Deep Multilayer Perceptrons for Dimensional Speech Emotion Recognition

Bagus Tris Atmaja*† and Masato Akagi†
* Sepuluh Nopember Institute of Technology, Surabaya, Indonesia
E-mail: bagus@ep.its.id
† Japan Advanced Institute of Science and Technology
E-mail: akagi@jaist.ac.jp

***Abstract*—Modern deep learning architectures are ordinarily performed in high performance computing facilities due to the large size of their input features and complexity of their models. This paper proposes traditional multilayer perceptrons (MLP) with deep layers and small input sizes to tackle this computation requirement limitation. This study compares a proposed deep MLP method to the more modern deep learning architectures with the same number of layers, batch size, and optimizer. The result shows that our proposed deep MLP outperformed modern deep learning architectures, i.e., LSTM and CNN, on the same number of layers and value of parameters. Both proposed and benchmark methods were optimized in the same way. The deep MLP exhibited the highest performance on both speaker-dependent and speaker-independent scenarios on IEMOCAP and MSP-IMPROV datasets.**

***Index Terms*—Affective computing, speech emotion recognition, multilayer perceptrons, neural networks, dimensional emotion**

I. Introduction

Speech emotion recognition is currently gaining interest from both academia and commercial industries. Researchers in the affective computing areas progressively proposed new methods to improve the accuracy of automatic emotion recognition. Commercial industries are trying to make this technology available to the market due to its potential applications. Previously, researchers have attempted to implement speech-based emotion recognition for wellbeing detection [1], call center application [2], and automotive safety [3].

One of the common requirements in computing speech emotion recognition is the availability of high performance computing facilities since the dataset usually is very large and the classifying methods are complicated. Graphical processing units (GPU)-based computers are more often used than CPU-based computers due to their processing speed to process the data, particularly, image-like data.

This paper proposes the use of deep multilayer perceptrons (MLP) to overcome the requirement of high computing power required by modern deep learning architectures. The inputs are high-level statistical functions (HSF) extracted from low-level descriptor (LLD), which are used to reduce the dimension of input features. The outputs are emotion dimensions, i.e., degree of valence, arousal, and dominance. The use of this classical technique with proper configurations shows promising results, measured in a correlation-based metric, over modern deep learning architectures.

According to research in psychology, two perspectives exist to model human emotions: dimensional emotion and categorical emotion. Russel [4] argued that emotion categories could be derived from dimensional emotion, particularly in valence-arousal space. Given this benefit, the ability to convert dimensional emotion to categorical emotion but not vice versa, predicting dimensional emotions is more beneficial than predicting categorical emotions. We added dominance to valence-arousal space, since it is suggested in [5] and this label is provided in the datasets. Dimensional emotion recognition is performed with speech input since the target applications are speech-based technologies like call center and voice assistant applications.

Deep neural network (DNN) is an extension of the neural network with deeper layers, i.e., more than one layer. Commonly DNNs were constructed with five or more layers [6]. This paper compares deep MLP and modern deep learning architecture, i.e., LSTM and CNN using the criteria of concordance correlation coefficient (CCC). This comparison is made on the same conditions. Our results show that on both speaker-dependent and speaker-independent scenarios (in three datasets), deep MLP obtained higher performances than LSTM and CNN from small-size input features.

II. Data and Feature Sets

This section describes data and feature sets used in this research.

a) IEMOCAP: The interactive emotional dyadic motion capture database [7] is used in this research. The database contains approximately 12 h of data with 10039 utterances. All of these data are used. Although the database consists of the measurement of speech, facial expression, head, and movements of affective dyadic sessions, only speech data are processed. The dimensional emotion labels are given in valence, arousal, and dominance attributes. The score of these attributes are in range [1-5], and they are normalized into [-1, 1] following the work in [8] when these labels are fed into the neural network system. Two versions of speech data are available in the dataset: stereo data per dialog and mono data per sentence (utterance). The mono version was used due to ease to process with the labels. The sampling rate and format of the data was 16 kHz and 16-bit PCM.

