electronics

MDPI

Article

An Efficient Hidden Markov Model with Periodic Recurrent Neural Network Observer for Music Beat Tracking

Guangxiao Song and Zhijie Wang *

College of Information Science and Technology, Donghua University, Shanghai 201620, China
* Correspondence: wangzj@dhu.edu.cn

Abstract: In music information retrieval (MIR), beat tracking is one of the most fundamental tasks. To obtain this critical component from rhythmic music signals, a previous beat tracking system of hidden Markov model (HMM) with a recurrent neural network (RNN) observer was developed. Although the frequency of music beat is quite stable, existing HMM based methods do not take this feature into account. Accordingly, most of hidden states in these HMM-based methods are redundant, which is a disadvantage for time efficiency. In this paper, we proposed an efficient HMM using hidden states by exploiting the frequency contents of the neural network's observation with Fourier transform, which extremely reduces the computational complexity. Observers that previous works used, such as bi-directional recurrent neural network (Bi-RNN) and temporal convolutional network (TCN), cannot perceive the frequency of music beat. To obtain more reliable frequencies from music, a periodic recurrent neural network (PRNN) based on attention mechanism is proposed as well, which is used as the observer in HMM. Experimental results on open source music datasets, such as GTZAN, Hainsworth, SMC, and Ballroom, show that our efficient HMM with PRNN is competitive to the state-of-the-art methods and has lower computational cost.

Keywords: hidden Markov model; periodic recurrent neural network; beat tracking; attention mechanism; deep learning

Citation: Song, G.; Wang, Z. An Efficient Hidden Markov Model with Periodic Recurrent Neural Network Observer for Music Beat Tracking. *Electronics* **2022**, *11*, 4186. https://doi.org/10.3390/electronics11244186

Academic Editor: Andrea Prati

Received: 6 November 2022
Accepted: 13 December 2022
Published: 14 December 2022

1. Introduction

Digital music is becoming increasingly ubiquitous in today's multimedia world. Different from other audio signals, there are patterns occurring recurrently in the flow of music, which is called rhythm. Rhythm is an indispensable element and the most basic aspect of music [1]. Automatic beat tracking becomes a fundamental part of music information retrieval (MIR) and has a wide range of applications, such as music tagging [2–4], genre classification [5–7], emotion recognition [8,9], and music transcription [10–12].

To recognize the periodic pulse-like beat in music signal, machine learning techniques usually extract features, firstly from spectrogram, which is a time-frequency representation of raw music, such as short-term Fourier transform (STFT) [13,14] and Mel-spectrogram [15–17]. Then, the tracking system predicts beat positions based on these features. In the stage of feature extraction, deep learning-based methods [13,18–21] have been adopted instead of the hand-engineered onset detection functions [22,23]. Böck [18] uses bidirectional recurrent neural network to automatically extract features from logarithmic Mel-spectrograms. Later, temporal convolutional networks [24,25] is developed for beat tracking task as well. These deep learning models are trained using ground-truth beat annotations, which is called supervised learning. In the post-processing stage, the period and phase of rhythmic patterns (tempo and beat positions) in musical signal are estimated based on extracted features. Autocorrelation [26,27], comb filters [15,16,28], tempograms [29], and also some sophisticated machine learning methods, such as hidden Markov model (HMM) [18,24,30,31], Gaussian mixture model [32], and support vector machine [33] are widely used.

As one of the most popular post-processing methods among the state-of-the-art beat tracking systems, HMM performs quite well [30,34–37] but with a high space and time complexity. Hidden states of the above models are usually defined as a two-dimensional space of tempo and beat position. Even the space has been optimized by several articles [30,36], but the HMM still has a high computational cost when it performs inference. In order to reduce the number of redundant hidden states of HMM-based beat tracking systems, in this paper, we propose a periodic recurrent neural network (PRNN) that obtains a more reliable tempo prediction and decreases the complexity of hidden space in HMM several times. In PRNN, a module that is able to globally aware the periodic features among the outputs in RNN. As a consequence, the activations of our PRNN result in a more stable and reliable frequency, which precisely estimates the beat period or the tempo in the post-processing stage. To the best of our knowledge, we are the first to leverage the characteristic of rhythmic music signal in end-to-end RNN training in the field of beat tracking. Experimental results on music datasets GTZAN, Hainsworth, SMC, and Ballroom show that our model consists of PRNN and HMM is competitive to the state-of-the-art algorithms in prediction accuracy and has several times lower computational complexity in the post-processing stage, which much reduce the inference time of beat tracking.

To summarize, the main contributions of this paper are as follows:

- We propose an efficient deep neural network and HMM based method to take advantage of the characteristic of period music beat;
- We propose a new module based on attention mechanism called PRNN, which can capture partial period states in RNN;
- We combine the PRNN module with HMM algorithm for beat tracking task and experimental results show the inference time is much reduced.

The rest of the paper is structured as follows. The proposed model is described in Section 3. The related works are introduced in Section 2. The experimental settings are given in Section 4. The results and discussion of experiments are presented in Section 5. The conclusions of this paper are summarized in Section 6.

2. Related Works

In the beat tracking system, music feature extraction is a core part for upcoming learning and beat decoding parts. Feature extraction reduces the size of raw music data. More importantly, it makes machine learning algorithms easier to process. In this section, we review the feature extraction methods including traditional or hand-crafted feature extraction and deep learning-based feature extraction. We review the neural network-based period patter recognition methods, as well as those that are related to our proposed module.

2.1. Traditional Feature Extraction

In music beat tracking algorithms, commonly used traditional music features are harmony, timbre, bass content, rhythmic pattern, and melody. These feature extraction methods are often implemented by time-frequency transform methods in digital signal processing. Harmony refers to the pitches, tones, notes or chords that occur simultaneously in music. To extract harmony, STFT with Hann window, constant-Q transform (CQT), and averaging operations are adopted to the raw music data. Another way to obtain the harmony feature is by using the chroma feature [38] at a constant frame rate, then synchronizing the features to the beat. Furthermore, the similarity of harmonic patterns is related to beat positions and is a commonly used feature. Timbre distinguishes different musical instruments and voices. It is determined by the physical characteristics of sound, including frequency spectrum and envelope. The timbre feature is often represented by Mel-frequency cepstral coefficients (MFCC) and the feature of timbre similarity is computed based on it. As a low-frequency feature, the bass content contains drum kicking and bass instruments that are highly related to the music beat. Low-frequency spectrogram is calculated to obtain this kind of feature. Rhythmic patterns frequently repeat including the sound and silences in music and are often used to represent the bar boundaries. Melodic

constant-Q transform is adopted in [39,40] for feature extraction, and melody consists of a linear succession of musical tones.

