ADAPTING DEEP LEARNING-BASED METER TRACKING TO CARNATIC MUSIC

Satyajeet PRABHU (satyajeet.prabhu01@estudiant.upf.edu) [1],
Thomas NUTTALL (thomas.nuttall@upf.edu) [1], **Vivek VIJAYAN** (vivek.vijayan01@estudiant.upf.edu) [1],
and **Martín ROCAMORA** (martin.rocamora@upf.edu) [1]

[1] *Music Technology Group*, **Universitat Pompeu Fabra**, Barcelona, Spain

ABSTRACT

Beat and downbeat tracking, jointly referred to as meter tracking, are fundamental tasks in the computational analysis of musical rhythm. However, state-of-the-art models often perform less reliably on musical traditions underrepresented in their training data. The South Indian musical tradition known as Carnatic music is one such case. In this study, we evaluate two state-of-the-art meter-tracking models–the *Temporal Convolutional Network* and *Beat This!*–in the Carnatic music context, and explore musically informed strategies to adapt them to this tradition.

Our results show that while off-the-shelf models do not always outperform the existing baseline for Carnatic music, their performance improves substantially with domain-specific training or transfer learning, matching or surpassing the existing state-of-the-art, and demonstrating that deep learning models can be effectively adapted to underrepresented traditions. Our code is made publicly available, and the Carnatic meter-tracking models are released as part of an existing tools repository for the computational analysis of Carnatic music.

1. INTRODUCTION

Rhythm in music is perceived as pulsations organized at multiple hierarchical levels of differing timespans, known as its *meter* or *metrical structure* [1, 2]. The *beat* (tactus) is the perceptually most salient pulse level to which a listener would naturally tap their foot and usually serves as the reference level for tempo. The *measure* (also known as a bar) refers to the grouping of beats into a recurring structure that establishes musical phrasing and form and shapes listeners' expectations of timing. In Western (Eurogenetic) music, meter is commonly represented by time signatures, such as 3/4, indicating three beats per measure, with each beat typically a quarter note. The first beat of a measure is known as the *downbeat*. It is often marked with a strong accent to convey the start of a measure.

Computational rhythm analysis–the extraction and analysis of rhythmic information from musical audio–is a central topic in Music Information Retrieval (MIR), and encompasses tasks such as tempo and time signature estimation, pattern discovery and microtiming analysis, among others. A key rhythm analysis task is *meter tracking*, the automatic estimation of beats and downbeats, important for downstream MIR tasks, as well as for applications including DJ mixing and automatic beat matching. State-of-the-art deep learning models for meter tracking typically require large amounts of high-quality annotated training data, which can be expensive and time-consuming to acquire, particularly in musical traditions with complex rhythmic structures. The South Indian music tradition of Carnatic music is one such example: its frequent use of uncommon meters, improvisation, microtiming, and syncopation poses a challenge for systems relying on fixed rhythmic patterns. And, as we demonstrate in this paper, state-of-the-art meter-tracking models do not generalise well to the Carnatic music context, likely due to the relative underrepresentation of Carnatic music in their training data.

Hence, Carnatic music offers an ideal case study for moving beyond one-size-fits-all approaches towards adapting meter-tracking methods to account for the complexities of non-Western musical traditions. In this work, we explore strategies such as musically informed training and fine-tuning of state-of-the-art meter-tracking models in the Carnatic music context, and show that, while these models underperform when applied out-of-the-box, they surpass the established benchmark for Carnatic music with customised training and transfer learning on Carnatic data. These improvements have important implications for higher-level MIR tasks such as music segmentation and structural analysis, and help bridge the gap between the state-of-the-art in computational rhythm analysis and the rhythmic diversity of global musical cultures.

1.1 Rhythm in Carnatic Music

Carnatic music, one of the two principal traditions of Indian art music, has become an important subject in MIR research as it presents unique challenges and opportunities for computational analysis [3]. Rhythmic organization in Carnatic music is governed by the *tāḷa* system [4]–a hierarchical framework of time cycles that underlies melodic and rhythmic phrasing, as well as improvisation. The tāḷa framework includes unique elements that lack direct equivalents in Western metrical theory, resulting in 175 theoretically possible tāḷas. Despite the differences, there exist some conceptual parallels with Western metrical struc-

Copyright: © 2026. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Tāḷa	#Beats
Ādi	8
Rūpaka	3
Miśra chāpu	7
Khaṇḍa chāpu	5

Table 1. Popular *tāḷas* in Carnatic music.

tures. The fixed-length tāḷa cycle, known as *āvartana*, is comparable to the measure. The basic equidistant pulse of an āvartana is called *akṣara*. The first akṣara pulse of the tāḷa cycle, *sama*, corresponds to the downbeat. While no direct equivalent of a beat exists, it is adapted to mean a grouping of akṣaras into equal length units [5]. Fig. 1 illustrates some of these concepts using the 8-beat ādi tāḷa cycle. These parallels, though not exact, provide a conceptual bridge between Western metrical hierarchies and Carnatic tāḷa organization, enabling the adaptation of meter-tracking systems originally developed for Western music to the Carnatic context. Table 1 lists the four commonly employed tāḷas in Carnatic music that are studied in this work, alongside their total number of beats per cycle.

