Understanding the Effectiveness of Different Machine Learning Algorithms for Emotion Classification with EEG Data

Tanmoy Sarkar Pias
Department of Computer Science
Virginia Tech
Blacksburg, VA 24061
tanmoysarkar@vt.edu

Jesse Harden
Department of Computer Science
Virginia Tech
Blacksburg, VA 24061
jessemh@vt.edu

Abstract

Emotion detection and classification is an interesting task area for machine learning that boasts diverse applications. One source of data used to classify emotions is that of EEG data. To better understand what machine learning methods work well with EEG data for binary emotion classification, the authors tried several methods on the confused student EEG dataset and then took the best-performing method to test on the DEAP dataset. The results of the authors' experiments suggest that 1D Convolutional Neural Networks can perform the task of binary emotion classification with EEG data quite well.

1 Introduction

Emotion detection and classification make up an interesting task area for machine learning; this task area boasts many diverse applications, including, but not limited to, detecting confusion in students [10], detecting anger in call-center work [13], and addressing challenges in human-robot interaction [1]. Emotion, as noted by Koelstra et al. [3], is a "psycho-physiological process triggered by conscious and/or unconscious perception of an object or situation." This process, which is generally linked to personality, mood, and more, is a key aspect of what and how humans communicate with each other [3] and attempt to communicate with machines. Given that emotional information can be valuable to many processes and tasks, such as ensuring a student is grasping the material presented and is not suffering from confusion, enabling machines to accurately detect and classify emotions may improve human-computer interactions in various contexts, leading to better teamwork between humans and machines as well as increased user satisfaction.

Thankfully, there exist tools to capture data from brain activity that can be used to detect emotions, such as EEGs. Indeed, Wang et al. [10] used EEG data to build a machine learning model predicting whether a student was confused by a particular video. Zheng et al. [14] used Deep Belief Networks (DBNs) to classify whether a subject was experiencing a positive or negative emotion with EEG data. These are just a couple of examples of the use of machine learning with EEG data to classify emotions.

To further understanding about what machine learning methods tend to work well for binary emotion classification with EEG data, the authors of this report sought to test various machine learning models on two EEG emotion detection and classification datasets to determine which model or models tend to perform best. The datasets used are as follows: the dataset for detecting student confusion gathered by Wang et al. [10], and the DEAP dataset gathered by Koelstra et al. [3]. Each of these datasets has its own attributes which make for interesting work.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

The confusion dataset provided by Wang et al. [10] is interesting for several reasons. With the rise of online classes due to the pandemic, the problems inherent with online learning delivery, such as not being able to see student faces and thus not be able to read their emotions, have become more widespread. While tools such as Zoom do allow face-to-face interaction, many students may choose not to show their faces during an online class. Although online classes can use tools such as "interactive forums and feedback quizzes" [10] to help bridge the gap between teachers and students, these tools aren't necessarily as good at delivering timely feedback as can be done in an in-person class. Thus, being able to detect confusion automatically from EEG data could assist professors, TAs, and others involved in online instruction by alerting them to who appears to be confused and what materials seem to be the most confusingly presented. In addition, binary classification on this dataset has proven challenging, with Ni et al. [5] achieving 73.3% accuracy with a Bidirectional LSTM model and Wang et al. [11] achieving 75.0% accuracy with a Confounder Filtering-improved Bidirectional LSTM model.

The DEAP dataset, as presented by Koelstra et al. [3], is a dataset which contains EEG data as well as participant ratings on valence and arousal according to Russell's model of affect [7], along with the feature of dominance, which ranges from a "helpless and weak feeling to an empowered feeling" [3]. Unlike the confused student dataset by Wang et al. [10], the DEAP dataset contains EEG data from multiple channels, resulting in more information being present with which to classify emotions. The three participant rating features mentioned above can be used to map nearly all emotions in a 3D space, as seen in the work of Verma and Tiwary [9]; each of these features (valence, arousal, and dominance) were all rated on a scale of 1 to 9. For the purposes of this work, the authors only considered valence, which was divided into 2 categories for binary classification: negative (1-4.99...) and positive (5-9).