The IEMOCAP dataset were arranged into two: speaker-dependent (SD) and speaker-independent scenarios. We split the dataset with ratio around 80/20 for training/test set on speaker-independent scenario. The last session, i.e., session five, is left for the test set (leave one session out, LOSO) on speaker-independent scenario. The number of utterances in each partition in the speaker-independent scenario is similar to that of speaker-dependent scenario shown in Table I.

b) MSP-IMPROV: We used the MSP-IMPROV corpus to generalize the impact of the evaluated methods. MSP-IMPROV is "an acted corpus of dyadic interactions to study emotion perception" [9]. This dataset consists of speech and visual recordings of 18 hours of affective dyadic sessions. Only the speech data, consisting of 8438 utterances, was used. The same split ratio was used in speaker-dependent scenario; the last session six is used for test set in speaker-independent scenario. The same labels scale and normalization were performed as in IEMOCAP dataset. While IEMOCAP labels are annotated by at least two annotators, these MSP-IMPROV labels were annotated by at least five annotators making the commonalities of the labels are higher than the previous IEMOCAP dataset. The 16 bits mono audio format sampled at 44 kHz were used.

Table I shows the number of utterances allocated for each set partition for both speaker-dependent and speaker-independent scenarios, including MSP-IMPROV dataset.

c) Mixed-corpus: We mixed both IEMOCAP and MSP-IMPROV datasets to create a new mixed-corpus dataset. We concatenated speaker-dependent scenario from IEMOCAP with speaker-dependent scenario from MSP-IMPROV. The concatenation were performed for each training, development, and test set. The same concatenation rules also applied for the speaker-independent scenario.

TABLE I
NUMBER OF UTTERANCES USED IN THE DATASETS ON EACH PARTITION

Scenarios	Training	Development	Test
IEMOCAP-SD	6431	1608	2000
IEMOCAP-LOSO	6295	1574	2170
IMPROV-SD	5256	1314	1868
IMPROV-LOSO	5452	1364	1622

d) Acoustic Feature Set: We used high statistical functions of the low-level descriptor (LLD) from Geneva Minimalistic Acoustic Parameter Set (GeMAPS), developed by Eyben et al. [10]. The HSF features were extracted per utterance in which the the labels were given. This HSF features were calculated from LLDs processed on a frame-based level with 25 ms window size and 10 ms of hop size. The use of HSF features reduces computation complexity since the feature size decreased from (3409 $\times$ 23) to (1 $\times$ 23 features), for the IEMOCAP dataset. To obtain the HSF feature, however, LLDs must obtained first. Then, HSF can be calculated as statistics of those LLDs for a fixed time, in this case, per utterance.

To add those statistical functions, we used a silent pause feature, which is also extracted per utterance. Silenct pause feature, in this paper, is defined as the portion of the silence frames compared to the total frames in an utterance. In human-human communication, this portion of silence in speaking might depends on the speaker's emotion. For example, high arousal emotion category like happy may have fewer silences (or pauses) than a sad emotion category. The ratio of silent pause per utterance is calculated as

TABLE II
GEMAPS FEATURE [10] AND ITS FUNCTIONALS; ONLY FUNCTIONALS (HSFS) USED IN THIS DIMENSIONAL SER.

LLDs	intensity, alpha ratio, hammarberg index, spectral slope 0-500 Hz, spectral slope 500-1500 Hz, spectral flux, 4 MFCCs, f_o, jitter, shimmer, Harmonics-to-Noise Ratio (HNR), Harmonic difference H1-H2, Harmonic difference H1-A3, F1, F1 bandwidth, F1 amplitude, F2, F2 amplitude, F3, and F3 amplitude.
HSFs	mean (of LLDs), standard deviation (of LLDs), silence

$$S = \frac{N_s}{N_t}, \tag{1}$$

where N_s is the number of frames to be categorized as silence (silence frames), and N_t is the number of total frames within an utterance. To be categorized as silence, the root mean square (RMS) energy of a frame must be less than a threshold. The threshold is a multiplication of *a factor* with RMS energy. Threshold (th) and RMS energy (X_{rms}) are formulated as

$$th = 0.3 \times \overline{X_{rms}} \tag{2}$$

and

$$X_{rms} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x[i]^2}, \tag{3}$$

where a silence factor of 0.3 is obtained from experiments. These equations are similar to that is proposed in [11] and [12]. In [11], the author used a different threshold factor for the same dataset used in this research. In [12], the author divided the total duration of disfluency over the total utterance length on n words and counted it as a disfluency feature.

III. BENCHMARK AND PROPOSED METHOD

LSTM and CNN are used as baselines, while MLP with deep layers is the proposed method. All evaluated methods used the same numbers of layers, units and value of parameters.

A. Benchmark Methods: LSTM and CNN

LSTM and CNN are two common deep learning architectures widely used in speech emotion recognition [13], [14], [15]. These two architectures were used as the baselines due to their reported effectiveness on predicting valence, arousal, and dominance. Both LSTM and CNN have the same five layers with the same number of units. We determined the size of kernel in CNN architecture in order to that the number of trainable parameters is similar to LSTM and MLP. The other parameters like batch size, feature and label standardization,

loss function, number of iterations, and callback criteria, are same for both architectures.

