Attentive RNNs for Continuous-time Emotion Prediction in Music Clips

Sanga Chaki[1], Pranjal Doshi[2],
Priyadarshi Patnaik[3], and Sourangshu Bhattacharya[2]

[1] Advanced Technology Development Centre, IIT Kharagpur, India
[2] Computer Science and Engineering Department, IIT Kharagpur, India
[3] Humanities & Social Sciences Department, IIT Kharagpur, India

Abstract. Continuous-time prediction of self reported musical emotions is a challenging problem with many applications. However, there are relatively few studies on design of Deep learning models for the above problem. Existing methods for the same problem has used LSTMs, with modest success. In this work, we describe an attentive LSTM based approach for emotion prediction from music clips. We postulate that attending to specific regions in the past gives the model, a better chance of predicting the emotions evoked by present notes. We validate our model through extensive experimentation on the standard 1000 Songs for Emotional Analysis of Music dataset, which is annotated with arousal and valence values in continuous time. We find that the attentive models significantly improve the prediction performance of arousal and valence over vanilla LSTM, both in terms of R^2 and Kendall-τ metrics.

Keywords: Attention · Emotion Prediction · LSTM · Music Emotion

1 Introduction

Music is well known as an effective means of eliciting emotions in listeners [8]. Automatic determination of the perceived emotion in music has become a major area of focus for the music information retrieval (MIR) community. It finds varied applications in the domains of personalized and/or generalized music recommendations, organizing music databases, automatic music creation etc. Many recent studies have used the Circumplex model of affect, proposed by Russel [11], to denote music emotions. According to the dimensional Circumplex model [11], emotion is mapped into a 2-D plane, spanned by two axes denoting *arousal* and *valence*, as points given by the pair of values <*arousal, valence*>. Thus, the problem of emotion recognition/prediction is turned into a two dimensional regression problem [18]. Keeping this in mind, a number of publicly available music clip datasets have been developed, which help to test novel methods for music emotion prediction, [13]. It is understandable that the emotions related to music are a time-continuous process, where the context of the sequential music frames play an immense role on the related emotion. Relating this to the machine learning perspective, one can discern the need of context sensitive models

 In: N. Chhaya, K. Jaidka, J. Healey, L. H. Ungar, A. Sinha (eds.): Proceedings of the 3rd Workshop of Affective Content Analysis, New York, USA, 07-FEB-2020, published at http://ceur-ws.org

like recurrent neural networks (RNNs) in music emotion prediction. RNNs can access previous step activations using the *hidden states*, which remember relevant information about a pertinent sequence, to predict current emotion. Long Short-Term Memory (LSTMs) are such a type of RNNs, which have performed well in several MIR tasks including music emotion regression [15].

In this study, we propose to use the attention mechanism with a deep RNN structure composed of LSTMs, to predict the perceived emotion in each defined time frame of music continuously. We use the well known ComPare 2013 [12] set of features, extracted using the openSMILE tool [6] and the *1000 Songs for Emotional Analysis of Music* [13] dataset for evaluation in the present study, as it has proved to produce significant results in many recent works [14].

2 Related Work

Current state-of-the-art methods for audio sentiment analysis are mostly based on deep neural network. RNNs are a class of neural networks that are suited for time series data. They use the outputs of network units at time t as input to other units at time $t + 1$. This allows RNNs to store temporal information present within the input data. Though in theory, RNNs can keep track of arbitrary long-term dependencies in the input sequences, practically they suffer from the problem of vanishing gradients [10]. RNNs using Long Short Term Memory (LSTM) [7] units partially solve this problem. LSTMs have been found to be extremely useful to capture long-term context or dependencies in data [10] and are now widely used to solve a large variety of problems, including MIR tasks. Recently, Coutinho et. al. [2] and Weninger et. al. [15] used RNN-LSTM networks successfully to perform continuous time music mood regression. Weninger et. al[15] reports performance by averaging predictions and achieving R^2 of upto 0.70 and 0.50 for continuous time arousal and valence respectively. Another of their works [17] also tries to improve on the performance by using a different as cost function.

