Investigating the Use of Acoustic Features to Understand Human Social Interaction From Speech

DAAN DIEPERINK, University of Twente, The Netherlands

Computer analysis of human speech can enrich our human-computer interactions. Aside from automatic speech recognition, which is about translating speech into text, there are other speech analysis tasks, that include predicting social or emotional characteristics about a speaker based on certain properties of the sound they produce. This research will investigate the application of various machine learning methods to predict different kinds of characteristics from acoustic features that are computed from speech audio signals.

Additional Key Words and Phrases: speech, audio, acoustic features, acoustic parameters, emotion recognition, machine learning, volume normalization

1 INTRODUCTION

1.1 Motivation

Social interaction certainly is an important aspect of human life, and speech is possibly the most natural form of communication. People may speak, on average, more than 15 thousand words every day [40]. However, speech consists not only of words themselves. The way in which those words are spoken can convey a subtext, attitude or emotion that was not apparent from the selection of words alone. Think of how different it can sound to hear the same words said with confidence as opposed to hesitation, or with sarcasm instead of sincerity. Or when an adult talks to an infant, they tend speak in a particular way, which can positively influence the development of the infant [23]. Humans intuitively change acoustic features in their speech depending on emotional or social context, and listeners naturally pick up these patterns to understand some of this context [3, 36].

This intuitive decoding of acoustic features in speech to understand context is no trivial task, and even our own perception of speech can be subjective [34]. Yet, automatic analysis of human social interactions based on these auditory features could be useful for human-computer interaction, as these features could provide more information than just the spoken words [17]. For example, robots employed in assisted-living environments could alert staff members when users show signs of distress, or chatbots could stop bothering users when they sense that their interactions are not appreciated. Furthermore, it could be used to make computer-generated speech sound more appropriate and natural for a given social or emotional context [17].

1.2 Objective

This research will investigate different methods of predicting emotional or social contexts based on certain acoustic features extracted from recorded speech. We will use acoustic features as defined in the extended Geneva Minimalist Acoustic Parameter Set (eGeMAPS) [21], which was developed to perform well on a variety of speech analysis tasks whilst containing a minimal amount of features. The extent to which the feature set has been applied to multiple datasets in direct comparison, however, has been limited. We will apply multiple machine learning techniques to train classifiers on these features and compare how effective they are at human social interaction understanding tasks. We will expand on existing research by analyzing the performance of five different classification algorithms, including our proposed neural net architecture based on long short-term memory (LSTM) [32], to six different speech analysis tasks. Moreover, we will investigate the effect that applying volume normalization as audio preprocessing technique has on classification performance for classifiers using eGeMAPS features.

Author's address: Daan Dieperink, University of Twente, The Netherlands, d.dieperink@student.utwente.nl.

1.3 Research Question

We will address the objective using the following research question:

RQ: *How does the eGeMAPS acoustic feature set perform on a variety of tasks within the topic of human social interaction understanding?*

To aid in answering the main research question, we propose two subquestions:

SQ1: *Which supervised learning model for classification performs best across the different tasks?*

SQ2: *Does audio volume normalization have an impact on classification performance for the eGeMAPS feature set?*

2 RELATED WORKS

Within the topic of understanding human social interactions from speech, speech emotion recognition (SER) is the most researched subtopic, having been studied for more than 25 years [55]. Other related tasks include the prediction of perceived personality from speech [22], recognizing social relationships from speech [60], as well as predicting characteristics such as social attitude [24].

Speech analysis based on audio signals is traditionally accomplished by extracting several acoustic features from the audio signal, before applying a machine learning classifier to those features. The acoustic features can relate to properties of the audio signals, such as pitch, energy, fundamental frequency or the signal's spectrogram, or they could consist of prosodic characteristics such as speech rate or syllable rate [62]. Such features can potentially capture the information needed for certain speech analysis tasks, while being much smaller than the entire audio signal.

The selection of these features is an important aspect, and different feature selection methods have been applied, without any such method being generally accepted as the best one [1]. These approaches to feature selection include the use of predetermined parameter sets such as eGeMAPS [21] (88 features), ComParE [54] (6373 features), emobase [20] (988 features), or IR-09 [53] (384 features). The eGeMAPS feature set in particular is useful due to its low cardinality, which allows classification algorithms to be more computationally efficient when compared to other feature sets, while offering similar performance [18]. Hence, we adopt the eGeMAPS

set for the purposes of this research. This feature set was primarily developed for analyzing emotionality in speech, but has been applied to other tasks as well, including autism spectrum disorder detection [6] and bipolar disorder detection [57]. Feature sets can additionally be combined with feature selection methods [38] that attempt to eliminate features that contribute little to the classification accuracy.

Many traditional classifiers have been used for speech analysis based on preprocessed audio features, such as Hidden Markov Models (HMM) [5, 52], Gaussian Mixture Models (GMM) [50, 13], Support Vector Machines (SVM) [15, 12], Multi-Layer Perceptrons (MLP) [42] and the *k*-Nearest Neighbors algorithm (KNN) [16, 7] [19]. Other classifiers include Bayesian Networks (BN) [47] and Random Forests (RF) [31, 43]. Out of these classifiers, the HMM has been the most often used for SER [19].

Instead of extracting features over an entire utterance at a time, some methods preserve the time dimension of the original audio data. This can be achieved, for example, by using audio spectrograms (or transformations thereof such as MFCC) [27] as inputs to the machine learning models, which typically apply convolutions to the input features. Such methods are generally more computationally intensive than the traditionally used classifiers, but can be capable of achieving better classification performance due to their ability to capture temporal dynamics within speech [45]. For instance, the deep convolutional TIM-Net architecture [63], using MFCC as input, achieved recent state-of-the-art results for speech emotion recognition on the RAVDESS [37] and SAVEE [28] datasets.

Alternatively, traditional features, such as from the eGeMAPS feature set, can be extracted from temporal segments (sometimes called frames or windows) of the audio signal, which allows recurrent neural networks such as LSTM [32] or GRU [14] to be used [2]. This approach could potentially allow classifiers to leverage the temporal information in the input features, while possibly benefiting from the ability of the features to efficiently encode properties that are relevant to the speech analysis task. However, such approaches have not been studied extensively.

Some previous research using the eGeMAPS parameter set has used volume normalization before feature extraction to eliminate difference in recording setup between utterances [26, 25]. However, eGeMAPS includes multiple features related to loudness [21], and a speaker's volume can be related to their emotional state. Hence, we will compare classification performance between setups with and without volume normalization.

3 METHODOLOGY

3.1 Audio Normalization

Before extracting eGeMAPS features from audio samples in our datasets, we normalize the sampling rates of all audio samples to 16kHz. Additionally, we normalize the volume of each audio sample such that each sample has a maximum amplitude of 1. Both resampling and amplitude normalization techniques are performed using the *librosa* Python library [39]. Later, we will leave out the volume normalization step on some experiments to measure the impact of this normalization.

3.2 Feature Extraction

Using the openSMILE library [20], we extract 88 acoustic features defined in the eGeMAPS feature set [21] from the audio signals in the datasets, after applying the audio normalization methods mentioned in section 3.1.

3.2.1 Regular feature extraction. The most straightforward method of feature extraction is to extract 88 features from every audio sample as a whole. These 88-dimensional feature vectors can be used directly as input to the regular classifiers (see section 3.3.1).

3.2.2 Segmented feature extraction. In addition to extracting features from utterances as a whole, we can also split each audio segment into multiple windows and extract eGeMAPS features from each window separately. Using this method of extracting features, we obtain 88-dimensional time series data from every audio sample, which allows us to use a recurrent neural network for classification and compare its performance to the classifiers that use the regular features for each sample. Each utterance is segmented by splitting it into windows of a fixed length (window size), where the starts of each segment are separated by another fixed length (hop size). Only the last window of each utterance is smaller than the window size in length, to ensure that the end of the last window coincides with the end of the sequence. Using a hop size smaller than the window size results in overlapping segments. We use the features extracted from each series of windows as input to our proposed recurrent model (see section 3.3.2). Since the number of windows generated by the segmentation method may differ between utterances of varying lengths, we ensure that the recurrent model is able to use sequences of varying lengths as inputs.

