Determining the Onsets in Audio Using EEG Signals

Shivani Butala
Language Technology Institute
Carnegie Mellon University
Pittsburgh, PA 15213
sbutala@andrew.cmu.edu

Suyash Chavan
Language Technology Institute
Carnegie Mellon University
Pittsburgh, PA 15213
schavan@andrew.cmu.edu

Sayali Moghe
Language Technology Institute
Carnegie Mellon University
Pittsburgh, PA 15213
smoghe@andrew.cmu.edu

Shruti Nair
Language Technology Institute
Carnegie Mellon University
Pittsburgh, PA 15213
shrutina@andrew.cmu.edu

Abstract

This report focuses on the objectives, methodology, challenges and major findings of our endeavor to observe electroencephalography (EEG) signals and their activity in response to onsets in natural music. We use a publicly available NMED-T dataset containing the EEG signals of 20 participants listening to a set of ten songs. We utilize deep learning techniques, mainly our own bidirectional long short-term memory network, which uses the activity of the EEG signals with respect to the ground truth timestamps of onsets in the songs to predict onsets of an audio. Through our novel approach, we were able to beat the baseline recall from 0.4 to 0.9245, F1 score from 0.54 to 0.6786, and Area Under the Curve (AUC) score from 0.54 to 0.605.

1 Introduction

The brain is one of the most complicated organs in the human body with several unsolved mysteries that surround it. The brain emits oscillating electrical voltages called EEG waves, that indicate brain activity. An abundance of information is present in these waves, and pave the way for several research initiatives to understand how this organ works.

The study of brain signals is a fascinating yet complicated domain. There are numerous scientists and researchers trying to learn more about the human brain every day and it is not an easy task. There is still a lot left to be understood. EEG signals are quite difficult to decipher and understand because they're incredibly noisy. But there are many tasks we can accomplish with them if we are able to understand them. We believe our project is trying to contribute more to the understanding of EEG signals. This is important because EEG signals capture everything your brain is processing, whether it is you blinking or speaking, or you listening to a familiar voice, or in our case, you hearing a new note. Observing how EEG signals change with respect to how you hear a new note can contribute to understanding how our brains work and react to new information or environments.

Music Information Retrieval, refers to systems and ideas which use audio/music as queries to extract required Information. Usual interfaces include "singing" or even "humming". Music Imagery Information Retrieval, on the other hand, is a nascent branch of research that attempts to identify music pieces from brain activity. This is done by attempting to rationalize and emulate the behaviour of the human brain when it is subjected to audio. By understanding this activity, the connection between audio input and brain activity can be uncovered and utilized.

An important aspect in Music Information Retrieval system is onset detection, detecting the beat or an "acoustic event" in audio. In layman terms, the onset marks the beginning of a musical note or beat. Expanding this problem to the Music Imagery Information Retrieval, is a little explored task, as the branch itself is quite new. Certain approaches in the field have considered the use of neural networks, but the work is very limited. For our project, we explore this domain and propose a novel methodology to detect audio onsets using EEG recordings from people listening to contemporary music. We will discuss what the baseline achieves, propose our own model design for the task, and discuss our results.

2 Related Work

2.1 Mind the Beat: Detecting Audio Onsets from EEG recordings of Music Listening [9]

This paper would serve as our primary reference, because we aim to improve on its results. The abstract of the paper claims that "Since there are no pre-existing works on this topic, the numbers presented in this paper may serve as useful benchmarks for future approaches to this research problem." This inspired us to tackle this and attempt to enhance the performance in this area. The dataset used for this paper is described in the Section 4 of this document (Dataset Description).

The paper describes their experimental set-up including how they obtained ground truth onset information for the audios, as well as how they pre-processed the EEG signals. They tried out 2 architectures: (1) a two-layer fully connected neural network and (2) a recurrent neural network. Both the proposed architecture were tried out in comparison to a standard spectral flux methodology, and a dummy method which would simply generate a beat every second. The performance of these models was judges based on Precision, recall and F-1 score. They reported that the RNN performed the best, having highest F-1 score of all. While the precision of the MLP was as good as the RNN, the recall was much lower, indicating that it had a much higher number of false negatives.