Though RNN-LSTMs are useful, it must be acknowledged that the difficulty of successfully capturing the context increases with length of input sequence [9]. This may become problematic for the neural network in case of longer input sequences like those in music. Here, inter-(musical)event relationships might play a bigger role in eliciting emotions than the actual sequence of (musical) events. Change in the order of the musical notes or other events might change the emotions considerably, much like context sensitive languages. To address this issue, Bahdanau et. al. [1] proposed the *attention* model, for the *encoder-decoder* architecture for neural machine translation. According to the *attention* model, to compute each output, the model will *attend* on those parts of the input sequence, which are more relevant for that particular output, by assigning higher weights to the associated encoder-side hidden states, using an *alignment* model. Though this model was originally proposed for the purpose of *encoder-decoder* based neural machine translation [1], it finds application in many different problems. Early works include use of LSTM in finding temporal structure in music [3],

music composition and generation [4] by Eck et. al. Recently, Coutinho et. al. [2] and Weninger et. al. [14], [15], [17] used RNN-LSTM networks successfully to perform continuous time music mood regression.

Most of the MIR tasks utilizing deep RNN-LSTM structures need considerable amount of training data to produce good results. In the domain of music emotion recognition, one such widely used dataset is the *1000 Songs for Emotional Analysis of Music* [13]. In the present work we use this dataset for evaluation. The openSMILE [6] toolkit is used to extract the ComParE [12] feature set for training.

3 Methodology

3.1 Dataset and Acoustic Features Used

In the present work, we use the *1000 Songs for Emotional Analysis of Music* dataset [13] for all experiments. Of the thousand clips,the dataset provides arousal and valence annotations for only 744 clips, which are used as *ground truth* values. Among these, 10% of the clips were assigned to the test set and the remaining formed the training set. We also use a set of purely acoustic affective features, given by the baseline feature set of the 2013 Computational Paralinguistics Evaluation (ComParE) tasks [12]. It has been shown by Weninger et. al. [14] that this set performs well in assessing emotion in terms of arousal and valence. The feature set contains 6670 features. These features are calculated by applying statistical functions to the contours of low-level descriptors (LLDs) of respective fixed length segments or time frames of the music audio signal, or the whole song. The statistical functionals include mean, moments etc. The LLDs include auditory weighted frequency bands, their sum, spectral measures such as centroid, roll-of point, skewness, sharpness, and spectral flux, MFCCs etc. The complete set of the LLDs, functionals and their detailed analysis can be found in [16], [5]. In the present work, we use TUM's open-source *openSMILE* feature extractor [6], to extract these features at non-overlapping intervals of 500 ms, for each music clip. The feature values for the datset were observed to be of different ranges. Thus, before performing multivariate regression, standard normalization was performed on the feature set. The features of the last 30 seconds of each clip from the dataset are used for this work. So, each clip is characterised by 61 feature vectors, each of size 6670. The arousal and valence annotations for each 500 ms time frame provided by the dataset [13] are used as the ground truth values.

3.2 LSTM-RNN

The key component of an LSTM is the cell state C, through which the relevant context/dependency information between the elements in the input sequence flows, with careful regulations by the *forget gate* (f), *input gate* (i) and the *output gate* (o). Intuitively, it can be understood that the *forget gate* (equation1)

decides which information is irrelevant and can be thrown away from the cell state.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

Next, the *input gate* (equation2) regulates what new information needs to be stored in the cell state, with the help of a vector of new candidate values (equation3).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

Thus, an update to the cell state is performed (equation4).

