IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 26, 2020, accepted November 2, 2020, date of publication November 4, 2020, date of current version November 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3035910

Ensemble Learning With Attention-Integrated Convolutional Recurrent Neural Network for Imbalanced Speech Emotion Recognition

XUSHENG AI[1], VICTOR S. SHENG[2], (Senior Member, IEEE), WEI FANG[3], CHARLES X. LING[4], AND CHUNHUA LI[1]

[1]Software and Service Outsourcing College, Suzhou Vocational Institute of Industrial Technology, Suzhou 215104, China
[2]Department of Computer Science, Texas Tech University, Lubbock, TX 79409, USA
[3]Jiangsu Engineering Center of Network Monitoring, School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China
[4]Department of Computer Science, Western University, London, ON N6A 5B7, Canada

Corresponding authors: Xusheng Ai (00754@siit.edu.cn) and Victor S. Sheng (victor.sheng@ttu.edu)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702351, in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 17KJB520036, in part by the Foundation of Key Laboratory in Science and Technology Development Project of Suzhou under Grant SZS201609, in part by the Suzhou Science and Technology Plan Project under Grant SYG201903, and in part by the Computer Basic Education Teaching Research Project under Grant 2018-AFCEC-328, 2019-AFCEC-288.

ABSTRACT This article addresses observation duplication and lack of whole picture problems for ensemble learning with the attention model integrated convolutional recurrent neural network (ACRNN) in imbalanced speech emotion recognition. Firstly, we introduce Bagging with ACRNN and the observation duplication problem. Then Redagging is devised and proved to address the observation duplication problem by generating bootstrap samples from permutations of observations. Moreover, Augagging is proposed to get oversampling learner to participate in majority voting for addressing the lack of whole picture problem. Finally, Extensive experiments on IEMOCAP and Emo-DB samples demonstrate the superiority of our proposed methods (i.e., Redagging and Augagging).

INDEX TERMS Imbalance learning, ensemble learning, convolutional neural network, recurrent neural network, speech emotion recognition.

I. INTRODUCTION

Emotion is important paralinguistic information in human communication. Emotion directs non-linguistic social signals (such as body language and facial expression) to express wants, needs and desires [1]. There are many applications of speech emotion recognition in different fields such as healthcare [2], services [3], and telecommunication [4]. In the healthcare field, speech emotion recognition can help clinicians assess patients' psychological disorders online. In the industry of customer call centers, speech emotion recognition (SER) can be used to detect customers' satisfaction. Speech emotion recognition can be also used to route 911 emergency call services for high priority emergency calls.

The associate editor coordinating the review of this manuscript and approving it for publication was Shuping He.

In recent years, convolutional recurrent neural network (CRNN) is widely used for SER [5]–[7]. At an early stage, CRNN simply assumes that a sequence of frames share the same emotion-relevant weight in an utterance. Later an attention model integrated into convolutional recurrent neural network (ACRNN) employs an attention layer to score weights of a sequence of frames [8], [9]. As a result, ACRNN can focus on emotion-relevant parts and produce discriminative utterance-level representations for SER. However, standard deep learning algorithms based on CE loss [10] does not consider the impact of uneven label distribution on the training set. That is, the number of the training examples labeled with an emotion is smaller than the number of examples labeled with another emotion. The former label is called a minority class and the latter one is a majority class. Although standard deep learning algorithms obtains a model with a minimum of train error, the resulting model is often biased toward the majority class. Consequently, many examples of

the minority class are misclassified and the overall accuracy of SER is low.

Researchers have proposed methods or algorithms to address the imbalance issue for standard machine learning algorithms, and some of them are also valid for deep learning algorithms [11], [12]. Among these solutions, Bagging [13] is one of widely-used methods for its simplicity and effectiveness. But when Bagging subsamples a training set with replacement to construct bootstrap samples, some training examples may appear more than once in a sample. As a result, duplicate training examples in a bootstrap sample may prevent a good generalization of the base learner. We call the problem ''observation duplication''. Correspondingly, this article devises a Random Evenly Distributed Aggregation (Redagging) method to address the observation duplication problem. Additionally, the Redagging base learners trained on partial data may be prevented from seeing the whole picture captured in the entire data. We call the problem ''lack of whole picture''. This article also proposes an Oversampling-Undersampling aggregation (Augagging) method to deal with the lack of whole picture problem.