3.3 Model Optimization

We fit two types of classifiers to the features as obtained in the feature extraction stage. Four regular classifiers are fit to the 88 features extracted by the regular feature extractor, and the recurrent model is trained on the time series features as extracted from the segmented audio. Hyperparameters are tuned using 25 iterations of Bayesian optimization, and performance metrics are evaluated with 5-fold cross-validation.

3.3.1 Regular classifiers. On the 88 features as extracted by the regular feature extraction method (section 3.2.1), we fit four regular classifiers: Random Forest (RF), C-Support Vector Machine (SVM), *k*-Nearest Neighbors (KNN) and Multi-Layer Perceptron (MLP). Implementations from the *scikit-learn* library [48] version 1.1.3 are used. The support vector machine uses the nonlinear radial basis function (rbf) as kernel, as is the default in its *scikit-learn* implementation. Other classifiers also use default *scikit-learn* parameters except for certain parameters that are tuned with Bayesian optimization using the *scikit-optimize* library [29]. An overview of the search space for these tuned parameters is provided in table 1. The multi-layer perceptron classifier uses only a single hidden layer between its input and output layers. The input features for each model are scaled using the *StandardScaler* from *scikit-learn*, except for the random forest classifier since feature scaling does not affect this algorithm.

Table 1. Hyperparameter search space of regular classifiers

Model	Parameter	Search Space	Search Scale
SVM	C	$(10^{-3}, 10^{4})$	logarithmic
	gamma	$(10^{-4}, 1)$	logarithmic
KNN	n_neighbors	$(1, 30)$	linear
	p	$[1, 2]$	categorical
RF	n_estimators	$(1, 800)$	linear
	criterion	['gini', 'entropy', 'log_loss']	categorical
MLP	hidden_layer_sizes	$[10, 30, 50]$	categorical
	activation	['tanh', 'relu']	categorical
	optimizer	['sgd', 'adam']	categorical
	max_iter	1024	constant

3.3.2 Recurrent model. On the time series data as obtained from the segmented feature extraction method (see section 3.2.2), we train a recurrent neural network implemented using the Pytorch framework [46]. Instead of the *StandardScaler* used by the regular classifiers, batch normalization [33] is applied to the model's inputs, such that every feature is normalized across all timesteps in a batch. These normalized features from each timestep are separately fed into a fully connected 'embedding' layer with ReLU activations. Thus, for input features from one timestep $x^{(t)}$, we compute the embeddings as $h^{(t)} = \max(0,\ x^{(t)} \cdot W + b)$, with W and b being the weight and bias matrices of the layer, respectively. The rest of the network architecture consists of three stacked bidirectional LSTM [32, 56] layers, followed by a fully-connected layer with softmax activations to predict class probabilities. The number of output classes differs between the datasets. During model training, we employ regularization techniques by adding dropout [30] after the embedding layer and after each LSTM layer, and we apply label smoothing [58]. During model evaluation, we use the argmax function on the output class probabilities to select the class with highest probability as output. The model's weights are updated during training using the Adam optimizer [35], with categorical cross-entropy as the loss function. Default parameters are used for the optimizer except for the learning rate, which is tuned. Like the regular classifiers, hyperparameters are tuned using 25 iterations of Bayesian optimization, although the *Ax* [4] library is used instead of *scikit-optimize* [29]. An overview of the search space in which parameters are tuned is provided in table 2.

Table 2. Hyperparameter search space for recurrent model

Parameter	Search Space	Search Scale
Learning rate	$(10^{-5}, 10^{-2})$	logarithmic
LSTM hidden size	$(32, 128)$	linear
Dropout rate	$(0.0, 0.5)$	linear
Label smoothing	$(0.0, 0.25)$	linear
Embedding size	$(64, 128)$	linear
LSTM layers	3	constant
Batch size	4096	constant
Training epochs	512	constant

4 EXPERIMENTAL SETUP

4.1 Datasets

Five datasets are used for the experiments. Each dataset contains speech utterances labeled with classes describing the speaker's emotion, sentiment or attitude. Some of the datasets may contain additional modalities such as text transcripts or video recordings to accompany the audio, but these are not used for this research. Additionally, some datasets contained a predefined division of data into *train*, *validation* and *test* splits. However, the experiments only use the *train* splits of the datasets where applicable. For every dataset, we define a set of groups such that every utterance belongs to one group. This division of the dataset is used for generating cross-validation splits in the experiments.

SAVEE. The Surrey Audio-Visual Expressed Emotion database (SAVEE) [28] consists of 480 utterances recorded from 4 British male actors. There are 30 unique English sentences spoken in one of seven acted emotions: anger, disgust, fear, happiness, neutral, sadness and surprise. The utterances are grouped by the spoken sentence for the experiments, yielding 30 groups.

CREMA-D. The Crowd-sourced Emotional Multimodal Actors Dataset (CREMA-D) [11] contains 7442 utterances of 91 different actors. Each recording is one of twelve English sentences acted in one of six emotions: happy, sad, anger, fear, disgust, and neutral. The labels also specify a levels of emotional intensity (low, medium, high, or unspecified) but this information is not used in this research, since it would require a different kind of classification task than the other datasets. Each utterance is assigned one of 1089 groups based on the combination of actor and sentence.

Emo-DB. The Emo-DB Database of German Emotional Speech [10] holds 535 recordings by ten actors, all in German. The recordings are labeled with the following emotions: anger, boredom, disgust, fear, happiness, sadness and neutral. The utterances are grouped by the unique combination of speaker and sentence, of which there are 100.

MELD. The Multimodal EmotionLines Dataset (MELD) [49] consists of footage from the *Friends* TV sitcom. The train set contains a total of 9989 utterances from 1039 different

dialogues. Every utterance contains speech from one actor, but may contain additional sound such as scene transition music or laughter from the audience. Sound waveforms were extracted from video files using the *ffmpeg* library [59].
The dataset provides two labels for each utterance. Each recording is labeled by the speaker's emotion out of anger, disgust, sadness, joy, neutral, surprise and fear. Every recording also has the speaker's sentiment annotated, as either positive, negative or neutral. From these two sets of labels we can create two different classification tasks: speech emotion recognition as well as speech sentiment recognition, which we will refer to as *MELD_emotion* and *MELD_sentiment*, respectively. The recordings are grouped by the dialogue in which they appear, so there are 1039 different groups.

Att-HACK. Att-HACK: An Expressive Speech Database with Social Attitudes [41] is a dataset of French speech labeled with social attitudes as opposed to primary emotions. The dataset contains 36634 utterances by 25 speakers, with the four social attitudes friendly, seductive, dominant, and distant. Each utterance is grouped by its sentence, of which there are 100.

4.2 Data Visualizations

4.2.1 Principal component analysis. From the features extracted using the regular feature extraction method (see section 3.2.1), we applied principal component analysis (PCA) [8] as dimensionality reduction technique to allow for data visualization. PCA was performed with three principal components that were plotted as 3-dimensional scatter plot, using as input the extracted features normalized using the *scikit-learn* [48] *StandardScaler*, This was repeated for each dataset, both with and without volume normalization (section 3.1) before feature extraction. The resulting plots are available in figures 1 and 2 in the appendix.

For the smaller datasets SAVEE and Emo-DB in figure 1, we can observe from the PCA plots that the principal component values might be somewhat correlated with some of the labels, suggesting that the eGeMAPS features can be used to recognize labels to some extent. However, the labels do not form well separated clusters, especially for the SAVEE dataset, which suggests that some classifiers may not be able to accurately predict labels on every example. Additionally, we may observe for all datasets except Emo-DB that the feature space for volume-normalized audio samples differs from the feature space where no volume normalization was applied. For Emo-DB, there appears to be little difference between both PCA plots, which could be caused by volume normalization or a similar technique being potentially already applied to the dataset by the dataset's authors. Unfortunately, the PCA plots for the larger datasets may seem as though they consist mostly of a single label. This is not the case, but results from the way the plots were made. Because the labels were processed one by one, data points from the last label added to the plot obscure the other labels. For this reason, we include an additional silhouette analysis, which does not suffer from this issue.