2.2 Detection of Note Onsets From EEG While Listening to Music [5]

This paper makes use of classification techniques such as Support Vector Machine and Logistic regression to predict the onsets of notes in music from EEG signals. The dataset consist of the EEG recordings of 15 participants who were subjected to 45 piano pieces. The paper labels every 100ms of EEG segments as "onset" and "non onset" depending on whether the segment includes a sustained note or not. Artifacts which include EEG noise were removed using visual inspection and a blind source separation algorithm called second-order blind identification. The input vector given to the classification model consisted of 280 features and a grid search, cross validated approach with regularization was utilized. Two different evaluations have been conducted in this paper namely segment level evaluation and music level evaluation. F1-score and area under the curve (AUC) were calculated for each predicted result from a 500-ms segment. Five out of fourteen participants area under the curve (AUC) metric indicated a value more than 0.7. The paper predicts the musical stimulus being listened for music evaluation and reports that maximum classification accuracy obtained was 91.7% when the predicted onset sequence was used to predict the musical stimulus being listened to. The paper thus claims that the obtained results suggest it is possible to decode the brain's response to each musical note from the brain waves while listening to music.

2.3 NMED-T: A Tempo-Focused Data Set of Cortical and Behavioral Responses to Naturalistic Music [6]

This paper introduces a Natural Music EEG Dataset, collected from 20 participants who heard a set of 10 commercially available musical works. The data set contains song stimulti from several genres and tempos, and all them contain electronically produced beats. They proposed that sometimes there are conflicting results across responses taken from various participants and also within participants' tapped responses. Research suggested that humans preferred certain frequencies related to natural movement. The paper also describes various trade-offs while collecting the data for generating the dataset. While collecting the data, they were limited in the number of songs. Collecting EEG signals for full length songs from every participant relegated their secondary tapping to shorter excerpts of music as they had to account for participant's fatigue. Hence, the paper aims at generating a dataset

which could potentially be used further in neuroscience and cognitive MIR, and further to process natural music in order to understand and determine audio recordings.

2.4 Universal Onset Detection with Bidirectional Long Short-Term Memory Neural Networks [4]

The paper presents an audio onset detector for all kinds of music using auditory spectral features and relative spectral differences using a Bi-directional LSTM. They use the Bi-directional LSTM as a reduction function and the method proposed by the authors does not make use of an onset detection method or its parameters to be tuned to any particular type of music. The audio data collected by the authors is transformed to the frequency domain using two parallel Short-Term Fourier Transforms with different window sizes. The magnitude spectra and their first order differences obtained from the STFTs is provided as an input to the Bi-LSTM network. The results were reported as correct if it was detected within a 100ms window around the annotated ground truth onset position. The authors reported two results, one for each set using the 100ms window for comparison on the first and 50ms window on the second. They compared the results with six other onset detection methods. It was concluded that the results were not as representative as other methods and vary depending on the parameters used in the model. The results were compared on the Bello data set. For complex music, the paper reported an improvement of 3.6%.

2.5 Online Real-time Onset Detection with Recurrent Neural Networks [2]

This paper discusses how to detect audio onsets in real-time without delay using RNNs, and claims that the performance of their model is comparable with the best offline models' performances. The baseline used is a state-of-the-art algorithm which won two MIREX onset detection algorithms. The authors aim to modify and make enhancements on this model in order to make the system useful for real-time online scenarios. The audio signal is processed frame-wise and is then transformed to a frequency domain the Short-Term Fourier Transform (STFT). The system proposes three parallel STFTs with different window lengths so that recent and old information is captured. The network was given an input as a dimensionally reduced linear magnitude spectrogram of STFTs. Dimensionality was reduced by passing each linear magnitude spectrogram through a filterbank of varied frequencies. As it was found out that bi-directional neural networks violate causality, it was not suitable for the task and was replaced with a unidirectional LSTM. The network was trained as a classifier with supervised learning and early stopping on 75% portion on the complete data set. 8-fold cross validation was used and metrics reported were precision, recall, and F1 score. The onset was determined to be correct if it was within 25ms range of the ground truth annotation. The detection window was kept very strict since it provided more significant results as compared to a detection window of 50ms.

