatch of ± 7.58 minutes between the digital scores and the audio recordings was found. This result leads to uncertainty when labelling portions of scores to a specific class. The notes which exceed the timestamps given in the annotated metadata have been discarded, removing around 5% of a score’s final section. Analysing the note distribution of the discarded slice against the whole dataset, we find that rests are approximately $3\times$ longer and 44.8% more frequent. Long rests (>4 quarters) are disproportionately represented in the slice (12.7% vs. 2.1% in corpus). This decision could also boost the accuracy of the models, as trimming a track’s last section can aid in classification accuracy [19]. After this selection, we found that each section lasts 966 ± 748 seconds (around 16 ± 12 minutes). There are small sections ranging from 4 to 200 notes, and sections with more than 7000 notes (909 ± 1049 notes).

Table 1: Example of note-stream dataset. *Nawba*, *tāb'*, are excluded for brevity.

	note	time	msr.	N.	dur.	bpm	bt.	Strg.	off.	Tie	chord	Key	length
A4	0.00	1			2.0	85	1.0		0.0	False	False	d minor	95.06
Rest	1.41	1			1.0	85	0.5		2.0	False	False	d minor	95.06
G4	2.12	1			1.0	85	0.25		3.0	True	False	d minor	95.06
G4	2.82	1			1.0	85	1.0		0.0	True	False	d minor	95.06
Rest	3.53	1			1.0	85	0.25		1.0	False	False	d minor	95.06
...		
C4	73	1098			1.0	60	0.125		10.0	True	False	C major	522.80

Another issue to be addressed regards the proper encoding for GNNs, as they need their input to be a graph data structure. Following on [20], a graph G is defined for each single track as a triple (V, E, R) , where V is the set of nodes, R is the set of edge labels, and $E \subseteq V \times V \times R$ is the set of labelled edges. Nodes correspond to the single notes in a track and the edge labels correspond the following relations between pairs of notes/nodes: consecutive notes, parallel notes, temporal overlap between notes and rest-note transitions. For the creation of these graphs, we employ *graphmuse* (<https://github.com/manoskary/graphmuse>), a Python library for Graph DL on Symbolic Music. Every node in the graph is composed of a feature vector with the following values: *[onset beat, duration beat, onset quarter, duration quarter, onset div, duration div, pitch, voice, is downbeat]*. These values are extracted through *partitura* (<https://github.com/CPJKU/partitura>). Each graph is assigned two graph-level features: key, and tempo.

3.3 Data Cleaning and Encoding

In order to guarantee data consistency, the dataset is cleaned by substituting formatting artefacts. For instance, the *Tie* feature (Table 1) uses *False* to represent absent connection between notes. String duration values ('8/3', '1/3') are translated to numerical representations. Pitches are folded onto octave 4, adhering to the Arab-Andalusian practice [15], and converted to MIDI pitch-number. Finally, a merged track discovered in Section 3.1 is discarded. Due to class imbalance, we exclude very under-represented classes from classification. For *nawba*, we keep classes 1–7, 10–13; for *tāb'*, classes 1, 2, 3, 5, 9, 10, 14, 16, and 18. This ID mapping is given in the metadata from the Zenodo dataset. After this selections, 127 and 136 pieces remain for *tāb'* and *nawba*, respectively.

3.4 Data Augmentation

The digital scores in the Arab-Andalusian are relatively few, extremely long and varying in length. Therefore, we augment the data to mitigate class under-representation and avoid overfitting by splitting train and test sets piece-wise. We employ the Sliding Window (SW) methodology to segment stream of notes.

SW is a common algorithmic approach to process streams of data, for which a moving “window” slides upon the stream processing a subset of the data. SW has been found to yield significant improvements of performance, particularly when used in conjunction with CNNs [12] or Pitch Histograms [3]. We decided upon twelve different stream-segmentation combinations given by the Cartesian product of the overlap set $\mathcal{O} = \{0\%, 10\%, 30\%, 50\%\}$ and the sequence length set $\mathcal{L} = \{256, 512, 1024\}$ obtaining 12 combinations total. Table 2 illustrates the results yielded by SW data augmentation for overlap 50%. Across all combinations, data augmentation reached a sample increase of around a maximum of 3067% and a minimum of 178%.

Table 2: SW augmentation. Increase (%) for sequence lengths, w/ 50% overlap.

