Beyond the Beat: Towards Metre, Rhythm and Melody Modelling with Hybrid Oscillator Networks

Andrew Lambert
City University London
andrew.lambert.1@city.ac.uk

Tillman Weyde
City University London
t.e.weyde@city.ac.uk

Newton Armstrong
City University London
newton.armstrong.1@city.ac.uk

ABSTRACT

In this paper we take a connectionist machine learning approach to the problem of metre perception and learning in musical signals. We present a hybrid network consisting of a nonlinear oscillator network and a recurrent neural network. The oscillator network acts as an entrained resonant filter to the musical signal. It 'perceives' metre by resonating nonlinearly to the inherent periodicities within the signal, creating a hierarchy of strong and weak periods. The neural network learns the long-term temporal structures present in this signal. We show that this hybrid network outperforms our previous approach of a single layer recurrent neural network in a melody prediction task.

We hypothesise that our hybrid system is enabled to make use of the relatively long temporal resonance in the oscillator network output, and therefore model more coherent long-term structures. A system such as this could be used in a multitude of analytic and generative scenarios, including live performance applications.

1. INTRODUCTION

Beat induction allows us to tap along to the beat of music, perceiving its pulse. This perceived pulse can be present in the stimulus, but it is often only implied by the musical events. Furthermore, performed music is rarely periodic and is subject to the performers' expressive timing. This makes beat induction difficult to model computationally.

Finding the pulse within a musical signal is a step towards achieving other music perception tasks, such as metre perception. Metre refers to the multi-layered divisions of time present in music, of which the referent layer is the pulse. Other layers in music divide the pulse into the smallest subdivisions of time, and extend it towards larger measures, phrases, periods, and even higher order forms. Thus, a single 'beat' can occur at one or more metrical levels, whereas the 'pulse' is the series of beats on the referent layer only. The more levels on which beat occurs, the 'stronger' that beat is perceived, creating a beat hierarchy, or metrical structure [1]. The individual components of music, the rhythmic events in time, lead to the formation of new macroscopic spatial, temporal and functional structures in metre. In performance, these structures vary and repeat with time in their own patterns.

The process through which humans achieve beat induction is known as entrainment. Entrainment is the coordination of temporally structured events through interaction where two or more periodic signals are coupled in a stable relationship. Many relationships are possible in entrained signals; exact synchronisation is considered to be a special case of entrainment. Ethnomusicologists are increasingly becoming aware of the importance of entrainment processes as an approach to understanding music making and music perception as a culturally interactive process [2].

Much prior work on pulse and metre perception has been concerned with abstract temporal information, such as crafted pulses in time [3, 4, 5, 6]. However, metre perception and preference develops through cultural learning and is determined by a multitude of musical signposts, including the melody and the tempo of the pulse [2, 7].

This project's aim is to design a hybrid network which is able to learn metrical structures, generalising on a corpus of sequences to make predictions about future musical events. This is therefore not a metre classification task, but an investigation into machine models of melody and rhythm. We are investigating if a music prediction task produces better results when utilising a certain model of metrical structure.

In section 2 we outline the models we have chosen for this task and the reasons behind these choices. Section 3 details the experiments we have conducted. Section 4 presents the results of our simulations. Finally, sections 5 and 6 offer insights and directions for future work.

2. MODELS

Our hybrid network consists of two connected networks. The first is a Gradient Frequency Neural Network (GFNN) [8], a nonlinear oscillator network. It acts as an entrained resonant filter to the musical signal and serves as a metre perception layer. The second is a Long Short-Term Memory network (LSTM) [9], a recurrent neural network, which is able to learn the kind of long-term temporal structures required in music signal prediction [3]. We use this layer for prediction and generation.

2.1 Metre Perception Layer

Oscillators have been used for beat induction in machines for over twenty years. Certain oscillator models lend them-

selves well to beat induction tasks due to their stable limit cycle and their entrainment properties [3]. By using oscillators to perceive beats, we have the ability to model beat induction as an emergent dynamical process, which changes over time as the signal itself evolves. Gasser et al.'s SONOR system, for instance, adds Hebbian learning to networks of adaptive oscillators, which can then learn to produce a metrical pattern [10].