While there are many machine learning models that could be tested on these two datasets, the authors chose to focus on the following models: SVM, Long Short Term Memory (LSTM) neural networks, including Bidirectional LSTM, 1D Convolutional Neural Networks, and autoencoders.

2 Related Work

This works considers prior work on each of the two datasets to help guide analysis. In this section, prior work on each dataset is explored.

2.1 Confused Student Dataset Prior Work

Starting with the initial work of Wang et al. [10], wherein this dataset was created, machine learning was used to model confusion. In this initial paper, 10 students watched 10 videos, 5 of them meant to be confusing to non-experts in the subject matter of the video, while wearing a single-channel Mindset designed to gather EEG data from the frontal lobe. After gathering and processing the data, Naive Bayes was used to model and predict whether a student was confused, with student-independent classifiers reaching an average of 57% accuracy. While this result meant the model was not much better than randomly guessing, further work would improve upon this initial attempt.

Ni et al. [5] would improve upon this initial result through the use of Bidirectional Long Short Term Memory (Bi-LSTM) neural networks. Bi-LSTM was used because of the ability of LSTM neural networks to analyze time series data, and the ability of Bi-LSTM neural networks to use both past and future data to improve predictions. Since EEG data is like time series data, this usage makes sense. Indeed, the Bi-LSTM model greatly improved upon Wang et al.'s results, generating 73.3% accuracy.

Wang et al. [11] would improve further upon the work by Ni et al. [5] by introducing a Confounder Filtering (CF) approach that effectively removes the effect of confounding variables by setting their weights to 0. Using this CF approach with a Bidirectional LSTM neural network appears to have improved the generalization of the model to unseen data points, as seen by the improved accuracy of 75.0% for the CF-Bi-LSTM model.

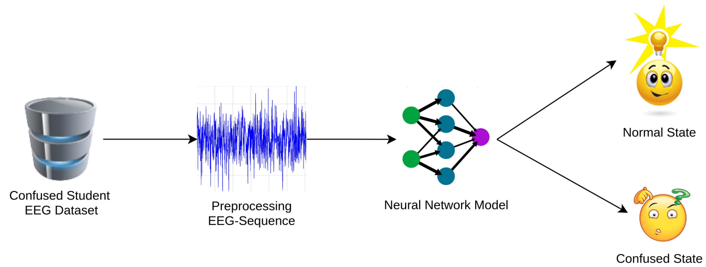

Figure 1: Process for Confused Student Dataset.

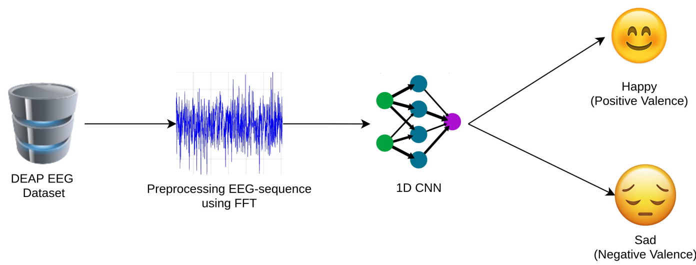

Figure 2: Process for DEAP Dataset.

2.2 DEAP Dataset Prior Work

The DEAP dataset compiled by Koelstra et al. [3] is a popular dataset for emotion classification. While not all papers which have used the DEAP dataset can be summarized here, a few relevant ones are noted.

In 2017, Tripathi et al. [8] used Deep Neural Networks and Convolutional Neural Networks to classify valence and arousal as binary (positive or negative) and trinary (positive, neutral, or negative) classes. The results with both types of neural networks improved accuracy (81.41% for CNN and 75.78% for DNN on binary valence classification) over earlier work, such as that by Rozgic et al. [6] in 2013.