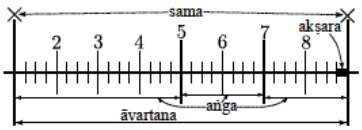


Figure 1. Illustration of an ādi tāḷa cycle; from [5].

Carnatic music poses unique challenges for meter-tracking systems due to its intricate rhythms. Western popular music primarily features time signatures such as 3/4 and 4/4, whereas irregular meters like 5- and 7-beat cycles are commonly encountered in the Carnatic repertoire (as seen in Table 1). The reliance on vocal cues (*konnakol*) and hand gestures to convey rhythm, rather than continuous percussion patterns, often leads to the absence of explicit percussion-based timekeeping characteristic of Western music. The frequent use of improvisation, along with subtle variations in timing (microtiming), present a challenge for systems relying on fixed rhythmic patterns. The use of syncopation, creative phase offsets (*eḍupu*), and polyrhythms further complicates accurate meter tracking. Lastly, the definition of a beat as an isochronous pulse in the western paradigm does not extend directly to Carnatic music. Based on the tāḷa, the foot tapping pulse can align either with the isochronous pulse or with non-isochronous section (*aṅga*) markers. For the purposes of MIR methodologies, we adopt the former interpretation.

2. RELATED WORK

Beats and downbeats are intrinsically linked and the natural relationship between them can be exploited. Probabilistic frameworks such as Probabilistic Graphical Models (PGM) [6] can be used to jointly model hidden variables (the beats and downbeats) from observations (the novelty function). A *Dynamic Bayesian Network* (DBN) is a type of PGM that specifically models temporal processes [7] and can also explicitly incorporate musical knowledge such as typical tempo ranges and metric structures learned from the training data. The *bar pointer model* [8] is a specific type of DBN that has been successfully applied to meter tracking. Its effectiveness in inferring meter has been demonstrated on culturally diverse music corpora, including ballroom dance music [9], as well as Turkish, Greek, and Carnatic music [10]. The model also features prominently in Srinivasamurthy's 2016 doctoral thesis on automatic rhythm analysis of Indian art music [5], which remains the most comprehensive study of meter tracking in Carnatic music to date.

Currently, data-driven deep learning approaches dominate the landscape for meter-tracking. Deep Neural Networks (DNNs) first learn features from audio and output an *activation* function containing possible beat and downbeat candidates. The output activations of DNNs are often noisy and cannot be directly used for predictions. DBNs, which impose temporal consistency and metrical structure, are commonly used as a post-processing step to infer beats and downbeats from the activations. However, due to their inherent properties, DBNs struggle with music containing tempo and time signature changes outside the prescribed range, as well as metric structures not represented in the state space. To overcome these limitations and generalise across music genres, recent efforts have attempted to remove this post-processing stage [11, 12].

Among DNNs, sequence models are best suited for beat and downbeat estimation as they capture the temporal structure of music. The Recurrent Neural Network (RNN), specifically BLSTM, has been applied to join beat and downbeat detection [13]. The Temporal Convolutional Network (TCN) is another powerful architecture that has been repurposed for beat tracking [14]. Unlike BLSTMs, TCNs enable parallel training across time steps, reducing training times and computational costs. More recently, the transformer architecture has been applied successfully to meter tracking [15]. This work focuses on two architectures–*TCN* [16], with an improved design that enables multi-task learning for beat and downbeat estimation, and *Beat This!* [12], a transformer-based system that removes the post-processing stage and achieves state-of-the-art beat and downbeat tracking performance on a number of Western music datasets.

Fig. 2 shows a high level comparison of the two architectures. A TCN-based meter tracker consists of a convolutional front-end for extracting features from spectrograms and a TCN sequence model that captures temporal structure using dilated convolutions, enabling efficient modelling of long contexts such as bars or phrases. The general representation of rhythmic content learned by these two blocks feeds separate task heads for beat and downbeat prediction in a multi-label classification setup. Finally, the activations from the individual heads are jointly post-processed using a DBN to predict discrete beat and downbeat positions. TCN-based systems achieve strong performance with modest data and have been successfully adapted to diverse styles and underrepresented me-

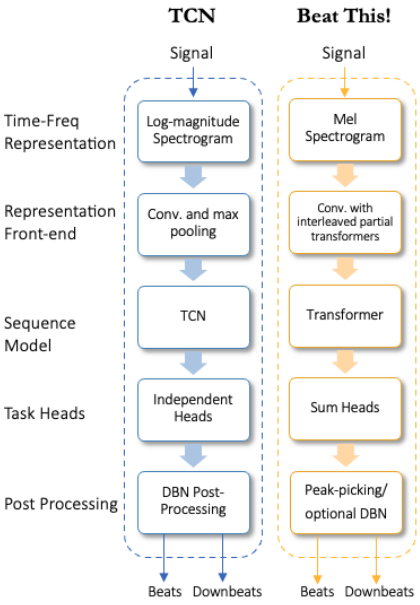


Figure 2. Comparison of TCN and Beat This! architectures.

ters through transfer learning and augmentation [17–19]. In contrast, the front-end in Beat This! combines convolutional layers with partial transformers enabling the model to integrate both timbral and rhythmic information early in the network. The transformer sequence model in Beat This! employs attention to capture both local and global dependencies directly. Additional mechanisms such as the Sum Head and a shift-tolerant BCE loss promote consistent, sharp predictions without post-processing. This approach, however, relies on extensive training data to generalise better across styles and requires more computational resources due to a larger model.

