Onset Detection for String Instruments Using Bidirectional Temporal and Convolutional Recurrent Networks

Maciej Tomczak
maciej.tomczak@bcu.ac.uk
Sound and Music Analysis Group (SoMA), Digital Media Technology Lab (DMT Lab), School of Computing and Digital Technology, Birmingham City University
Birmingham, United Kingdom

Jason Hockman
jason.hockman@bcu.ac.uk
Sound and Music Analysis Group (SoMA), Digital Media Technology Lab (DMT Lab), School of Computing and Digital Technology, Birmingham City University
Birmingham, United Kingdom

ABSTRACT

Recent work in note onset detection has centered on deep learning models such as recurrent neural networks (RNN), convolutional neural networks (CNN) and more recently temporal convolutional networks (TCN), which achieve high evaluation accuracies for onsets characterized by clear, well-defined transients, as found in percussive instruments. However, onsets with less transient presence, as found in string instrument recordings, still pose a relatively difficult challenge for state-of-the-art algorithms. This challenge is further exacerbated by a paucity of string instrument data containing expert annotations. In this paper, we propose two new models for onset detection using bidirectional temporal and recurrent convolutional networks, which generalise to polyphonic signals and string instruments. We perform evaluations of the proposed methods alongside state-of-the-art algorithms for onset detection on a benchmark dataset from the MIR community, as well as on a test set from a newly proposed dataset of string instrument recordings with note onset annotations, comprising approximately 40 minutes and over 8,000 annotated onsets with varied expressive playing styles. The results demonstrate the effectiveness of both presented models, as they outperform the state-of-the-art algorithms on string recordings while maintaining comparative performance on other types of music.

CCS CONCEPTS

• **Information systems** → **Music retrieval**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

Music information retrieval, onset detection, recurrent convolutional neural networks, temporal convolutional networks

ACM Reference Format:
Maciej Tomczak and Jason Hockman. 2023. Onset Detection for String Instruments Using Bidirectional Temporal and Convolutional Recurrent Networks. In *Audio Mostly 2023 (AM '23), August 30–September 01, 2023, Edinburgh, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3616195.3616206

AM '23, August 30–September 01, 2023, Edinburgh, United Kingdom

ACM ISBN 979-8-4007-0818-3/23/08.
https://doi.org/10.1145/3616195.3616206

1 INTRODUCTION

Onset detection is a crucial task in the field of music information retrieval, as it provides information about the timing and rhythm of a musical piece. Although the rapid advancements in deep learning have led to the development of powerful algorithms that can understand and process music at a much deeper level, onset detection still holds an important position in variety of areas in the educational [16], analytical [17] and creative industries [18]. While onset detection algorithms for percussion and monophonic instruments have been well established, detecting onsets in string instruments remains challenging. In this paper, we evaluate several approaches that combine bidirectional temporal and convolutional recurrent networks to detect onsets in general and specialised musical scenarios which include string instrument recordings. We evaluate our methods on a new dataset of quartet, trio, duet, and solo (QTDS) ensemble recordings with note onset annotations, as well as an extended version of the Böck dataset [7] which includes a range of solo instrumental recordings and musical mixtures. In this study, we propose deep learning models, designed to exceed the performance of existing state-of-the-art methods concerning onset detection accuracy on varied instrumentation as well as for string instruments, with a particular emphasis on their application to the newly proposed QTDS set with onset annotations.

1.1 Background

Event detection is a fundamental task in music information retrieval (MIR), which aims to identify the exact time points when musical events, such as notes or chords, begin. Accurate onset detection is crucial for various applications, including music transcription, beat tracking, and music synchronization. Over the years, numerous methods have been proposed for onset detection, ranging from digital signal processing techniques [2, 11] to more recent machine learning approaches [3, 22, 25]. Among these, deep learning-based methods have shown great potential in improving the performance of onset detection systems.