In 2019, Xing et al. [12] combined a Stacked autoencoder (SAE) with an LSTM neural network to analyze the DEAP dataset and classify valence and arousal according to a binary (positive or negative) class. Xing et al. achieved similar levels of accuracy for the mean accuracy to Tripathi et al., with their SAE+LSTM model achieving 81.10% mean accuracy on valence.

3 Approach / Method

To investigate what kinds of methods perform best for binary classification on EEG datasets for emotion recognition, the authors first did some data preparation and examination of both the confused student dataset from Wang et al. [10] and the DEAP dataset. Next, the authors experimented with different methods on the confused student dataset due to it being a simpler dataset. Finally, the machine learning method which worked the best on the confused student dataset was taken and used on the DEAP dataset. In this section, the authors cover the data examination and preparation and the machine learning methods used in detail for each dataset. An overview of our process can be seen in Figure 1 and Figure 2.

3.1 Confused Student Dataset Preparation

The first dataset to be examined was the confused student dataset provided by Wang et al. [10]. This dataset consists of EEG Data from 10 different students as they watched each of 10 different videos. Thus, this dataset contains EEG data for 100 different combinations of student and video. In addition, the raw EEG data this dataset is based on has already been decomposed into several different features based on the frequency bands for EEG data: Delta (<4 Hz), Theta (4-7 Hz), Alpha (8-11 Hz), Beta

(12-29 Hz), and Gamma (30-100 Hz). Alpha, Beta, and Gamma are further divided into two features each, one for the lower end of their spectrum and one for the higher end. Futhermore, two features, Attention and Mediation, were pre-calculated by Wang et al. [10] for this dataset. The dataset also contains labels for the subject, as well as the video. Finally, there are two confusion labels, each normalized to either confused (1) or not confused (0): one for the expected, predefined value for the video, and one for participant's actual rating of how confused they were.

To prepare for the experiments with this dataset, the authors first sought to determine how many rows there were for each individual video-subject pair. To do this, the authors took the video id (0-9 for the ten videos) and combined it with the subject id (0-9 for the ten subjects) to form a unique combined identifier. Then, a count for each of the unique feature sets was taken, with the smallest count being 112 rows. To ensure that all video-subject pairs are equally represented in our training, the authors limited each video-subject pair to the first 112 rows.

Next, feature selection on the confused student dataset was performed manually. Since, as noted by Wang et al. [11], subject identity information was found to be a confounding factor, we removed the column that represented the subject's identity, as well as the combined column for each video-subject pair, from the data the models would be trained on. In addition, we also removed the column identifying the particular video a subject was watching. Finally, since we wanted to train on subject's self-rating of confusion rather than the expected result, we removed the predefined value for confusion column from the data.

Lastly, the now-prepared dataset was divided into training and testing sets, with 80% of the data going to the training set and 20% going to the testing set.

3.2 DEAP Dataset Preparation

The second dataset to be examined was the DEAP dataset created by Koelstra et al. [3]. This dataset consists of EEG data as well as participant ratings from 14-16 participants of how music videos made them feel in terms of arousal, valence, and dominance, among other items. Unlike the confused student dataset, the DEAP dataset's EEG data comes from multiple channels, 45 to be precise. In addition, while the dataset does contain ratings for arousal, valence, and dominance, the authors chose to focus on valence.

To prepare for experiments with this dataset, the individual participant data, stored in .dat files, was first converted to a NumPy data format for speed of processing. Once this was done, Fourier analysis was done and the data was combined into a single dataset that was then split into two datasets: one for training, containing 80% of the data, and one for testing, containing the rest of the data. In addition, the participant rating categories (arousal, valence, and dominance), which are normally from 1-9, were divided into two binary categories, with 1-4.99... labeled as negative and 5-9 labeled as positive.