Fig. 1 shows the structures of both LSTM and CNN. On the first layer, 256 neurons are used and decreased by half for the subsequent level. The decrease of the number of neurons on each layer is based on assumption that the model is supposed to learn better along with these layers. Five LSTM layers used `tanh` as activation function, while five CNN layers used ReLU activation function. We kept all output from the last LSTM layer and flattened it before splitting it into three dense layers for obtaining the prediction of valence, arousal, and dominance. For the CNN architecture, the same `flatten` layer is used before entering three one-unit dense layers.

All architectures used the same standardization for input features and labels. A z-score normalization is used to standardize feature, while `minmax scaler` [16] is used to scale the labels into [-1, 1] range. Apart from using MSE as a loss function in LSTM and CNN, we optimized the benchmark methods by using CCC [17] loss with multitask learning (MTL) approach, in which the prediction of valence, arousal, and dominance are done simultaneously. The calculation of CCC loss (CCCL) begins from the following CCC evaluation to measure the performance of recognition,

$$CCC = \frac{2\rho\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2 + (\mu_x - \mu_y)^2}. \quad (4)$$

Then, the CCC loss can be formulated as the opposite:

$$CCCL = 1 - CCC, \quad (5)$$

where ρ is Pearson's correlation between gold standard y and and predicted score x, σ is standard deviation, and μ is the mean score. The total loss ($CCCL_T$) for three variables is then defined as the sum of CCCL for these three as follows:

$$CCCL_T = CCCL_V + CCCL_A + CCCL_D \quad (6)$$

where subscript V, A and D denote valence, arousal, and dominance, respectively. The average CCC is used to evaluate all architectures including the proposed deep MLP method.

All architectures (LSTM, CNN, MLP) used a mini-batch size of 200 utterances (by shuffling the orders), maximum number iteration of 180, and 10 iterations patience of early stopping criteria (callback). An Adam optimizer [18] is used to adjust the learning rate during the training process. Both LSTM and CNN architectures run on GPU-based machine using CuDNN implementation [19] within Keras toolkit [20]. The MLP architecture was implemented in scikit-learn toolkit [16].

B. Proposed Method: Deep MLP

Fig. 2 shows our proposed deep MLP structure. The MLP used here similar to the definition of connectionist learning proposed by Hinton [21]. As the benchmark methods (LSTM and CNN), deep MLP also has five layers with the same number of units as previous. The layer structure differs from benchmark methods in the absence of flatten layer since the output of the last MLP layers can be coupled directly to three

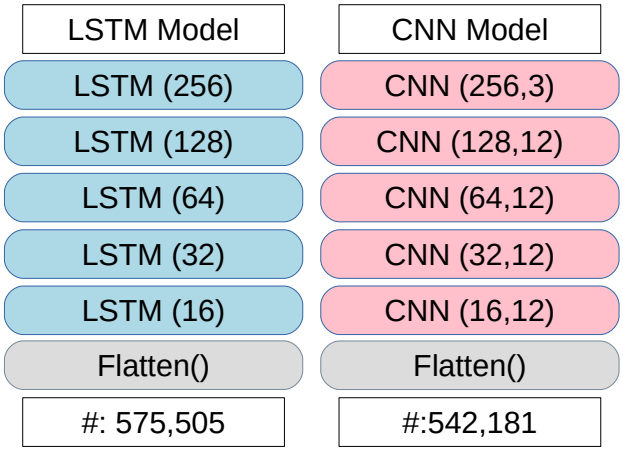

Fig. 1. Diagram of LSTM and CNN used for benchmarking of proposed deep MLP; the number inside the bracket denoted the number of units (nodes); the second number on CNN denotes kernel size, # denotes number of trainable parameters.

one-unit dense layers. While the previous LSTM and CNN used batch normalization layer in the beginning (input) to speed-up computation process, this deep MLP structure did not use that layer since the implementation only require a minute to run on a CPU-based machine.

We used the same batch size, tolerance for early stopping criteria, optimizer, and maximum number of iteration as the benchmark methods. While the benchmark methods are optimized with CCC loss function beside MSE, the proposed deep MLP method used MSE only as the cost function,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2. \quad (7)$$

The total loss function is given as an average of MSE scores from valence, arousal, and dominance,

$$MSE_T = MSE_V + MSE_A + MSE_D. \quad (8)$$

There are no weighting factors used here for emotional attributes since we assume all emotion dimensions are equally contributed to the emotional state. The Python implementation of the proposed deep MLP method and benchmark methods are available at https://github.com/bagustris/deep_mlp_ser.