Eyben et al. [2010] employed an adaptive thresholding algorithm combined with a recurrent neural network (RNN) for onset detection, which represented a significant advancement in the field of MIR. Prior to their work, onset detection mainly relied on signal processing techniques and hand-crafted features, which often lacked

the ability to adapt to varying music genres, recording conditions, or instrumentations. Schlüter and Böck [2014] introduced the use of convolutional neural networks (CNN) for onset detection, enabling their system to recognize local patterns and structures within the input data that were difficult to capture with RNNs. Although CNNs can struggle to fully encapsulate the long-range temporal dependencies present within music signals, RNNs are capable of modelling longer dependencies at the cost of encountering challenges related to vanishing gradients during training [19]. Vogl et al. [2017] proposed the use of convolutional bidirectional recurrent neural networks (CBRNN) implementing a general onset detection pipeline of feature extraction, event classification and peak picking for the task of drum transcription. The authors reported higher classification results using the CRNN models compared to the baseline CNN and BRNN systems.

More recently, temporal convolutional networks (TCN) have emerged as an alternative approach that combines the benefits of both convolutional and recurrent layers, offering a more suitable solution for modelling temporal sequences in music [1]. Fonseca et al. [2021] showed promising results using a TCN architecture proposed in Böck and Davies [2020] for onset detection of percussion instruments. The original TCN was used in a multi-task learning approach for the task of joint beat, downbeat and tempo detection which demonstrated higher performance compared to other deep learning architectures in the literature. To date, these neural network (NN) architectures have not been evaluated together, making it difficult to assess the extent of performance differences in the task of onset detection across a variety of challenging musical data. In this paper we implement and compare the different architectures using a general purpose Böck dataset [7] as well as a newly proposed dataset of string quartet instruments.

1.2 Motivation

This study aims to investigate the effectiveness of different RNN, CRNN, and TCN architectures for the task of onset detection. We seek to evaluate the performance of these architectures in comparison to methods proven to be well suited for musical event detection, as well as models proposed for other music information retrieval tasks (e.g., beat and downbeat detection). We additionally explore the influence of incorporating string quartet recordings into the training sets on the model performance and propose a new dataset of string ensemble performances to advance future work in MIR.

The reasoning for comparing RNN, CRNN, and TCN methods in the context of music onset detection is as follows. RNNs are designed to manage sequential input, making them particularly suitable for tasks involving time-series data, such as natural language processing or audio signal processing, including music onset detection. While CNNs are adept at identifying local patterns and spatial information, CRNNs combine the strengths of both CNNs and RNNs, enabling them to capture both spatial and temporal information. This fusion of capabilities can be advantageous in applications like music onset detection, where processing complex data is required. To illustrate the differences between these models, the comparison between modes of operation of RNNs, CNNs, and CRNNs is illustrated in Figure 1 using an audio spectrogram input. Temporal convolutional networks offer another suitable solution for

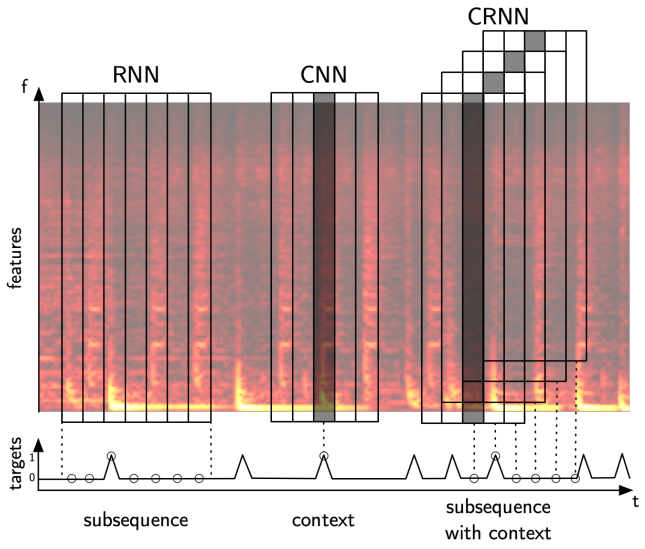

Figure 1: Demonstration of differences in modes of operation of RNNs, CNNs, and CRNNs (adapted from the original version [25]) on an audio spectrogram excerpt. Shaded regions denote target spectral frames within input subsequences.

modelling temporal sequences in music and are capable to model longer-range dependencies with less memory and computational requirements to RNNs. TCNs are comprised of dilated (i.e., with holes) convolution layers that maintain consistent input and output lengths, employing convolutions across the temporal dimension to discern time-dependent relationships in the input. TCNs are designed with skip connections, which help maintain a smooth gradient flow during training. This feature mitigates the vanishing gradient problem that often plagues RNNs [14], and allows TCNs to learn complex patterns in data more effectively.