2.6 mir_eval: A Transparent Implementation of Common MIR Metrics [7]

This paper illustrates how to evaluate and produce metrics for different MIR (Music Information Retrieval) tasks. Section 3.6 of the paper discusses the evaluation for onset detection. Precision, Recall, and F1 scores can be computed using mir_eval, and the "correctness" depends on a certain small window of time. We will be using mir_eval to evaluate the performance of our models for this project.

2.7 Improved Musical Onset Detection with Convolutional Neural Networks [8]

This paper aims at using convolutional neural networks to identify musical onsets with a data set of about 100 minutes of music with 26k annotated onsets. They claim that CNNs perform better than the state-of-art methods used previously, all the while requiring less manual preprocessing. They make use of different detectors for percussive and harmonic onsets, and combine results with many variations of the same scheme. A CNN is trained on an spectrogram excerpts centered on the frame, giving binary labels as "onsets" and "non-onsets". The training is done on spectrograms of different window sizes while testing is done using by computing the sectrograms of signals and feeding them to the network with an onset activation function. The system is designed in a way that it can be used as a black box for further exploration. The architecture achieves an F1 score of 88.5%, which is higher than the state-of-the-art RNN model. It was concluded that CNN provided better insights

using visualization techniques and the network was able to combine multiple minor variations of the same approach which is not possible using hand-designed algorithmic techniques.

2.8 madmom: A New Python Audio and Music Signal Processing Library [3]

The paper describes the functionality and usage of the *madmom* library. It is an open-source audio processing and music information retrieval library (MIR) consisting of numerous state-of-the-art algorithms for onset detection detection, beat, downbeat and meter tracking, tempo estimation, and piano transcription. The *madmom.features* package included high level functionalities used for onset detection including functions such as the *madmom.features.onsets* which contains functionality related to the onset detection. Overall, the paper provides an introduction to the *madmom* library and it's design principles and structure.

2.9 Learning Representations From EEG with Deep Recurrent-Convolutional Neural Networks [1]

The paper discusses on a novel approach to learn EEG signals using multi-channel EEG time series. It also discusses the advantages of the classification task in the field of learning EEG signals. The first step is to transform the EEG signals to a sequence of multispectral images. It is opposed to the standard EEG analysis techniques since this also takes into account the spatial information. The next step is to use a deep recurrent-convolutional neural network to learn representations from these sequence of images. This method is inspired by the video classification techniques. The stated method is observed to to preserve the spatial, spectral, and temporal structure of EEG which helps in finding the features which are less sensitive to variations and distortions. Two approaches were taken int consideration - Single frame approach where each image was constructed over the complete trial duration, and Multi-frame approach where each trial was divided into 0.5s windows and constructed an image over each time window, delivering 7 frames per trial. The results of the Single frame approach included extracting features by applying FFT on complete duration of the trial which led to a single 3-channel image per trial. The Multi-frame approach included extracting features separately for each window leading to conservation of temporal information rather than averaging into a single slice.

3 Methodology

3.1 Dataset Description

We are using NMED-T, a public domain data set of dense array electroencephalography (EEG) recordings from various subjects listening to natural music [6]. The song lengths vary between four and five minutes. There are ten songs and EEG recordings received from twenty participants in the data set. They provided a rating of familiarity and enjoyment for each of these songs. Along with this information, the data set also contains demographic information about the participants. Clean and aggregated information is stored in Matlab format while the tapping data is stored in the form of text files. The data set contains 55 files and the total size of the data set is 39 GB.

3.2 Setting Up Our Data

Because the actual songs were not provided to us, we obtained them and converted them to .wav files before using them. The song data was restructured using the MADMOM library's OnsetPeakPicking-Processor() function and RNNOnsetProcessor() function. By inputting an audio file into the function, we receive a series of timestamps that indicate an audio onset, and use these timestamps to create a sequence consisting of 0's and 1's where a 1 indicates an onset, and 0 indicates otherwise. Two set of files were created. The first one was for the binary classified onsets of different songs per user and the second was the timestamps of the songs. The first set of truths were used for training the neural network, and the timestamps were used for the purpose of evaluation which will be discussed later.