3. DATASETS

We use a diverse set of beat and downbeat annotated datasets that are commonly employed in meter tracking research to pre-train the TCN on Western music for subsequent fine-tuning. The RWC Classical, Jazz, and Pop datasets [20] comprise 50 classical, 50 jazz, and 100 pop music tracks, totaling approximately 17 hours of audio. The Beatles dataset [21] consists of 180 songs by The Beatles, mainly in the pop style, with a total duration of about 8 hours. The Ballroom dataset [9, 22] includes 698 tracks spanning eight ballroom dance styles, each with a mean duration of approximately 30 seconds. The GTZAN dataset [23, 24] contains 1,000 audio excerpts evenly distributed across 10 genres–blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock–each 30 seconds long, for a total of approximately 8.3 hours. In total, these datasets amount to approximately 39 hours of audio.

For training and transfer learning on Carnatic data, we use the Carnatic Music Rhythm dataset (CMR_f) [25], a rhythm-annotated corpus of Carnatic music performances. Annotation timestamps mark metrical positions in the tala cycle, including the sama (downbeat) and other beats. The

Tāḷa	#Pieces	Total Duration hrs (min)	Median Time	#Ann.	#Sama
Ādi (8)	50	4.21 (252.78)	4m51s	22793	2882
Rūpaka (3)	50	4.45 (267.45)	4m37s	22668	7582
Miśra chāpu (7)	48	5.70 (342.13)	6m35s	54309	7795
Khaṇḍa chāpu (5)	28	2.24 (134.62)	4m25s	21382	4387
Total	176	16.61 (996.98)	5m04s	121602	22646

Table 2. Summary of Carnatic Music Rhythm dataset.

dataset also provides metadata specifying the tāḷa employed in each piece. The corpus comprises both vocal and instrumental recordings spanning a variety of musical forms, with all pieces containing percussion, and totals approximately 16.6 hours of audio. A summary of the dataset is provided in Table 2. We utilize the *mirdata* Python library [26] for standardized dataloading of all datasets used in this study.

4. METHODOLOGY

In this paper, we evaluate TCN and Beat This! for beat and downbeat detection in Carnatic music and assess how well they can be adapted to this tradition using transfer learning strategies previously applied to Latin American music [17]. We compare the performance of these models out-of-the-box to the existing Carnatic meter tracking benchmark, the bar pointer model [5], and explore two adaptation strategies–*fine-tuning* and *training from scratch*–to determine whether they further improve performance.

4.1 Bar Pointer Model Baseline

The baseline for this study is the bar pointer model evaluated on the uninformed meter tracking task using the CMR_f dataset [5]. We use results reported for two different inference strategies–*Viterbi Algorithm (BP-HMM)* and *Auxiliary Mixed Particle Filter (BP-AMPF)*. The experimental setup in the reference work employs two-fold cross-validation using pre-determined train and test folds, ensuring no overlap between the data in each fold. Each fold is constructed such that both the training and test sets contain an equal number of pieces (88 each). To account for variability in probabilistic modeling, the original study performs three runs of the cross-validation process, and the final performance metrics are reported as the mean over these runs. The training data is pooled from all tāḷas in the dataset, ensuring that the model is exposed to a diverse range of rhythmic patterns, while preserving the tāḷa distribution within each fold. Specifically, the tāḷa distribution in the training and test folds mirrors that of the full dataset.

4.2 Training Strategies

For both models under evaluation, we replicate the data splits and training setup used in the bar pointer baseline described above. In addition, we establish common procedural guidelines to ensure a fair and consistent comparison between the benchmark and the evaluated models. The train fold is subdivided into training (80%) and validation

(20%) subsets resulting in 70 training examples and 18 validation examples within each fold. To ensure reproducibility of validation splits and network initializations, we use fixed seeds (42, 52, 62) for the three runs. Validation loss is employed as the primary metric for monitoring training progress and for early stopping. Training is terminated when no improvement in validation loss is observed.