Ovr.	Len.	<i>nawba</i>		<i>ṭāb‘</i>	
		Samples Increase		Samples Increase	
50%	256	4307	3066.91%	3846	2928.35%
	512	1792	1217.65%	1482	1038.07%
	1024	663	387.50%	591	365.35%

Graph wise, data augmentation utilizes *graphmuse*’s sampling process. Given graph $G = (V, E, R)$, we extract sub-graph $G' = (V', E', R')$ such that $|V'| \leq B$, where B is the sub-graph size parameter s . Thus, sub-graph extraction selects central nodes and samples 3-hop neighbourhoods of decreasing size $\{s \times 0.6, s \times 0.3, s \times 0.1\}$. The sample increase for sub-graphs is shown in Table 3.

Table 3: Graph sub-sampling augmentation. Increase (%) for sub-graph sizes.

Sub. Size	<i>ṭāb‘</i>		<i>nawba</i>	
	Samples Increase		Samples Increase	
128	9972	7748.03%	9972	7223.53%
256	6996	5494.49%	6996	5041.18%

3.5 Feature Engineering and Selection

Machine Learning methods require features. The GNN-based method leverages the information in the graphs, whereas other learning methods require feature engineering. Following [6], we used features taken from these categories: pitch, rhythm, harmony and timbre. As the MusicXML files in the Arab-Andalusian dataset lack the instrument information, no timbre-related features were extracted. Nonetheless, we extract totally 64 engineered features, listed in the supplementary material, Appendix A, Table 4.

4 Methodology

Section 4.1 explains the baseline experiment from [15]. Moreover, Section 4.2 presents the general architecture for DL models and the details for Long Short-Term Memory (LSTM), 1DCNN and GNN.

4.1 Model-based AI

This paper builds upon [15], for which a model matching approach was employed for *nawba* prediction. The baseline experiment was conducted with a total of 77 samples equally distributed across classes. The 12-TET PCD histograms are stretched into 1200 cents. One cent equals $1/100^{th}$ of a semitone and is a common notation to accurately describe changes in pitch. For each class, a model is generated from the training set.

In the original experiment, the Leave One Out (LOO) methodology is employed during cross validation. At every fold one sample per class is left for testing, while other samples are used for training. For each class a model is generated by computing a PCD and applying Gaussian Smoothing. Subsequently, each PCD model of the test set's samples is classified by evaluating it against the class models generated by the training set using L2 distance [15]. However, in this work, we decide to replicate the experiment using the complete, although class-unbalanced, 136 samples for *nawba* and 127 samples for *ṭāb*.

4.2 Deep Learning models for Sequence Data

We define an abstract DL architecture which takes as input a stream of notes designed according to the model. Otherwise, all DL models treat global features, key embedding and classification layers equally with some parameter changes. 1DCNN Classification layer is slightly different (see supplementary material, Appendix B, Table 6). This deliberate decision was made to fairly evaluate the model's capabilities to abstract and characterize sections of tracks.

Figure 2 illustrates the commonalities in architecture between the DL models: LSTM, 1DCNN and GNN. The upper branch labeled *Neural Network for Note-Level Embedding* stands for the model-specific layers which process the temporal notes. That is, LSTM layers for the LSTM Network, 1D Convolutional Layers for the 1DCNN, and SAGEConv layers for the GNN. The *Global Context Vector* represents the engineered features which are then processed by a *Shallow Multilayer Perceptron (MLP)*. Meanwhile, the key is fed to an Embedding, a dense vector which learns key relationships without assuming an inherent order. Finally, the vectors are concatenated into two *Fully Connected Classification Layer* which outputs classes' probabilities. Note that batch normalization is applied only for the GNN and 1DCNN, as Layer Norm was employed for the LSTM model. We employ a Long Short-Term Memory (LSTM) model, an enhanced RNN which replaces the hidden state with a single memory cell composed of three functional gates.

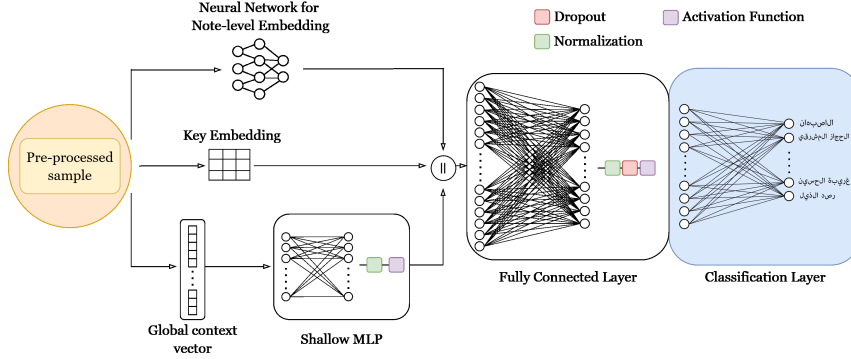


Figure 2: General architecture employed for the three specific DL models.