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

Finally, the *output gate* (equation5) decides the output of the network (equation6), based on a filtered version of the cell state.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

In the above equations, x_t and h_t denote the input and output at time t. Each W and b denote the associated weights and biases of each of the gates. The LSTM gates use the sigmoid function as the activation function. Though RNN-LSTMs (equation 7) provide a great way to carry relevant information from one step to the next through the cell state, the difficulty increases with length of input sequence. Basically, the neural network has to compress all the necessary information of an input sequence into a single fixed length vector, the last hidden state (equation 8). This may become problematic for the neural network in case of longer input sequences.

$$h_t = f(x_t, h_{t-1}) \tag{7}$$

$$c = q(h_1, h_2, \ldots, h_T) = h_T \tag{8}$$

3.3 Attention Mechanism

To address this issue, Bahdanau et. al. [1] proposed the *attention* model, for the *encoder-decoder* architecture for neural machine translation. Let x_i and y_i denote the i^{th} input and output of the model; h_i and s_i are the hidden states of the encoder and decoder associated with i^{th} input and output respectively, each annotation h_i contains information about the whole input sequence with strong focus on the parts surrounding the i^{th} input. c_i is the unique context vector associated with the i^{th} input; $g()$ is a function of y_{i-1}, s_i and c_i. According to the *attention* model [1], to compute each output (equation 9), a distinct context vector (equation 11) is used, which is a function of all the hidden states at the encoder side and not just the last one. Here, equation 10 is a modified form of equation 7.

$$p(y_i|y_1, y_2, \ldots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i) \tag{9}$$

$$s_i = f(s_{i-1}, y_i, c_i) \tag{10}$$

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{11}$$

Each time, the context vector c_i is calculated as a weighted sum of all the hidden states (equation 12). The idea being, for each output, the context vector will *attend* on those parts of the input sequence, which are more relevant for that particular output, by assigning higher weights to the associated encoder-side hidden states, using an *alignment* model. In equation 13, e_{ij} is the score of how well inputs around position j and the output at position i align or match.

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \tag{12}$$

$$e_{ij} = a(s_{i-1}, h_j) \tag{13}$$

We use a modified form of attention, as the problem we are tackling is music mood regression and not machine translation, thus there is no need for a decoder side architecture. The encoder encodes the input into a set of hidden states and the attention is applied on them to produce target arousal and valence values. Generally, the encoder in neural machine translation reads the input sequence $\mathbf{x} = (x_1, x_2, \ldots, x_T)$ – which is a sequence of vectors – and produces the hidden states $(h_1, h_2, \ldots, h_T)$, using some RNN approach like in equation 7. In case LSTM is used, equation 7 takes the specific form of equation 6. In most cases, the whole set of hidden states $(h_1, h_2, \ldots, h_T)$ are available to compute the context vector for the translation. So, all the hidden states are used for the context vector , either with attention (equation 11) or without (equation 8). This also makes sense for natural language processing, as the translation of an input x_t might depend on any input x_i, where both $i < t$ or $i > t$ are possible.

But, when we listen to music, the emotion associated with the music at t^{th} second is seldom influenced by the music following it. Rather, it might be argued, that the associated emotions at the t^{th} second will be more dependent on any music preceding it. Let the output be $\mathbf{y} = (y_1, y_2, \ldots, y_T)$. For the t^{th} output, y_t, it will be a function of a) the present hidden state h_t, b) the previous output y_{t-1}, c) the unique context vector c_t.

$$p(y_t | y_1, y_2, \ldots, y_{t-1}, \mathbf{x}) = g_1(h_t, y_{t-1}, c_t) \tag{14}$$

The unique context vector c_t depends on the sequence of annotations $(h_1, h_2, \ldots, h_{t-1})$, and is computed as a weighted sum of these annotations h_j. So, the model is *attending* to each h_j, corresponding to each of the inputs.

$$c_t = \sum_{j=1}^{t-1} \alpha_{tj} h_j \tag{15}$$

As in Bahdanau et. al.'s [1] work, referring to our equations 12 and 13, for each output y_t, we calculate the alignment between the corresponding h_{t-1} and each