More recently, the phenomenon of nonlinear resonance has been applied to metre perception and categorisation tasks. Large et al. [8] have introduced the Gradient Frequency Neural Network (GFNN), which is a network of oscillators whose natural frequencies are distributed across a spectrum. When a GFNN is stimulated by a signal, the oscillators resonate nonlinearly, producing larger amplitude responses at certain frequencies along the spectrum. Nonlinear resonance can account for pattern completion, the perception of the missing fundamental, tonal relationships and the perception of metre [11].

When the frequencies in a GFNN are distributed within a rhythmic range, resonances occur at integer ratios to the pulse. These resonances can be interpreted as a hierarchical metrical structure. Rhythmic studies with GFNNs include rhythm categorisation [4], beat induction in syncopated rhythms [5] and polyrhythms [12].

2.2 Temporal Structure Layer

There have been many connectionist approaches to musical tasks [13, 14, 3, 15, 16]. Whilst recurrent neural networks are good at learning temporal patterns, they often lack global coherence due to the lack of long-term memory. Long Short-Term Memory (LSTM) networks were designed to overcome this problem. A simplified diagram of an LSTM memory block can be seen in Figure 1. A self-connected node known as the Constant Error Carousel (CEC) ensures constant error flow back through time. The input and output gates control how information flows into and out of the CEC, and the forget gate controls when the CEC is reset. The input, output and forget gates are connected via 'peepholes'. To describe the LSTM model in further detail would be out of the scope of this paper; for a full specification, see [9].

LSTMs have already had some success in music applications. Eck and Schmidhuber [3] trained LSTMs which were able to improvise chord progressions in the blues and more recently Coca et al. [16] used LSTMs to generate melodies that fit within user specified parameters.

3. EXPERIMENTS

All experiments operate on monophonic symbolic music data. We have used a corpus of 100 German folk songs from the Essen Folksong Collection [17].

We implemented the GFNN in MATLAB [1], using the standard differential equation solvers, and the LSTM in Python using the PyBrain [2] library.

[1] http://www.mathworks.co.uk/

[2] http://pybrain.org/

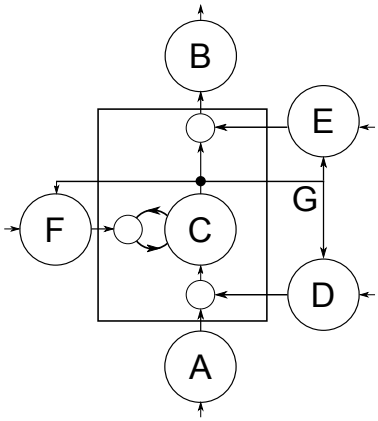


Figure 1: A single LSTM memory block showing (A) input, (B) output, (C) CEC, (D) input gate, (E) output gate, (F) forget gate and (G) peephole connections.

3.1 GFNN

The GFNN consists of 128 Hopf oscillators defined by the following differential equation:

$$\frac{dz}{dt} = z(\alpha + i\omega + \frac{\beta\varepsilon|z|^4}{1-\varepsilon|z|^2}) + \frac{x}{1-\sqrt{\varepsilon}x}\cdot\frac{1}{1-\sqrt{\varepsilon}\bar{z}} \quad (1)$$

where z is the complex valued output, $\bar{z}$ is its complex conjugate, ω is the driving frequency in radians per second, α is a damping parameter, β is an amplitude compressing parameter, ε is a scaling parameter and x is a time-varying stimulus. This oscillator is a complex valued model which spontaneously oscillates according to its parameters, and entrains to and resonates with an external stimulus.

For all experiments, parameter values were fixed as follows:

$$\alpha = -0.1, \beta = -0.1, \varepsilon = 0.5 \quad (2)$$

This gives a sinusoid-like oscillation whose amplitude is gradually dampened over time (see Figure 2). The gradual dampening of the amplitude allows the oscillator to maintain a long temporal memory of previous stimulation.