4.2.1 TCN setup and training:

The TCN experiments are conducted using a reference implementation [1] of [17] provided in the LAMIR 2024 Hackathon tutorial on adapting deep learning models to Latin American Music tasks [27], with modifications for our specific experimental setup. We employ a simple loss function defined as the sum of the binary cross-entropy (BCE) loss for beat and downbeat predictions and do not use any data augmentation schemes. For post-processing network activations, we employ the joint beat and downbeat DBN post-processor [13,28] provided by the *madmom* Python library [29]. We evaluate three training strategies for the TCN ($\approx 72.3k$ parameters). In *Baseline (TCN-BL)*, we train the model on the RWC (Classical, Jazz and Popular), Ballroom, Beatles and GTZAN datasets for Western music to establish its baseline performance on CMR_f. In the *Fine-tuning (TCN-FT)* strategy, the model with the lowest validation loss from TCN-BL is used as a starting point for fine-tuning on CMR_f with the assumption that, although pre-trained on Western datasets, it has learned a representation adaptable to a different musical tradition. *Training from Scratch (TCN-FS)* trains the model directly on CMR_f without any prior pre-training.

4.2.2 Beat This! setup and training:

We adopt only two training strategies for Beat This!. In *Baseline (BeatThis-BL)*, we use the stock implementation [2] provided by the original authors for baseline evaluation. The three pre-trained checkpoints provided with the stock model (*final0*, *final1*, *final2*), trained on a large corpus spanning 18 Western music datasets are evaluated on CMR_f and their mean performance across checkpoints is reported as the Beat This! baseline. In *Fine-tuning (BeatThis-FT)*, we modify the stock implementation to enable fine-tuning, while retaining the default configurations, such as the data augmentation schemes and the built-in shift-tolerant weighted BCE loss. The fine-tuning process begins with the *final0* checkpoint as the pre-trained baseline, which is then fine-tuned on CMR_f. Beat This! ($\approx 20.3M$ parameters), being a transformer-based architecture, is highly data-intensive and makes training from scratch on a single dataset impractical.

For models trained from scratch (TCN-BL and TCN-FS), training is performed for 100 epochs with early stopping patience of 20 and a learning rate of 0.005. For the fine-tuning strategies (TCN-FT and BeatThis-FT), models are fine-tuned over 50 epochs with early stopping patience of 10 and a reduced learning rate of 0.001.

[1] `https://lamir-workshop.github.io/lamir_hackathon/intro.html`

[2] `https://github.com/CPJKU/beat_this`

4.3 Evaluation

The evaluation of meter tracking systems typically involves comparing predicted beat and downbeat times against annotated ground truth [30]. We use F-measure [31] as a measure of accuracy by comparing predictions to ground truth annotations within a fixed temporal tolerance window (commonly ±70 ms). We also use the continuity-based metrics [32, 33], CML_t (Correct Metrical Level - total) and AML_t (Allowed Metrical Levels - total). CML_t is the ratio of correct predictions at the annotated metrical level to the total annotations. AML_t is same as CML_t, but allows metrical ambiguities such as predictions at double tempo, half tempo and at same metrical level but off-phase that are perceptually acceptable. We use the *mir_eval* Python library [34] for computing these measures.

4.4 Musically Informed Training and Post-Processing

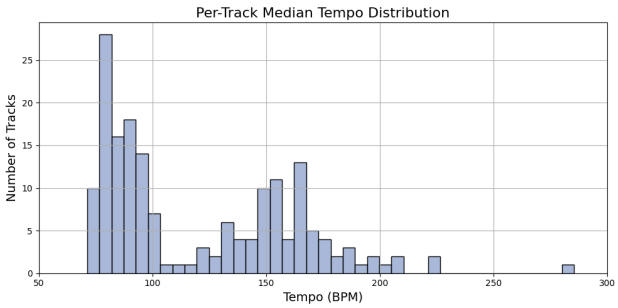


Figure 3. Distribution of tempos in the CMR_f dataset.

We incorporate musicological insights into the meter tracking systems during training and post-processing to enhance performance and improve adaptation to Carnatic Music. To ensure consistent performance across tāḷas, we implement a stratified train/validation split based on tāḷas. This strategy balances the representation of all tāḷas in both training and validation sets, enabling per-tāḷa error analysis and targeted improvements for underperforming tāḷas.

Post-processing is a particularly impactful area for improving meter tracking performance. The DBN post-processor from the *madmom* library allows us to set parameters to better reflect the musical characteristics of the data being processed. We set the beats_per_bar = [3, 5, 7, 8] corresponding to the four tāḷas in our dataset, instead of the default [3, 4]. We constrain the tempo range by setting the minimum tempo to 55 bpm and the maximum tempo to 230 bpm, reflecting the tempo distribution observed in the dataset (Fig. 3). This range, selected based on preliminary experiments, covers 99% of all tempos, while providing a constrained search space for the post-processor.

5. RESULTS

The code required to reproduce the results, analyses, and plots is provided in the respective repositories for the

Model	Beat			Downbeat		
	F-measure	CML_t	AML_t	F-measure	CML_t	AML_t
BP-HMM	71.8	—	72.2	44.0	—	—
BP-AMPF	82.5	—	90.6	57.4	—	—
TCN-BL	77.1	51.6	77.9	28.9	21.6	33.8
TCN-FT	80.7	50.2	**91.9**	52.8	35.3	57.8
TCN-FS	84.6	62.9	87.9	64.2	**52.3**	**67.1**
BeatThis-BL	71.3	39.2	56.8	27.6	2.0	8.7
BeatThis-FT	**90.3**	**78.0**	80.0	**66.8**	38.2	53.7

Table 3. Model-wise performance comparison.