Table 1: Summary of best results obtained across different models

Network Name	T = Topology				Arousal			Valence		
	#L	L1_Size	L2_Size	Attn	R^2_A	$\overline{\tau}_A$	MAE_A	R^2_V	$\overline{\tau}_V$	MAE_V
LSTM_NAT_700	1	700	-	-	0.70	0.21	0.13	0.39	0.10	0.15
LSTM_NAT_1024	1	1024	-	-	0.73	0.12	0.12	0.32	0.05	0.15
LSTM_NAT_700_128	2	700	128	-	0.69	0.20	0.12	0.11	0.11	0.16
LSTM_AT_300	1	300	-	Y	0.75	0.15	0.13	0.44	0.05	0.17
LSTM_AT_400	1	400	-	Y	**0.75**	0.07	0.13	**0.53**	0.05	0.16
LSTM_AT_300_128	2	300	128	Y	0.71	0.16	0.13	0.51	0.04	0.16

of h_j, where $1 \leq j \leq (t-2)$. So, the alignment model, when attending to h_j, is given by

$$e_{tj} = a(h_{t-1}, h_j) \tag{16}$$

Each of these scores e_{tj} are used to calculate the attention weights for each h_j as below

$$\alpha_{tj} = \frac{exp(e_{tj})}{\sum_{k=1}^{t-1} exp(e_{tk})} \tag{17}$$

4 Experimental Setup

4.1 Training and Evaluation

10-fold cross validation was used on the training and test sets. Evaluation measures are computed and reported on the entire test set and not by averaging across folds. We compare the proposed attention approach to the more traditional LSTM-RNN approach, which has provided good results in the past [15]. Both use same input features, standardized to zero mean and unit variance. Neural networks with one or two hidden layers were used for the experiments. The number of LSTM units (linear activation) used in each case varied from 32 to 1024. For the attention networks, the attention layer is added after the hidden layers, using sigmoid attention activation. Root Mean Square Propagation (RMSProp) optimization with 10 sequences per weight update is used for training. Training is done for maximum 30 epochs. An early stopping strategy is also used, making use of a validation set from each fold's training set. If validation error shows no improvement over 10^{-4} after 5 epochs, processing is stopped. Mean squared error (MSE) is used to calculate loss. Sequences are presented in random order during training. All hyper-parameters not explicitly mentioned here are left to their default values as in Tensorflow 1.14.

4.2 Models Used

The networks used for the current work are assigned names depending on whether they apply attention (AT) or not (NAT), followed by the layer sizes. For example, for an LSTM, no attention network with 1 layer of 128 hidden units, the

Table 2: Effect of different network topologies on LSTM_NAT regression models

Network Name	T = Topology				Arousal			Valence		
	#L	L1_Size	L2_Size	Attn	R^2_A	$\overline{\tau}_A$	MAE_A	R^2_V	$\overline{\tau}_V$	MAE_V
LSTM_NAT_128	1	128	-	-	0.56	0.11	0.13	0.20	0.11	0.15
LSTM_NAT_300	1	300	-	-	0.57	0.16	0.12	0.22	0.12	0.16
LSTM_NAT_400	1	400	-	-	0.60	0.14	0.11	0.29	0.08	0.16
LSTM_NAT_512	1	512	-	-	0.62	0.19	0.14	0.20	0.09	0.17
LSTM_NAT_700	1	700	-	-	**0.70**	**0.21**	0.13	**0.39**	0.10	0.15
LSTM_NAT_1024	1	1024	-	-	**0.73**	0.12	0.12	0.32	0.05	0.15
LSTM_NAT_700_128	2	700	128	-	0.69	0.20	0.12	0.11	0.11	0.16
LSTM_NAT_700_400	2	700	400	-	0.68	0.14	0.11	0.28	0.09	0.16