The remainder of this paper is structured as follows: Section 2 introduces the proposed neural network models, while Section 3 provides a summary of the evaluation process. Section 4 presents the results and accompanying discussion, and finally, Section 5 presents the conclusions and directions for future work.

2 METHOD

Four different NN based systems are implemented. The systems use the same onset detection pipeline, where input features are fed into a model followed by a peak picking stage to determine the onset candidates as defined in Equation (3). The systems are inspired by architectures used for drum transcription in [25] and beat detection in [5]. All models are implemented using TensorFlow Python library[1] unless specified otherwise.

2.1 CRNN-a

In the proposed CRNN architecture, the model consists of two convolutional layers, followed by a bidirectional gated recurrent unit (GRU) layer, and a final softmax layer with two outputs. The

[1] https://www.tensorflow.org/

recurrent layers are implemented after the initial convolutional layers to provide the network with information about the structural patterns present in the data. All layers are time distributed to process every time-step of a sequence independently and use rectified linear unit (ReLU) activations. The two convolutional layers use 32 filters each, with 2×2 kernels with batch normalisation and 2×2 max pooling with same padding. A single recurrent GRU cell layer uses 96 filters with 2×2 kernels and dropout rate [24] of 0.15. The network uses a spectral context of 10 frames either side (κ=21) of the current target frame ($x^{t-10} : x^{t+10}$) for each time-step and is trained on sequences of length 100 (illustratively shown in Figure 1). The context of 21 frames and sequences of length 100 have been shown to perform well on individual instrument recordings in [25].

2.2 BTCN-a

The proposed bidirectional TCN architecture consists of a series of time-distributed layers, followed by a bidirectional GRU layer and an output softmax layer. The main component of TCN is implemented non-causally, meaning that dilated convolutions extend in both directions (i.e., backwards and forwards in time of the current frame). The first two convolutional layers share the same parameters as CRNN-a. The TCN uses a single residual block with filter size of 64 and 6 dilation rates ranging from 2^0 to 2^5 time frames with a dropout of 0.15. The output layer uses softmax activation to obtain the onset predictions. The same spectral context of κ frames and sequence length of 100 is used.

2.3 TCN-b

The TCN-b system is based on the architecture proposed in [5] and adapted for the task of onset detection. The baseline TCN-b consists of 2D convolutions, max pooling, dropout, and TCN architecture from [5]. The network uses the same spectral context as previous systems with 10 frames either side of the current target frame. A dropout of 0.15 is used with exponential linear unit (ELU) [10] activations. The network is implemented with 11 layers and filter size of 16 with dilations ranging from 2^0 to 2^{10} time frames. Finally, a softmax layer with 2 units is applied to produce the final output.

2.4 BTCN-b

The BTCN-a system uses the same general architecture as TCN-b system proposed in [5] but instead uses time distributed layers with κ frames of spectral context and is trained with sequences of length 100 to be comparable with the CRNN-a and BTCN-a models.

2.5 CNN and BRNN

In addition to the above, a BRNN [4] approach and a CNN [20] approach available in the madmom Python library [6] were used. Both algorithms are used with default parameters and use built-in peak picking method.

2.6 SuF and CoF

The SuperFlux (SuF) [9] algorithm calculates the difference between the short-time magnitudes of adjacent spectral frames by incorporating a spectral trajectory-tracking stage to the common spectral flux algorithm. This method was specifically designed for music signals featuring soft onsets and the vibrato effect in string instruments. The ComplexFlux (CoF) [8] algorithm builds upon the SuF approach by incorporating a local group delay, which enhances the method's robustness to fluctuations in the loudness of stable tones. Both approaches are used with default algorithm parameters and default logarithmic spectrogram parameters from the madmom Python library. Peak picking for SuF and CoF approaches is described in Section 2.8.