2.2. Deep Learning-Based Feature Extraction

The traditional feature extraction methods mentioned above work well when dataset is small. As digital music has grown rapidly in recent years, the complexity and diversity of music data has also improved. The performance of beat tracking system with hand-crafted features becomes unsatisfactory. On demand of the automatic feature recognition in music data, deep neural network based learning methods are utilized in MIR. In [18], bi-directional RNN extracts the features in two directions forming a reversible context for beat decoding processing. Convolutional recurrent neural network (CRNN) was investigated in [19] for the downbeat tracking task. A multi-model of RNNs approach [20] is used for beat tracking. The temporal convolutional network (TCN) was taken for tracking beat [24,25], and in [21] the authors only used visual information. Tempo-invariant convolutional neural networks was proposed in [41]. The model learns rhythmic patterns at the tempi presented in training dataset, which brings a better tempo generalization and network efficiency. A variant of transformer, named spectral-temporal transformer in transformer (SpecTNT), is proposed in [42] for a beat tracking task. The self-supervised training strategy and fully convolutional networks were used in [43] for small and unbalance datasets, and the model improved cross-dataset generalization at no extra annotation cost.

2.3. Periodic Pattern Recognition

Periodic pattern recognition is extended from frequent item mining because the periodic behaviors reveal more important information. For instance, analyzing and identifying the periodic movements of stock market, improving the performance of online recommender system by extracting customer's periodic shopping habits, and finding genes that frequently correlated in DNA sequences, these tasks all heavily rely on cyclic features. Recently, several neural network-based periodic pattern mining methods have been proposed. Pyramidal convolutional recurrent model with periodic representations are fused in [44] for crowd density prediction. In [45], a hop residual recurrent graph neural network are designed to improve the traffic prediction that can find the periodic patterns among the time series information. For learning long-term correlations and periodic relationships from historical traffic data, spatio-temporal convolutional neural network [46] is proposed. The CPU workload of loud virtual machine exhibits both periodic and non-periodic patterns with the sudden peak of load, and Karim et al. [47] propose a hybrid RNN to tackle this problem. In the field of software development, a spatial-temporal graph neural network [48] is utilized for bug triaging process of dynamic developer collaboration network that includes hourly-periodic, daily-periodic, and weekly-periodic components. Attention based encoder–decoder neural network [49] is proposed to predict freeway traffic speed, and the attention weights are extracted which highlights the contribution of daily and weekly periodic input towards speed prediction.

3. Proposed Model

Most of deep learning-based beat tracking system are modeled as a sequence-to-sequence problem with spectrogram input and post-processing for beat correction. In a popular framework proposed in [18], the neural network extracts features from the spectrogram to predict the probability of each frame being a beat. Specifically, normalized activations of bidirectional long-short term memory (LSTM) [50] are used. HMM decodes the beat time using Viterbi algorithm based on observations of previous activations. The HMM and the decoding Viterbi algorithm are explained in Appendix A. In this paper, we improve this framework as shown in Figure 1 by introducing an attention based period mechanism into RNN, called PRNN. Then hidden space in the HMM part is much reduced by leveraging the benefits of PRNN. Next, we describe the PRNN part and its role in framework in detail.

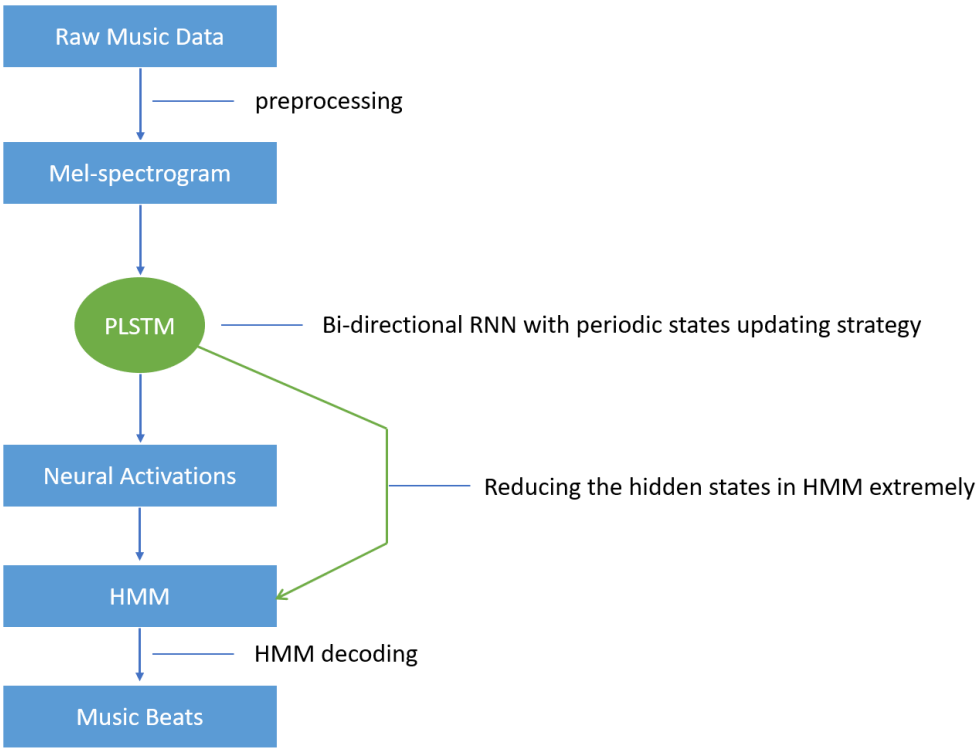

Figure 1. Data flow diagram of the proposed beat tracking system.

3.1. Periodic Recurrent Neural Network

In conventional RNNs, such as LSTM, gated recurrent unit (GRU) [51] mainly solve the problem of gradient vanishing. Hidden states of current time step are based on the states of previous time step and the input of current time step. In music information systems, a large amount of similar features occurs periodically, which is highly related to music beats. Existing RNN methods lack a mechanism to deal with periodic features. In this paper, a variation of attention mechanism is proposed to tackle this problem.

Attention mechanism is widely used in sequence-to-sequence tasks which performs a soft-selection over features in neural network and obtain variable context for different step or time. Formally, given a query vector q and a source sequence $x = [x_1, x_2, \ldots, x_n]$, attention uses an annotation function $\theta(x_i, q)$ that calculates the alignment score between q and x_i. The result of $\theta(x_i, q)$ represents the attention of q to x_i. Next, a softmax function is as follows

$$p(z = i|x, q) = \frac{e^{(\theta(x_i,q))}}{\sum_{i=1}^{n} e^{(\theta(x_i,q))}} \tag{1}$$

normalizes the scores $[\theta(x_i, q)]_{i=1}^{n}$ over the whole source sequence to a probability distribution $z \sim p(z|x, q)$. The attention mechanism described above results in a weighted sum of the input x. Function θ is usually defined as a neural network and we adopt scaled dot-product as the following Equation (2).

$$\theta(x_i, q) = \frac{x_i^T q}{\sqrt{n}} \tag{2}$$

The score is calculated by dot-product of query q and source sequence x_i. A scaling factor of $\sqrt{n}$ is introduced for preventing the case that the softmax function has an extremely small gradient which leads to inefficient learning when the input is large.