TCN [3] and the modified Beat This! [4]. In addition, the best-performing models trained on the full CMR_f dataset are made available through the CompIAM Python library [35] for computational analysis of Indian art music.

5.1 Model Performance

Table 3 shows the overall performance of the two models and their strategies against the bar pointer model baseline with the highest performing metrics highlighted in bold.

Both the TCN-BL and BeatThis-BL models, trained only on Western music datasets, fail to achieve baseline performance set by the bar pointer model in meter tracking for Carnatic music. While their beat tracking accuracy approaches the baseline, their downbeat tracking accuracy remains substantially lower, despite being trained on extensive datasets. This disparity reveals the fundamental differences in rhythmic structures between Western and Carnatic music and illustrates the challenges faced by the two networks in directly transferring learned knowledge to the Carnatic tradition.

The TCN-FT model nearly attains baseline performance, notably achieving the highest beat AML_t score among all evaluated models. Interestingly, preliminary experiments demonstrated comparable results when fine-tuning a model pre-trained only on the GTZAN dataset, indicating that the quantity of data used during pre-training may be less significant compared to the subsequent fine-tuning on Carnatic music. In fact, the performance of the TCN-FT model may be hindered by being undertrained. Additional fine-tuning could enhance the results, effectively equating to training the model from scratch.

Both TCN-FS and BeatThis-FT significantly outperform the DBN baseline in beat and downbeat tracking, with BeatThis-FT establishing itself as the most effective approach for achieving raw accuracy in meter tracking of Carnatic music. Meanwhile, TCN-FS excels in maintaining temporal continuity, particularly in the downbeat tracking task. The difference in performance between the two models is as expected, given their respective architectures and the application of post-processing in the TCN model. The Beat This! architecture emphasizes the accuracy of local predictions, while the TCN model paired with the postprocessor promotes globally coherent predictions. Additionally, the powerful transformer architecture used by Beat This! is able to extract more meaningful features from the Carnatic data compared to the relatively lightweight TCN model, although this advantage comes with increased computational demands.

[3] https://github.com/satyajeetprabhu/tcn-carnatic-tracker

[4] https://github.com/satyajeetprabhu/beat-this-carnatic

Tāḷa	Beat			Downbeat		
	F-measure	CML_t	AML_t	F-measure	CML_t	AML_t
Ādi (8)	77.8	52.7	84.8	62.9	42.5	84.4
Rūpaka (3)	75.8	32.8	84.9	40.7	19.5	23.8
Miśra chāpu (7)	95.6	92.3	94.0	86.7	88.8	94.5
Khaṇḍa chāpu (5)	93.5	84.5	88.7	69.8	65.9	66.6
Overall	84.6	62.9	87.9	64.2	52.3	67.1

Table 4. TCN-FS: Tāḷa-wise performance comparison.

Tāḷa	Beat			Downbeat		
	F-measure	CML_t	AML_t	F-measure	CML_t	AML_t
Ādi (8)	86.6	74.3	77.5	49.3	2.2	55.8
Rūpaka (3)	89.5	74.7	77.5	81.6	68.3	68.3
Miśra chāpu (7)	94.2	86.0	86.1	72.8	49.7	50.6
Khaṇḍa chāpu (5)	91.4	76.6	78.5	61.1	28.8	29.3
Overall	90.3	78.0	80.0	66.8	38.2	53.7

Table 5. BeatThis-FT: Tāḷa-wise performance comparison.

5.2 Tāḷa-wise Performance

With TCN-FS and BeatThis-FT identified as the two leading strategies for tracking Carnatic meter, we proceed to analyse their performance on each tāḷa, shown in Tables 4 and 5 respectively.

In terms of beat tracking, BeatThis-FT demonstrates relatively consistent performance across tālas compared to TCN-FS. This consistency is also evident in the continuity scores. TCN-FS struggles particularly with ādi (8) and rūpaka (3) tālas, especially the latter. Interestingly, even in Beat This!, both these tālas score lower in beat tracking accuracy than miśra chāpu (7) and khaṇḍa chāpu (5). This may seem counter-intuitive, as one might expect systems to struggle more with rarer and more complex meters. However, the difference is due to the variety of patterns within a given tāla. In Carnatic music, performers often improvise and vary grouping structures within a cycle, while maintaining the core framework and overall length. As explained in [5], popular tālas like ādi and rūpaka tend to have more such variations, making them more difficult for beat tracking systems to generalize. This complexity is visible in the plots in Fig. 4, which compares the average cycle length spectral flux patterns for ādi and miśra chāpu tāḷas in the CMR_f dataset. The patterns indicate varying accent strengths at different metrical positions, reflecting the rhythmic variation within each tāla. Also, both models achieve high performance on miśra chāpu for both beat and downbeat detection. This consistency implies that miśra chāpu's rhythmic pattern is relatively easier to model accurately for both architectures.