3.3 Data Preprocessing

As suggested by the main paper, we pass the EEG data with a bandpass filter (0.1–40 Hz) using SciPy. Later, we padded our dataset with zeros to a length of the longest song (37500 samples, 5 min) for

the sake of uniformity. This data is then segmented into one second blocks. We used k-fold cross validation to train and validate our models.

3.4 Baseline Selection

As our work builds upon the "MIND THE BEAT" project, we implemented the two novel models that were presented in the paper as the baseline. We implemented a multi-layer perceptron network as per the paper's suggestions, with a 2 layer implementation with a single ReLU activation function in between the layers. As per the original paper, the outputs of the model are not pushed through a sigmoid layer. This is done so that the model outputs can be sent to the mir_evaluate function call. The training data was explained as (x_i, y_i) where x_i is a EEG data of shape 125 x 125 which indicates the data collected from 125 channel for 1 second of data, divided into 125 frames and y_i is 1 x 125 shaped vector where each of the dimensions indicates the ground truth of that frame of second being an onset. Since the MLP expects flattened data, x_i is flattened before being passed through the MLP, which is done in the models forward pass.

The second baseline we implemented was the RNN model. This was comparatively trickier to understand based on the information we had. In the end, using the model diagram mentioned in the paper, we designed a Recurrent Neural Network, with feeds into 2 GRU layers with a hidden dimension of 64. The outputs of each are fed into a linear layer which outputs the onset logits. There was originally some confusion regarding the implementation at this point, as the granularity of the sequence of signal needed to be determined. After internal discussion, we decided to use each second's data as a sequence of frames that will be fed into the RNN. Thus every datapoint would be a sequence of 125 frames, each 125 dimensional. At every timestamp, the models outputs the raw logits of that time sequence being an onset.

3.5 Model Architecture

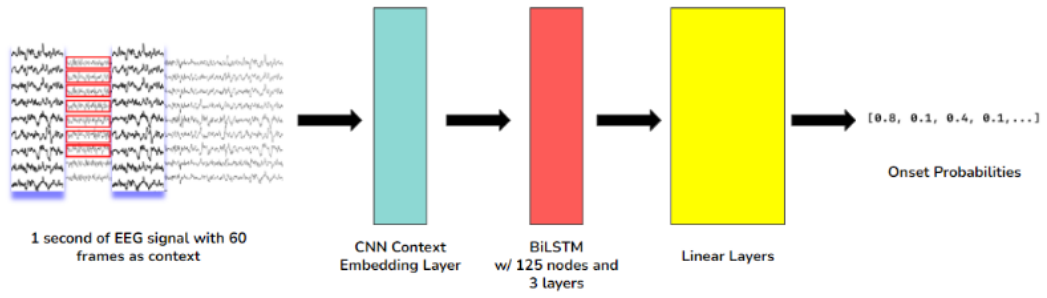

Figure 1: BiLSTM Architecture

Please refer to Figure 1 for our proposed model's architecture. The implementation details have been discussed in Section 4.4.

3.6 Evaluation Metrics

We decided to use precision, recall, F1 scores, and AUC scores as our evaluation metrics. As per the baseline, we are using a window of confidence to determine whether we are predicting an onset correctly or not. This is done using mir_eval libraries inbuilt function, mir_eval.onset.f_measure which takes a predicted and true series of timestamps in seconds, and uses a window to evaluate the measures.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{3}$$

Although the paper we are referring to only discusses these measures, we will also be observing and reporting AUC as a metric to dictate how well our model is performing. You will notice that two types of AUC scores are discussed. One of the scores was calculated using sklearn. The issue was that the scores came out to be lower than what they ideally should have been given our precision and recall. This happens because sklearn's function, sklearn.metrics.roc_auc_score does not observe the same window of confidence that mir_eval takes into account.