In order to make the attention mechanism being capable of perceiving or capturing period features in RNN, we set a matrix $S \in \mathbb{R}^{w \times n}$ for storing historical hidden states, and a hyperparameter w of S, given a fix-length window for retrieving potential similar period

patterns. As shown in Algorithm 1, when time step t is less than window size w, each hidden state h_t has been stored in matrix S. When the length of S equals the window size, scaled dot-product attention described in Equation (2) compares the current hidden state to previous hidden states in S, which captures the similar parts among historical states and stores the result in vector $p \in \mathbb{R}^n$. Then the algorithm fuses the current hidden state with the vector p that mainly contains the similar parts of previous hidden states using a learnable parameter $W \in \mathbb{R}^{n \times n}$. After the first update of hidden state h_t, we do this update when t is in multiples of w until the maximum time step T. When t is not in multiples of w, hidden state h_t is stored into matrix S as a queue operation shown in Algorithm 1.

Algorithm 1 Update of *period states* in PRNN

Require: $1 \leq t \leq T, (t, T) \in \mathbb{Z}, S \in \mathbb{R}^{w \times n}, h_t \in \mathbb{R}^n$
▷ t the time step, T the total time step, S the matrix saves the past hidden states of window size w and h_t is the hidden state of time step t
Ensure: h_t ▷ updated h_t with *period states*
for $1 \leq t < w$ **do** ▷ Initialization
$S_{t,n} \leftarrow h_t$
end for
for $w \leq t \leq T$ **do** ▷ Update *period states* vector $p \in \mathbb{R}^n$
if $t \bmod w = 0$ **then** ▷ Window size w
$a \leftarrow softmax(h_t S^T / \sqrt{n})$ ▷ Scaled dot-product attention
$p \leftarrow aS$ ▷ Attention scores $a \in \mathbb{R}^w$
$h_t \leftarrow h_t + Wp$ ▷ $W \in \mathbb{R}^{n \times n}$ is a learnable parameter
end if
$S_{w-1,n} \leftarrow S_{w,n}$ ▷ Dequeue the first state vector
$S_{w,n} \leftarrow h_t$ ▷ Enqueue the current hidden state h_t
end for

3.2. Periodic LSTM in HMM

LSTM [50] is introduced to overcome gradients vanishing in RNN. In this work, we adopt LSTM embedded with the period states updating module, called PLSTM, in the beat-tracking system. The PLSTM is formalized as following.

$$\begin{aligned} \begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} &= \begin{pmatrix} \sigma(W_i[x_t, h_{t-1}] + b_i) \\ \sigma(W_f[x_t, h_{t-1}] + b_f) \\ \sigma(W_o[x_t, h_{t-1}] + b_o) \\ f(W_g[x_t, h_{t-1}] + b_g) \end{pmatrix} \\ c_t &= f_t * c_{t-1} + i_t * g_t \\ \overline{h}_t &= o_t * f(c_t) \\ h_t &= period\_states\_update(\overline{h}_t) \end{aligned} \tag{3}$$

As shown in Figure 2, the PLSTM part is used in the last layer in multi-layer bi-directional LSTM network. In the stage of HMM decoding, frequency component analysis of the neural activations is processed by discrete Fourier transformation. Then the potential tempi obtained from first k frequency components are used for reducing the number of hidden states in HMM. Importantly, with the proposed PLSTM the neural network is able to capture the similar features in the hidden states, which contributes to the analysis of frequency components and makes a prediction of the potential tempi more precisely. We remove the redundant hidden states in HMM decoding phase using the prior knowledge of the potential tempi. According to [30], the time complexity of the inference part is $T \times F$, where F is the total frames of the music signal and T is transitions at each time step determined by the transition model. In our model, by using the potential beat tempo, the intervals of state space change from 1 to a set of integers $\mathbb{I}$ with larger value than 1 and the length of the set is the hyperparameter k representing the first k-th potential beat tempo.

The time complexity of our post-processing part is roughly $T \times F \ / \ min(\mathbb{I})$. By using Viterbi algorithm and the observation of PLSTM in HMM (Appendix A), the final music beats are decoded.

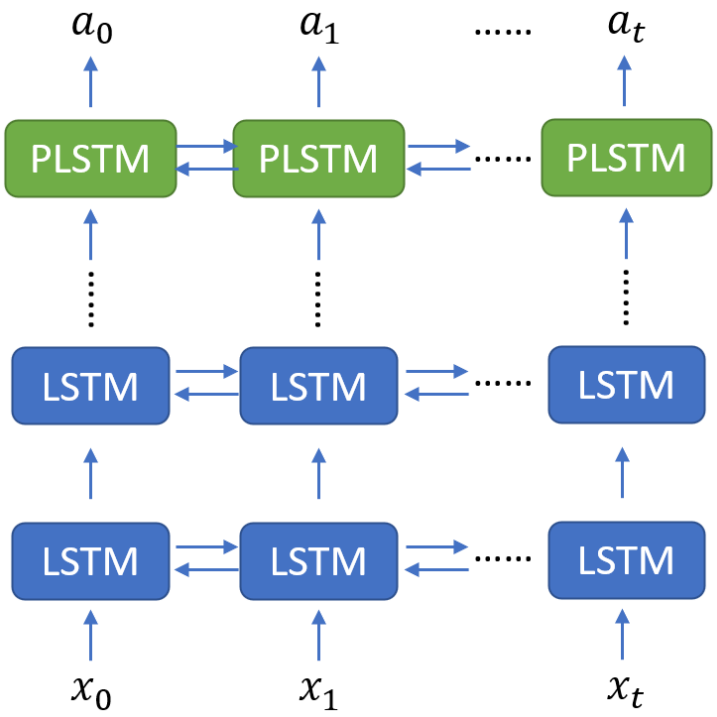

Figure 2. Period states updating module in multi-layer bi-directional LSTM network.

4. Experimental Settings

In experiments, we evaluate the performances of accuracy and inference time of the proposed model with other state-of-the-art models on these datasets as shown in Table 1. Our model is compared with three models, two of them have the same preprocessing method called log magnitude spectrograms using STFT and post processing method of dynamic Bayesian network (DBN) called a bar pointer model [30]. The other model is proposed in [52], the authors process the audio with harmonic/percussive source separation (HP-separation) and multiple fundamental frequency (MF0) estimation. Then use these features are input in a cepstroid invariant neural network and post-processed mainly using CQT. The settings of hyperparameter k representing the number of potential beat frequencies are discussed. The relationships between accuracy and inference time of different k settings are tested as well.