2.7 Input Features

To process an audio file with different NN models, the file must be systematically divided into frame-by-frame spectral features. Initially, the input audio (16-bit 44,100 kHz mono WAV files), is segmented into T frames utilizing an n-sample (n = 2048) Hanning window with a specified $\frac{n}{4}$ hopsize. Subsequently, a frequency representation for each frame is generated using the magnitudes of a discrete Fourier transform (DFT), resulting in an $\frac{n}{2} \times T$ spectrogram. A logarithmic frequency representation for each frame is generated following a process similar to the one described in [25]. By employing twelve triangular filters per octave, the magnitudes of a DFT are converted to a logarithmic scale ranging from 20 Hz to 20 kHz. This produces a F x T logarithmic spectrogram (F=84) used in as input fed into the NN models in separate configurations for a TCN and the models that implement bidirectional connections.

2.8 Peak Picking

As in most of the related work, peak picking is used to identify discrete onset candidate locations from the framewise activation function $\tilde{y}$ with time-steps t, which represents the output of the NN-based onset detection models. The output peaks are selected using the maximum value within a local window and above a threshold τ. We employ a peak picking method by [12] and revised in [21] to determine τ. This involves computing the mean of a window using a user-defined constant λ, a maximum value $t_{\max}$, a minimum value $t_{\min}$, and a window width controlled by δ as follows:

$$\tau^t = \text{mean}(\tilde{y}^{t-\delta} : \tilde{y}^{t+\delta}) * \lambda, \tag{1}$$

$$\tau^t = \begin{cases} t_{\max}, & \tau > t_{\max} \\ t_{\min}, & \tau < t_{\min}. \end{cases} \tag{2}$$

To obtain the onset classification vector O, we evaluate each time-step of $\tilde{y}$ to determine if it is greater than or equal to every other value within a specified number of frames, controlled by θ, and if it surpasses the threshold value τ:

$$O^t = \begin{cases} 1, & \tilde{y}^t == \max(\tilde{y}^{t-\theta} : \tilde{y}^{t+\theta}) \quad \& \quad \tilde{y}^t > \tau^t \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

2.9 Training

All models are trained using Adam optimiser [15] and an initial learning rate of 0.0001. The learning rate is fixed and is reduced by a factor of 0.2 whenever the F-measure (i.e., a standard metric assessing onset detection algorithms) on the validation set shows no improvement for 3 consecutive epochs. The data is divided by track into 70% training, 15% validation, and 15% testing sets. The training set is used to optimise each model and the validation set is used to prevent overfitting and to optimise the peak-picking

parameters for the F-measure calculation. Training is stopped if no improvement in F-measure on the validation set is observed for 10 consecutive epochs or if a maximum of 100 epochs have elapsed. All bidirectional models were trained with minibatch gradient descent and batch size of 32. The TCN model is trained on full sequences (i.e., recordings) with a batch size of 1.

3 EVALUATION

In this section, we present the evaluation methodology used to assess the performance of various NN models in detecting onsets across a range of real world recordings of musical mixtures as well as string quartet instrument recordings containing challenging playing styles. The evaluation is conducted on two datasets: the newly constructed quartet, trio, duet, and solo (QTDS) set, and the Böck set, a general-purpose dataset from [7]. The QTDS set features recordings of string ensembles performing in different musical conditions, while the Böck set contains a variety of solo and mixed music excerpts. Similarly to other work in event detection [7, 25] a cross validation strategy is implemented. If testing data is biased towards certain examples, it may misrepresent system performance. To prevent this, cross-validation divides the data into subsets. This mitigates bias and enhances result reliability by averaging performance across multiple subsets. We follow a 5-fold cross validation strategy on all datasets and report on mean precision, recall and F-measure scores which are taken across tested folds for each system under evaluation.