In order to compensate for this lack of flexibility, we manually calculated a rough estimate of the AUC score by plotting our (recall, precision) point , and drawing a line from (0,0) to (recall, precision), and then from (recall, precision) to (1,1). In order to get the AUC score, we calculate the area under the line formed by these points.

We choose these measures over something simpler like accuracy due to the imbalance of our dataset.

4 Experiments

All experiments were conducted using the PyTorch library. We made this decision as we all had experience working with PyTorch, and the framework is a widely accepted deep learning framework. We hope that our code contributions can help ease new research in this domain.

4.1 Fully Connected Neural Network

A fully connected network (FCN) was used by the first paper in the Literature Review as a baseline comparison network. The 125 channels × 125 time-step signal is turned into a vector with shape 1 × 15625. A 1 × 125 sequence is outputted to represent the equivalent binary onset output sequence in the audio example. Due to lack of a starter code, we implemented this network from scratch. The neural network consisted of 2 layers, and had a hidden layer of size 256.

The output of the FCN is modelled as:

$$y_{hat} = Linear(W_2 * ReLU(Linear(W_1 * flattened(x_i) + b_1)) + b_2) \tag{4}$$

4.2 Recurrent Neural Network

As this is a time-series, a Recurrent neural network makes intuitive sense. The baseline paper recommends a basic GRU based 2 layer recurrent network. It takes 1 second blocks of the data which is then recurrently passed through the network. The entire network was implemented by us taking a toll of our time span.

Our baseline network consists of an input layer of 15625 nodes, a hidden layer of 256 nodes, and an output layer of 125 nodes. We use ReLU as our activation. The input layer is 15625 nodes because it is taking in one second of EEG data, which originally has a dimension of 125 channels * 125 time steps. This array is flattened to give us a 1d array of size 15625. The output is an array of size 125 consisting of logits.

4.3 Convolutional Neural Network

The network consisted of two 1-dimensional convolutional layers and a ReLU activation after each layer. The first convolutional layer took the 125 channel input and outputted a 2500 channel vector. After applying the activation function, the second convolutional layer took the 2500 channel input and returned 125 channel vector back as the output having output size 65. The final classifier was a linear layer which took the 65 features as the input and resulted a single array of size 125.

4.4 Bi-LSTM Network

Each data point in our dataset can be explained as (x_i, y_i) where x_i is a 125 x 125 frame which represents 1 second of the EEG signal, with 125 channels and y_i is a 1x125 ground truth onsets.

Improvising on the baseline RNN model, a bidirectional LSTM is used along with cross validation to boost the performance of the model. We use a 1D convolutional layer to form the contextual embeddings for the Bi-LSTM model. The input has 125 channels and the output is kept with the same channels to maintain consistency in the embeddings. The bidirectional LSTM network consists of an input layer of 15625 nodes, a hidden size of 125 with 3 layers and a dropout of 0.4. The output of the Bi-LSTM layer is fed into a sequential layer consisting of 3 linear layers with LeakyReLU activation and dropout of 0.3 between layers. The output is a single array of size 125 consisting of logits.

We originally used binary cross entropy loss, which is calculated as shown below:

$$BCE = -((y)\log(y_hat) + (1-y)\log(1-y_hat)) \tag{5}$$

Later we used binary cross entropy loss with logits, where Sigmoid is applied along with the loss function call. We used madmom.features.onsets.peak_picking with a threshold of 0.055 to determine whether a certain time step contained an onset or not.

To incorporate more intelligence into our model, we added context which resulted in significant improvement to our performance. 1 second of the EEG signal, along with the context (best context of 60 for us), is inputted into a convolutional neural network to give us the right embeddings to pass into the next layer.

We have a bidirectional LSTM with 125 hidden nodes and 3 layers. This network takes in our context embeddings and passes the output into a set of linear layers for the final step. This network of linear layers has 250 input nodes, a hidden layer of 1028 nodes, followed by a DropOut layer (probability = 0.4) , another hidden layer of 4096 nodes with a Dropout layer of probability 0.4, and then our final output node.