Lastly, feature selection on the confused student dataset was performed manually. Given the wealth of channels available, the authors wished to narrow the subset to what might be the most valuable channels. Based on the channels chosen for the EPOC+ device [2], the authors chose the corresponding 14 EEG channels from the DEAP dataset to use for predictions.

3.3 Machine Learning Methods

To investigate what kinds of machine learning methods work well for binary classification of emotion with EEG data, the authors prepared four different machine learning methods: an SVM as a baseline, an autoencoder-based method, an LSTM, a Bidirectional LSTM, and 1-dimensional CNNs. In this subsection, the authors cover the methods and architectures used.

3.3.1 SVM Method

To develop a baseline for success on the confused student dataset, the authors chose to try a Support Vector Machine (SVM) method. Regularization parameter was set to 1, and the data was standardized using the transform method of the StandardScaler library.

3.3.2 Autoencoder Method

Based on the success of the work of Xing et al. [12], the authors chose to try an autoencoder method, with two autoencoder architectures tried in the course of this project. Both architectures started by taking in the data, feeding it to the autoencoder, which used the ReLU activation function, and then passing the data to a dense layer using the sigmoid activation function with the same number of nodes as the number of columns originally passed in: 11.

The first autoencoder architecture started by taking the data and passing it to 50 nodes in the first encoding layer, going down to 5 nodes in the second encoding layer. The first decoding layer contained 5 nodes, and then went up to 50 nodes in the second decoding layer. Finally, the last layer was a dense layer.

The second autoencoder architecture started by taking the data and passing it to 128 nodes in the first encoding layer, going down to 64 and 32 in the second and third encoding layers respectively. The next layer, the first decoding layer, contained 64 nodes, and the second decoding layer contained 128 nodes. Finally, the last layer was a dense layer.

3.3.3 LSTM Methods

Based on the success of the work of Ni et al. [5] and Wang et al. [11], the authors also chose to try an LSTM and Bidirectional LSTM model.

The basic LSTM architecture used 3 LSTM layers, each using the ReLU activation function, followed by a dense layer and then an output layer; the output layer used the sigmoid activation function. This model was trained with Binary Cross-Entropy loss function and RMSprop optimizer.

The Bidirectional LSTM models started with a Batch Normalization layer followed by a Bidirectional LSTM layer, with a dropout layer with 30% dropout and a dense layer using the hard sigmoid activation function finishing the architecture in that order. These models were also trained with Binary Cross-Entropy loss function and RMSprop optimizer.

3.3.4 1D CNNs

Based on the success of the work of Tripathi et al. [8], the authors chose to try using Convolutional Neural Networks (CNN) in addition to the other two methods.

The first 1D CNN architecture started by taking the data and passing it sequentially to 6 1D CNN layers; each of these layers had 64 filters, a kernel size of 3, and used the ReLU activation function. Following those layers was a dropout layer for regularization with a fairly strong dropout percentage of 50%, and then a max pooling layer with a pool size of 2. Finally, the data is then flattened and passed to a dense layer of size 100 with the ReLU activation function before being passed to the output layer using the Hard Sigmoid activation function.

The second 1D CNN architecture started by taking the data and passing it sequentially to 4 1D CNN layers using the ReLU activation function, with each 1D CNN layer followed by a batch normalization layer and a max pooling 1D layer with a pool size of 2. The first 1D CNN layer had 1024 filters and a kernel size of 9. The second 1D CNN layer had 512 filters and a kernel size of 6. The third 1D CNN layer had 256 filters and a kernel size of 6. The last 1D CNN layer had 128 filters and a kernel size of 6. After the 1D CNN layers, the data was flattened and passed to 3 dense layers, using the ReLU activation function, of sizes 1024, 256, and 64 respectively; each dense layer was followed by a dropout layer for regularization with a dropout percentage of 20%. Finally, the output layer made the predictions with the Softmax activation function.

4 Experiments

As mentioned in the beginning of the prior section, the confused student dataset was experimented on first, and the strongest method was then tested and refined on the DEAP dataset. In this section, the authors go over the experiments run on each dataset and their results.