Table 1. Comparison table between the proposed method and other state-of-the-art models.

Models	Preprocessing	Neural Network Type	Post Processing
Bock [25]	STFT	Bi-directional LSTM	DBN
Davies [24]	STFT	Temporal Convolutional Network	DBN
Elowsson [52]	HP separation MF estimation	Cepstroid Neural Network	CQT
Ours	STFT	Period LSTM	DBN

4.1. Datasets

For training and evaluating our model, the following datasets with available beat annotations are used.

- GTZAN
  The GTZAN (http://marsyas.info/downloads/datasets.html, accessed on 11 January 2022) dataset contains 1000 excerpts of 10 genres evenly. Each clip lasts 30 s and the total length is 8 h 20 min. This dataset was originally used for music genre classification and its annotations were extended for beat tracking in [53].
- Ballroom
  The Ballroom (http://mtg.upf.edu/ismir2004/contest/tempoContest/node5.html, accessed on 11 January 2022) dataset has 698 instances of ballroom dancing music with different genres, such as Samba, Tango, Waltz, etc. The duration of each excerpt is 30 s and total length is 5 h 50 min.
- Hainsworth
  In total, 222 excerpts taken direct from CD recordings are in Hainsworth (http://www.marsyas.info/tempo, accessed on 11 January 2022) [54] dataset. The examples have been categorized into different genres, such as dance, pop, and choral. This dataset consists of 3 h 20 min of audio, therefore giving a length of about 60 s for each clip.
- SMC_MIRUM
  The SMC_MIRUM (http://smc.inescporto.pt/data/SMC_MIREX.zip, accessed on 11 January 2022) [55,56] dataset was formed with a purpose of challenging beat tracking systems. Several properties including changes in tempo, wide dynamic range, pauses, poor audio quality, etc., make this set more difficult. In total, 217 valid beat annotations are in this dataset. Each sample lasts 40 s and the total length of audio recordings in this dataset is 2 h and 25 min.

4.2. Preprocessing and Training Settings

We use the same preprocessing and training settings in the four datasets. Then, 8-fold cross validation is used and final test results are the average of all the validations. A Mel-spectrogram with 512 samples of window-size, 256 samples of hop-size, and 96 Mel-bins is applied to each music sample uniformly. In the training process, cross-entropy loss is employed as loss function. The network is trained with Adam [57] optimizer and the momentum is 0.9. We set the learning as 0.01 at the beginning of the training and is reduced by exponential decay function and the factor is 0.1. In total, 50 epochs are trained and we select the model whose validation loss is the minimum value as the best model. In the decoding part, we set the first k frequency components as {3, 5, 10, 15, 20} to explore a balanced value for improving efficiency while keeping accuracy. We also half the length of intervals 2 times when initialising the space of hidden states to create more options for Viterbi decoding. The experiments are implemented on the platform of Intel Core i7-10870H CPU, Nvidia GeForce RTX 3070 GPU, CUDA 11.1, and Pytorch 1.8 framework.

5. Results and Discussion

Table 2 shows the evaluation results on the four music datasets. In each dataset, the first result of Bock is the baseline of our experiments. The F1 scores upgrade on the three datasets except for the GTZAN dataset for music beat tracking. For the CMLt, the proposed model performs the best only on the SMC dataset and performs slightly worse than the other models on GTZAN, Ballroom, and Hainsworth datasets. The last column is the result of AMLt, and shows our model performs better than the others on Ballroom and SMC datasets. On GTZAN dataset, 0.004, 0.014, and 0.002 of F-measure, CMLt, and AMLt values are less than the Bock's model, respectively, but 66 times faster inference process has been obtained when $k = 5$. The similar performances are observed on Ballroom, Hainsworth, and SMC datasets and the accuracies of the proposed model improved due to the reliable frequency of music beat determined by PRNN. Figure 3 displays the beat prediction result on a test music clip with the ground truth values and the neural activations of the proposed PRNN. The green waves represent the activation values that are the outputs of PRNN, and the pulses suggest the potential beat frequency of the music clip. The predicted beats that are represented by the green dots are close to the ground-truth that is represented by

the red dash lines. Table 3 and Figure 4 describes the detailed prediction accuracy and inference efficiency on different settings of k in frequency component analysis with the input of neural activations. As shown, $k = 5$ is a appropriate choice of this hyperparameter and the average F-measure on evaluation datasets reaches a high range. Table 4 indicates that our model has more efficient inference time and needs lower computational resources as well. The average hidden states number in the proposed model is one-tenth less than the baseline model when set $k = 5$ in frequency component analysis. Consequently, the inference time is 70 times faster on these four evaluation datasets. The PLSTM outputs better neural activations than original LSTM for frequency component analysis because of the additional periodic pattern recognition module.

Table 2. Evaluation Results on GTZAN, Ballroom, Hainsworth, and SMC datasets for beat tacking. Evaluation metrics of F-measure, CMLt, and AMLt are shown.

	F-Measure	**CMLt**	**AMLt**
		GTZAN	
Bock [25]	0.864	0.768	0.927
Davies [24]	0.843	0.715	0.914
Ours	0.860	0.754	0.925
		Ballroom	
Bock [25]	0.938	0.892	0.953
Elowsson [52]	0.925	0.903	0.932
Davies [24]	0.933	0.881	0.929
Ours	0.941	0.902	0.959
		Hainsworth	
Bock [25]	0.884	0.808	0.916
Elowsson [52]	0.742	0.676	0.792
Davies [24]	0.874	0.795	0.930
Ours	0.892	0.796	0.928
		SMC	
Bock [25]	0.529	0.428	0.567
Elowsson [52]	0.375	0.225	0.332
Davies [24]	0.543	0.432	0.632
Ours	0.563	0.450	0.685