Table 3: Effect of different network topologies on LSTM_AT regression models

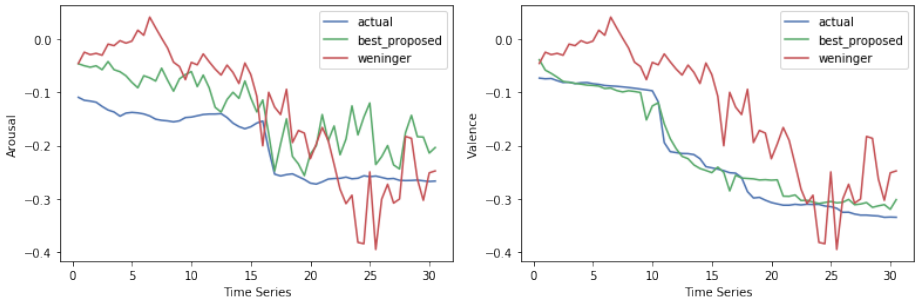
Network Name	T = Topology				Arousal			Valence		
	#L	L1_Size	L2_Size	Attn	R^2_A	$\overline{\tau}_A$	MAE_A	R^2_V	$\overline{\tau}_V$	MAE_V
LSTM_AT_32	1	32	-	Y	0.57	0.14	0.13	0.21	0.06	0.18
LSTM_AT_64	1	64	-	Y	0.63	0.14	0.14	0.22	0.09	0.18
LSTM_AT_128	1	128	-	Y	0.69	0.13	0.13	0.48	0.06	0.16
LSTM_AT_300	1	300	-	Y	**0.75**	**0.15**	0.13	0.44	0.05	0.17
LSTM_AT_400	1	400	-	Y	**0.75**	0.07	0.13	**0.53**	0.05	0.16
LSTM_AT_300_128	2	300	128	Y	0.71	0.16	0.13	0.51	0.04	0.16
LSTM_AT_400_128	2	400	128	Y	0.70	0.10	0.12	0.47	0.08	0.17

name is *LSTM_NAT_128*. For an LSTM, attention network with 2 layers of 700 and 128 hidden units each, the name is *LSTM_AT_700_128*. Thus, all networks belonging to each proposed model class are assigned the suffixes a) LSTM, no attention is *LSTM_NAT*, b) LSTM with attention is *LSTM_AT*. We replicate one of the best models proposed in Weninger et.al's work [15], with a single layer LSTM-RNN, though using the whole dataset [13] and entire feature set [12]. We get comparable results for layer size of 400 units. This is named *LSTM_NAT_400* and used as a baseline in this work.

4.3 Evaluation metrics

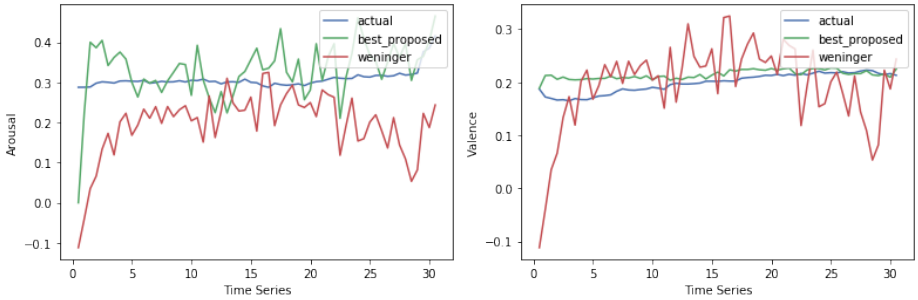
The metrics used for reporting the results are Coefficient of determination (R^2), average Kendall's τ per song ($\overline{\tau}$)and mean absolute error (MAE). The determination coefficient (R^2) is a key output of regression analysis, which provides a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model. Best possible score is 1.0. It can also be negative. If a data set has n values marked ($y_1 \dots y_n$), and each associated with a predicted value ($f_1 \dots f_n$). So, R^2 is defined as

$$R^2 \equiv 1 - \frac{SS_{res}}{SS_{tot}} \tag{18}$$



(a) Clip 2 - Arousal Comparison (b) Clip 2 - Valence Comparison

Fig. 1: Comparison of Arousal and Valence Predictions with the ground truth and baseline models for Clip 2