Table 1: Results of LSTM Experiments

Batch Size	Epochs	Train Accuracy	Test Accuracy
5	30	76	70
5	40	70	77
5	50	87	63
10	25	67	75
10	30	74	74
10	30	73	75
10	30	75	78
20	30	77	70
20	30	72	60
20	50	85	57
100	30	72	70
1000	30	72	71

4.1 Confused Student Dataset Experiments

On the confused student dataset provided by Wang et al. [10], the authors tried to train an SVM, autoencoders, LSTM methods, and 1D CNN methods. Here the authors detail their experiments with this dataset and the results.

4.1.1 SVM Baseline

The first experiment run was with the SVM baseline, which achieved approximately 57.67% accuracy on the testing set.

4.1.2 Autoencoder Attempts

The second experiments run were with the autoencoder architectures. The authors intended to successfully train an autoencoder on the data and then feed the autoencoded version of the data to an LSTM or other neural network, similar to the work of Xing et al. [12]. However, attempts to train an autoencoder on the confused student dataset resulted in an ever-ballooning loss with both the first and, later, the second autoencoder architecture, meaning that the autoencoder methods we tried did not learn to encode and decode the data successfully. That being said, there may be autoencoder methods which can learn from the confused student dataset. Still, given the repeated setbacks with autoencoders, the authors decided not to pursue the autoencoder ideas further.

4.1.3 Working with LSTMs

The next set of experiments with the confused student dataset involved LSTM models. First, the authors tested the regular LSTM model with 30 epochs for training and a batch size of 50; this resulted in approximately 61.96% accuracy on the test set. With this baseline in min, the authors then tried the Bidirectional LSTM architecture with several different combinations of batch sizes and epochs, with results as seen in Table 1

4.1.4 Finding Success with 1D CNNs

Finally, the authors tried to see if 1D CNNs could provide better results than the LSTMs did. The first CNN architecture mentioned, the one with 6 CNN layers, was developed and tested on the confused student dataset. Much to the authors' surprise, this architecture perfectly classified both the training and testing sets with 100% accuracy. After checking for any sign of a mistake in the code, such as accidentally passing in the label to be predicted, the authors found, surprisingly, nothing amiss. Upon repeated runs of the method with randomized starting points, the authors found that, generally, the 1D CNN method did produce a very high accuracy, if not perfect accuracy. That being said, a couple times the method resulted in 0 accuracy; this was a relatively rare occurrence over the repeated testing.

Table 2: Best Test Accuracy for Each Model Tested

Method	Confused Student Dataset	DEAP Dataset
SVM	57.67%	-
Autoencoders	Failure to learn	-
LSTM	61.96%	-
Bi-LSTM	78%	-
1D CNN	100%	94%

4.2 DEAP Dataset Experiments

Given the incredible success of the 1D CNN architecture with the confused student dataset, the authors decided to stick with that method for the DEAP dataset. However, the initial CNN architecture used did not produce nearly the same level of results with the DEAP dataset as with the confused student dataset. Thus, the authors developed the second 1D CNN architecture mentioned through trial and error.

The second 1D CNN architecture mentioned resulted in accuracies above 90% on the testing set over several trials, with the highest accuracy being approximately 94% on the testing set. Given this incredible performance and the lackluster results of the other methods on the simpler dataset, the authors concluded that a proper 1D CNN architecture for a given binary emotion classification problem would perform better than the other methods tested in this paper.

5 Analysis and Discussion

The results of the experiments run, summarized in Table 2, suggest that 1D CNNs can be used with EEG data for binary emotion classification with surprisingly accurate results, better than that of one of the state-of-the-art methods, Bidirectional LSTMs. This presumes, of course, that a good local minimum is reached, and not a bad one, as happened in a couple cases with the 1D CNN on the confused student dataset.