4.5 Bi-LSTM Network with Positive Weights

We originally used the Binary Cross Entropy loss with logits as our loss function. The issue is that our dataset is very skewed. The ratio of onsets to non-onsets is quite small. In an attempt to solve this problem, we used positive weights along with BCELogitLoss to account for the class imbalance. We tried positive weights to be of ratio 0.64. Architecture remained the same.

5 Results

	Precision	Recall	F1 Measure	AUC
FCN	0.24	0.18	0.21	0.15
RNN	0.54	0.60	0.52	0.54
CNN	0.28	1.00	0.43	0.14
BiLSTM	0.5570	0.9245	0.6786	0.7390

Model	Precision	Recall	F1 Measure	AUC using Manual Calculations	AUC using sklearn
BiLSTM	0.5570	0.9245	0.6786	0.7390	0.6050
BiLSTM trained on weighted loss	0.7063	0.6931	0.6877	0.5066	0.6150

As mentioned earlier, we report the precision, recall and the F1 score using the mir_eval library's f-measure function. We also report the AUC values for each model. However, since the evaluation is done in a different manner as compared to standard binary classification task, the AUC reporting is done basically on this one operating point. Upon analyzing the model outputs, it becomes evident that the model is predicting more offsets than there actually are. This results in a high recall and low precision. We train each model with early stopping, and for each model, the best metric is reported. This trade-off in precision and recall can be considered using a loss which penalizes differently for each class. As explained earlier, this led us to experiment with Weighted BCELoss with Logits. Based on the evaluation metrics, our BiLSTM approach with a context outperforms the baseline comfortably in terms of the F-measure and AUC. Adding positive weights, improves the model's performance. The F1 score gets better and reaches 0.6877. However, the change in the precision and recall is quite large. With BCELogitLoss, we have a precision of 0.557 and recall of 0.9245. But with positive weights, we had a precision of 0.7063 and recall of 0.6931.

6 Challenges

We had a great learning experience working on this project and applying the concepts from the course in order to solve our problem. However, we also faced different challenges throughout the project that, although helped us learn, did end up taking up a lot of our time and because of which, we did not get a chance to work on more approaches that we would have liked to explore.

6.1 Domain Knowledge

The topic chosen for this project relies heavily on the usage of EEG signals and understanding how brain waves work. This requires a huge deal of understanding about EEG itself and being in a field that does not focus on these aspects, understanding them and then working through the problems of EEG signals required a bit of extra time.

6.2 Dataset

The open source NMED-T dataset of electroencephalographical (EEG) signals containing responses from 20 people over 10 songs is used as an input to our models. The EEG signals were in the form of .mat files where signals for every song were stored in a dictionary format. On extracting the data from the signal, every audio file had to be restructured to form two separate files. Another challenge we faced was that the dataset did not have true labels of the songs for this particular purpose. As per the baseline, we use the MADMOM library's inbuilt functions for generating this ground truths.

6.3 Model Architecture

While the paper provides architectures for both the baseline implementations, certain things were missed out from the paper. The baseline architecture for fully connected network although used a simple design for implementation, there was no mention of the activation function used. Along with this, the paper also failed to mention if any normalization layer was inserted between two FC layers. To understand the complete architecture, we got in touch with the authors of [9] who provided insight into not only the FCNN architecture but also the RNN implementation. The description provided for the RNN architecture in the paper proved to be incomplete and not clear enough to implement the baseline.

6.4 Lack of Starter Code

A significant amount of time and effort was spent on reproducing baseline results because we had no starter code to work with, so we had to code everything from scratch. As stated in the previous section, we also reached blocks in our progress due to the dependency on the author to reply to our emails when a detail of the architecture or experiment was missing.