3.1 Datasets

3.1.1 QTDS Set. To investigate the performance of different NN models on the challenging onsets produced by string instrument in a variety of performance styles, a new dataset of note onset annotations is constructed from the selected quartet, trio, duet, and solo ensemble recordings part of the Virtuoso Strings dataset.[2] We refer to the chosen recordings, along with their corresponding onset annotations curated for the purposes of this project, as the QTDS dataset. The contents of QTDS are summarized in Table 1. The dataset includes isolated recordings of first and second violins, viola, and cello performing the following four QTDS excerpts: a quartet excerpt from Ludwig van Beethoven's Op. 59 No. 3 Finale (bars 210–271); a trio excerpt from Ernő Dohnányi's Op. 10, Marcia (bars 1–20); a duet excerpt from Amadeus Mozart's K424 mvt. 3 var. 2 (bars 32–48); and solo performances of the entire Finale of Joseph Haydn's string quartet Op. 74 No. 1 (285 bars). The QTDS excerpts were performed under three different musical conditions representing various interpretation instructions, chosen to span a wide range of performance types. The normal condition (NR) represents a concert-style performance. The speed condition (SP) represents performances that include spontaneous accelerando and decelerando initiated by a single musician (i.e., a designated leader). The deadpan (DP) condition represents performances with minimal expression in tempo and articulation. The 29 files in the QTDS set have a total length of approximately 41 minutes and have 8,254 annotated onsets. To enable future comparisons, the manually curated annotations for all instruments are publically available for download together with music score excerpts through the accompanying website.[3]

Ensemble	Composer	Excerpt	Bars	Dur.	Onsets
Quartet	Beethoven	Op. 59 No. 3 Finale	210–271	12.32	2908
Trio	Dohnányi	Op. 10, Marcia	1–20	7.01	1297
Duet	Mozart	K424 Mvt. 3 Var. 2	32–48	3.47	644
Solo	Haydn	Op. 74 No. 1	1–285	18.24	3405
				41.05	8254

Table 1: Quartet, trio, duet, and solo (QTDS) dataset information showing durations (in minutes) and numbers of annotated onsets for each set of ensemble recordings. Bottom row shows total number of onsets and length of recordings.

3.1.2 Böck Set. This dataset proposed in [7] consists of 321 files with a total length of 102 minutes and 25,927 onsets. It is a general purpose dataset which includes a variety of solo and mixed music excerpts. The types of audio recordings in the dataset are grouped into the following categories: complex mixtures (193 files), pitched percussive (60 files), non-pitched percussive (17 files), wind instruments (25 files), bowed strings (23 files), and vocal (3 files). To evaluate pretrained onset detection models from the publically available madmom Python library we separate 63 recordings from the Böck set for testing. For reproducibility, we select only recordings labelled as test files and mark this test set with a † symbol. While these files might have been used in the training of madmom models, the results on this subset should demonstrate a general performance of these methods on the Böck set.

To test the improvement of the systems with new training string instrument data marked with an * symbol, the Böck dataset [7] is extended with 12 recordings of a quartet (Q) ensemble performing together the excerpt of the Haydn Op. 74 No. 1 Finale (bars 1–49) under three different conditions (i.e., NR, SP, DP). The added data consists of approximately 10 minutes of violin, viola and cello recordings and 1,598 note onsets. While these recordings are separate from the ones used in the QTDS test set, there exists an overlap with the score in the first 49 bars of the test set performances of the Haydn's Finale.

3.2 Evaluation Methodology

We present a detailed analysis of the performance of the evaluated neural network models from two perspectives. Firstly, we explore the performance of the models on various subsets of the datasets, considering different ensemble types, performance styles, and musical conditions. This analysis aims to provide a deeper understanding of the models' strengths and weaknesses in detecting onsets under diverse circumstances. Furthermore, it offers valuable insights into the generalisability of the models and their ability to adapt to various musical contexts, ultimately guiding future research and enhancements in onset detection techniques. Secondly, we investigate the model capabilities in detecting onsets for each instrument in the string ensembles, including the first and second violins, viola, and cello. This examination helps to better understand performance of different models with respect to distinct timbral characteristics and

[2] https://github.com/arme-project/virtuoso-strings

[3] https://github.com/arme-project/onset-detection-for-strings

Böck set	F-measure	Precision	Recall
BTCN-a	0.816	0.835	0.836
CRNN-a	0.816	0.855	0.815
TCN-b [5]	0.812	0.868	0.793
BTCN-b [5]	0.786	0.844	0.774
BTCN-a*	0.827	0.895	0.795
CRNN-a*	0.822	0.882	0.800
TCN-b* [5]	0.816	0.875	0.791
BTCN-b* [5]	0.823	0.887	0.797

Table 2: Onset detection results on the Böck dataset [7] for models trained (where applicable) using the Böck dataset [7]. Models marked with an * included string quartet (Q) recordings in their training sets.