The rest of the paper is organized as follows. In Section II, we introduce related work. In Section III, we describe the base classifier ACRNN and Bagging. In Section IV, we introduce our Redagging and Augagging. In Section V, we show our experimental results on an English database of emotional speech IEMOCAP and a German database of emotional speech Emo-DB. Finally, we conclude our study in Section VI.

II. RELATED WORK

In this section, we introduce imbalance issue on deep learning techniques and ensemble learning methods.

A. IMBALANCE ISSUE ON DEEP LEARNING TECHNIQUES

At an early stage, traditional machine learning methods were used for speech emotion recognition. Specifically, after speech signals are transformed into statistical features, speech emotion recognition can be transformed into multi-classification problems [1], [14]–[17]. Recently, with the success in image recognition, CRNNs [5]–[7] have been adopted for SER tasks. Generally, CRNN includes three main parts. Firstly, convolutional layer transforms input features into high-dimension feature representations. Then BiLSTM layer generates a sequence of low-dimension feature representations. Finally, full connected (FC) layer outputs a probability array.

In CRNN, BiLSTM layer assumes that each frame has the same influence on the target emotion. However, some scholars argued that the most influence often comes from a few frames. Then some attention mechanisms are published [8], [9]. An attention model integrated convolutional recurrent neural network assigns each frame with a different weight. With the weights of a sequence of frames, ACRNN can focus on emotion-relevant parts and produce discriminative utterance-level representations for SER. Thus, compared with CRNN, ACRNN can more precisely recognize the target emotion. In 2018, Chen *et al.* [18] proposed to feed Mel-spectrogram, Mel-spectrogram with deltas and Mel-spectrogram with delta-deltas into an attention model integrated CNN. In 2019, Latif *et al.* [19] devised parallel convolution layers with multiple filter widths to achieve good experimental results by directly capturing various contextual information. In 2020, Kwon and Mustaqeem [20] proposed a deep-step convolutional neural network (DSCNN) to improve the accuracy of prediction by focusing on the salient and descriptive features of speech signals.

However, those work does not consider imbalanced speech emotion recognition. Classical deep learning algorithms iteratively update weights of models to reduce CE loss [10]. When the label distribution of a train data set is uneven, the final model easily biases toward the majority class [21], [22]. Hensman and Masko [11] empirically studied the impact of imbalanced data on CNN training. They concluded that oversampling is a viable way to counter the negative impact. Later, Buda *et al.* [12] systematically investigated the impact of class imbalance of CNNs in terms of the classification accuracy. They stated that the effect of class imbalance on classification performance is detrimental and oversampling should be applied to eliminate the imbalance. Zheng *et al.* [23] suggested generating local features, comprehensive information in local data, and global features from speech signals, and then they used an ensemble to perform speech emotion recognition. This article focuses on data-level methods that address class imbalance by changing data distribution. Feature augmentation or algorithm enhancement is beyond the scope of this article.

Ensemble learning methods are one of the most popular methods of addressing class imbalance for its simplicity and easy-migration. So far, there is no systematic research on ensemble learning methods to address the class imbalance of CNN. Actually, Ensemble learning is a good way to counter the class imbalance of CNN for the independence of algorithms or features. This article also proposes two ensemble learning methods to improve the performance of CNN.

B. ENSEMBLE LEARNING METHODS

There are three popular classes of ensemble learning methods: Bagging [13], Boosting [24], and Stacking [25]. Bagging repeatedly subsamples a training set with replacement to construct bootstrap samples, on which a learning algorithm trains a sequence of base learners. After obtaining the base learners, Bagging combines them by majority voting and the most-voted class is predicted. Boosting is in fact a family of algorithms in that there are many variants. In Boosting algorithms, each learner takes into account its previous learner's success. After each training step, the weights are redistributed. Misclassified data increases its weights to emphasize the most difficult cases. In this way, subsequent learners will focus on them during their training. The final learner is derived by weighted majority voting of the base learners. Stacking first generates a number of first-level

individual learners on a training set by employing different learning algorithms. Those individual learners are then combined by a second-level learner which is called as meta learner. Eventually, the meta learner makes the prediction.

Boosting improves the generalization error by reducing the training error. However, a neural network (i.e, ACRNN) may be set up with thousands of parameters. A deep learning based learner can be easily over-trained on the training set and lose its generalization properties. For instance, when we used ACRNN as the base learner, we often saw that the training error of the based learner is near zero, but the generation error is still high. Therefore, Boosting is difficult to improve generalization by reducing the training error of the deep learning based learner. Stacking needs to construct base learners using different learning algorithms. However, the algorithm selection in Stack is not a simple problem. When inappropriate algorithms are selected to train stacking learners, the performance may be not as good as we expect. Hence, this article concentrates on the improvement of Bagging for SER tasks.