6.5 Evaluation Strategies

The baseline paper makes use of the mir_eval library's F1 measure to determine the precision, recall, and F1 scores between the ground truth and predicted timestamps. The function used from mir_eval libraries uses a window size to calculate these metrics. The paper discusses using different window sizes for calculating the precision and recall values. We observed that using different window sizes changed the values of precision, recall, and F1 score. Sklearn, on the other hand, counts positive values only when the predicted onset and the ground truth onset coincide completely. This affected the precision, recall values and were observed to be vastly different than the values obtain from mir_eval. Besides these metrics, we also intended to use ROC-AUC score to understand the performance of the model with respect to the predicted values. The mir_eval library did not contain the required function and using the standard roc_auc_score from the sklearn library did outputted results that didn't truly illustrate the performance of our model because of its strictness and inability to consider a window. These limitations made it difficult for us to plot an ROC curve that was truly indicative of our model's performance.

7 Proposed Extensions

The class imbalance for a task of this nature leads to skewed results as we saw. In the future, techniques such as oversampling and undersampling can be used to deal with an imbalanced datasetsince they tend to work well with biomedical signals. To handle the imbalanced dataset problem, further specialized losses can be looked into, such as focal loss, which has been shown to work well for imbalanced datasets for object detection.

One key direction to build on this research is to consider the integration of prior information about the signals. For example we know that one onset cannot be immediately followed by another onset. This can be incorporated in the calculation of peaks. One way to go about this could be to use a dynamic programming based approach for introducing transition probabilities which are based on the priors. Another approach could be based on making Markov assumptions about the sequence probabilities.

An interesting extension we wished to look more into was the use of auto-regressive models. Auto-regressive models have historically performed well on sequential data, so it may hold promising results while working on EEG signal data.

8 Conclusion

The study of EEG signals is an interesting but highly complicated domain with more and more research being conducted everyday. In this project, we are addressing the problem statement of how to predict the onsets in music using physiological data obtained from participants listening to the music. We solve the problem using Deep Learning technologies such as Recurrent Neural Networks and Long Short Term Memory models to extract onsets in music using EEGs. We consider a window of one second of EEG obtained from participants during music listening sessions using the NMED dataset. The dataset consisted of 10 songs and concurrently recorded EEG from 20 users. We improved upon the existing baseline using the concept of context which considers the previous and future locations of onsets to correctly detect a current onset, obtaining a AUC score of 0.74, precision of 0.55, recall value of 0.925 and f1 score of 0.67. The results illustrate the feasibility of building a deep learning network to identify onsets of acoustic events in music and provide a basis for future work.

9 Acknowledgement

We would like to thank our mentor and teacher, Dr. Bhiksha Ramakrishnan, for his constant guidance, valuable insights, and patience with our project. We would also like to thank Roshan Ram, our TA mentor, for his advice and time with the project and report.

References

[1] Pouya Bashivan, Irina Rish, Mohammed Yeasin, and Noel Codella. Learning representations from eeg with deep recurrent-convolutional neural networks, 2015.

[2] Sebastian Böck, Andreas Arzt, Florian Krebs, and Markus Schedl. Online real-time onset detection with recurrent neural networks. 2012.

[3] Sebastian Böck, Filip Korzeniowski, Jan Schlüter, Florian Krebs, and Gerhard Widmer. madmom: a new python audio and music signal processing library. 2016.

[4] Florian Eyben, Sebastian Böck, Björn Schuller, and Alex Graves. Universal onset detection with bidirectional long-short term memory neural networks. pages 589–594, 01 2010.

[5] Yuiko Kumagai and Toshihisa Tanaka. Detection of note onsets from eeg while listening to music. pages 400–405, 2021.

[6] Steven Losorelli, Duc T. Nguyen, Jacek P Dmochowski, and Blair Kaneshiro. Nmed-t: A tempo-focused dataset of cortical and behavioral responses to naturalistic music. 2017.

[7] Colin Raffel, Brian Mcfee, Eric Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel Ellis. mir_eval: A transparent implementation of common mir metrics. *Proceedings - 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, 10 2014.

[8] Jan Schlüter and Sebastian Böck. Improved musical onset detection with convolutional neural networks. pages 6979–6983, 2014.

[9] Ashvala Vinay, Alexander Lerch, and Grace Leslie. Mind the beat: Detecting audio onsets from eeg recordings of music listening. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 231–235, 2021.