However, the authors also learned from their failure with autoencoders, their work with LSTMs, and from the ways this work differs from past work on the chosen datasets. Finally, the authors also consider several threats to the validity of the conclusion that 1D CNN architectures may be better than other methods for binary emotion classification with EEG data.

5.1 Failing with Autoencoders

The failure to produce results with autoencoders was hard for the authors, as one of our ideas was to combine autoencoders with other methods. While this setback was unfortunate, the authors have attempted to learn from this and hypothesize why this might have happened. One possibility is that the learning rate was too large, resulting in the steps made for learning being too large, skipping over local minimums entirely. Another possibility is that perhaps the optimizer (Adam) or the loss function (binary cross-entropy) were perhaps not appropriate for the autoencoder method. The authors suspect that the former explanation is more likely than the latter.

5.2 Learning from LSTM Work

While the LSTM models did not produce the best accuracy, the experiments with different batch sizes and epochs did result in some interesting findings. One interesting note is that a relatively large number of epochs (50, in the case of our experiments) resulted in a great deal of variance between the training accuracy and the testing accuracy. In short, it seems that running too many epochs results in overfitting. Batch size, on the other hand, did not seem to have nearly as monumental an effect, as various batch sizes were tried, ranging from 5 to 1000, without much noticeable different in results. That being said, the best performance seemed to come from the combination of a batch size of 10 and 30 epochs of training, with results that were both low in variance and decent in accuracy on the testing set, ranging from 74% to 78%.

Table 3: Comparing Our Work to Other Works

Authors	Method	Confused Student Dataset	DEAP Dataset
Ni et al. [5]	Bi-LSTM	73.3%	-
Wang et al. [11]	CF-Bi-LSTM	75.0%	-
Tripathi et al. [8]	CNN	-	81.41%
Xing et al. [12]	SAE+LSTM	-	81.10%
Nath et al. [4]	LSTM	-	94.69%
Pias and Harden	1D CNN	100%	94%

5.3 Good Feature Selection and Confounder Filtering

One of the ways this work differs significantly from previous works on these datasets, which, along with our work, are summarized in Table 3, is in the manual selection and exclusion of features for consideration. The fact that such success was found with 1D CNNs may owe partially to the exclusion of noisy, irrelevant, and/or confounding features, such as the columns identifying the subject and the video in the confused student dataset. Given that some methods perform less well in the presence of noisy, irrelevant, and/or confounding features, being able to remove these features from consideration boasts great potential for future work. Thus, the authors suggest that feature selection and exclusion algorithms, such as the aforementioned confounder filtering of Wang et al. [11], are critical to refining the successes of state-of-the-art methods.

5.4 Threats to Validity

The authors of this paper recognize that their experimental methods are not perfect; in this section, we discuss several threats to the validity of the conclusion made about 1D CNNs.

First, the fact that other methods were not tested on the DEAP dataset means that it is theoretically possible that one of those methods might somehow outshine 1D CNNs. While the authors, based on the performance of 1D CNNs on both datasets, concluded that 1D CNNs are the way to go, we do acknowledge this seemingly slight possibility.

Second, the fact that we used 80% of the data for training and only 20% for testing may have made it easier to achieve such spectacular results, given that there was less data which needed to be accurately predicted in the testing set. The authors propose trying other splits, such as 70-30, 60-40, and 50-50, as future work.

Finally, the authors' interpretation of the anomalous 1D CNN results as being due to reaching a bad local minimum may be flawed. If there is some other reason for the fluctuations in performance, then the authors cannot guarantee that a 1D CNN will generally produce consistently good results on binary emotion classification.

6 Conclusion

Binary emotion classification, and emotion classification more generally, is an interesting area for machine learning that has many potential applications. In this paper, the authors tried several machine learning methods, including autoencoders, LSTMs, and CNNs, and found that 1D CNNs work very well for binary emotion classification when good feature selection and exclusion is done. While the authors acknowledge that our experiments aren't perfect, we find that the stunning level of accuracy present with CNNs suggests that 1D CNNs an excellent tool for the job of binary emotion classification with EEG data.