4.2.2 Silhouette analysis. In addition to principal component analysis, we applied silhouette analysis [51] on the same extracted features normalized using the *StandardScaler*. The silhouette coefficient of every sample in the dataset represents how similar it is to samples of the same label and how dissimilar it is to samples of other labels. This score ranges from a value of -1 (the sample is very similar to samples from other labels) to 1 (the sample is very similar to other samples of the same label). A value of 0 means that a sample lies on the decision boundary between its own label's cluster and another cluster. The plotted silhouette scores can be found in figures 3 and 4 of the appendix.

From the plots, we can observe how well features for labels in each dataset lie in nonoverlapping clusters. The plots for the Emo-DB dataset show the highest mean silhouette coefficient out of all plots, which could mean that the eGeMAPS features work relatively well to distinguish between labels. In general, we might expect labels with higher average silhouette values to be more easily recognized by the classifiers.

4.3 Validation Metrics

Classifier performance metrics are evaluated for each classifier using the best hyperparameters as found through Bayesian search. On a 5-fold cross-validation split, we measure the balanced accuracy [9], (unbalanced) accuracy and F1-macro [44] scores across each fold.

4.4 Implementation Details

Each classifier training run consists of training 5 times using 5-fold cross-validation. The folds are obtained by shuffling the training data and splitting it into five equally sized folds. The folds are stratified, so they contain the same number of examples per output class (class ratios are preserved as much as possible). Additionally, the splits preserve the groups that were defined for each dataset (see section 4.1), such that no group spans multiple folds. The purpose of this grouping is to ensure that the models are not able to 'remember' as easily how certain sentences or speakers sound for certain emotions. This forces the classifiers to generalize to unseen sentences or combinations of speaker and sentence, as determined by the grouping of each dataset. For each cross-validation split, the models are fitted to data from four folds and validated against the one remaining fold. We find the best hyperparameters for each model with 25 iterations of Bayesian optimization, taking a new random cross-validation split every iteration. The Bayesian search attempts to maximize the balanced accuracy for each classifier.

5 RESULTS

5.1 Regular Classifiers

The best hyperparameters for the four regular classifiers were found using Bayesian optimization. An overview of the best hyperparameters found is available in table 4 in the appendix, and cross-validation metrics for these hyperparameters are available in table 3 in the appendix. The experiments were repeated on each dataset, both with and without applying volume normalization to the audio before extracting the eGeMAPS features (see section 3.1). Performance metrics for an additional 'dummy' classifier were added for comparison of classifier performance to random guessing, with the dummy

classifier using the same cross-validation method as the other models. The dummy classifies used stratified random sampling of the target classes, such that the chance of selecting any class is the same as the frequency that the class appears in the data. Since the stratification is based on on the class ratios of the train data obtained from the cross-validation split, and not the validation fold, there may be some variation in the dummy's classification performance, although the cross-validation splitting strategy attempts to keep the class ratios consistent across all folds.

The performance metrics in table 3 show the widely varying performance of the classifiers between the different datasets. Out of the different classifiers, the SVM classifier seemed to perform best in general, although the MLP achieved the single highest balanced accuracy, scoring 77.4% on the Emo-DB dataset with volume normalization. For every dataset, each classifier achieved higher performance than the dummy classifier. However, for the datasets MELD_emotion and MELD_sentiment, no classifier was able to improve over the dummy classifier's balanced accuracy by more than 6 percent. Every regular classifier achieved higher balanced accuracy on the dataset where no volume normalization was applied, compared to the experiments with volume normalization. The exception to this is the Emo-DB dataset, where two regular classifiers happened to perform better with volume normalization, and two regular classifiers happened to perform better without volume normalization, with regard to the balanced accuracy scores.

In figure 5, we additionally provide confusion matrices for the SVM's predictions on each dataset, without volume normalization being applied. The figures show the model's predicted labels compared to the ground truth labels. These matrices show us the frequency of some misclassifications. For instance, we can see that, for the Emo-DB dataset, speech labeled with 'happiness' was most commonly misclassified as 'anger' by the SVM.

5.2 Recurrent model

The recurrent model was trained and tuned using the segmented features extracted from the SAVEE, Emo-DB and CREMA-D datasets (see section 4.1). Audio without volume normalization was used, and the experiments were repeated for segmentation windows with sizes of both 1000 ms and 500 ms, taking half the window size as hop size (see section 3.2.2). An overview of the hyperparameters found in the experiments and the achieved cross-validation scores is available in table 5 in the appendix. From these results we can see that the window size of 500 ms performed better than 1000 ms on the SAVEE and CREMA-D datasets, but not on Emo-DB. The highest achieved balanced accuracy by the recurrent model was 74.5% on Emo-DB, using 1000 ms window size. On the CREMA-D dataset, the recurrent model using 500 ms window size outperformed all regular classifiers by at least 1.9% balanced accuracy.

6 DISCUSSION

6.1 Research Questions

SQ1: *Which supervised learning model for classification performs best across the different tasks?*

First, we will comment on the difficulty of the different speech analysis tasks. Then, we will discuss the performance of the classifiers, and compare the best performing regular classifier to the recurrent classifier.

By comparing the results across all datasets, it becomes apparent that some datasets were much more difficult to classify using eGeMAPS features than others. Whereas all classifiers achieved over 65% balanced accuracy on the Emo-DB datasets (much higher than the dummy classifier, which scored about 15%), none of the models were able to improve by much over the dummy for the MELD datasets, on both its tasks. The eGeMAPS parameter set was devised to work well for speech emotion analysis tasks on many datasets, but was tuned specifically on a handful of German, English and French speech datasets. One of these datasets was Emo-DB itself, so it is no surprise that the classifiers perform relatively well on this dataset.

If we compare the general classification performance achieved by each classifier, it seems that the SVM generally achieved the best results across the six different tasks. The SVM has already been widely used for speech emotion recognition research [1], and these results seem to confirm its effectiveness.

The recurrent neural net architecture as proposed in section 3.3.2 was unfortunately not trained on all datasets, but only on the smallest three. This was due to an unfortunate bug in our implementation which caused a memory leak for larger datasets. Combined with time constraints put upon this research, we were unable to run the Bayesian optimization on those datasets. Since deep neural models tend to require more data to achieve high performance, the recurrent approach may have worked better on larger datasets. By extracting the features as time-series data from multiple segments, recurrent classifiers can take into account the changes in acoustic features throughout an entire utterance to potentially make more accurate predictions about emotion. Although this approach requires much more computation, both in the feature extraction stage as well as during model training and inference, this is somewhat counteracted by the small dimensionality of the eGeMAPS feature set.

Out of the three different datasets and two different segmentation parameters used for training the recurrent models, the recurrent strategy performed worse than the support vector machine on all but one combination of dataset and segmentation parameters. For the CREMA-D dataset, using a segmentation window size of 500 ms and hop size of 250 ms, the recurrent network outperformed all four regular classifiers on all three metrics by more than 15 percent. However, we cannot determine with certainty whether the recurrent model with window size 500 ms and hop size 250 ms is truly better than the SVM on the CREMA-D dataset without volume normalization, since the difference in balanced accuracy could have been caused by the different random cross-validation splits used for the experiments (the recurrent model could have theoretically performed no better than the SVM, but gotten 'lucky' with the cross-validation splits it used for performance evaluation). Nevertheless, the results do suggest that such a segmented strategy could be used to improve classification performance with eGeMAPS features in some situations. Especially given more epochs for training, the LSTM-based recurrent model might obtain better results.

SQ2: *Does audio volume normalization have an impact on classification performance for the eGeMAPS feature set?*

Volume normalization changes some of the features as extracted from the samples. This difference is apparent from the PCA plots and silhouette scores of features using volume normalization, compared to features extracted without volume normalization applied (see figures 2, 3, 1, 4). With the exception of the MLP classifier on the Emo-DB dataset, all classifiers obtained higher balanced accuracy without volume normalization on every dataset. Using a one-tailed pairwise Wilcoxon signed-rank test [61], we can determine whether volume normalization significantly diminished balanced accuracy for the four regular classifiers (excluding the dummy classifier) on the six evaluated datasets. We use the balanced accuracy scores (disregarding the standard deviations measured across folds) from KNN, SVM, RF and MLP on all six classification tasks, pairing the scores achieved using features with volume normalization applied to the non-normalized audio features from identical combinations of classifier and dataset. The test shows that for these datasets, the regular classifiers perform significantly better when using the features without volume normalization applied, in terms of balanced accuracy ($N = 24$, $Z = -3.5286$, $p = 0.00021 < 0.01$).