The results for downbeat detection performance are more nuanced. While Beat This-FT outperforms TCN-FS in overall F-measure (66.8 vs 63.9), both exhibit mixed re-

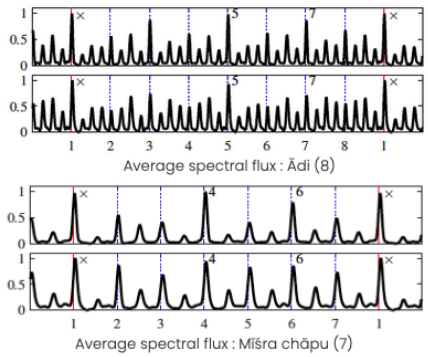


Figure 4. Spectral flux comparison of tālas; from [5].

sults with wide variation across tāḷas. For example, Beat This-FT excels dramatically on rūpaka, whereas it struggles on ādi downbeats. This suggests that each model may be more adept at handling certain rhythmic structures but less consistent across all tāḷa types.

5.3 Error Analysis

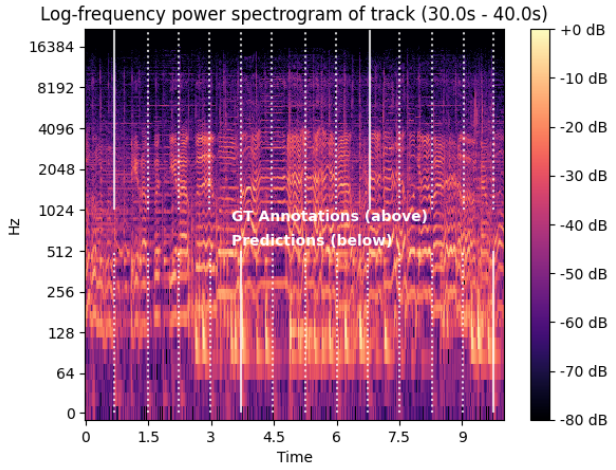


Figure 5. Track '10048': Lowest downbeat F-measure for ādi tāḷa.

Due to the relatively poor performance of TCN-FS on the ādi (8) and rūpaka (3) tāḷas, we analyse the tracks with the lowest beat and downbeat F-measure scores for these two tāḷas. For the ādi tāḷa track with the lowest beat F-measure (track ID '10047'), the beat predictions are consistently shifted by half a beat. For the ādi tāḷa track with the lowest downbeat F-measure (track ID '10048'), the detected downbeats are consistently displaced by half a cycle. Fig. 5 shows the ground-truth annotations and model predictions overlaid on the spectrogram of track '10048'. Although its downbeat F-measure is zero, the track achieves a perfect AML_t score. In both cases, the predictions are systematically offset rather than incorrect making the AML_t score more representative of performance. This discrepancy is likely due to the use of post-processing, which can result in scores unrepresentative of the model's actual performance.

The rūpaka tāḷa outliers were analysed through listening,

Tāḷa	min_tempo=55, max_tempo=125		min_tempo=125, max_tempo=230	
	Beat F	**Downbeat F**	**Beat F**	**Downbeat F**
Ādi (8)	**86.5**	**63.0**	63.0	55.6
Rūpaka (3)	**93.1**	**57.6**	64.5	28.3
Miśra chāpu (7)	64.4	56.1	**96.6**	**85.7**
Khaṇḍa chāpu (5)	64.3	38.7	**95.3**	**71.6**

Table 6. TCN-FS: Tāḷa-wise performance for different post-processor tempo range settings.

as the causes of their poor performance are more difficult to identify visually. In the beat tracking outlier, the percussion alternates between triple and quadruple meter through metric modulation, challenging the post-processing's ability to handle these rapid shifts. The downbeat outlier is particularly difficult because the percussion is performed at twice the tempo of the ground-truth annotations while simultaneously employing polymeter. The subpar performance of TCN-FS on rūpaka tāḷa is likely due to the challenges these variations pose for the post-processing stage.

5.4 Tempo and Cycle Duration Effects

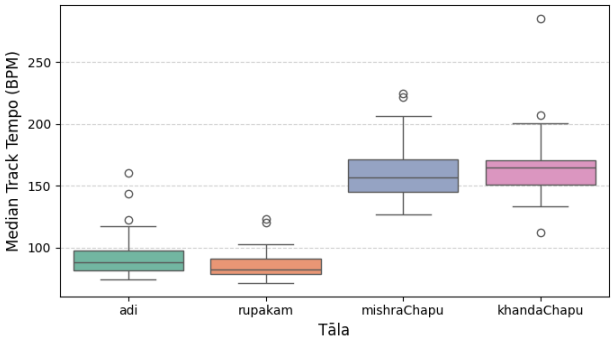


Figure 6. Distribution of median track tempo by tāḷa.