Furthermore, we design a 1DCNN inspired by [12]. Three stacked 1D Conv. layers of increasing size elaborate the sequences of notes involving: batch normalization, dropout, activation function (ReLU) and Max Pooling, which is used to reduce the dimensionality of the input by pooling the greatest element block-wise. Finally, we build upon a GNN class of *graphmuse* to implement a task-specific network. The architecture incorporates both node-level and engineered features which are pooled and concatenated with global features. The model is composed of multiple SAGEConv layers which operate on a shared set of edge type through Hetero Convolution. Each convolution is followed by batch normalization and dropout layers to stabilize training and prevent over fitting.

Across all DL models, we use decreasing learning rate to prevent the model from getting stuck in local minima, with weight decay and dropout to prevent over fitting. Finally, we use class-balanced Cross Entropy. The details of the models' architectures along with their training parameters are listed in the supplementary material, Appendix B, Tables 5 - 9.

5 Evaluation

The experiments aim at showing the differences at *nawba* and *ṭāb'* classification between the baseline [15] (modified to classify also *ṭubū'*), traditional Machine (Deep) Learning methods and GNNs. The Jupyter Notebooks to replicate the experiments are available on GitHub (https://github.com/aleski017/tab-nawba-symbolic_recognition).

5.1 Experimental Setup and Baseline Results

The data is split into training and test sets based on the track prefix ID to prevent data leakage. This ensures that no sections from the same track appear in both the training and test sets as tracks can have choruses, or repeated

Table 4: Baseline Classification Performance.

Model	Prec	Rec	F1
<i>nawba</i> (Original)	78.9	75.3	74.2
$\bar{t}ab'$ (Weighted)	71.3	71.6	71.3
<i>nawba</i> (Weighted)	72.8	72.1	71.6

motifs happening throughout. The experiments used Grid Search for parameter tuning and cross-validated to ensure robust results. We employ k-fold for K-Nearest Neighbors (KNN), Random Forest Classifier (RF) and Support Vector Machine (SVM), while Monte Carlo Cross Validation (MCCV) for Deep Learning (DL) models as trade-off between computational costs and efficiency [18]. High dropout and weight decay was employed to prevent overfitting.

To set a baseline, we use the template matching strategy described in 4.1 to also perform $\bar{t}ab'$ detection. The metrics are computed based on weighted-averaging. Table 4 shows the performances achieved by [15] for the evenly-distributed training set (first row), and the results with class distribution stated in Section 3.3 (second and third row).

5.2 Results for Traditional ML models

Firstly, KNN delivers surprisingly strong classification results, achieving the highest F1 score of 86.6 for $\bar{t}ab'$, while *nawba* classification is slightly less accurate at 85.8. Furthermore, the sequence length and overlap have minimal impact on performance. The accuracy slightly declines as k increased. Instead, RF is sensitive to overlap, with shorter sequences leading to better performance. However, it performs noticeably worse than KNN for both $\bar{t}ab'$ and *nawba* classification tasks, with maximum F1 scores of 82.6 and 82.5 respectively. Finally, Support Vector Classifier (SVC) with a *radial basis function* kernel shows similar results to KNN, achieving maximum F1 scores of 85.6 for $\bar{t}ab'$ and is surprisingly better for *nawba* classification, reaching F1 at 86.9. Similar to KNN, higher overlap slightly improves accuracy, while sequence length has little to no effect.

5.3 Results for Deep Learning models

For 1DCNN, the best F1 score for $\bar{t}ab'$ is 87.0%, whereas *nawba* caps at 83.3%. Sequence lengths of 256 and 512 yield superior performance, while 1024 tends to reduce model effectiveness. High overlap consistently improves results, as the best configuration for both classification tasks is achieved at 50% overlap.

The LSTM model for both classification tasks demonstrates high sensitivity to longer sequences, benefiting from its ability to capture long-term dependencies. Overlap impact does not display any significant pattern. For $\bar{t}ab'$, the highest F1 score is observed at 10% overlap and 1024-length sequences, with an F1 of 85.8%. For *nawba*, the best result is obtained at 50% overlap and 1024-length sequences, with an F1 of 85.4%. The GNN model surpasses both the

Table 5: Cross-model classification summary across all configurations. For GNN the sub-graph size is listed.