We should note that, although volume normalization hurt performance on these datasets, volume normalization still may improve performance in other settings. The reason for applying volume normalization is to reduce loudness variation between different recording settings, and this may still be beneficial for other datasets.

RQ: *How does the eGeMAPS acoustic feature set perform on a variety of tasks within the topic of human social interaction understanding?*

In general, the eGeMAPS parameters seem well suited for speech analysis tasks, as they achieved decent results across different datasets and classifier algorithms. Moreover, the low number of features allowed for training models with lower memory usage when compared to models using larger parameter sets. Especially deep neural networks, such as our recurrent architecture, may benefit from the smaller input sizes. However, the feature set might not be useful for every speech analysis task. On the MELD dataset, no classifier was able to improve much over random guessing. Utterances in the MELD dataset did not just contain speech, but contained sounds from multiple different sources in some cases, which may have reduced the effectiveness of eGeMAPS features.

6.2 Limitations

For this research, samples in the datasets were grouped (see section 4.1), and these groupings were used in generating the cross-validation splits to make the classifiers less prone to memorization of specific combinations of e.g. speaker and sentence. Intuitively, this makes the classification tasks more difficult, and we would expect to obtain lower performance on the tasks as a result of this grouping. However, it is unclear to what extent this impacts classification scores. Without the effect of the grouping to classification performance being known, it is difficult to directly compare classification metrics to other research where the same grouping strategy was not applied.

Moreover, the experiment setup may have led to high variation between measured performance metrics. Each classifier used 5-fold cross-validation to compute classification performance, but the division of the data into the 5 folds was different every time, which may have resulted in some performance variance. The standard deviations for the performance metrics that are included in the results (tables 3 and 5) were computed only from the 5 trials on a single 5-fold cross-validation split and thus do not capture this variation across different splits. This is also the reason why we cannot determine with certainty from our results whether the recurrent model using a window size of 500 ms and hop size of 250 ms truly outperformed the SVM on the CREMA-D dataset without volume normalization. Furthermore, it may have introduced noise that could have negatively impacted the ability of the Bayesian search strategy to find the optimal hyperparameters. These issues could be alleviated by using the same cross-validation splits for every experiment, albeit that the selection of splits may introduce some potentially undesired bias.

6.3 Future Work

Future research could include the evaluation of the recurrent models on larger datasets, since these models tend to benefit from more data. Alternatively, data augmentation techniques could be used to increase the amount of data available. Only two segmentation parameter settings were tested in this research, and more could be tried to determine an optimal window and hop size for the recurrent model. However, such an approach would require re-extraction of the acoustic features on every change of the segmentation settings, which might render a Bayesian search of the optimal window and hop sizes unfeasible. Although volume normalization diminished classification performance in our experiments, the technique could potentially still be useful for datasets with large variations in amplitude. As an alternative to volume normalization, future research could analyze the effectiveness of dynamic range compression, which might reduce difference in amplitude without removing the variation altogether.

7 CONCLUSION

In this research, the performance of various classification methods for speech emotion recognition, sentiment recognition and attitude recognition were evaluated for different datasets, using the eGeMAPS feature set. We proposed a recurrent architecture for processing eGeMAPS features as time series data by segmenting the input audio, and found a case in which this approach may have improved classification performance over traditional approaches. Moreover, we found in our experiments that applying volume normalization to audio signals before extracting eGeMAPS features significantly diminishes classification performance.

REFERENCES

[1] Mehmet Berkehan Akçay and Kaya Oğuz. "Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers". In: *Speech Communication* 116 (2020), pp. 56–76.

[2] Bagus Tris Atmaja. "RNN-based dimensional speech emotion recognition". In: (2020).

[3] Jo-Anne Bachorowski. "Vocal expression and perception of emotion". In: *Current directions in psychological science* 8.2 (1999), pp. 53–57.

[4] Eytan Bakshy et al. "AE: A domain-agnostic platform for adaptive experimentation". In: *Conference on Neural Information Processing Systems*. 2018, pp. 1–8.

[5] Leonard E Baum and Ted Petrie. "Statistical inference for probabilistic functions of finite state Markov chains". In: *The annals of mathematical statistics* 37.6 (1966), pp. 1554–1563.

[6] Federica Beccaria, Gloria Gagliardi, and Dimitrios Kokkinakis. "Extraction and Classification of Acoustic Features from Italian Speaking Children with Autism Spectrum Disorders." In: *Proceedings of the RaPID Workshop-Resources and ProcessIng of linguistic, para-linguistic and extra-linguistic Data from people with various forms of cognitive/psychiatric/developmental impairments-within the 13th Language Resources and Evaluation Conference*. 2022, pp. 22–30.

[7] Anuja Bombatkar et al. "Emotion recognition using Speech Processing Using k-nearest neighbor algorithm". In: *Int. J. Eng. Res. Appl* 4 (2014), pp. 2248–9622.

[8] Rasmus Bro and Age K Smilde. "Principal component analysis". In: *Analytical methods* 6.9 (2014), pp. 2812–2831.

[9] Kay Henning Brodersen et al. "The balanced accuracy and its posterior distribution". In: *2010 20th international conference on pattern recognition*. IEEE. 2010, pp. 3121–3124.

[10] Felix Burkhardt et al. "A database of German emotional speech." In: *Interspeech*. Vol. 5. 2005, pp. 1517–1520.

[11] Houwei Cao et al. "Crema-d: Crowd-sourced emotional multimodal actors dataset". In: *IEEE transactions on affective computing* 5.4 (2014), pp. 377–390.

[12] Yashpalsing Chavhan, ML Dhore, and Pallavi Yesaware. "Speech emotion recognition using support vector machine". In: *International Journal of Computer Applications* 1.20 (2010), pp. 6–9.

[13] Xianglin Cheng and Qiong Duan. "Speech emotion recognition using gaussian mixture model". In: *2012 International Conference on Computer Application and System Modeling*. Atlantis Press. 2012, pp. 1222–1225.

[14] Kyunghyun Cho et al. "On the properties of neural machine translation: Encoder-decoder approaches". In: *arXiv preprint arXiv:1409.1259* (2014).

[15] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine learning* 20 (1995), pp. 273–297.

[16] Thomas Cover and Peter Hart. "Nearest neighbor pattern classification". In: *IEEE transactions on information theory* 13.1 (1967), pp. 21–27.

[17] Roddy Cowie et al. "Emotion recognition in human-computer interaction". In: *IEEE Signal processing magazine* 18.1 (2001), pp. 32–80.

[18] Cem Doğdu et al. "A Comparison of Machine Learning Algorithms and Feature Sets for Automatic Vocal Emotion Recognition in Speech". In: *Sensors* 22.19 (2022), p. 7561.

[19] Moataz El Ayadi, Mohamed S Kamel, and Fakhri Karray. "Survey on speech emotion recognition: Features, classification schemes, and databases". In: *Pattern recognition* 44.3 (2011), pp. 572–587.

[20] Florian Eyben, Martin Wöllmer, and Björn Schuller. "openSMILE – The Munich Versatile and Fast Open-Source Audio Feature Extractor". In: Jan. 2010, pp. 1459–1462. DOI: 10.1145/1873951.1874246.

[21] Florian Eyben et al. "The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing". In: *IEEE Transactions on Affective Computing* 7.2 (2016), pp. 190–202. DOI: 10.1109/TAFFC.2015.2457417.

[22] Laura Fernández Gallardo and Benjamin Weiss. "Towards Speaker Characterization: Identifying and Predicting Dimensions of Person Attribution." In: *INTERSPEECH*. 2017, pp. 904–908.