The oscillator frequencies in the network were logarithmically distributed from 0.25Hz to 16Hz. The GFNN was stimulated by rhythmic time-series data in the form of a decay envelope on note onsets, synthesised from the symbolic data. All sequences in the corpus were synthesised at a tempo of 120bpm (2Hz), meaning that our metrical periodicities the GFNN ranged from a demisemiquaver (32nd note) to a breve (double whole note).

An example output can be seen in Figure 3; stronger and weaker oscillations can clearly be seen. Performing a Fourier transform on the GFNN output reveals that there is energy at many frequencies in the spectrum, including the pulse (Figure 4). Often this energy is located at integer ratios to the pulse, implying a perception of the metrical structure.

3.2 LSTM

We constructed five different LSTMs for our experiment, all of which followed the standard LSTM model with peep-

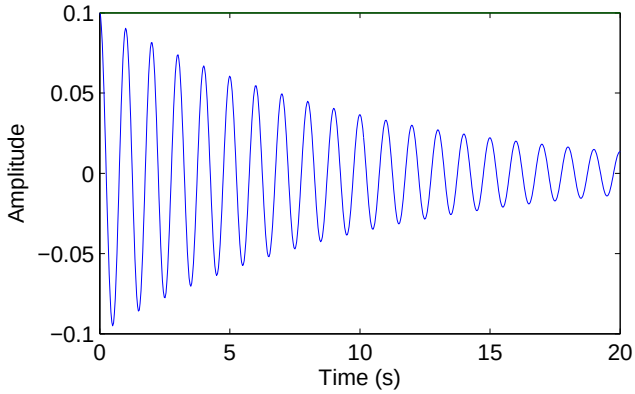


Figure 2: A Hopf oscillator with the following parameters, $\omega = 2\pi, \alpha = -0.1, \beta = -0.1, \varepsilon = 0.5$. The amplitude has decayed by half in approximately 6.5 seconds.

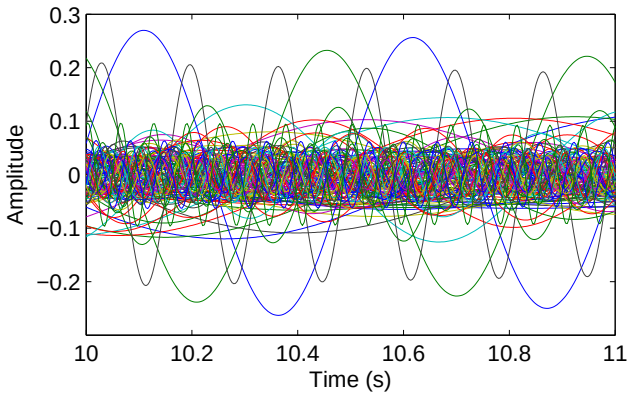


Figure 3: A 1 second excerpt from the GFNN output.

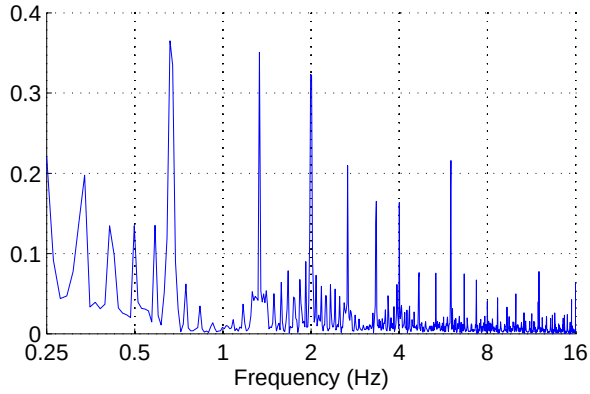


Figure 4: The magnitude spectrum of the summed GFNN output from Figure 3. The energy peaks can be interpreted as a perception of metrical structure.