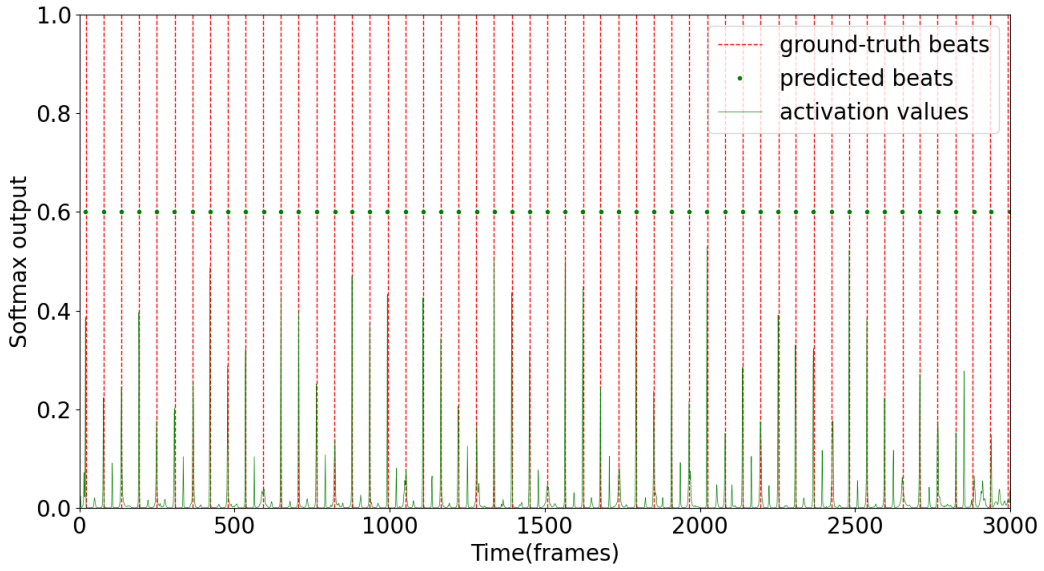

Figure 3. Beat tracking result of our model on a test music clip.

Table 3. "Average inference time | Average F-measure" on different settings of first k frequency components on evaluation datasets.

k	GTZAN	Ballroom	Hainsworth	SMC
3	0.0052 0.795	0.0047 | 0.855	0.0082 | 0.771	0.0065 | 0.482
5	0.0116 | 0.860	0.0106 | 0.941	0.0185 | 0.892	0.0117 | 0.563
10	0.0585 | 0.863	0.0515 | 0.935	0.0311 | 0.881	0.0619 | 0.557
15	0.2509 | 0.861	0.2112 | 0.910	0.3672 | 0.885	0.2781 | 0.560
20	0.4115 | 0.863	0.3932 | 0.925	0.4691 | 0.890	0.4203 | 0.558

Table 4. The average state number and inference time of the proposed model compared with baseline model. The upper two rows are the results of the proposed model with $k = 5$, and the lower two rows the baseline model.

	GTZAN	Ballroom	Hainsworth	SMC
Avg. State Number	400	344	342	414
Avg. Inference Time	0.0116	0.0106	0.0185	0.0117
State Number		5617		
Avg. Inference Time	0.775	0.778	1.282	0.776

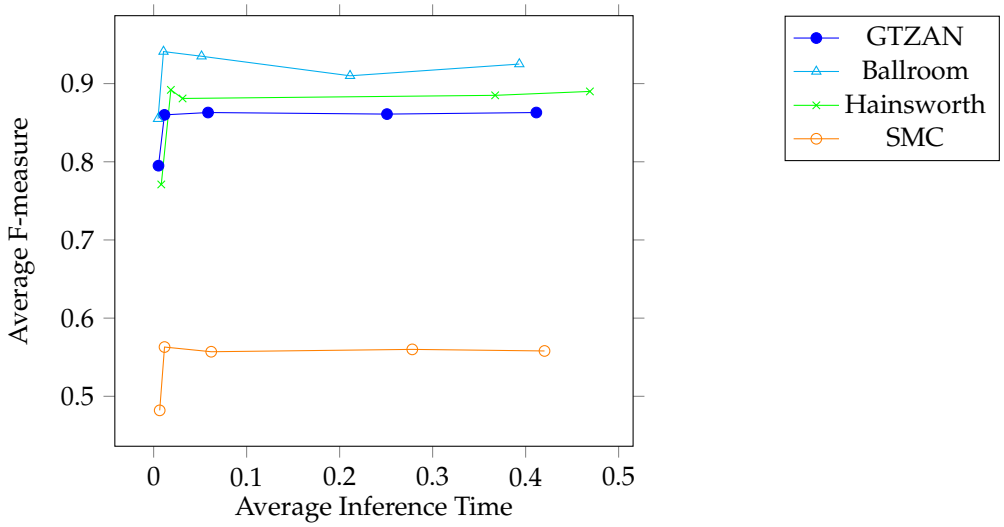

Figure 4. Average F-measure and average inference time on different settings of first k ($k = \{3, 5, 10, 15, 20\}$ from left to right in each line) frenquency components on GTZAN, Ballroom, Hainsworth, and SMC datasets.

6. Conclusions

In this paper, we propose an efficient music beat tracking model. Firstly, PRNN that can perceive the period features of input information is presented. In the beat tracking framework of RNN and HMM, we utilized the frequencies of neural activation values. The PLSTM can capture the periodic patters in the hidden layers and output neural activations with more accurate frequency components for potential tempo estimation. Then the number of hidden space in HMM part is much reduced, which leads shorter inference time and lower computational complexity. The performances of our model on different datasets are close to the state-of-the-art beat tracking models and are much faster in the meanwhile. The main limitation of our method is that the period attention module needs re-training when used in a new dataset. We will explore the applicability and migration feature of the PRNN in the future work.

Author Contributions: Conceptualization, G.S. and Z.W.; methodology, G.S. and Z.W.; software, G.S.; validation, G.S. and Z.W.; formal analysis, Z.W.; investigation, Z.W.; resources, Z.W.; data curation, G.S.; writing—original draft preparation, G.S.; writing—review and editing, Z.W.; visualization, G.S.; supervision, Z.W.; project administration, Z.W.; funding acquisition, Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Key Research and Development program of China (No. 2019YFC1521300) and the Fundamental Research Funds for the Central Universities (No. CUSF-DH-D-2018097).

Data Availability Statement: Publicly available datasets were analyzed in this study. GTZAN data can be found here: [http://marsyas.info/downloads/datasets.html, accessed on 11 January 2022]. Ballroom data can be found here: [http://mtg.upf.edu/ismir2004/contest/tempoContest/node5.html, accessed on 11 January 2022]. Hainsworth data can be found here: [http://www.marsyas.info/tempo, accessed on 11 January 2022]. SMC_MIRUM data can be found here: [http://smc.inescporto.pt/data/SMC_MIREX.zip, accessed on 11 January 2022].

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MIR	Music Information Retrieval
HMM	Hidden Markov Model
RNN	Recurrent Neural Network
PRNN	Periodic Recurrent Neural Network
STFT	Short-term Fourier Transform
MFCC	Mel-frequency Cepstral Coefficients
CRNN	Convolutional Recurrent Neural Network
TCN	Temporal Convolutional Network
LSTM	Long-short Term Memory
GRU	Gated Recurrent Unit
JND	Just Noticeable Difference
BPM	Beats Per Minute

Appendix A. Hidden Markov Model in Music Beat Tracking

In music beat tracking field, a popular post-processing method is a bar pointer model that is based on hidden Markov model. In this part, we describe how we use this probabilistic state-space model to solve metrical structure analysis problem, specifically in this paper, the situation of beat tracking. In the model, the regularly recurring patterns and accents of an audio piece are represented by a sequence of hidden variables. These hidden variables are inferred from another sequence of variables extracted from the music signal by the proposed deep neural network and the network usually called an observer in HMM.