(a) Clip 584 - Arousal Comparison (b) Clip 584 - Valence Comparison

Fig. 2: Comparison of Arousal and Valence Predictions with the ground truth and baseline models for Clip 584

where, $SS_{res} = \sum_i (y_i - f_i)^2$ and $SS_{tot} = \sum_i (y_i - \overline{y})^2$, given $\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. Kendall's τ per song ($\overline{\tau}$) is a measure of how well the emotional profile of each song is captured by the regressor, as opposed to overall correlation. It measures the correspondence between two rankings. Values close to 1 indicate strong agreement, values close to -1 indicate strong disagreement. It is defined as

$$\overline{\tau} = \frac{P - Q}{\sqrt{(P + Q + T) * (P + Q + U)}} \tag{19}$$

where, P is the number of concordant pairs, Q the number of discordant pairs, T the number of ties only in target set $(y_1 \dots y_n)$, and U the number of ties only in predicted set $(f_1 \dots f_n)$. The mean absolute error (MAE) is given for reference. In the next section, we report the results of applying the proposed model for dynamic music emotion regression.

5 Experimental Results

5.1 Comparison of methods for emotion prediction

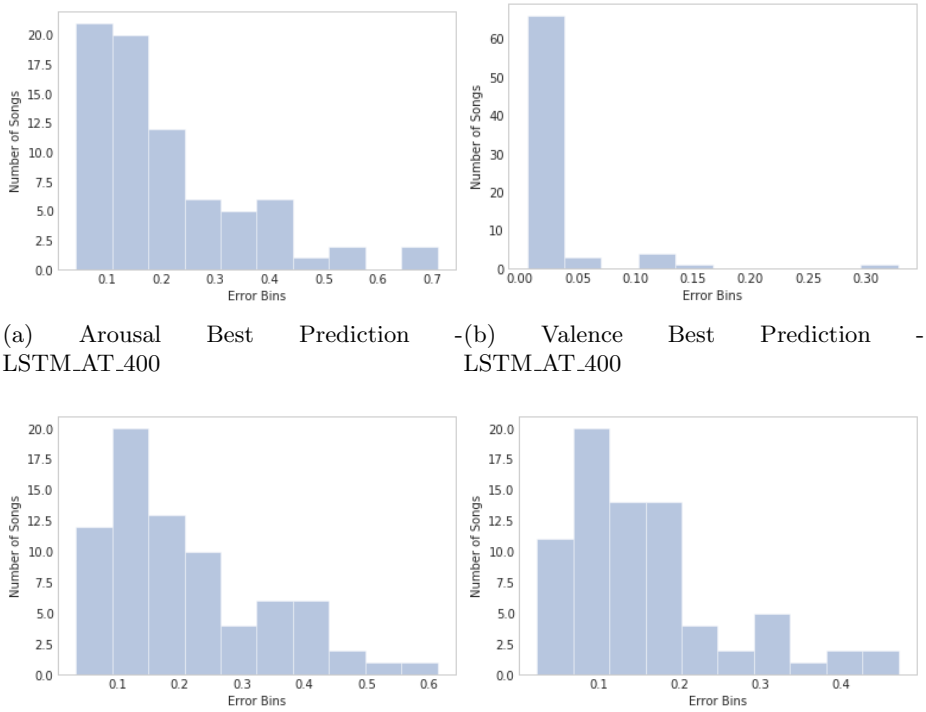
In the current work, we use four main types of LSTM-RNN models, which are

- No Attention (LSTM_NAT)
  - Single Layer: Eg. LSTM_NAT_128
  - Double Layer: Eg. LSTM_NAT_700_128
- With Attention (LSTM_AT)
  - Single Layer: Eg. LSTM_AT_128
  - Double Layer: Eg. LSTM_AT_400_128

In the first set of experiments, the performances of different models using different network topologies are compared. The best results obtained from each of these models are summarized in table 1.