Böck test set†	F-measure	Precision	Recall
SuF† [9]	0.745	0.717	0.845
CoF† [8]	0.735	0.706	0.844
BRNN† [4]	0.703	0.902	0.613
CNN† [20]	0.878	0.944	0.835

Table 3: Onset detection results on the fixed test subset (†) from the Böck dataset [7] for models trained (where applicable) using the Böck dataset.

playing techniques of various instruments, and also highlights potential areas for improvement and optimization in onset detection tailored to each specific instrument.

We employ the conventional F-measure evaluation metric, which is derived from precision and recall. True positive, false positive and false negative onset candidates are considered valid if they occur within a 50ms tolerance window of the ground truth annotations. Onsets that occur within 30ms of each other are merged into a single onset at the middle position. The reported results are generated by sweeping the threshold parameter λ in the peak picking phase and selecting the value that produces the highest F-measure on the corresponding dataset.

4 RESULTS AND DISCUSSION

4.1 Subset Results

4.1.1 Böck Set. The results of the onset detection models on the Böck dataset [7] are presented in Table 2. Overall, the results demonstrate that the evaluated models yield comparable performance in detecting onsets on the Böck dataset. Models that include string quartet recordings in their training sets tend to have slightly better performance, suggesting that incorporating diverse training data may improve onset detection capabilities. The F-measure values range from lowest 0.786 for the baseline BTCN-b to highest 0.827 for the proposed BTCN-a architecture. Additionally, the inclusion of string quartet (Q) recordings in the training set resulted in a performance improvement for all proposed models, demonstrating the value of using additional string instrument data for training. In comparison to the baseline methods, the proposed models achieved competitive or better performance, highlighting the effectiveness of the bidirectional temporal and convolutional recurrent networks in onset detection tasks.

QTDS set	F-measure	Precision	Recall
BTCN-a	0.882	0.858	0.909
CRNN-a	0.886	0.856	0.923
TCN-b [5]	0.870	0.835	0.911
BTCN-b [5]	0.891	0.872	0.914
SuF [9]	0.898	0.869	0.930
CoF [8]	0.892	0.865	0.922
BRNN [4]	0.778	0.926	0.679
CNN [20]	0.867	0.787	0.973
BTCN-a*	0.906	0.896	0.918
CRNN-a*	0.907	0.887	0.930
TCN-b* [5]	0.874	0.847	0.907
BTCN-b* [5]	0.897	0.878	0.919

Table 4: Results tested on the QTDS set and trained (where applicable) using the modified Böck dataset [7]. Models marked with an * included string quartet (Q) recordings in their training sets.

Among the baseline methods evaluated on a fixed Böck test set presented in Table 3, the CNN model [20] achieved the highest F-measure, followed by ComplexFlux [8], and SuperFlux [9]. The BRNN model [4] obtained the lowest F-measure, however maintained a high precision indicating a low number of false positive onsets relative to the number of true positive onsets at the cost of missing many true positives indicted by a low recall score.

4.1.2 QTDS Set. The onset detection results for the models tested on the QTDS set, consisting solely of string instrument recordings, are presented in Table 4. Models that include string quartet recordings in their training sets tend to have slightly better performance, suggesting the importance of incorporating diverse training data. In comparison to some baseline models, the evaluated models perform well in terms of F-measure, precision, and recall. These results demonstrate the potential of the proposed models in onset detection tasks and provide insights into the importance of training data diversity in improving model performance. In particular, the proposed BTCN-a* and CRNN-a* models yield the highest F-measure and recall scores among all evaluated models, indicating better performance in detecting onsets in string ensembles. These results suggest that even a small amount of string quartet training data can significantly enhance the models' capability to detect onsets accurately in various musical contexts.

Including string quartet recordings (Q) in the training set resulted in performance improvements for all proposed models when tested on the QTDS set. In comparison to the baseline methods, the proposed models achieved better performance, highlighting the effectiveness of the bidirectional temporal and convolutional recurrent networks for onset detection in string instrument recordings.