Now we describe the bar point model using a toy example. We consider a state-space of two hidden variables, one is the position within a bar and the other is the tempo.

The hypothetical pointer moves through the space of the hidden variables across a clip of music. For each frame n the hidden state of the bar pointer with two hidden variables mentioned above is referred as $\mathbf{x}_n = [\phi_n, \dot{\phi}_n]$. $\phi_n \in \{1, 2, \ldots, M\}$ indicates the position within a bar where$M$ is the total number of discrete positions per bar, and $\dot{\phi}_n \in \{\dot{\phi}_{\min}, \dot{\phi}_{\min} + 1, \ldots, \dot{\phi}_{\max}\}$ represents the tempo in bar positions of each time frame. $\dot{\phi}_{\min}$ and $\dot{\phi}_{\max}$ are the lowest and the highest tempo, respectively. The observation features extracted from the proposed deep RNN with PLSTM module are denoted as y_n.

In general, our goal is to obtain the most likely hidden state sequence $\mathbf{x}^*_{1:N} = \{\mathbf{x}^*_1, \mathbf{x}^*_2, \ldots, \mathbf{x}^*_N\}$ given a sequence of observations $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$ for every music clip that is formulized as the following,

$$\mathbf{x}^*_{1:N} = \arg\max_{x_{1:N}} P(\mathbf{x}_{1:N} \mid \mathbf{y}_{1:N}). \tag{A1}$$

with

$$P(\mathbf{y}_{1:N} \mid \mathbf{x}_{1:N}) \propto P(\mathbf{x}_1) \prod_{n=2}^{N} P(\mathbf{x}_n \mid \mathbf{x}_{n-1}) P(\mathbf{y}_n \mid \mathbf{x}_n) \tag{A2}$$

where $P(\mathbf{x}_n \mid \mathbf{x}_{n-1})$ is the transition model, $P(\mathbf{y}_n \mid \mathbf{x}_n)$ is the observation model and $P(\mathbf{x}_1)$ is the initial state distribution. Equation (A1) can be solved by the Viterbi algorithm. Finally, the beat frames set $\mathcal{B}$ can be extracted from the sequence of bar positions as

$$\mathcal{B} = \{n : \phi_n^* = 1\} \tag{A3}$$

which corresponds to a bar position that matches a beat position. Arbitrary prior knowledge such as tempo distributions can be incorporated into this bar point model. In this paper, we adopt a uniform distribution.

The transition model $P(\mathbf{x}_n \mid \mathbf{x}_{n-1})$ can be decomposed into a distribution for each of the two hidden variables ϕ_n, and $\dot{\phi}_n$ by the following equation,

$$P(\mathbf{x}_n \mid \mathbf{x}_{n-1}) = P(\phi_k \mid \phi_{n-1}, \dot{\phi}_{n-1}) \cdot P(\dot{\phi}_n \mid \dot{\phi}_{n-1}) \tag{A4}$$

The first factor is

$$P(\phi_k \mid \phi_{n-1}, \dot{\phi}_{n-1}) = \mathbb{1}_x \tag{A5}$$

where $\mathbb{1}_x$ is an indicator function that equals one if $\phi_k = (\phi_{n-1} + \dot{\phi}_{n-1} - 1) \bmod +1$ and zero otherwise. The bar position is cyclic because of the modulo operator. The second factor $P(\dot{\phi}_n \mid \dot{\phi}_{n-1})$ is expressed by

$$P(\dot{\phi}_n \mid \dot{\phi}_{n-1}) = \begin{cases} 1 - p_{\dot{\phi}}, & \dot{\phi}_n = \dot{\phi}_{n-1} \\ \frac{p_{\dot{\phi}}}{2}, & \dot{\phi}_n = \dot{\phi}_{n-1} + 1 \\ \frac{p_{\dot{\phi}}}{2}, & \dot{\phi}_n = \dot{\phi}_{n-1} - 1 \end{cases} \tag{A6}$$

when $\dot{\phi}_{\min} \leq \dot{\phi}_n \leq \dot{\phi}_{\max}$, otherwise $P(\dot{\phi}_n \mid \dot{\phi}_{n-1}) = 0$. $p_{\dot{\phi}}$ is the probability of a tempo change. Equation (A6) means that the pointer can perform three tempo transitions from each state. For observation models, in this work we uses RNN with the PLSTM module to derive a probability of a frame being a beat or not.

The tempo-position state space is divided into equidistant points in the original implementations of the bar pointer model and each point aligned to a bar position and tempo with integer value. In this discretisation way, the number of position points per bar is constant across the tempi, which means a bar at low tempo has a lower time resolution than a bar at high tempo. Consequently, using a constant frame, more observation rates are available for a bar at low tempo than at high tempo. A mismatch between the time resolution of the discretized bar position and the time resolution of the feature extraction from neural network is caused.

It is not consistent with human tempo sensitivity experiments that two adjacent tempo grid points have a constant distance in the bar pointer model described above. Two adjacent tempo grid points have a constant distance in original bar pointer model, which is not consistent with human tempo sensitivity experiments. With the just noticeable difference (JND) around 2–5% of the beat interval, human is able to notice tempo changes proportionally. Therefore, to reach a sufficient high tempo resolution at low tempi, we enlarge the number of tempo states. The tempo state is independent of all tempo states as the tempo model forms a first-order Markov chain. The model can not reflect any long term dependencies between tempo states, which possibly leads unstable trajectories of tempo.

We use one bar position state per audio frame while the number of discrete bar positions M is dependent on the tempo. The number of observations per bar or every four beats at a tempo T in beats per minute (BPM) is

$$M(T) = \text{round}\left(\frac{4 \times 60}{T * \Delta}\right) \tag{A7}$$

where Δ is the length of audio frame. Using Equation (A7), we compute the number of bar positions of the minimum tempo $M(T_{\min})$ and that of maximum tempo $M(T_{\max})$. Then we model all $N_{\max}$ tempi that correspond to bar positions of integer values in the interval $[M(T_{\max})\ M(T_{\min})]$, with

$$N_{\max} = M(T_{\min}) - M(T_{\max}) + 1 \tag{A8}$$

For $N < N_{\max}$, we mimic the human behavior of auditory system by distributing N states logarithmically within the range of beat intervals when choose the tempo states. Within a bar, transitions at beat positions are allowed only to stablize the tempo trajectories. Three tempo transitions we used in this work are defined as If $\Phi_k \in \mathcal{B}$, else

$$\begin{aligned} P(\dot{\phi}_n \mid \dot{\phi}_{n-1}) &= f(\dot{\phi}_n, \dot{\phi}_{n-1}) \\ P(\dot{\phi}_n \mid \dot{\phi}_{n-1}) &= \begin{cases} 1, & \dot{\phi}_n = \dot{\phi}_{n-1} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{A9}$$