In the case of LSTM_NAT networks, separate models for arousal and valence are trained using the LSTM-RNN architecture. Performances of networks having one hidden layer with 128, 300, 400, 512, 700, 1024 units and two hidden layers with (700, 128) and (700, 400) units are calculated. Table 2 reports the results for regression without attention, using different network topologies. For the single-layer topologies, a clear trend can be seen for arousal. With the increase in layer size (L1_Size), R^2_A increases. $\overline{\tau}_A$ increases till L1_Size = 700, but decreases for L1_Size = 1024. The metrics for valence does not follow a clear trend. It can be seen that *LSTM_NAT_700* performs best in terms of all the evaluation metrics considered, for both arousal and valence, giving $R^2_A = 0.70$, $\overline{\tau}_A = 0.21$, $R^2_V = 0.39$, and $\overline{\tau}_V = 0.10$. Though, *LSTM_NAT_1024* performs better for arousal ($R^2_A = 0.73$), its performance dips for valence ($R^2_V = 0.39$). $\overline{\tau}$ is also reduced for both arousal and valence. The two-layer topologies of this model, *LSTM_NAT_700_128* and *LSTM_NAT_700_400* perform comparable to the best single layer network *LSTM_NAT_700* and *LSTM_NAT_1024* for arousal, both in terms of R^2_A and $\overline{\tau}_A$. The performance for valence decreases in the 2-layer topologies. Thus, increasing layer size might help improve performance for arousal, but not for valence. Also, increasing the number of hidden layers might be unable to produce any significant improvement in performance for both arousal and valence.

The performances of of LSTM_AT networks, using different network topologies are presented in table 3. Performances of networks having one hidden layer with 32, 64, 128, 300, 400 units and two hidden layers with (300, 128) and (400, 128) units are calculated. A clear trend for performances of arousal and valence predictions are observed in this case. For arousal, among the single-layer topologies, best performance is recorded for the networks *LSTM_AT_300* and *LSTM_AT_400*, for R^2. It can be seen that addition of the attention mechanism improves the performance according to both metrics. For both arousal and valence, the best performances among all the models used is recorded for *LSTM_AT_400*, with $R^2_A = 0.75$ and $R^2_V = 0.53$. Henceforth, for all comparison purposes, we use this model as the best proposed model of this study. Increase in number of layers produce comparable performance for both arousal and valence and no significant change is observed.



(a) Arousal Best Prediction - LSTM_AT_400

(b) Valence Best Prediction - LSTM_AT_400

(c) Arousal Baseline Prediction - LSTM_NAT_400

(d) Valence Baseline Prediction - LSTM_NAT_400

Fig. 3: Arousal and Valence Error histograms over validation set using the proposed best methods and the baseline method

5.2 Comparison of fine-grained emotion prediction

In the second set of experiments, the best models for arousal and valence predictions, as obtained in the previous section, are used for fine-grained (per 500 ms) emotion prediction of some music clips. For arousal and valence predictions, we use the *LSTM_AT_400* model (table 1). We choose two clips from the *1000 Songs for Emotional Analysis of Music* [13] dataset, with clip ids 2 and 584 respectively. Clip 2 is of the genre Blues and negative valence (gloomy). Clip 584 is of the Folk genre, and significantly upbeat and positive valence (happy). We compare the predicted values with a) The ground truth values as provided by the *1000 Songs for Emotional Analysis of Music* dataset [13], and b) the baseline model [15], as represented by *LSTM_NAT_400*. Figures 1(a) and 2(a) denote the time varying arousal predictions, and figures 1(b) and 2(b)denote the time varying predictions for valence. In case of clip 2, 1(a) shows that the arousal prediction errors are lower for the proposed model initially, for the first 20 seconds. In the last 10 seconds, the errors of the proposed model and the baseline model

are comparable. But for valence prediction, the errors of the proposed model are significantly lower, as seen in figure 1(b). For clip 584, figure 2(a) shows that the arousal prediction errors are lower across the entire clip for the proposed model, thus matching the ground truth more. For valence prediction, as seen in figure 2(b) the errors of the proposed attention model are significantly low for the entire clip.