4.2 Results Per Instrument

Figure 2 shows F-measures from different NN models for each instrument (i.e., first and second violins, viola, and cello). The results show that the CRNN model achieved the highest average F-measure

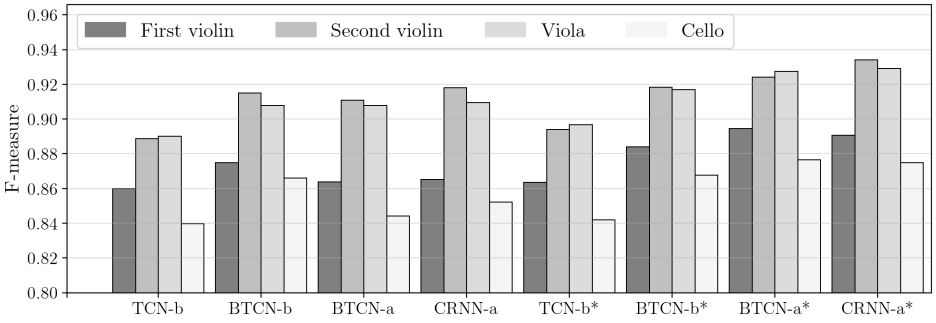

Figure 2: Mean F-measures per instrument tested on QTDS set.

score for all instruments. For the first violin, the BTCN-a* model achieved the highest F-measure score, while for the second violin, the CRNN-a* model obtained the highest score. For viola and cello, the CRNN-a* and the BTCN-a* models achieved the highest F-measure scores, respectively. The results suggest that the CRNN model is generally effective in detecting onsets in string instruments, while the BTCN-a* and CRNN-a* models are effective for the first and second violin, respectively. The improvement in F-measure scores for models that include a small amount of string training data, marked with an * indicates that the incorporation of string data into the training is capable of generalising to different playing styles present in the QTDS set. Additionally, the significantly lower onset detection performance on the cello recordings is in line with similar findings in [23], where cello's onsets were found to be the most difficult to manually annotate and to detect automatically. This could potentially be attributed to the cello's relatively thicker, low-tension strings which might lead to less distinct note attack characteristics when fingers or the bow engage and disengage with a string to initiate a new note. This finding should be considered during further work on note event detection process of string instruments as well as curation of new datasets.

5 CONCLUSIONS

In this paper, we proposed two new models for onset detection using bidirectional temporal and convolutional recurrent neural networks. We evaluated these models on a benchmark dataset from the MIR community and a newly proposed dataset of string instrument recordings with varied expressive playing styles. The results showed that the proposed models outperformed state-of-the-art algorithms on string recordings while maintaining comparable performance on other real world music examples. These results demonstrate the effectiveness of the models in tackling the challenges posed by the detection of onsets in string instruments. Additionally, the evaluations highlighted the importance of training data in achieving high accuracy in onset detection, particularly for string instruments where expert annotations are relatively scarce. We found that including a small amount of string training data in the proposed models improved their performance, emphasizing the need for further data collection and annotation efforts in this area. The new dataset with onset annotations represents a significant contribution to this research field and will strengthen the resources available to the community.

ACKNOWLEDGMENTS

This project is kindly funded by Engineering and Physical Sciences Research Council (EPSRC) grant with reference EP/V034987/1. We would like to thank Susan Li for helping to review the annotations.