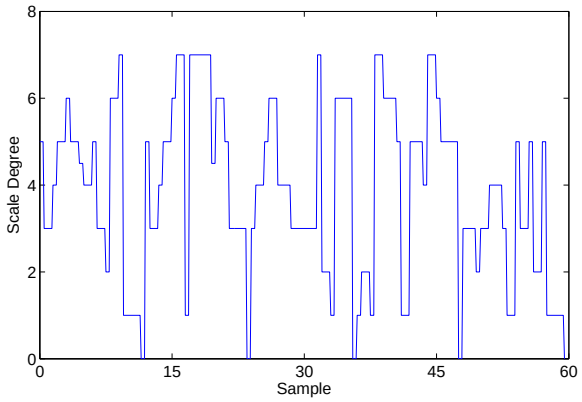


Figure 5: Example scale degree time-series data.

hole connections enabled. In all cases, the task was to predict the next sample in time-series music data. Therefore, a single output node was used for all models.

The melodies of the sequences in the corpus are in varying keys and octaves, so we abstracted the absolute pitch values to their relative scale degrees. We first inserted scale degree numbers, their onsets and offsets into the data stream and then re-sampled the data using the zero-order hold method. The data was re-sampled such that one sample was equivalent to a semiquaver (16th note). Accidentals were encoded by adding or subtracting 0.5 from the scale degree and rests were encoded as 0 values. An example data stream can be seen in Figure 5. The GFNN output data was re-sampled to match the target data's sample rate.

The number of hidden LSTM blocks was fixed at 10 for all experiments. Training was done by backpropagation through time [18] using RProp$^-$ [19].

During training we used k-fold cross-validation [20]. In k-fold cross validation, the dataset is divided into k equal parts, or 'folds'. A single fold is retained as the test data for testing the model, and the remaining k - 1 folds are used as training data. The cross-validation process is then repeated k times, with each of the k folds used exactly once as the test data. For our experiments k was fixed at 4, and a maximum of 2500 training epochs was set per fold, but never reached.

3.2.1 LSTM 1

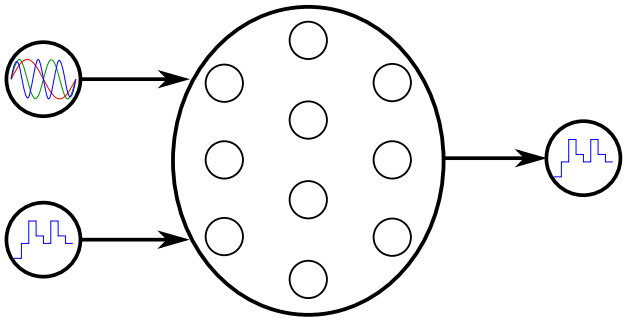
LSTM1 was designed as a baseline to measure the hybrid networks against. It did not take input from the GFNN, and so consisted of single input containing the time-series scale degree data from the corpus.

3.2.2 LSTM 2

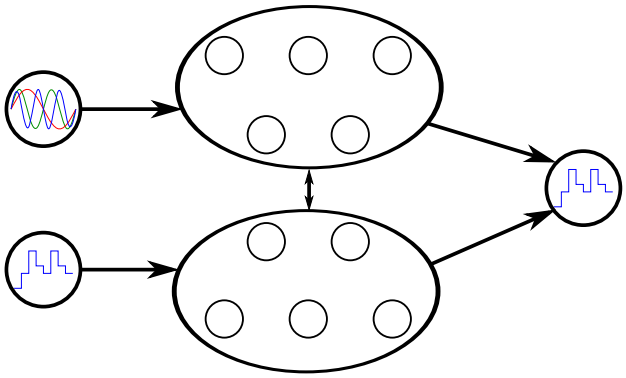
LSTM2 had 128 additional inputs compared with LSTM1, one for each oscillator in the GFNN (see Figure 6a). This brings the total number of inputs to 129.

3.2.3 LSTM 3

LSTM3 had 8 additional inputs compared with LSTM1 (see Figure 6a). These inputs consisted of a pre-filtered GFNN output, containing the strongest resonant oscillations. Our hypothesis here was that better predictions could be made by pre-filtering out some less resonant oscillations. Oscillations were filtered by averaging the GFNN output over the corpus and finding the oscillators with the largest amplitude response over the final 25% of the piece. We ensured a spread of frequencies by ignoring oscillators if another oscillator of near frequency was already included in the filtered result. Once these frequencies were found,



(a) Network diagram for LSTM2 and LSTM3. Input and hidden layers are fully connected.