5.3 Cross analysis of errors

In the third set of experiments, we use the best proposed model LSTM_AT_400 and the baseline model LSTM_NAT_400 on the validation set, to group the clips into error bins for arousal and valence prediction. These are shown as histograms in figure 3. Comparing figures 3(a) and 3(c), it can be seen that, for the proposed model, the number of clips with higher values of errors are less, in case of arousal. In case of valence, for the proposed model, almost all the clips are grouped into the error bins ≤ 0.05 (figure 3(b). Whereas for the baseline model, a significant number of clips across bins are present.

6 Conclusion

We demonstrate that the state of the art models for continuous-time emotion prediction perform modestly, thus emphasizing the need for further research in this area. We have proposed an attentive LSTM based model which improves the state of the art performance significantly, on standard benchmark dataset with standard metrics.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015)
2. Coutinho, E., Weninger, F., Schuller, B.W., Scherer, K.R.: The munich lstm-rnn approach to the mediaeval 2014" emotion in music'" task. In: MediaEval (2014)
3. Eck, D., Schmidhuber, J.: Finding temporal structure in music: Blues improvisation with lstm recurrent networks. In: Proceedings of the 12th IEEE workshop on neural networks for signal processing. pp. 747–756. IEEE (2002)
4. Eck, D., Schmidhuber, J.: A first look at music composition using lstm recurrent neural networks. Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale **103**, 48 (2002)
5. Eyben, F.: Real-time speech and music classification by large audio feature space extraction. Springer (2015)
6. Eyben, F., Wöllmer, M., Schuller, B.: Opensmile: the munich versatile and fast open-source audio feature extractor. In: Proceedings of the 18th ACM international conference on Multimedia. pp. 1459–1462. ACM (2010)

7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
8. Juslin, P.N.: From mimesis to catharsis: expression, perception, and induction of emotion in music. Musical communication pp. 85–115 (2005)
9. Koehn, P., Knowles, R.: Six challenges for neural machine translation. In: Proceedings of the First Workshop on Neural Machine Translation. pp. 28–39. Association for Computational Linguistics (Aug 2017). https://doi.org/10.18653/v1/W17-3204
10. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International conference on machine learning. pp. 1310–1318 (2013)
11. Russell, J.: A circumplex model of affect. Journal of Personality and Social Psychology **39**(6), 1161–1178 (1980)
12. Schuller, B., Steidl, S., Batliner, A., Vinciarelli, A., Scherer, K., Ringeval, F., Chetouani, M., Weninger, F., Eyben, F., Marchi, E., et al.: The interspeech 2013 computational paralinguistics challenge: social signals, conflict, emotion, autism. In: Proceedings INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France (2013)
13. Soleymani, M., Caro, M.N., Schmidt, E.M., Sha, C.Y., Yang, Y.H.: 1000 songs for emotional analysis of music. In: Proceedings of the 2nd ACM international workshop on Crowdsourcing for multimedia. pp. 1–6. ACM (2013)
14. Weninger, F., Eyben, F., Schuller, B.: The tum approach to the mediaeval music emotion task using generic affective audio features. In: Proceedings MediaEval 2013 Workshop, Barcelona, Spain (2013)
15. Weninger, F., Eyben, F., Schuller, B.: On-line continuous-time music mood regression with deep recurrent neural networks. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5412–5416. IEEE (2014)
16. Weninger, F., Eyben, F., Schuller, B.W., Mortillaro, M., Scherer, K.R.: On the acoustics of emotion in audio: what speech, music, and sound have in common. Frontiers in psychology **4**, 292 (2013)
17. Weninger, F., Ringeval, F., Marchi, E., Schuller, B.W.: Discriminatively trained recurrent neural networks for continuous dimensional emotion recognition from audio. In: IJCAI. vol. 2016, pp. 2196–2202 (2016)
18. Yang, Y.H., Lin, Y.C., Su, Y.F., Chen, H.H.: A regression approach to music emotion recognition. IEEE Transactions on audio, speech, and language processing **16**(2), 448–457 (2008)