IV. Experiment Results

A. MLP vs. DNN results

CCC is the standard metric used in affective computing to measure the performance of dimensional emotion recognition. We presented our results on that metric in two different groups: within-corpus and mixed-corpus evaluations. The results are shown in Table III and IV.

Table III shows CCC scores of valence (V), arousal (A), dominance (D) and their average from different datasets, scenarios, and methods. The proposed deep MLP method outperforms benchmark methods by significant margins. On both every emotion dimension and averaged score, the proposed deep MLP attained the highest CCC score for both

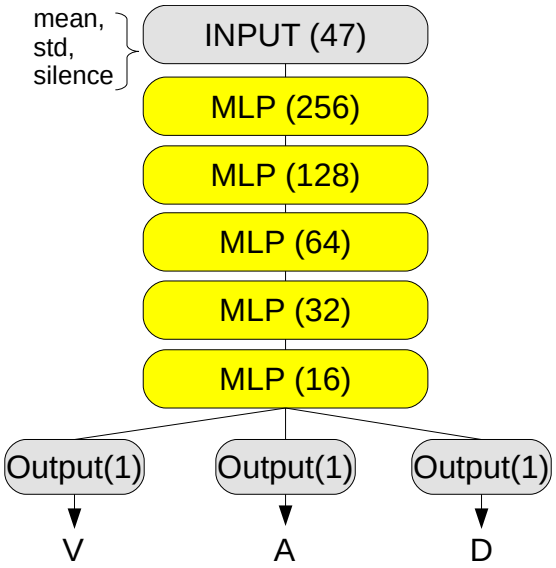

Fig. 2. Diagram of proposed deep MLP with five layers; the number inside the bracket denoted the number of units.

speaker-dependent and speaker-independent scenarios (bold). On the IEMOCAP dataset, the speaker-dependent score is only slightly higher than speaker-independent score due to the nature of the dataset structure. The utterances in the IEMOCAP dataset is already in order by its session when it is sorted by file names. The change from speaker-dependent to speaker-independent is done by changing the number of train/test partitions. In contrast, the file naming of utterances in MSP-IMPROV made the arrangement of the sessions not in order when utterances are sorted by its file names. We did the sorting process to assure the pair of features and labels. The gap between speaker-dependent and speaker-independent scenarios in MSP-IMPROV is larger than in IEMOCAP, which may be caused by these different files organization.

TABLE III
RESULTS OF CCC SCORES ON WITHIN-CORPUS EVALUATION WITH MSE LOSS

Dataset	Method	V	A	D	Mean
		speaker-dependent			
IEMOCAP	LSTM	0.140	0.485	0.381	0.335
	CNN	0.060	0.381	0.340	0.260
	MLP	**0.335**	**0.599**	**0.473**	**0.469**
		speaker-independent			
	LSTM	0.142	0.468	0.363	0.324
	CNN	0.068	0.398	0.330	0.265
	MLP	**0.316**	**0.488**	**0.454**	**0.453**
		speaker-dependent			
MSP-IMPROV	LSTM	0.350	0.599	0.462	0.471
	CNN	0.300	0.580	0.439	0.440
	MLP	**0.438**	**0.650**	**0.519**	**0.536**
		speaker-independent			
	LSTM	0.182	0.492	0.336	0.337
	CNN	0.174	0.461	0.327	0.321
	MLP	**0.271**	**0.553**	**0.406**	**0.410**

Table IV shows the results from the mixed-corpus dataset. This corpus is a concatenation of IEMOCAP with MSP-IMPROV for both speaker-dependent and speaker-independent scenarios, as listed in Table I. In this mixed corpus, the proposed deep MLP method also outperformed LSTM and CNN on all emotion dimensions and averaged CCC scores. The score on speaker-dependent scenario is in between the score of speaker-dependent scenario in IEMOCAP and MSP-IMPROV corpus. For speaker-independent scenario, the score of mixed corpus is lower than in within corpus. This low score suggests that speaker variability (in different sessions) affected the result, even with the z-normalization process. Instead of predicting one different session (LOSO), the test set in the mixed-corpus consists of two different sessions, each from IEMOCAP and MSP-IMPROV, which made the regression task more difficult.

In Table III and IV, results for CNN are average scores of 20 runs. This normalization is performed due to random weighting initiation in CNN layers. LSTM and MLP show the same result for each run thanks to initiation of fixed random seed numbers.