[23] Roberta Michnick Golinkoff et al. "(Baby) talk to me: the social context of infant-directed speech and its effects on early language acquisition". In: *Current Directions in Psychological Science* 24.5 (2015), pp. 339–344.

[24] Fasih Haider and Saturnino Luz. "Attitude recognition using multi-resolution cochleagram features". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 3737–3741.

[25] Fasih Haider et al. "Affective speech for Alzheimer's dementia recognition". In: *LREC: Resources and ProcessIng of linguistic, para-linguistic and extra-linguistic Data from people with various forms of cognitive/psychiatric/developmental impairments (RaPID)* (2020), pp. 67–73.

[26] Fasih Haider et al. "Emotion recognition in low-resource settings: An evaluation of automatic feature selection methods". In: *Computer Speech & Language* 65 (2021), p. 101119.

[27] Noushin Hajarolasvadi and Hasan Demirel. "3D CNN-based speech emotion recognition using k-means clustering and spectrograms". In: *Entropy* 21.5 (2019), p. 479.

[28] Sanaul Haq, Philip JB Jackson, and J Edge. "Speaker-dependent audio-visual emotion recognition." In: *AVSP*. Vol. 2009. 2009, pp. 53–58.

[29] Tim Head et al. *scikit-optimize/scikit-optimize*. Version v0.9.0. Oct. 2021. DOI: 10.5281/zenodo.5565057. URL: https://doi.org/10.5281/zenodo.5565057.

[30] Geoffrey E. Hinton et al. "Improving neural networks by preventing co-adaptation of feature detectors". In: *CoRR* abs/1207.0580 (2012). arXiv: 1207.0580. URL: http://arxiv.org/abs/1207.0580.

[31] Tin Kam Ho. "Random decision forests". In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE. 1995, pp. 278–282.

[32] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[33] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *CoRR* abs/1502.03167 (2015). arXiv: 1502.03167. URL: http://arxiv.org/abs/1502.03167.

[34] Sudarsana Reddy Kadiri and Paavo Alku. "Subjective Evaluation of Basic Emotions from Audio–Visual Data". In: *Sensors* 22.13 (2022), p. 4931.

[35] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[36] Nadine Lavan et al. "Flexible voices: Identity perception from variable vocal signals". In: *Psychonomic bulletin & review* 26 (2019), pp. 90–102.

[37] Steven R Livingstone and Frank A Russo. "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English". In: *PloS one* 13.5 (2018), e0196391.

[38] Iker Luengo et al. "Automatic emotion recognition using prosodic parameters". In: *Ninth European conference on speech communication and technology*. Citeseer. 2005.

[39] Brian McFee et al. "librosa: Audio and music signal analysis in python". In: *Proceedings of the 14th python in science conference*. Vol. 8. 2015, pp. 18–25.

[40] Matthias R Mehl et al. "Are women really more talkative than men?" In: *Science* 317.5834 (2007), pp. 82–82.

[41] Clément Le Moine and Nicolas Obin. "Att-HACK: An Expressive Speech Database with Social Attitudes". In: *arXiv preprint arXiv:2004.04410* (2020).

[42] Panuwit Nantasri et al. "A light-weight artificial neural network for speech emotion recognition using average values of MFCCs and their derivatives". In: *2020 17th International conference on electrical engineering/electronics, computer, telecommunications and information technology (ECTI-CON)*. IEEE. 2020, pp. 41–44.

[43] Fatemeh Noroozi et al. "Vocal-based emotion recognition using random forests and decision tree". In: *International Journal of Speech Technology* 20.2 (2017), pp. 239–246.

[44] Juri Opitz and Sebastian Burst. "Macro f1 and macro f1". In: *arXiv preprint arXiv:1911.03347* (2019).

[45] Anda Ouyang et al. "Speech based emotion prediction: Can a linear model work?" In: *INTERSPEECH*. 2019, pp. 2813–2817.

[46] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[47] Judea Pearl. "Bayesian netwcrks: A model cf self-activated memory for evidential reasoning". In: *Proceedings of the 7th conference of the Cognitive Science Society, University of California, Irvine, CA, USA*. 1985, pp. 15–17.

[48] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[49] Soujanya Poria et al. "Meld: A multimodal multi-party dataset for emotion recognition in conversations". In: *arXiv preprint arXiv:1810.02508* (2018).

[50] Douglas A Reynolds et al. "Gaussian mixture models." In: *Encyclopedia of biometrics* 741.659-663 (2009).

[51] Peter J Rousseeuw. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis". In: *Journal of computational and applied mathematics* 20 (1987), pp. 53–65.

[52] Björn Schuller, Gerhard Rigoll, and Manfred Lang. "Hidden Markov model-based speech emotion recognition". In: *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)*. Vol. 2. Ieee. 2003, pp. II–1.

[53] Björn Schuller, Stefan Steidl, and Anton Batliner. "The interspeech 2009 emotion challenge". In: (2009).

[54] Björn Schuller 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.

[55] Björn W Schuller. "Speech emotion recognition: Two decades in a nutshell, benchmarks, and ongoing trends". In: *Communications of the ACM* 61.5 (2018), pp. 90–99.

[56] Mike Schuster and Kuldip K Paliwal. "Bidirectional recurrent neural networks". In: *IEEE transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.
[57] Zafi Sherhan Syed, Kirill Sidorov, and David Marshall. "Automated screening for bipolar disorder from audio/visual modalities". In: *Proceedings of the 2018 on Audio/Visual Emotion Challenge and Workshop*. 2018, pp. 39–45.
[58] Christian Szegedy et al. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.
[59] Suramya Tomar. "Converting Video Formats with FFmpeg". In: *Linux J.* 2006.146 (June 2006), p. 10. ISSN: 1075-3583.
[60] F.R. Vossebeld. *Towards understanding social interactions through audio signals*. B.S. thesis. July 2022. URL: http://essay.utwente.nl/91976/.
[61] Frank Wilcoxon. "Individual comparisons by ranking methods". In: *Breakthroughs in statistics*. Springer, 1992, pp. 196–202.
[62] Chung-Hsien Wu, Jui-Feng Yeh, and Ze-Jing Chuang. "Emotion perception and recognition from speech". In: *Affective Information Processing* (2009), pp. 93–110.
[63] Jiaxin Ye et al. "Temporal Modeling Matters: A Novel Temporal Emotional Modeling Approach for Speech Emotion Recognition". In: *arXiv preprint arXiv:2211.08233* (2022).