7 Future Work

In addition to testing the methods with other train-test data splits, the authors suggest that 1D CNNs be tried with more complex emotion classification problems, such as with 3 or more classes, and that other methods be tested with the particular set of channels from the DEAP dataset that were selected.

Contributions

Jesse Harden contributed mainly to the planning and analysis of experiments, as well as taking responsibility for the writing of this report. Jesse Harden also designed the Spotlight Video presentation and took part in it.

Tanmoy Sarkar Pias contributed mainly to the coding and running of experiments. He did contribute knowledge that was helpful for the writing, and helped create diagrams and tables. In addition, Tanmoy did take part in the Spotlight Video presentation.

Broader Impact

The authors find that the ethical concerns usually mentioned in this kind of section are not applicable to this work.

Acknowledgments and Disclosure of Funding

The authors would like to thank the TAs, Yiran Xu and Akash Patil, and the professor, Dr. Jia-Bin Huang, for their helpful guidance during this project. This class project was not a recipient of any kind of funding.

References

[1] F. Cavallo, F. Semeraro, L. Fiorini, G. Magyar, P. Sinčák, and P. Dario. Emotion modelling for social robotics applications: a review. *Journal of Bionic Engineering*, 15(2):185–203, 2018.

[2] EMOTIV. Electrode arms - alternative reference - epoc+. `https://emotiv.gitbook.io/epoc-user-manual/troubleshooting/electrode_arms`, 2019. Accessed: 2021-04-27.

[3] S. Koelstra, C. Muhl, M. Soleymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras. Deap: A database for emotion analysis; using physiological signals. *IEEE transactions on affective computing*, 3(1):18–31, 2011.

[4] D. Nath, M. Singh, D. Sethia, D. Kalra, S. Indu, et al. An efficient approach to eeg-based emotion recognition using lstm network. In *2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA)*, pages 88–92. IEEE, 2020.

[5] Z. Ni, A. C. Yuksel, X. Ni, M. I. Mandel, and L. Xie. Confused or not confused? disentangling brain activity from eeg data using bidirectional lstm recurrent neural networks. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 241–246, 2017.

[6] V. Rozgić, S. N. Vitaladevuni, and R. Prasad. Robust eeg emotion classification using segment level decision fusion. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 1286–1290. IEEE, 2013.

[7] J. A. Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

[8] S. Tripathi, S. Acharya, R. D. Sharma, S. Mittal, and S. Bhattacharya. Using deep and convolutional neural networks for accurate emotion classification on deap dataset. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4746–4752, 2017.

[9] G. K. Verma and U. S. Tiwary. Affect representation and recognition in 3d continuous valence–arousal–dominance space. *Multimedia Tools and Applications*, 76(2):2159–2183, 2017.

[10] H. Wang, Y. Li, X. Hu, Y. Yang, Z. Meng, and K.-m. Chang. Using eeg to improve massive open online courses feedback interaction. In *AIED Workshops*, 2013.

[11] H. Wang, Z. Wu, and E. P. Xing. Removing confounding factors associated weights in deep neural networks improves the prediction accuracy for healthcare applications. In *BIOCOMPUTING 2019: Proceedings of the Pacific Symposium*, pages 54–65. World Scientific, 2018.

[12] X. Xing, Z. Li, T. Xu, L. Shu, B. Hu, and X. Xu. Sae+ lstm: A new framework for emotion recognition from multi-channel eeg. *Frontiers in neurorobotics*, 13:37, 2019.

[13] W.-J. Yoon and K.-S. Park. A study of emotion recognition and its applications. In *International Conference on Modeling Decisions for Artificial Intelligence*, pages 455–462. Springer, 2007.

[14] W.-L. Zheng, J.-Y. Zhu, Y. Peng, and B.-L. Lu. Eeg-based emotion classification using deep belief networks. In *2014 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2014.