Model	$\mathfrak{t}\bar{a}b'$			$nawba$		
	Config	F1	Increase	Config	F1	Increase
Model-based	—	71.3	—	—	71.6	—
KNN	1024, 50%	86.6	+15.3	512, 30%	85.8	+14.2
RF	256, 50%	82.6	+11.3	256, 50%	82.5	+10.9
SVC	256, 50%	85.6	+14.3	512, 30%	86.0	+14.4
1DCNN	512, 30%	87.0	+15.7	512, 50%	83.3	+11.7
LSTM	1024, 10%	85.8	+14.5	1024, 30%	85.4	+14.2
GNN	128	89.7	+18.4	256	87.0	+15.4

1DCNN and LSTM, attaining the highest F1 scores: 89.7% for $\mathfrak{t}\bar{a}b'$ (sub-graph size 128) and 87.0% for $nawba$ (sub-graph size 256). Interestingly, larger sub-sampling sizes reduce performance, suggesting denser graphs are more effective.

5.4 Discussion

On average, $\mathfrak{t}\bar{a}b'$ detection performs around 0.91 ± 1.84 better than that of $nawba$. The consistent gap between $\mathfrak{t}\bar{a}b'$ and $nawba$ classification across all models (besides SVC) infers differences in task complexity. Short-term pattern analysis (1DCNN) achieves higher accuracy on $\mathfrak{t}\bar{a}b'$, while capturing long-term dependencies (LSTM) performs comparably on both tasks; GNNs excel on $\mathfrak{t}\bar{a}b'$ with smaller graphs but on $nawba$ with larger graphs, suggesting that $nawba$ classification requires a longer-range context compared to $\mathfrak{t}\bar{a}b'$. Notably, the GNN model achieves top performance across all configurations, supporting the hypothesis that graph representations express music qualities better than statistical features alone. Moreover, graph sub-sampling boosts data quantity, contributing in higher results.

The best models, vastly improve on the baseline model-based AI accuracies. The 1DCNN achieves a maximum of +15.7% accuracy improvement, and the GNN on $\mathfrak{t}\bar{a}b'$ achieves +18.4% in the best case. In Table 5 we present the best configuration of each model for $\mathfrak{t}\bar{a}b'$ and $nawba$ respectively. Moreover, a comprehensive list of performances for all overlap and sequence length combinations can be found in Appendix C, Tables 10 - 15. Notably, all models process the same global features. However, adding the temporal dimension does not always imply accuracy increase. On the contrary, for LSTM it appears that the note-patterns disturb the model for $\mathfrak{t}\bar{u}b\bar{u}'$.

6 Conclusion

This study aimed to improve upon $nawba$ classification accuracy of model-based AI, while including $\mathfrak{t}\bar{a}b'$ detection. The experimental results clearly demonstrate

that data-driven models in conjunction with engineered features possess good discriminative power and significantly outperform the baseline model-based AI experiment. Data augmentation aids classification of all classes, especially those under-represented. Furthermore, the addition of temporal features slightly enhanced classification performance. Particularly, our best results were achieved with a recent state-of-the-art model, the Graph Neural Network (GNN), which, to our knowledge, has never before been applied to Non-Western music.

Despite the documented limitations of the dataset, the results found are promising—especially in *tāb*’ detection, which consistently outperformed *nawba* classification. We created a more structured and richer dataset from the raw digital sheets that can be used as a basis for future research. Additionally, we include Jupyter notebooks to facilitate exploratory analysis.

As further work, we plan to resolve formatting inconsistencies between MusicXML and audio files by adopting multimodal approaches. Furthermore, the proposed methodology could be extended to other Non-Western traditions.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Caro Repetto, R., Pretto, N., Chaachoo, A., Bozkurt, B., Serra, X.: An open corpus for the computational research of Arab-Andalusian music. In: Proceedings of the 5th International Conference on Digital Libraries for Musicology (DLfM). p. 78–86. Paris, France (2018). <https://doi.org/10.1145/3273024.3273025>
2. Chaachoo, A.: La música andalusí al-Ála: historia, conceptos y teoría musical. Almuzara (2011)
3. Chawin, D., Rom, U.: Sliding-Window Pitch-Class Histograms as a Means of Modeling Musical Form. Transactions of the International Society for Music Information Retrieval **4**, 223–235 (2021). <https://doi.org/10.5334/tismir.83>
4. Chen, T.P., Su, L.: Functional Harmony Recognition of Symbolic Music Data with Multi-task Recurrent Neural Networks. In: Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR). pp. 90–97. Paris, France (2018), http://ismir2018.ircam.fr/doc/pdfs/178_Paper.pdf
5. Chordia, P., Rae, A.: Raag Recognition Using Pitch-Class and Pitch-Class Dyad Distributions. In: Proceedings of the International Society for Music Information Retrieval Conference (ISMIR). pp. 431–436. Vienna, Austria (2007). <https://doi.org/10.5281/zenodo.1416888>
6. Corrêa, D.C., Rodrigues, F.A.P.: A survey on symbolic data-based music genre classification. Expert Systems with Applications **60**, 190–210 (2016). <https://doi.org/10.1016/j.eswa.2016.04.008>
7. Cosenza, E., Valenti, A., Bacciu, D.: Graph-based polyphonic multitrack music generation. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI). pp. 643–651. Macao, P.R. China (2023). <https://doi.org/10.24963/ijcai.2023/643>
8. Dzhambazov, G.B., Yang, Y., Caro Repetto, R., Serra, X.: Automatic alignment of long syllables in a cappella Beijing opera. In: Proceedings of the International