$\mathcal{B}$ is the set of bar positions that corresponds to beats, and $f(\cdot)$ is a function that models the tempo change probabilities. We use this exponential distribution as the following

$$f(\dot{\phi}_n, \dot{\phi}_{n-1}) = \exp\left(-\lambda \times \left|\frac{\dot{\phi}_n}{\dot{\phi}_{n-1}} - 1\right|\right) \tag{A10}$$

where the rate parameter $\lambda \in \mathbb{Z}_{\geq 0}$ determines the steepness of the distribution. $\lambda = 0$ means that transitions to all tempi are equally probable. In practice, for music with roughly constant tempo, we set $\lambda \in [1, 300]$.

References

1. Lenc, T.; Keller, P.E.; Varlet, M.; Nozaradan, S. Neural tracking of the musical beat is enhanced by low-frequency sounds. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 8221–8226. [CrossRef] [PubMed]
2. Song, G.; Wang, Z.; Han, F.; Ding, S.; Iqbal, M.A. Music auto-tagging using deep Recurrent Neural Networks. *Neurocomputing* **2018**, *292*, 104–110. [CrossRef]
3. Kim, K.L.; Lee, J.; Kum, S.; Park, C.L.; Nam, J. Semantic Tagging of Singing Voices in Popular Music Recordings. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 1656–1668. [CrossRef]
4. Song, G.; Wang, Z.; Han, F.; Ding, S.; Gu, X. Music auto-tagging using scattering transform and convolutional neural network with self-attention. *Appl. Soft Comput.* **2020**, *96*, 106702. [CrossRef]
5. Wu, W.; Han, F.; Song, G.; Wang, Z. Music genre classification using independent recurrent neural network. In Proceedings of the 2018 Chinese Automation Congress (CAC), Xi'an, China, 30 November–2 December 2018; pp. 192–195.
6. Yu, Y.; Luo, S.; Liu, S.; Qiao, H.; Liu, Y.; Feng, L. Deep attention based music genre classification. *Neurocomputing* **2020**, *372*, 84–91. [CrossRef]
7. Yadav, A.; Vishwakarma, D.K. A unified framework of deep networks for genre classification using movie trailer. *Appl. Soft Comput.* **2020**, *96*, 106624. [CrossRef]
8. Dong, Y.; Yang, X.; Zhao, X.; Li, J. Bidirectional convolutional recurrent sparse network (BCRSN): An efficient model for music emotion recognition. *IEEE Trans. Multimed.* **2019**, *21*, 3150–3163. [CrossRef]
9. Panda, R.; Malheiro, R.M.; Paiva, R.P. Audio features for music emotion recognition: A survey. *IEEE Trans. Affect. Comput.* **2020**, *early access*. [CrossRef]
10. Sigtia, S.; Benetos, E.; Dixon, S. An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 927–939. [CrossRef]
11. Benetos, E.; Dixon, S.; Duan, Z.; Ewert, S. Automatic music transcription: An overview. *IEEE Signal Process. Mag.* **2018**, *36*, 20–30. [CrossRef]
12. Wu, Y.T.; Chen, B.; Su, L. Multi-Instrument Automatic Music Transcription With Self-Attention-Based Instance Segmentation. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 2796–2809. [CrossRef]
13. Böck, S.; Krebs, F.; Widmer, G. Joint Beat and Downbeat Tracking with Recurrent Neural Networks. In Proceedings of the ISMIR, New York, NY, USA, 7–11 August 2016; pp. 255–261.
14. Müller, M.; McFee, B.; Kinnaird, K.M. Interactive learning of signal processing through music. *IEEE Signal Process. Mag.* 2021, *accepted for publication*. [CrossRef]
15. Gkiokas, A.; Katsouros, V. Convolutional Neural Networks for Real-Time Beat Tracking: A Dancing Robot Application. In Proceedings of the ISMIR, Suzhou, China, 23–28 October 2017; pp. 286–293.