(b) Network diagram for LSTM2a and LSTM3a showing reduced connectivity between the input and hidden layers.

Figure 6: Network diagrams showing connections between input, hidden and output layers. LSTM2 and 2a had full connections of 128 oscillations from the GFNN, LSTM3 and 3a had filtered connections of 8 oscillations from the GFNN.

they were fixed for all sequences. This brings the total inputs to 9.

3.2.4 LSTM 2a and 3a

For these networks we experimented with partial connections between the input layer and the hidden layer. The hidden blocks were split into two groups of equal size (see Figure 6b). In a similar way to Eck and Schmidhuber's [3] treatment of chords and melody parts, one group was connected only to the scale degree time-series data input and the other was connected only to the GFNN data inputs. The hidden layer remained fully connected with itself and the output layer. LSTM2a used the full GFNN output and LSTM3a used the pre-filtered GFNN output. Our hypothesis here was that by forcing the hidden blocks to process only rhythm or melody data, better predictions would be made.

4. RESULTS

Networks were evaluated by activating each of them with the sequences in the corpus (ground truth). We activated the networks with the ground truth throughout the sequence, and for the last 25% of inputs the network output was compared to the target data.

The results have been evaluated by focusing on melody and rhythm. For melody, we have the "Sequence" metric. This has been calculated as a proportion of samples where the output, rounded to the nearest half, matches the target value. Higher numbers therefore indicate better predictions. For rhythm we have the "Precision", "Recall" and "F-measure" metrics. These metrics all refer to the onset prediction in the network, where changes of value in the target and the output are concurrent. The metrics are calculated with the following formulae:

$$precision = \frac{correctly\,predicted\,onsets}{all\,predicted\,onsets} \quad (3)$$

$$recall = \frac{correctly\,predicted\,onsets}{ground\,truth\,onsets} \quad (4)$$

$$F = 2 \cdot \frac{precision.recall}{precision + recall} \quad (5)$$

Melody and rhythm are highly related, but have been singled out here to more fully understand the GFNNs effect on the network. The sequence metric represents timing and value, whereas the onset metrics of precision, recall and F-measure represent timing only.

Table 1 shows the results when the networks are tested against the training folds, and Table 2 shows the results when the networks are tested against the test folds. The values shown the mean values calculated over the 4 folds in the cross-validation. Generally, the results for the training data indicate how well a network has adapted that data. The test data results indicate how well a model generalises to data that it has not been trained on, but that is drawn from the same distribution. The results on the test data

Network	Sequence	Precision	Recall	F-measure
LSTM1	0.39842	0.91955	0.45575	0.60645
LSTM2	0.38229	0.93898	0.45729	0.61274
LSTM3	**0.49428**	0.92890	0.45214	0.60555
LSTM2a	0.38644	**0.95247**	**0.45953**	**0.61735**
LSTM3a	0.44366	0.92402	0.45849	0.60988

Table 1: Results of all LSTM experiments on the training dataset.

Network	Sequence	Precision	Recall	F-measure
LSTM1	0.39071	0.91962	0.45623	0.60599
LSTM2	0.32831	0.93313	0.45739	0.61157
LSTM3	**0.49273**	0.92689	0.45298	0.60582
LSTM2a	0.35777	**0.94818**	**0.46885**	**0.62507**
LSTM3a	0.44010	0.92421	0.46349	0.61420

Table 2: Results of all LSTM experiments on the test dataset.

compared to the training data are correlated and are no more than 5.2% higher on average, indicating a good generalisation without over-fitting.

Sequence prediction was fairly poor for all networks, but LSTM3 achieved the highest score here. This was 9.6% more accurate than LSTM1 on the training data and 10.2% more accurate on the test data. LSTM3 consistently outperformed LSTM1 in sequence modelling across training and test datasets and is a statistically significant result in both cases. This provides some evidence that melody modelling benefits from the nonlinear resonance model.