REFERENCES

[1] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv:1803.01271. Retrieved from https://arxiv.org/abs/1803.01271* (2018).
[2] Juan P. Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike E. Davies, and Mark B. Sandler. 2005. A Tutorial on Onset Detection in Music Signals. *IEEE Transactions on Speech and Audio Processing, 13(5)* (2005), 1035–1047.
[3] Sebastian Böck. 2016. *Event Detection in Musical Audio.* Ph. D. Dissertation. Johannes Kelper University Linz.
[4] Sebastian Böck, Andreas Arzt, Florian Krebs, and Markus Schedl. 2012. Online Real-time Onset Detection with Recurrent Neural Networks. In *Proceedings of the International Conference on Digital Audio Effects (DAFx), York, UK.*
[5] Sebastian Böck and Matthew E.P. Davies. 2020. Deconstruct, Analyse, Reconstruct: How to improve Tempo, Beat, and Downbeat Estimation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Montréal, Canada.* 574–582.
[6] Sebastian Böck, Filip Korzeniowski, Jan Schlüter, Florian Krebs, and Gerhard Widmer. 2016. madmom: A New Python Audio and Music Signal Processing Library. *In Proceedings of the ACM International Conference on Multimedia (ACM-MM), Amsterdam, Netherlands* (2016), 1174–1178.
[7] Sebastian Böck, Florian Krebs, and Markus Schedl. 2012. Evaluating the Online Capabilities of Onset Detection Methods. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Porto, Portugal.* 49–54.
[8] Sebastian Böck and Gerhard Widmer. 2013. Local Group Delay Based Vibrato and Tremolo Suppression for Onset Detection. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Curitiba, Brazil.* 361–366.
[9] Sebastian Böck and Gerhard Widmer. 2013. Maximum Filter Vibrato Suppression for Onset Detection. In *Proceedings of the International Conference on Digital Audio Effects (DAFx), Maynooth, Ireland.*
[10] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *arXiv:1511.07289. Retrieved from https://arxiv.org/abs/1511.07289* (2015).
[11] Simon Dixon. 2006. Onset Detection Revisited. In *Proceedings of the International Conference on Digital Audio Effects (DAFx), Montréal, Canada.* 133–137.
[12] Florian Eyben, Sebastian Böck, Björn Schuller, and Alex Graves. 2010. Universal Onset Detection with Bidirectional Long Short-Term Memory Neural Networks. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Utrecht, Netherlands.* 589–594.
[13] João Fonseca, Magdalena Fuentes, Filippo B. Bonini, and Matthew E. P. Davies. 2021. On the Use of Automatic Onset Detection for the Analysis of Maracatu de Baque Solto. *In Perspectives on Music, Sound and Musicology* (2021), 209–225.
[14] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. 2015. An Empirical Exploration of Recurrent Network Architectures. In *International Conference on Machine Learning.* 2342–2350.
[15] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980. Retrieved from https://arxiv.org/abs/1412.6980* (2014).
[16] Başak Esin Köktürk-Güzel, Osman Büyük, Barış Bozkurt, and Ozan Baysal. 2023. Automatic Assessment of Student Rhythmic Pattern Imitation Performances. *In Digital Signal Processing, 133(1)* (2023).

[17] Miriam D. Lense, Eniko Ladányi, Tal-Chen Rabinowitch, Laurel Trainor, and Reyna Gordon. 2021. Rhythm and Timing as Vulnerabilities in Neurodevelopmental Disorders. *In Philosophical Transactions of the Royal Society B, 376(1835)* (2021).

[18] Marco A. Martínez-Ramírez, Wei-Hsiang Liao, Giorgio Fabbro, Stefan Uhlich, Chihiro Nagashima, and Yuki Mitsufuji. 2022. Automatic Music Mixing with Deep Learning and Out-of-domain Data. *In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Bengaluru, India* (2022).

[19] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the Difficulty of Training Recurrent Neural Networks. In *International Conference on Machine Learning, Atlanta, Georgia, USA*. 1310–1318.

[20] Jan Schlüter and Sebastian Böck. 2014. Improved Musical Onset Detection with Convolutional Neural Networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 6979–6983.

[21] Carl Southall, Ryan Stables, and Jason Hockman. 2017. Automatic Drum Transcription for Polyphonic Recordings Using Soft Attention Mechanisms and Convolutional Neural Networks. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Suzhou, China*.

[22] Carl Southall, Ryan Stables, and Jason Hockman. 2018. Improving Peak Picking Using Multiple Time-step Loss Functions. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Paris, France*.

[23] Maciej Tomczak, Min Susan Li, Adrian Bradbury, Mark Elliott, Ryan Stables, Maria Witek, Tom Goodman, Diar Abdlkarim, Massimiliano Di Luca, Alan Wing, and Jason Hockman. 2022. Annotation of Soft Onsets in String Ensemble Recordings. *arXiv:2211.08848. Retrieved from https://arxiv.org/abs/2211.08848* (2022).

[24] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. 2015. Efficient Object Localization Using Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 648–656.

[25] Richard Vogl, Matthias Dorfer, Gerhard Widmer, and Peter Knees. 2017. Drum Transcription via Joint Beat and Drum Modeling Using Convolutional Recurrent Neural Networks.. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), Suzhou, China*. 150–157.