16. Cheng, T.; Fukayama, S.; Goto, M. Convolving Gaussian kernels for RNN-based beat tracking. In Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO), Eternal, Rome, 3–7 September 2018; pp. 1905–1909.
17. Istvanek, M.; Smekal, Z.; Spurny, L.; Mekyska, J. Enhancement of Conventional Beat Tracking System Using Teager–Kaiser Energy Operator. *Appl. Sci.* **2020**, *10*, 379. [CrossRef]
18. Böck, S.; Schedl, M. Enhanced beat tracking with context-aware neural networks. In Proceedings of the International Conference Digital Audio Effects, Paris, France, 19–23 September 2011; pp. 135–139.
19. Fuentes, M.; McFee, B.; Crayencour, H.; Essid, S.; Bello, J. Analysis of common design choices in deep learning systems for downbeat tracking. In Proceedings of the 19th International Society for Music Information Retrieval Conference, Paris, France, 23–27 September 2018.
20. Cano, E.; Mora-Ángel, F.; Gil, G.A.L.; Zapata, J.R.; Escamilla, A.; Alzate, J.F.; Betancur, M. Sesquialtera in the colombian bambuco: Perception and estimation of beat and meter. In Proceedings of the International Society for Music Information Retrieval Conference, Montreal, QC, Canada, 11–16 October 2020; pp. 409–415.
21. Pedersoli, F.; Goto, M. Dance beat tracking from visual information alone. In Proceedings of the International Society for Music Information Retrieval Conference, Montreal, QC, Canada, 11–16 October 2020; pp. 400–408.
22. Holzapfel, A.; Stylianou, Y. Beat tracking using group delay based onset detection. In Proceedings of the ISMIR-International Conference on Music Information Retrieval (ISMIR), Philadelphia, PA, USA, 14–18 September 2008; pp. 653–658.
23. Laroche, J. Efficient tempo and beat tracking in audio recordings. *J. Audio Eng. Soc.* **2003**, *51*, 226–233.
24. MatthewDavies, E.; Böck, S. Temporal convolutional networks for musical audio beat tracking. In Proceedings of the 2019 27th European Signal Processing Conference (EUSIPCO), A Coruña, Spain, 2–6 September 2019; pp. 1–5.
25. Böck, S.; Davies, M.E.; Knees, P. Multi-Task Learning of Tempo and Beat: Learning One to Improve the Other. In Proceedings of the ISMIR, Delft, The Netherlands, 4–8 November 2019; pp. 486–493.
26. Ellis, D.P. Beat tracking by dynamic programming. *J. New Music. Res.* **2007**, *36*, 51–60. [CrossRef]
27. Lartillot, O.; Grandjean, D. Tempo and metrical analysis by tracking multiple metrical levels using autocorrelation. *Appl. Sci.* **2019**, *9*, 5121. [CrossRef]
28. Böck, S.; Krebs, F.; Widmer, G. Accurate Tempo Estimation Based on Recurrent Neural Networks and Resonating Comb Filters. In Proceedings of the ISMIR, Malaga, Spain, 26–30 October 2015; pp. 625–631.
29. Cemgil, A.T.; Kappen, B.; Desain, P.; Honing, H. On tempo tracking: Tempogram representation and Kalman filtering. *J. New Music. Res.* **2000**, *29*, 259–273. [CrossRef]
30. Krebs, F.; Böck, S.; Widmer, G. An Efficient State-Space Model for Joint Tempo and Meter Tracking. In Proceedings of the ISMIR, Malaga, Spain, 26–30 October 2015; pp. 72–78.
31. Chuang, Y.C.; Su, L. Beat and Downbeat Tracking of Symbolic Music Data Using Deep Recurrent Neural Networks. In Proceedings of the 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Auckland, New Zealand, 7–10 December 2020; pp. 346–352.
32. Peeters, G.; Flocon-Cholet, J. Perceptual tempo estimation using GMM-regression. In Proceedings of the Second International ACM workshop on Music Information Retrieval with User-Centered and Multimodal Strategies, Nara, Japan, 2 November 2012; pp. 45–50.
33. Percival, G.; Tzanetakis, G. Streamlined tempo estimation based on autocorrelation and cross-correlation with pulses. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2014**, *22*, 1765–1776. [CrossRef]
34. Whiteley, N.; Cemgil, A.T.; Godsill, S.J. Bayesian Modelling of Temporal Structure in Musical Audio. In Proceedings of the ISMIR, Victoria, BC, Canada, 8–12 October 2006; pp. 29–34.
35. Krebs, F.; Böck, S.; Widmer, G. Rhythmic Pattern Modeling for Beat and Downbeat Tracking in Musical Audio. In Proceedings of the ISMIR, Curitiba, Brazil, 4–8 November 2013; pp. 227–232.
36. Srinivasamurthy, A.; Holzapfel, A.; Cemgil, A.T.; Serra, X. Particle filters for efficient meter tracking with dynamic bayesian networks. In Proceedings of the ISMIR 2015, Malaga, Spain, 26–30 October 2015.
37. Krebs, F.; Holzapfel, A.; Cemgil, A.T.; Widmer, G. Inferring metrical structure in music using particle filters. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2015**, *23*, 817–827. [CrossRef]
38. Müller, M.; Ewert, S. Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features. In Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR), Miami, FL, USA, 24–28 October 2011.
39. Fuentes, B.; Liutkus, A.; Badeau, R.; Richard, G. Probabilistic model for main melody extraction using constant-Q transform. In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 5357–5360.
40. Durand, S.; Bello, J.P.; David, B.; Richard, G. Robust downbeat tracking using an ensemble of convolutional networks. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *25*, 76–89. [CrossRef]
41. Di Giorgi, B.; Mauch, M.; Levy, M. Downbeat tracking with tempo-invariant convolutional neural networks. *arXiv* **2021**, arXiv:2102.02282.
42. Hung, Y.N.; Wang, J.C.; Song, X.; Lu, W.T.; Won, M. Modeling beats and downbeats with a time-frequency Transformer. In Proceedings of the ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) Singapore, 23–27 May 2022; pp. 401–405.

43. Desblancs, D.; Hennequin, R.; Lostanlen, V. Zero-Note Samba: Self-Supervised Beat Tracking; hal-03669865, 2022. Available online: https://hal.archives-ouvertes.fr/hal-03669865/file/desblancs2022jstsp_supplementary.pdf (accessed on 11 January 2022).
44. Zonoozi, A.; Kim, J.j.; Li, X.L.; Cong, G. Periodic-CRN: A convolutional recurrent model for crowd density prediction with recurring periodic patterns. In Proceedings of the IJCAI, Stockholm, Sweden, 13–19 July 2018; pp. 3732–3738.
45. Chen, C.; Li, K.; Teo, S.G.; Zou, X.; Wang, K.; Wang, J.; Zeng, Z. Gated residual recurrent graph neural networks for traffic prediction. In Proceedings of the AAAI conference on artificial intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 485–492.
46. He, Z.; Chow, C.Y.; Zhang, J.D. STCNN: A spatio-temporal convolutional neural network for long-term traffic prediction. In Proceedings of the 2019 20th IEEE International Conference on Mobile Data Management (MDM), Hong Kong, 10–13 June 2019; pp. 226–233.
47. Karim, M.E.; Maswood, M.M.S.; Das, S.; Alharbi, A.G. BHyPreC: A novel Bi-LSTM based hybrid recurrent neural network model to predict the CPU workload of cloud virtual machine. *IEEE Access* **2021**, *9*, 131476–131495. [CrossRef]
48. Wu, H.; Ma, Y.; Xiang, Z.; Yang, C.; He, K. A spatial–temporal graph neural network framework for automated software bug triaging. *Knowl.-Based Syst.* **2022**, *241*, 108308. [CrossRef]
49. Abdelraouf, A.; Abdel-Aty, M.; Yuan, J. Utilizing attention-based multi-encoder-decoder neural networks for freeway traffic speed prediction. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 11960–11969. [CrossRef]
50. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
51. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv* **2014**, arXiv:1409.1259.
52. Elowsson, A. Beat tracking with a cepstroid invariant neural network. In Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR 2016), New York, NY, USA, 7–11 August 2016; pp. 351–357.
53. Marchand, U.; Fresnel, Q.; Peeters, G. Gtzan-Rhythm: Extending the Gtzan Test-Set with Beat, Downbeat and Swing Annotations. In Proceedings of the Extended abstracts for the Late-Breaking Demo Session of the 16th International Society for Music Information Retrieval Conference, Malaga, Spain, 26–30 October 2015.
54. Hainsworth, S.W. Techniques for the Automated Analysis of Musical Audio. Ph.D. Dissertation, Cambridge University, Cambridge, UK, 2004.
55. Holzapfel, A.; Davies, M.E.; Zapata, J.R.; Oliveira, J.L.; Gouyon, F. Selective sampling for beat tracking evaluation. *IEEE Trans. Audio Speech Lang. Process.* **2012**, *20*, 2539–2548. [CrossRef]
56. Holzapfel, A.; Davies, M.E.; Zapata, J.R.; Oliveira, J.L.; Gouyon, F. On the automatic identification of difficult examples for beat tracking: Towards building new evaluation datasets. In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 89–92.
57. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.