Fig. 1. Principal Component Analysis plots for datasets SAVEE, Emo-DB, CREMA-D

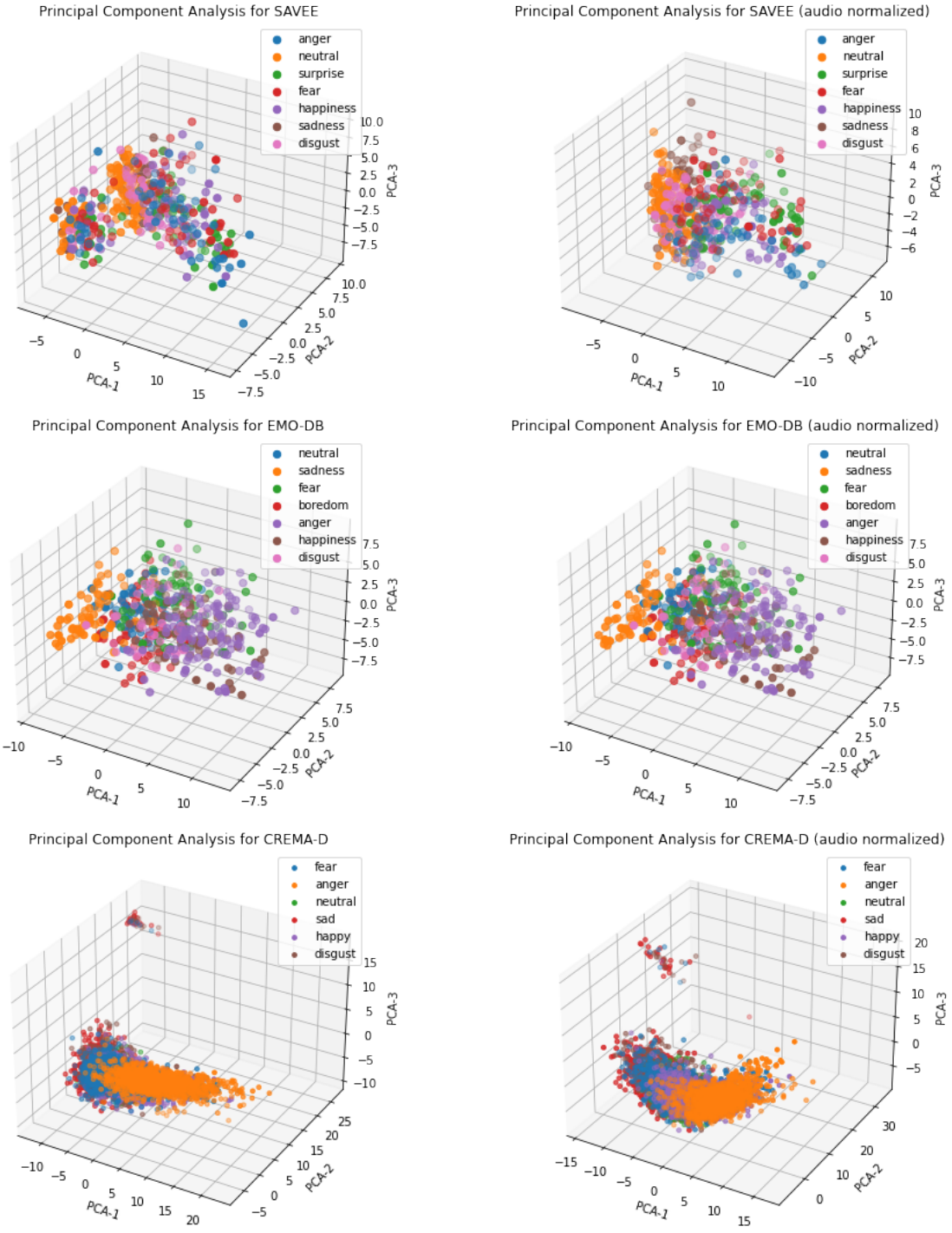

Fig. 2. Principal Component Analysis plots for datasets MELD, Att-HACK

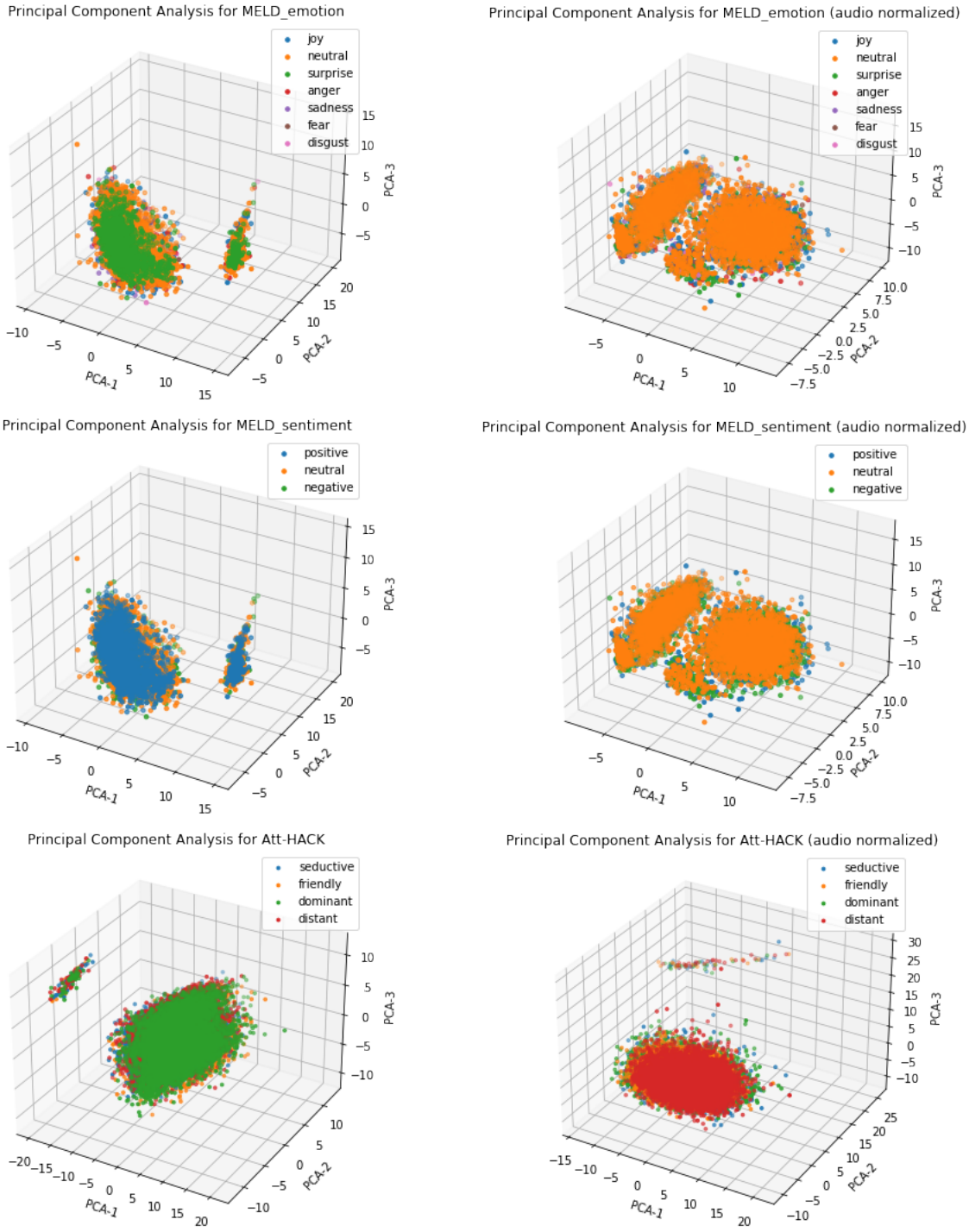

Fig. 3. Silhouette plots for datasets SAVEE, Emo-DB, CREMA-D

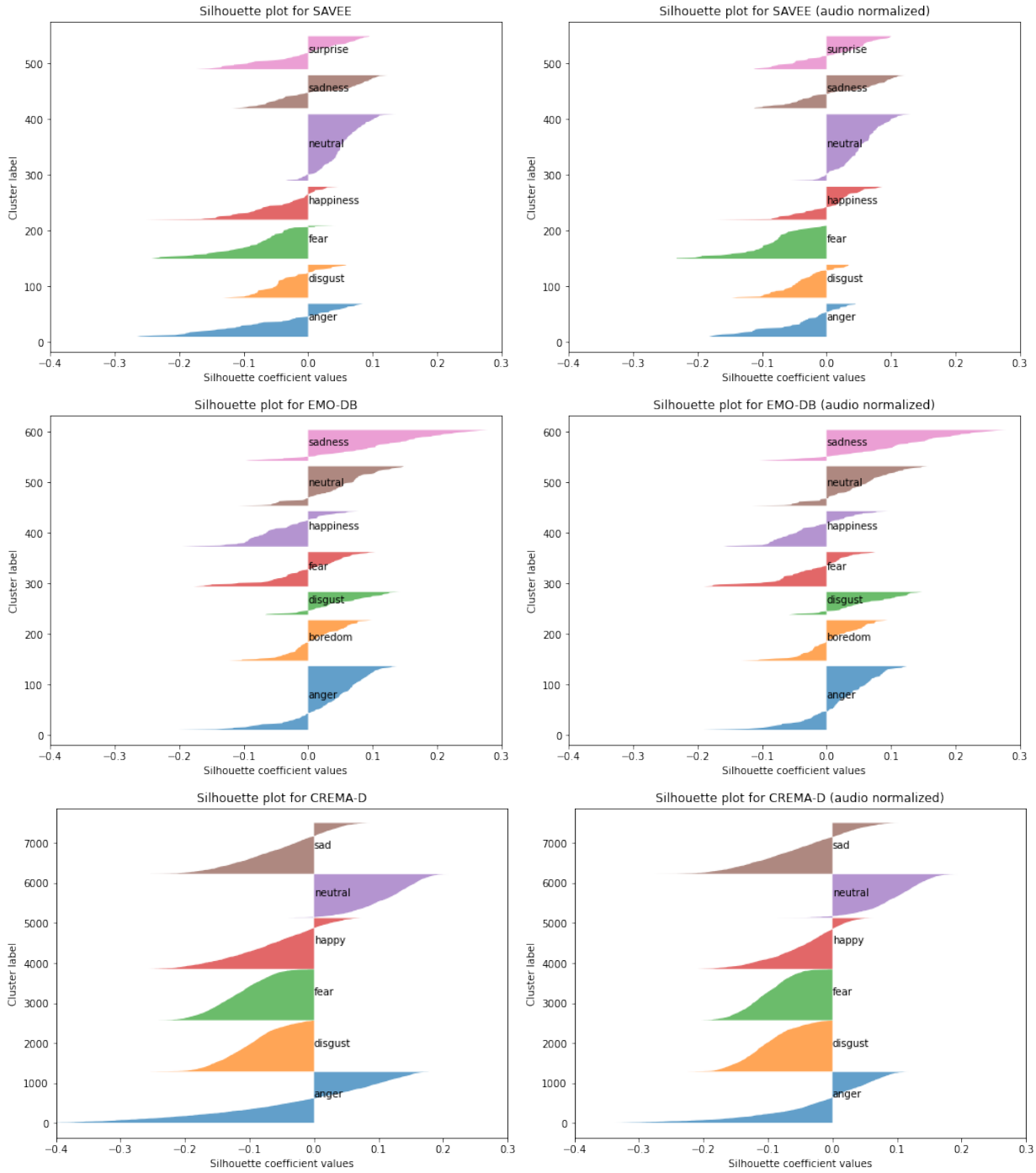

Fig. 4. Silhouette plots for datasets MELD, Att-HACK

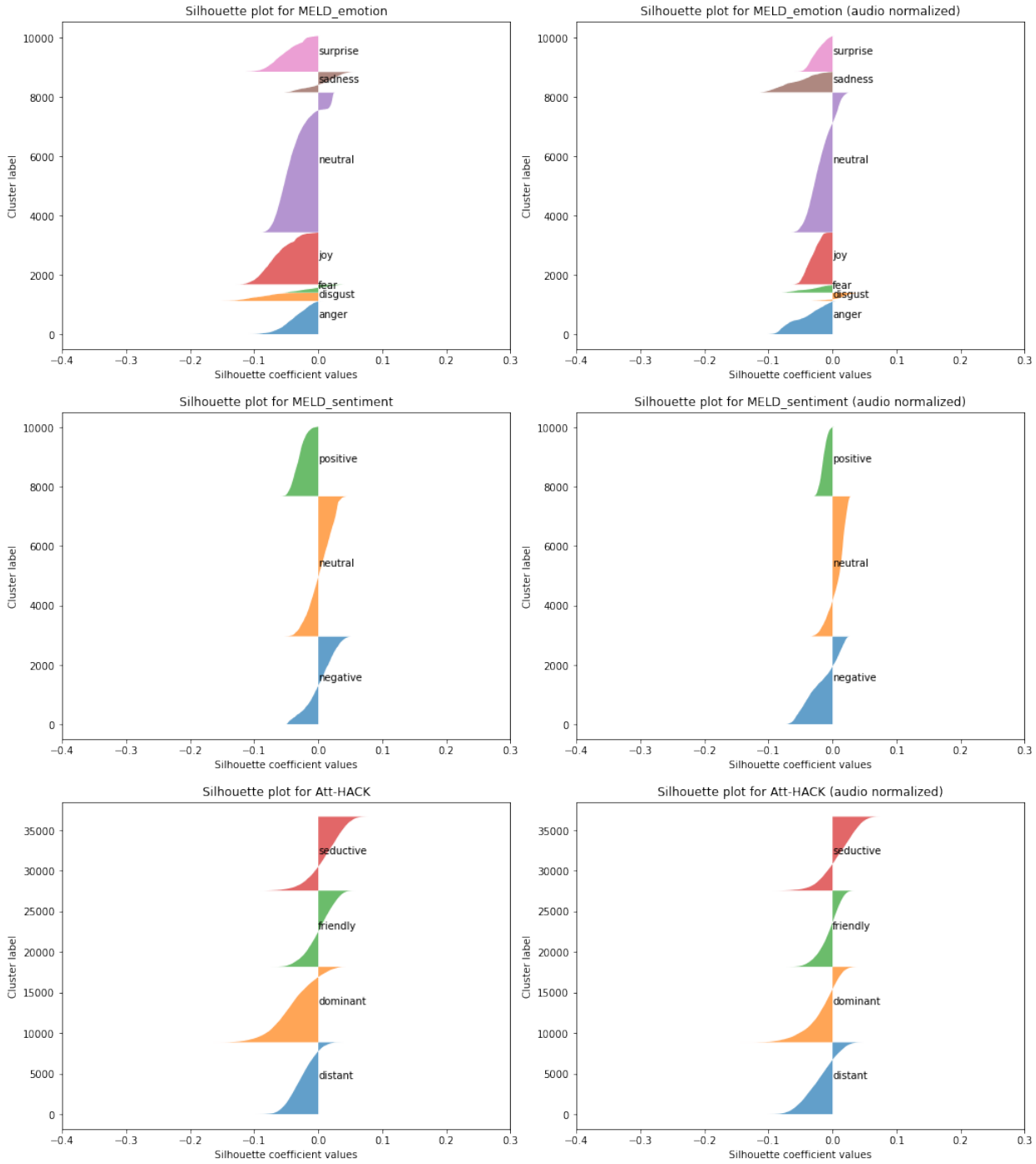

Table 3. Cross-validation metrics for various tuned classifiers and standard deviation ($N = 5$) across folds, including dummy classifier.

SAVEE (audio normalized)

Classifier	Balanced Accuracy	Accuracy	F1 macro
KNN	0.593 ± 0.037	0.621 ± 0.022	0.591 ± 0.042
SVM	**0.646 ± 0.043**	0.668 ± 0.041	0.648 ± 0.042
RF	0.644 ± 0.052	**0.688 ± 0.047**	**0.655 ± 0.056**
MLP	0.578 ± 0.039	0.597 ± 0.049	0.568 ± 0.037
Dummy	0.122 ± 0.013	0.127 ± 0.014	0.121 ± 0.012

SAVEE (not normalized)