We show that our proposed deep MLP functioned to overcome the requirement of modern neural network architectures since it surpassed the results obtained by these architectures. Using a small dimension of feature size, i.e., 47-dimensional data, our deep MLP with five layers, excluding input and output layers, achieved the highest performance. Modern deep learning architectures require high-end hardware which costs expensive (e.g., GPU card). We showed that using a small deep MLP architecture, which does not require a high computation load, can achieve a better performance. Our proposed deep MLP method gained a higher performance than benchmark methods not only on both within-corpus and mixed-corpus datasets but also on both speaker-dependent and speaker-independent scenarios.

TABLE IV
RESULTS OF CCC SCORES ON MIXED-CORPUS EVALUATION; ALL METHODS USE THE SAME MSE LOSS FUNCTION.

Method	V	A	D	Mean
	speaker-dependent			
LSTM	0.224	0.498	0.392	0.371
CNN	0.148	0.413	0.342	0.301
MLP	**0.395**	**0.640**	**0.461**	**0.499**
	speaker-independent			
LSTM	0.081	0.272	0.216	0.190
CNN	0.038	0.234	0.194	0.155
MLP	**0.212**	**0.402**	**0.269**	**0.294**

B. Optimizing the benchmark methods

Two strategies were performed to optimize the benchmark methods: the use of CCC loss and different layer number. These optimizing methods are intended to evaluate whether the benchmark methods could close or surpass the performance of the proposed method by using different parameters value. First, the CCC loss is used instead of the MSE. Second, we varied the number of layers. The first method is utilized due to the evaluation is measured in CCC; the second method is evaluated to avoid overfitting data by using smaller layer numbers.

Table V shows the CCC scores of the benchmark methods with CCC loss instead of MSE. The results show significant improvement over the previous method shown in Table III. For instance, the averaged CCC score on IEMOCAP dataset improved from 0.335, 0.260, 0.324, 0.265 to 0.379, 0.387, 0.382, 0.334 for LSTM SD, CNN SD, LSTM LOSO, and CNN LOSO, respectively. These improvements are reasonable since the loss function is correlated with the evaluation metric. The improved CCC scores are, however, still far from the scores obtained by MLP, i.e., 0.469 for IEMOCAP SD and 0.453 for IEMOCAP LOSO.

On the MSP-IMPROV dataset, the optimized benchmark methods also fail to surpass the deep MLP results. In this dataset, the gap between DNN and MSP becomes smaller. There is also a case where a DNN method gained a higher score in the dominance part of MSP-IMPROV speaker-dependent scenario. In this case, however, the average CCC score by MLP is still the highest. The highest averaged scores by DNN in this dataset are 0.531 (MLP=0.536) for speaker-dependent with LSTM and 0.380 (MLP=0.410) for speaker-independent with CNN. In case of CNN, the results presented in Table VI are also average scores of 20 runs.

The results of optimizing benchmark methods on mixed-corpus dataset are similar to IEMOCAP and MSP-IMPROV datasets as shown in Table IX. The performance scores obtained by LSTM and CNN with CCC loss improved the previous score with MSE, but they are still lower than MLP. On both speaker-dependent and speaker-independent scenarios, LSTM obtained higher scores than CNN. Given the fact that LSTM performs better than CNN in most cases, we performed evaluation on the use of different layer numbers.

TABLE V
CCC SCORES OF OPTIMIZED BENCHMARK METHODS ON WITHIN-CORPUS EVALUATION WITH CCC LOSS FUNCTION

Dataset	Method	V	A	D	Mean
		speaker-dependent			
IEMOCAP	LSTM	0.213	0.506	0.418	0.379
	CNN	0.135	0.451	0.407	0.331
		speaker-independent			
	LSTM	0.204	0.520	0.422	0.382
	CNN	0.127	0.467	0.409	0.334
		speaker-dependent			
MSP-IMPROV	LSTM	0.413	0.639	0.541	0.531
	CNN	0.375	0.623	0.517	0.506
		speaker-independent			
	LSTM	0.210	0.487	0.333	0.344
	CNN	0.226	0.519	0.382	0.376

Table VII shows the number of layers and their corresponding number of units/nodes on each layer. These layer numbers variation were used to evaluate LSTM and MLP. Although the baseline results shown in Table III and IV used five layers, it is a worth to evaluate the performance over different number of layers. In most cases, overfitting is the common problem in neural network to be avoided. A simple method to avoid overfitting is by reducing the number of layers, which reduces the number of trainable parameters.