Fig. 6 shows a box plot of the median track tempos in CMR_f, grouped by tāḷa. Notably, majority of tracks in the ādi (8) and rūpaka (3) tāḷas fall within a narrow tempo range of approximately 50 to 125 bpm. In contrast, the two best-performing tāḷas, miśra chāpu (7) and khaṇḍa chāpu (5), cluster around 160 bpm but exhibit a wider tempo distribution. TCN, which relies on post-processing, operates over a broad range of tempos to accommodate the four tāḷas. Narrowing this range based on specific tāḷas significantly enhances TCN performance on these tāḷas, as shown in Table 6. For instance, constraining the tempo range to suit ādi and rūpaka tālas results in significant gains and even pushes TCN performance on these tāḷas past Beat This!. This is useful in cases such as tāḷa-informed meter tracking but also highlights the importance of tempo profiling as a preparatory step for post-processor dependent tracking systems.

Next, we measure the cycle duration for each track as the interval between consecutive downbeats. Table 7 summarizes the cycle durations by tāḷa in CMR_f. The ādi tāḷa distinctly features longer and slower cycles compared to

Tāla	Min. cycle duration (s)	Max. cycle duration (s)	Median cycle duration (s)
Ādi (8)	2.9	7.1	5.4
Rūpaka (3)	1.2	3.1	2.1
Miśra Chāpu (7)	1.6	3.6	2.6
Khaṇḍa Chāpu (5)	0.9	2.9	1.8

Table 7. Tāla-wise summary of cycle durations.

the others, with a median cycle duration of 5.4 seconds–more than double that of rūpaka (2.1s) and khaṇḍa chāpu (1.8s). This has important implications for downbeat tracking: longer cycle lengths demand a larger context window for accurate detection, which can increase the complexity of the model's task. Additionally, longer cycles mean fewer downbeat annotations per track, potentially limiting the amount of training data available to the network for learning. Another insight is that the variability in cycle duration is also greatest in ādi tāḷa, with a range from 2.9 to 7.1 seconds. Taken together, the longer cycle lengths and higher variability of ādi tāḷa may be contributing factors to its lower performance in both models, which we aim to examine in detail in future work.

6. CONCLUSION

In this paper, we adapt two state-of-the-art deep learning models for meter tracking–TCN and Beat This!–to Carnatic music. These models, trained primarily on Western music repertoires, fail to match the performance of the current state-of-the-art for this task in Carnatic music. However we demonstrate that when trained or fine-tuned using Carnatic data, they substantially surpass it. We note that whilst the finetuned, transformer-based, Beat This! exhibits the strongest and most consistent performance, the TCN combined with a musically informed post-processing achieves comparable results with much fewer parameters.

The final best performing models, and new state-of-the-art for meter tracking in Carnatic Music, are published as part of the compIAM repository of computational tools for Indian art music analysis.

Acknowledgments

The authors would like to thank Dr. Ajay Srinivasamurthy for his assistance in replicating the experimental setup of his original work on meter tracking for Carnatic music using the bar pointer model [5].

7. REFERENCES

[1] J. A. Bilmes, "Timing is of the essence : perceptual and computational techniques for representing, learning, and reproducing expressive timing in percussive rhythm," Master's thesis, Massachusetts Institute of Technology, 1993.

[2] J. London, *Hearing in Time: Psychological Aspects of Musical Meter*. Oxford University Press, 2012.

[3] P. Rao, H. Murthy, and S. Prasanna, *Indian Art Music: A Computational Perspective*. Sriranga Digital Software Technologies Pvt. Ltd., 2023.

[4] P. Sambamoorthy, *South Indian Music*, 7th ed. Madras, India: The Indian Music Publishing House, 1973, vol. 3.

[5] A. Srinivasamurthy, "A Data-driven Bayesian Approach to Automatic Rhythm Analysis of Indian Art Music," Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, Spain, 2016.

[6] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.

[7] K. P. Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning," Ph.D. dissertation, University of California, Berkeley, 2002.

[8] N. Whiteley, A. T. Cemgil, and S. J. Godsill, "Bayesian modelling of temporal structure in musical audio," in *Proceedings of ISMIR 2006, 7th International Conference on Music Information Retrieval*, Victoria, Canada, Oct. 2006, pp. 29–34.

[9] F. Krebs, S. Böck, and G. Widmer, "Rhythmic Pattern Modeling for Beat and Downbeat Tracking in Musical Audio," in *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, Curitiba, Brazil, 2013, pp. 227–232.

[10] A. Holzapfel, F. Krebs, and A. Srinivasamurthy, "Tracking the "Odd": Meter Inference in a Culturally Diverse Music Corpus," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014, pp. 425–430.

[11] T.-P. Chen and L. Su, "Toward Postprocessing-Free Neural Networks for Joint Beat and Downbeat Estimation," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022, pp. 27–35.