The effect of the modified architecture in LSTM2a and LSTM3a was mixed. LSTM2a outperformed LSTM2 on both datasets and had the highest F-measures of all the hybrid networks, but its sequence prediction was still lower than LSTM1. LSTM3a had a better onset prediction compared with LSTM3 across both datasets, but was not able to match LSTM3's sequence predictions. LSTM3a is better than LSTM1 in all measures, but does not have a single overall best measure.

In terms of rhythm alone there was no clear improvement made with the hybrid networks. However, LSTM2a consistently outperformed LSTM1 and also scored the highest F-measure out of all networks, though this is not a statistically significant improvement.

5. DISCUSSION

The GFNN output, with its strong and weak nonlinear resonances at frequencies related to the pulse, can be interpreted as a perception of metre. Our results show that providing this data helped to improve melody prediction with an LSTM. We hypothesise that this is due to the LSTM being able to make use of the relatively long temporal resonance in the GFNN output, and therefore model more coherent long-term structures.

GFNNs rely on a large number of oscillators spread across a frequency spectrum to improve the accuracy of the output. However, we have shown that LSTMs trained with RProp$^-$ can struggle to filter out some of the noise that is produced as a result of this, as can be seen by the poor performance of LSTM2. There are two potential solutions to this problem which we have explored. The first is to prefilter the GFNN output, greatly reducing the amount of less relevant resonances (LSTM3). This produced the best results in our experiments, but may not be a good solution when dealing with varying tempos or expressive timing, as it introduces an assumption of a metrically homogeneous corpus. The results of LSTM3 show a small degradation in the F-measure, which may be due to this issue. The second solution explored here is to design the LSTM's topology to segment the connections between the input and hidden layers, and therefore have some LSTM blocks processing rhythm data, and others processing melody. This did improve the rhythmic results in LSTM2a and LSTM3a, with LSTM2a having the best rhythmic score overall, however the melody modelling suffered as a result. More work is needed to discover the best filtering or training method.

There is a striking imbalance between the precision and recall scores for all networks, suggesting a chaotic output from the LSTM with too many events being triggered. This lead to results that were not impressive overall, with melodic prediction improved, but not rhythmic prediction in this case. There is a clear need to make outputs more stable, perhaps utilising a better threshold strategy the output nodes.

Both Eck and Schmidhuber's [3] and Coca et al.'s [16] LSTMs either operate on note-by-note data, or quantised time-series data. By inputting metrical data, our system can be extended to work with real time data, as opposed to the metrically quantised data we are using here. This opens up the system for use with a multitude of different tempos and live performance applications.

6. FUTURE WORK

There is much work that we would like to do with our hybrid GFNN-LSTM model.

We would like to perform a study on a bigger corpus where there are more structural elements to learn. Sequences in the Essen Folksong Collection tend to be relatively short, around 16 bars in length. Whilst there are patterns and structures present in these sequences, especially when an entire geographical region is considered, analysing a longer piece with more repeated motifs within a song would be a fruitful exercise.

Performance improvements could be made in the GFNN layer. Currently each oscillator in the network is stimulated by the signal, but is not connected to the other oscillators in the network. Implementing local coupling, where each oscillator receives stimulus from its neighbours, could improve the response of GFNNs with fewer oscillators, particularly in sparse, syncopated, or polyrhythmic signals [21, 5].

In a similar way to Gasser et al. [10] parameters in the GFNN could also be targeted for learning, such as coupling weights between oscillators, stimulus strength, and even frequency. This could lead to a more coherent sense of genre specific frequency distributions within the network. This learning could act in a similar way to our prefiltered networks, LSTM3 and LSTM3a, without introducing forced assumptions.

7. CONCLUSION

We have presented a hybrid network consisting of a metre perception layer (GFNN), and a temporal prediction layer (LSTM). We feel this initial experiment gives some indication that better melody models can be created by modelling metrical structures. By using an oscillator network to track the metrical structure of a performance data, we can move towards real-time processing of signals and close the loop in the GFNN-LSTM, creating an expressive, metrically aware, generative real-time model.

Acknowledgments

Andrew Lambert is supported by a PhD studentship from City University London.