TABLE VI
CCC SCORES OF OPTIMIZED BENCHMARK METHODS ON MIXED-CORPUS EVALUATION WITH CCC LOSS FUNCTION

Method	V	A	D	Mean
	speaker-dependent			
LSTM	0.278	0.533	0.450	0.420
CNN	0.202	0.477	0.432	0.371
	speaker-independent			
LSTM	0.102	0.300	0.252	0.218
CNN	0.076	0.271	0.248	0.198

TABLE VII
NUMBER OF LAYER AND CORRESPONDING UNITS/NODES ON EACH LAYER

# layers	# units
1	(16)
2	(32, 16)
3	(64, 32, 16)
4	(128, 64, 32, 16)
5	(256, 128, 64, 32, 16)
6	(512, 256, 128, 64, 32, 16)

Table VIII shows the effect of using a different number of layers in LSTM method (with CCC loss). The result shows that reducing the number of layers improves the performance of LSTM method. This result reveals that overfitting exists in the results with larger number of layers. The averaged CCC scores improved from 0.379, 0.382, 0.531, and 0.344 to 0.397, 0.393, 0.532, and 0.372 for IEMOCAP SD, IEMOCAP LOSO, MSP-IMPROV SD, and MSP-IMRPV LOSO, respectively. Although the results are improved, this second strategy to optimize the benchmark methods fails to surpass score of the MLP method.

A similar improvement on reducing number of layers was also observed in mixed-corpus dataset. Table IX shows the improved CCC scores from the previous Table VI. The highest CCC score by this layer number variation is 0.423 (MLP=0.499) for speaker-dependent scenario and 0.235 (MLP=0.294) for speaker-independent scenario. To this end, two methods to optimize the benchmark methods failed to surpass the performance obtained by the proposed MLP method.

C. Optimizing the proposed MLP method

The proposed MLP method is re-evaluated with different number of layers as the benchmark methods. Table X and XI shows the evaluation result on within-corpus and mixed-corpus datasets.

Table X shows that by using smaller number of layers the improved CCC scores could be obtained. In a speaker-dependent scenario, the highest CCC score for both IEMOCAP and MSP-IMPROV datasets were obtained by using three layers; for speaker-independent scenario, it was two layers. Although the improvement was small, using smaller number of layers reduces the computation cost of the MLP method. In this within-corpus evaluation, the CCC scores improved from 0.469, 0.453, 0.536, and 0.416 to 0.472, 0.455, 0.562, and 0.416, for IEMOCAP SD, IEMOCAP LOSO, MSP-IMPROV SD, and MSP-IMPRPOV LOSO, respectively.

TABLE VIII
CCC SCORES OF OPTIMIZED LSTM METHOD ON WITHIN-CORPUS EVALUATION WITH DIFFERENT NUMBER OF LAYERS

Dataset	# layers	V	A	D	Mean
		speaker-dependent			
IEMOCAP	6	0.203	0.493	0.422	0.373
	5	0.213	0.506	0.418	0.379
	4	0.219	0.512	0.422	0.384
	3	0.216	0.550	0.425	**0.397**
	2	0.203	0.537	0.425	0.389
	1	0.180	0.520	0.414	0.372
		speaker-independent			
	6	0.199	0.478	0.402	0.360
	5	0.204	0.520	0.422	0.382
	4	0.191	0.528	0.413	0.378
	3	0.201	0.511	0.397	0.370
	2	0.193	0.552	0.432	0.392
	1	0.194	0.555	0.431	**0.393**
		speaker-dependent			
MSP-IMPROV	6	0.405	0.640	0.543	0.529
	5	0.413	0.639	0.541	0.531
	4	0.415	0.638	0.543	**0.532**
	3	0.418	0.639	0.536	0.531
	2	0.414	0.633	0.536	0.528
	1	0.412	0.623	0.530	0.522
		speaker-independent			
	6	0.208	0.501	0.341	0.350
	5	0.210	0.487	0.333	0.344
	4	0.191	0.503	0.346	0.347
	3	0.217	0.481	0.309	0.336
	2	0.264	0.507	0.344	**0.372**
	1	0.236	0.494	0.326	0.352