- Workshop on Folk Music Analysis. pp. 88, 91. Málaga, Spain (2016). <https://doi.org/10.21427/d7gr04>
9. Garcia Casado, M.: Computational Analysis of Arab-andalusian Music Through Symbolic Pattern Discovery Algorithms. Master Thesis, Universitat Pompeu Fabra, Barcelona (2020). <https://doi.org/10.5281/zenodo.4091429>
 10. Gedik, A., Bozkurt, B.: Pitch-frequency histogram-based music information retrieval for Turkish music. *Signal Processing* **90**, 1049–1063 (2010). <https://doi.org/10.1016/j.sigpro.2009.06.017>
 11. Gulati, S., Serrà, J., Ishwar, V., Serra, X.: Discovering rāga motifs by characterizing communities in networks of melodic patterns. In: *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pp. 286–290. Shanghai, China (2016). <https://doi.org/10.1109/ICASSP.2016.7471682>
 12. Han, D., Caro Repetto, R., Jeong, D.: Finding tori: Self-supervised learning for analyzing korean folk song. In: *Proceedings of the 24th International Society for Music Information Retrieval Conference (ISMIR)*. pp. 440, 447. Milan, Italy (2023). <https://doi.org/10.5281/zenodo.10265319>
 13. Huang, Y.F., Liang, J.I., Wei, I.C., Su, L., et al.: Joint analysis of mode and playing technique in Guqin performance with machine learning. In: *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*. pp. 85–92. Montreal, Canada (2020). <https://doi.org/10.5281/zenodo.4245378>
 14. Karystinaios, E., Widmer, G.: Cadence Detection in Symbolic Classical Music using Graph Neural Networks. In: *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*. pp. 917, 924. Bengaluru, India (2022). <https://doi.org/10.5281/zenodo.7316814>
 15. Pretto, N., Bozkurt, B., Caro Repetto, R., Serra, X.: Nawba recognition for Arab-Andalusian music using templates from music scores. In: *Proceedings of the 15th Sound and Music Computing Conference, SMC 2018*. p. 394 – 399. Limassol, Cyprus (2018). <https://doi.org/10.5281/zenodo.1257388>
 16. Serra, X.: A Multicultural Approach in Music Information Research. In: *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*. pp. 151–156. Miami, Florida, USA (2011). <https://doi.org/10.5281/zenodo.1416591>
 17. Serra, X.: Creating research corpora for the computational study of music: The case of the CompMusic project. In: *Proceedings of the 53rd AES International Conference on Semantic Audio*. Audio Engineering Society, London, UK (2014)
 18. Wilimitis, D., Walsh, C.G.: Practical Considerations and Applied Examples of Cross-Validation for Model Development and Evaluation in Health Care: Tutorial. *JMIR AI* **2**, e49023 (2023). <https://doi.org/10.2196/49023>
 19. Yesiler, F., Bozkurt, B., Serra, X.: Makam recognition using extended pitch distribution features and multi-layer perceptrons. In: *Proceedings of the 15th Sound and Music Computing Conference*. pp. 249–253. Limassol, Cyprus (2018). <https://doi.org/10.5281/zenodo.1422577>
 20. Zhang, H., Karystinaios, E., Dixon, S., Widmer, G., Cancino-Chacón, C.: Symbolic Music Representations for Classification Tasks: A Systematic Evaluation. In: *Proceedings of the 24th International Society for Music Information Retrieval Conference (ISMIR)*. pp. 848–858. Milan, Italy (2023). <https://doi.org/10.5281/zenodo.10265421>
 21. Ünal, E., Bozkurt, B., Karaosmanoğlu, M.K.: A Hierarchical Approach to Makam Classification of Turkish Makam Music, Using Symbolic Data. *Journal of New Music Research* **43**(1) (2014). <https://doi.org/10.1080/09298215.2013.870211>