8. REFERENCES

[1] F. Lerdahl and R. Jackendoff, "An overview of hierarchical structure in music," *Music Perception: An Inter-*

disciplinary Journal, vol. 1, no. 2, pp. 229–252, Dec. 1983.

[2] M. Clayton, R. Sager, and U. Will, "In time with the music: the concept of entrainment and its significance for ethnomusicology," *European Meetings in Ethnomusicology*, vol. 11,, pp. 3–142, Aug. 2005.

[3] D. Eck and J. Schmidhuber, "Finding temporal structure in music: blues improvisation with LSTM recurrent networks," in *Proceedings of the 2002 12th IEEE Workshop on Neural Networks for Signal Processing, 2002*, 2002, pp. 747–756.

[4] R. Bååth, E. Lagerstedt, and P. Gärdenfors, "An oscillator model of categorical rhythm perception," in *Proceedings of the 35th Annual Conference of the Cognitive Science Society*, M. Knauff, M. Pauen, N. Sebanz, and I. Wachsmuth, Eds. Austin, TX: Cognitive Science Society, 2013, pp. 1803–1808.

[5] M. J. Velasco and E. W. Large, "Pulse detection in syncopated rhythms using neural oscillators," in *12th International Society for Music Information Retrieval Conference*, Miami, FL, 2011, pp. 185–190.

[6] D. Temperley, "An evaluation system for metrical models," *Computer Music Journal*, vol. 28, no. 3, pp. 28–44, Oct. 2004.

[7] J. A. Grahn, "Neural mechanisms of rhythm perception: Current findings and future perspectives," *Topics in Cognitive Science*, vol. 4, no. 4, pp. 585–606, Oct. 2012.

[8] E. W. Large, F. V. Almonte, and M. J. Velasco, "A canonical model for gradient frequency neural networks," *Physica D: Nonlinear Phenomena*, vol. 239, no. 12, pp. 905–911, Jun. 2010.

[9] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.

[10] M. Gasser, D. Eck, and R. Port, "Meter as mechanism: A neural network model that learns metrical patterns," *Connection Science*, vol. 11, no. 2, pp. 187–216, 1999.

[11] E. W. Large, "Neurodynamics of music," in *Music Perception*, ser. Springer Handbook of Auditory Research, M. R. Jones, R. R. Fay, and A. N. Popper, Eds. Springer New York, Jan. 2010, no. 36, pp. 201–231.

[12] V. Angelis, S. Holland, P. J. Upton, and M. Clayton, "Testing a computational model of rhythm perception using polyrhythmic stimuli," *Journal of New Music Research*, vol. 42, no. 1, pp. 47–60, 2013.

[13] P. M. Todd, "A connectionist approach to algorithmic composition," *Computer Music Journal*, vol. 13, no. 4, pp. 27–43, Dec. 1989.

[14] M. C. Mozer, "Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing," *Connection Science*, vol. 6, no. 2-3, pp. 247–280, 1994.

[15] A. Kalos, "Modeling MIDI music as multivariate time series," in *IEEE Congress on Evolutionary Computation, 2006. CEC 2006*, 2006, pp. 2058–2064.

[16] A. Coca, D. Correa, and L. Zhao, "Computer-aided music composition with LSTM neural network and chaotic inspiration," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, Aug. 2013, pp. 1–7.

[17] H. Schaffrath. (1995) The essen folksong collection in kern format. [Online]. Available: http://www.esac-data.org/

[18] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[19] C. Igel and M. Hüsken, "Improving the rprop learning algorithm," in *Proceedings of the second international ICSC symposium on neural computation (NC 2000)*. Citeseer, 2000, pp. 115–121.

[20] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *IJCAI*, vol. 14, no. 2, 1995, pp. 1137–1145.

[21] D. Eck, "A network of relaxation oscillators that finds downbeats in rhythms," in *Artificial Neural Networks — ICANN 2001*, ser. Lecture Notes in Computer Science, G. Dorffner, H. Bischof, and K. Hornik, Eds. Springer Berlin Heidelberg, Jan. 2001, no. 2130, pp. 1239–1247.