Classifier	Balanced Accuracy	Accuracy	F1 macro
KNN	0.630 ± 0.048	0.649 ± 0.036	0.632 ± 0.043
SVM	0.650 ± 0.048	0.681 ± 0.029	0.654 ± 0.051
RF	**0.670 ± 0.034**	**0.707 ± 0.038**	**0.674 ± 0.041**
MLP	0.610 ± 0.059	0.634 ± 0.063	0.602 ± 0.055
Dummy	0.125 ± 0.040	0.146 ± 0.045	0.121 ± 0.040

Emo-DB (audio normalized)

Classifier	Balanced Accuracy	Accuracy	F1 macro
KNN	0.679 ± 0.049	0.712 ± 0.049	0.678 ± 0.049
SVM	0.752 ± 0.048	0.759 ± 0.037	0.747 ± 0.045
RF	0.739 ± 0.039	0.765 ± 0.033	0.738 ± 0.041
MLP	**0.774 ± 0.040**	**0.782 ± 0.043**	**0.768 ± 0.040**
Dummy	0.138 ± 0.042	0.143 ± 0.047	0.131 ± 0.040

Emo-DB (not normalized)

Classifier	Balanced Accuracy	Accuracy	F1 macro
KNN	0.692 ± 0.036	0.722 ± 0.045	0.691 ± 0.037
SVM	**0.764 ± 0.022**	**0.768 ± 0.027**	**0.766 ± 0.026**
RF	0.730 ± 0.043	0.759 ± 0.036	0.734 ± 0.047
MLP	0.738 ± 0.040	0.743 ± 0.037	0.735 ± 0.032
Dummy	0.159 ± 0.037	0.163 ± 0.046	0.152 ± 0.039