TABLE X
CCC SCORES OF OPTIMIZED MLP METHOD ON WITHIN-CORPUS EVALUATION WITH DIFFERENT NUMBER OF LAYERS

Dataset	# layers	V	A	D	Mean
		speaker-dependent			
IEMOCAP	6	0.311	0.614	0.475	0.467
	5	0.335	0.599	0.473	0.469
	4	0.342	0.582	0.476	0.467
	3	0.341	0.615	0.459	**0.472**
	2	0.332	0.612	0.458	0.468
	1	0.295	0.632	0.461	0.463
		speaker-independent			
	6	0.260	0.605	0.437	0.434
	5	0.316	0.588	0.454	0.453
	4	0.326	0.575	0.448	0.450
	3	0.313	0.601	0.437	0.450
	2	0.301	0.611	0.453	**0.455**
	1	0.254	0.608	0.443	0.435
		speaker-dependent			
MSP-IMPROV	6	0.243	0.595	0.483	0.441
	5	0.438	0.650	0.519	0.536
	4	0.478	0.672	0.535	0.561
	3	0.479	0.669	0.539	**0.562**
	2	0.453	0.674	0.538	0.555
	1	0.441	0.659	0.524	0.541
		speaker-independent			
	6	0.272	0.563	0.403	0.413
	5	0.258	0.545	0.388	0.397
	4	0.279	0.510	0.363	0.384
	3	0.270	0.549	0.392	0.404
	2	0.290	0.556	0.402	**0.416**
	1	0.266	0.537	0.375	0.393

TABLE IX
CCC SCORES OF OPTIMIZED LSTM METHOD ON MIXED-CORPUS EVALUATION WITH DIFFERENT NUMBER OF LAYERS

# layers	V	A	D	Mean
	speaker-dependent			
6	0.259	0.529	0.423	0.404
5	0.278	0.533	0.450	0.420
4	0.272	0.534	0.459	0.422
3	0.254	0.527	0.441	0.408
2	0.264	0.553	0.451	**0.423**
1	0.241	0.544	0.453	0.413
	speaker-independent			
6	0.125	0.276	0.239	0.213
5	0.102	0.300	0.252	0.218
4	0.116	0.303	0.246	0.222
3	0.109	0.297	0.239	0.215
2	0.127	0.321	0.250	0.233
1	0.124	0.326	0.255	**0.235**

TABLE XI
CCC SCORES OF OPTIMIZED MLP METHOD ON MIXED-CORPUS EVALUATION WITH DIFFERENT NUMBER OF LAYERS

# layers	V	A	D	Mean
	speaker-dependent			
6	4.1e-6	2.3e-7	2.1e-7	1.6e-7
5	0.432	0.638	0.453	**0.508**
4	0.429	0.636	0.437	0.501
3	0.405	0.640	0.457	0.501
2	0.370	0.635	0.456	0.487
1	0.292	0.625	0.450	0.456
	speaker-independent			
6	0.230	0.422	0.283	0.312
5	0.200	0.418	0.313	0.310
4	0.212	0.410	0.326	**0.316**
3	0.228	0.394	0.287	0.303
2	0.232	0.389	0.285	0.302
1	0.130	0.355	0.268	0.251

Table XI shows optimization of the MLP method in mixed-corpus evaluation. In the speaker-dependent scenario, the use of larger six layers resulted in very small numbers of CCC scores due to convergence problem. This mixed-corpus dataset also needs to be trained in whole iteration number instead of using a stopping criterion. The previous results, shown in Table IV and others, were obtained using 10 patience of earlystopping criteria. The results were slightly improved from 0.499 to 0.508 for speaker-dependent data and 0.294 to 0.316 for speaker-independent data.

This research has thrown up several issues in need of further investigation. First, the complexity of mixed datasets might increase the complexity of training process. This different corpus evaluation is a challenge in SER, particularly on the use of cross datasets: different datasets are used for training and testing. Second, the finding that MLP performs better than DNN may be shown on small datasets, as used in this research. It is necessary to confirm these findings on larger datasets. Third, the proposed method works effectively on the small size of feature; this small features may be a limitation of our proposed deep MLP method. While we optimized the MLP method with a different number of layers, the optimization method with a different loss function, i.e., CCC loss, is left for the future research. We presume that the use of CCC loss will improve the performance over MSE and MAE as observed in the benchmark methods.

V. Conclusions

This paper demonstrated that the use of deep MLP with proper parameter choices outperformed the more modern neural network architectures with the same value of parameters. We compared MLP to DNN (LSTM and CNN) with the same number of layers (five layers), batch-size (200 batch) and solver (adam). For both speaker-dependent and speaker-independent scenarios, the proposed deep MLP attained the consistent highest performance among the evaluated methods. The proposed deep MLP also attained the highest score on both within-corpus and mixed-corpus scenarios. We improved both MLP and benchmark DNN method with different number of layers. Although, the benchmark methods were optimized with different loss function and different number of layers, the obtained scores could not surpass the scores obtained by MLP. Reducing number of layers in both MLP and benchmark methods improve the performance scores slightly. Based on the results of these investigations, there are no high requirements on computing power to obtain state-of-the-art results on dimensional speech emotion recognition. The proper choice of feature (i.e., using small size feature) and the classifier can leverage the performance of conventional neural networks. Future research should be directed to investigate the performance of the proposed method on cross-corpus evaluations and on the use of different loss functions, which are not evaluated in this paper.