[12] F. Foscarin, J. Schlüter, and G. Widmer, "Beat this! Accurate Beat Tracking Without DBN Postprocessing," in *Proceedings of the 25th International Society for Music Information Retrieval Conference (ISMIR)*, San Francisco, United States, 2024, pp. 962–969.

[13] S. Böck, F. Krebs, and G. Widmer, "Joint Beat and Downbeat Tracking with Recurrent Neural Networks," in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, New York City, United States, 2016, pp. 255–261.

[14] M. E. P. Davies and S. Böck, "Temporal convolutional networks for musical audio beat tracking," in *2019 27th European Signal Processing Conference (EUSIPCO)*, A Coruña, Spain, 2019, pp. 1–5.

[15] Y. Hung, J. Wang, X. Song, W. T. Lu, and M. Won, "Modeling Beats and Downbeats with a Time-Frequency Transformer," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, 2022, pp. 401–405.

[16] S. Böck and M. E. P. Davies, "Deconstruct, Analyse, Reconstruct: How to improve Tempo, Beat, and Downbeat Estimation," in *Proceedings of the 21th International Society for Music Information Retrieval Conference (ISMIR)*, Montreal, Canada, 2020, pp. 574–582.

[17] L. S. Maia, M. Rocamora, L. W. P. Biscainho, and M. Fuentes, "Adapting meter tracking models to Latin American music," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022, pp. 361–368.

[18] A. Rapini and A. Jordanous, "Beat Tracking for Salsa Music: Adapting and Benchmarking Models Using a Newly Introduced Salsa Dataset," in *Proceedings of the 1st Latin American Music Information Retrieval Workshop (LAMIR)*, Rio de Janeiro, Brazil, 2024, pp. 19–23.

[19] G. Morais, B. McFee, and M. Fuentes, "Skip That Beat: Augmenting Meter Tracking Models for Underrepresented Time Signatures," in *Proceedings of the 1st Latin American Music Information Retrieval Workshop (LAMIR)*, Rio de Janeiro, Brazil, 2024, pp. 34–39.

[20] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Popular, Classical and Jazz Music Databases," in *Proceedings of ISMIR 2002, 3rd International Conference on Music Information Retrieval*, Paris, France, 2002.

[21] M. Mauch, C. Cannam, M. Davies, S. Dixon, C. Harte, S. Kolozali, D. Tidhar, and M. Sandler, "OMRAS2 Metadata Project 2009," in *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, Kobe, Japan, 2009.

[22] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, "An experimental comparison of audio tempo induction algorithms," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1832–1844, 2006.

[23] U. Marchand and G. Peeters, "Swing Ratio Estimation," in *Digital Audio Effects 2015 (Dafx15)*, Trondheim, Norway, Nov. 2015.

[24] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[25] A. Srinivasamurthy and X. Serra, "A Supervised Approach to Hierarchical Metrical Cycle Tracking from Audio Music Recordings," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014, pp. 5237–5241.

[26] R. M. Bittner, M. Fuentes, D. Rubinstein, A. Jansson, K. Choi, and T. Kell, "mirdata: Software for reproducible usage of datasets," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019, pp. 99–106.

[27] G. Morais, R. Namballa, X. Juanola, M. Rocamora, and M. Fuentes, *LAMIR HAckathon: Adapting Deep Learning Models for Latin American Music Tasks*, Dec. 2024. [Online]. Available: https://lamir-workshop.github.io/lamir_hackathon/

[28] F. Krebs, S. Böck, and G. Widmer, "An Efficient State Space Model for Joint Tempo and Meter Tracking," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, Malaga, Spain, 2015.

[29] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "madmom: a new Python Audio and Music Signal Processing Library," in *Proceedings of the 24th ACM International Conference on Multimedia*, Amsterdam, The Netherlands, 2016, pp. 1174–1178.

[30] M. E. P. Davies, N. Degara, and M. D. Plumbley, "Evaluation Methods for Musical Audio Beat Tracking Algorithms," Centre for Digital Music, Queen Mary University of London, London, UK, Technical Report C4DM-TR-09-06, Oct 2009.

[31] S. Dixon, "Automatic Extraction of Tempo and Beat From Expressive Performances," *Journal of New Music Research*, vol. 30, no. 1, pp. 39–58, 2001.

[32] A. P. Klapuri, A. J. Eronen, and J. T. Astola, "Analysis of the meter of acoustic musical signals," *Trans. Audio, Speech and Lang. Proc.*, vol. 14, no. 1, p. 342–355, Dec. 2006.

[33] S. W. Hainsworth, "Techniques for the Automated Analysis of Musical Audio," Ph.D. dissertation, Department of Engineering, University of Cambridge, Cambridge, UK, 2004.

[34] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "mir_eval: A Transparent Implementation of Common MIR Metrics," in *Proceedings of the 15th International Conference on Music Information Retrieval (ISMIR)*, 2014, pp. 367–372.

[35] G. Plaja-Roglans, T. Nuttall, and X. Serra, *compIAM*, 2024. [Online]. Available: https://mtg.github.io/compIAM/