CREMA-D (audio normalized)

Classifier	Balanced Accuracy	Accuracy	F1 macro
KNN	0.484 ± 0.008	0.479 ± 0.009	0.458 ± 0.009
SVM	**0.562 ± 0.010**	**0.561 ± 0.010**	**0.559 ± 0.011**
RF	0.529 ± 0.016	0.527 ± 0.016	0.513 ± 0.017
MLP	0.539 ± 0.015	0.539 ± 0.015	0.537 ± 0.015
Dummy	0.168 ± 0.010	0.169 ± 0.010	0.168 ± 0.010

CREMA-D (not normalized)

Classifier	Balanced Accuracy	Accuracy	F1 macro
KNN	0.496 ± 0.008	0.491 ± 0.008	0.473 ± 0.006
SVM	**0.571 ± 0.011**	**0.570 ± 0.011**	**0.567 ± 0.011**
RF	0.537 ± 0.017	0.533 ± 0.016	0.521 ± 0.017
MLP	0.550 ± 0.009	0.549 ± 0.010	0.547 ± 0.009
Dummy	0.174 ± 0.006	0.175 ± 0.006	0.174 ± 0.006

MELD_emotion (audio normalized)

Classifier	Balanced Accuracy	Accuracy	F1 macro
KNN	0.160 ± 0.005	**0.394 ± 0.012**	0.148 ± 0.007
SVM	**0.165 ± 0.009**	0.319 ± 0.004	**0.163 ± 0.009**
RF	0.148 ± 0.007	0.287 ± 0.012	0.148 ± 0.006
MLP	0.164 ± 0.009	0.381 ± 0.012	0.160 ± 0.014
Dummy	0.144 ± 0.012	0.286 ± 0.010	0.143 ± 0.011

MELD_emotion (not normalized)

Classifier	Balanced Accuracy	Accuracy	F1 macro
KNN	0.163 ± 0.006	0.398 ± 0.015	0.153 ± 0.009
SVM	**0.170 ± 0.011**	0.360 ± 0.009	**0.169 ± 0.011**
RF	0.160 ± 0.005	**0.471 ± 0.023**	0.132 ± 0.013
MLP	**0.170 ± 0.004**	0.403 ± 0.012	0.166 ± 0.004
Dummy	0.144 ± 0.009	0.283 ± 0.008	0.144 ± 0.009

MELD_sentiment (audio normalized)

Classifier	Balanced Accuracy	Accuracy	F1 macro
KNN	0.371 ± 0.009	0.452 ± 0.011	0.345 ± 0.013
SVM	0.370 ± 0.014	0.448 ± 0.014	**0.356 ± 0.017**
RF	0.372 ± 0.007	**0.472 ± 0.009**	0.333 ± 0.009
MLP	**0.377 ± 0.008**	0.468 ± 0.015	0.349 ± 0.017
Dummy	0.329 ± 0.010	0.359 ± 0.011	0.329 ± 0.010

MELD_sentiment (not normalized)

Classifier	Balanced Accuracy	Accuracy	F1 macro
KNN	0.373 ± 0.009	0.462 ± 0.014	0.343 ± 0.009
SVM	0.384 ± 0.012	0.464 ± 0.020	**0.369 ± 0.015**
RF	0.382 ± 0.010	0.481 ± 0.012	0.346 ± 0.011
MLP	**0.391 ± 0.006**	**0.487 ± 0.008**	0.360 ± 0.017
Dummy	0.336 ± 0.008	0.366 ± 0.008	0.336 ± 0.008

Att-HACK (audio normalized)

Classifier	Balanced Accuracy	Accuracy	F1 macro
KNN	0.560 ± 0.006	0.561 ± 0.006	0.562 ± 0.006
SVM	**0.639 ± 0.011**	**0.639 ± 0.011**	**0.639 ± 0.012**
RF	0.602 ± 0.009	0.601 ± 0.009	0.600 ± 0.009
MLP	0.614 ± 0.010	0.613 ± 0.010	0.613 ± 0.010
Dummy	0.248 ± 0.008	0.248 ± 0.008	0.248 ± 0.008

Att-HACK (not normalized)

Classifier	Balanced Accuracy	Accuracy	F1 macro
KNN	0.616 ± 0.010	0.617 ± 0.010	0.616 ± 0.010
SVM	**0.686 ± 0.018**	**0.686 ± 0.018**	**0.686 ± 0.018**
RF	0.666 ± 0.009	0.666 ± 0.009	0.666 ± 0.009
MLP	0.664 ± 0.010	0.664 ± 0.010	0.664 ± 0.010
Dummy	0.248 ± 0.005	0.248 ± 0.005	0.248 ± 0.005

Table 4. Hyperparameters found by Bayesian optimization for regular classifiers.

		Hyperparameters found for dataset (audio volumes normalized)					
Model	**Parameter**	**SAVEE**	**Emo-DB**	**CREMA-D**	**MELD_emotion**	**MELD_sentiment**	**Att-HACK**
SVM	C	42.993294	10000.0	10000.0	10000.0	780.590539	56.538504
	gamma	0.009697	0.002096	0.0001	0.014551	0.002897	0.00466
KNN	n_neighbors	1	13	30	7	16	30
	p	1	1	1	1	1	1
RF	n_estimators	800	484	614	1	88	800
	criterion	'log_loss'	'entropy'	'entropy'	'gini'	'log_loss'	'entropy'
MLP	hidden_layer_sizes	50	50	30	50	10	50
	activation	'relu'	'relu'	'relu'	'tanh'	'relu'	'tanh'
	optimizer	'adam'	'adam'	'sgd'	'adam'	'adam'	'sgd'

		Hyperparameters found for dataset (no volume normalization)					
Model	**Parameter**	**SAVEE**	**Emo-DB**	**CREMA-D**	**MELD_emotion**	**MELD_sentiment**	**Att-HACK**
SVM	C	3507.290038	157.772406	23.963558	80.898058	10000.0	2.203558
	gamma	0.013161	0.01177	0.002701	0.014317	0.000363	0.02889
KNN	n_neighbors	1	14	23	7	19	30
	p	1	1	1	1	1	1
RF	n_estimators	659	291	657	89	80	770
	criterion	'gini'	'entropy'	'log_loss'	'gini'	'entropy'	'entropy'
MLP	hidden_layer_sizes	30	50	30	50	10	50
	activation	'relu'	'relu'	'relu'	'relu'	'tanh'	'relu'
	optimizer	'adam'	'sgd'	'sgd'	'adam'	'sgd'	'sgd'

Table 5. Metrics with standard deviation ($N = 5$) across folds and found hyperparameters for segmented model (no volume normalization).

Dataset	**SAVEE**		**Emo-DB**		**CREMA-D**	
Segment window size	1000 ms	500 ms	1000 ms	500 ms	1000 ms	500 ms
Segment hop size	500 ms	250 ms	500 ms	250 ms	500 ms	250 ms
Learn rate	0.00141	0.001771	0.003293	0.001021	0.001992	0.001987
LSTM hidden size	105	100	32	75	128	82
Dropout rate	0.065976	0.32155	0.429366	0.263525	0.259807	0.174893
Label smoothing	0.152388	0.247527	0.25	0.235065	0.024273	0.145705
Embedding size	75	104	119	121	77	100
Balanced accuracy	0.510 ± 0.026	**0.560 ± 0.054**	**0.745 ± 0.056**	0.708 ± 0.031	0.562 ± 0.010	**0.590 ± 0.006**
Accuracy	0.561 ± 0.058	**0.595 ± 0.061**	**0.752 ± 0.045**	0.721 ± 0.034	0.561 ± 0.009	**0.587 ± 0.007**
F1 macro	0.503 ± 0.027	**0.546 ± 0.053**	**0.728 ± 0.062**	0.694 ± 0.033	0.561 ± 0.008	**0.587 ± 0.006**

Fig. 5. Confusion matrices for predictions on each dataset by SVM classifier using tuned hyperparameters (no volume normalization).