References

[1] Y. Gao, Z. Pan, H. Wang, and G. Chen, "Alexa, My Love: Analyzing reviews of amazon echo," in *2018 IEEE SmartWorld, Ubiquitous Intell. Comput. Adv. Trust. Comput. Scalable Comput. Commun. Cloud Big Data Comput. Internet People Smart City Innov.* IEEE, oct 2018, pp. 372–380. [Online]. Available: https://ieeexplore.ieee.org/document/8560072/

[2] V. A. Petrushin, "Emotion In Speech: Recognition And Application To Call Centers," *Proc. Artif. neural networks Eng.*, vol. 710, pp. 22–30, 1999.

[3] C. Nass, I. M. Jonsson, H. Harris, B. Reaves, J. Endo, S. Brave, and L. Takayama, "Improving automotive safety by pairing driver emotion and car voice emotion," in *Conf. Hum. Factors Comput. Syst. - Proc.*, 2005.

[4] J. A. Russell, "Affective space is bipolar," *J. Pers. Soc. Psychol.*, 1979.

[5] I. Bakker, T. van der Voordt, P. Vink, and J. de Boon, "Pleasure, Arousal, Dominance: Mehrabian and Russell revisited," *Curr. Psychol.*, vol. 33, no. 3, pp. 405–421, 2014.

[6] B. T. Atmaja, D. Arifianto, and M. Akagi, "Speech recognition on Indonesian language by using time delay neural network," *ASJ Spring Meet.*, pp. 1291–1294, 2019.

[7] C. Busso, M. Bulut, C.-C. C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," *Lang. Resour. Eval.*, vol. 42, no. 4, pp. 335–359, 2008.

[8] S. Parthasarathy and C. Busso, "Jointly Predicting Arousal, Valence and Dominance with Multi-Task Learning," in *Interspeech*, 2017, pp. 1103–1107.

[9] C. Busso, S. Parthasarathy, A. Burmania, M. AbdelWahab, N. Sadoughi, and E. M. Provost, "MSP-IMPROV: An Acted Corpus of Dyadic Interactions to Study Emotion Perception," *IEEE Trans. Affect. Comput.*, vol. 8, no. 1, pp. 67–80, jan 2017.

[10] F. Eyben, K. R. Scherer, B. W. Schuller, J. Sundberg, E. Andre, C. Busso, L. Y. Devillers, J. Epps, P. Laukka, S. S. Narayanan, and K. P. Truong, "The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing," *IEEE Trans. Affect. Comput.*, vol. 7, no. 2, pp. 190–202, apr 2016.

[11] G. Sahu and D. R. Cheriton, "Multimodal Speech Emotion Recognition and Ambiguity Resolution," Tech. Rep. [Online]. Available: http://tinyurl.com/y55dlc3m

[12] J. D. Moore, L. Tian, and C. Lai, "Word-level emotion recognition using high-level features," in *Int. Conf. Intell. Text Process. Comput. Linguist.* Springer, 2014, pp. 17–31.

[13] Y. Xie, R. Liang, Z. Liang, and L. Zhao, "Attention-based dense LSTM for speech emotion recognition," *IEICE Trans. Inf. Syst.*, vol. E102D, no. 7, pp. 1426–1429, 2019.

[14] M. Schmitt, N. Cummins, and B. W. Schuller, "Continuous Emotion Recognition in Speech - Do We Need Recurrence?" in *Interspeech 2019*. ISCA: ISCA, sep 2019, pp. 2808–2812.

[15] B. T. Atmaja and M. Akagi, "Speech Emotion Recognition Based on Speech Segment Using LSTM with Attention Model," in *2019 IEEE Int. Conf. Signals Syst.* IEEE, jul 2019, pp. 40–44.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825—-2830, 2011.

[17] L. I.-K. Lin, "A concordance correlation coefficient to evaluate reproducibility," *Biometrics*, vol. 45, no. 1, pp. 255–68, 1989.

[18] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, dec 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[19] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, "cuDNN: Efficient Primitives for Deep Learning," Tech. Rep., oct 2014.

[20] F. Chollet and Others, "Keras," https://keras.io, 2015.

[21] G. E. Hinton, "Connectionist learning procedures," *Artif. Intell.*, vol. 40, no. 1-3, pp. 185–234, 1989.