Bagging builds heterogeneous base learners by random drawing examples with replacement from a training set. A learning algorithm can generate multiple base learners in a parallel style. But the observation duplication problem of the base learners may cause overfitting [26], [27]. Thus, this article devises Redagging to prevent from observation duplication in bootstrap samples and proves the superiority of the proposed method. However, Bagging-based Redagging utilizes undersampling of the source data set to produce the multiple samples and hence, the learners on these multiple samples may be prevented from seeing the ''whole picture'' captured by the entire data set. So this article also propose Redagging-based Augagging to combine oversampling and Redagging by majority voting for addressing the lack of whole picture problem.

III. BAGGING WITH ACRNN

In this section, we introduce Bagging using ACRNN as the base classifier and observation duplication problem. Firstly, we briefly overview the construction of the network input. Then we describe the architecture of ACRNN. After that, we introduce Bagging using ACRNN as the base classifier. Finally, we analyze the observation duplication problem in Bagging with ACRNN.

A. NETWORK INPUT

To train a model on audio data, we extracted spectral features from the speech signal and then converted them into array. At first, let us define a few notations or functions as follows.

T: number of classes,

S: time (frame) length,

F: number of Mel-filter bank,

$\phi(m, i)$: inserts a new axis into m at position i,

$[v_1, v_2, \ldots, v_c]_i$: join a sequence of arrays $\{v_1, v_2, \ldots, v_c\}$ along an axis at position i,

q_i: the Mel filterbank energies through a Mel filterbank i.

Then, the log-Mel filterbank energies (log-Mels) m_i, the delta features, and the delta-deltas features are produced according to (1), (2), (3).

$$m_i = \log(q_i) \tag{1}$$

where q_i is the Mel filterbank energy through the ith Mel filterbank.

$$m_i{}' = \frac{\sum_{n=1}^{N} n(m_{i+n} - m_{i-n})}{2\sum_{n=1}^{N} n^2} \tag{2}$$

where n is an incremental number and N is the number of the preceding (or following) frames that are calculated.

$$m_i{}'' = \frac{\sum_{n=1}^{N} n(m_{i+n}{}' - m_{i-n}{}')}{2\sum_{n=1}^{N} n^2} \tag{3}$$

With m_i, $m_i{}'$, and $m_i{}''$, $x \in R^{S\times F\times 3}$ is produced according to (4). Then x is used as the input of the aftermentioned ACRNN.

$$x = [\varphi(m_i, 3), \varphi(m_i{}', 3), \varphi(m_i{}'', 3)]_3 \tag{4}$$

B. ACRNN

In the paper, ensembles use an attention model integrated convolutional recurrent neural network (ACRNN) as the base classifier. The architecture of ACRNN is shown in Fig. 1. Firstly, three ConvPool layers transform input feature x into 3-D high-level representations. Secondly, the first full-connected (FC) layer transforms the 3-D representations into 2-D space. Thirdly, BiLSTM layer obtains a sequence of high-level representations $h_s = (\overrightarrow{h_s}, \overleftarrow{h_s})$. Fourthly, the second FC layer calculates weight ratio of each frame α_s according to (5). Fifthly, Fun layer performs the weighted sum of the high level features to get utterance-level features g according to (6). Sixthly, the third and fourth FC layer reduce the number of dimensions. Finally, Softmax layer outputs the probability of T class labels $p = (p_0, p_1, \ldots, p_{T-1})$ where p_t represents the probability of emotion e_t. In convention, the position of the maximal element of p is obtained as the label of x. Note that the three ConvPool layers, the first FC layer, and the third FC layer are all followed by a BatchNorm layer [28] and a LeakyLeRU [29] activation layer. Since the BatchNormalization layer and the LeakyLeRU layer do not change the dimension, they are combined into the preceding layer to save space.

$$\alpha_s = \frac{\exp(W \cdot h_s)}{\sum_{s=1}^{S} \exp(W \cdot h_s)} \tag{5}$$

where W represents network layer parameters.

$$g = \sum_{s=1}^{S} \alpha_s h_s \tag{6}$$

However, some researchers [11], [12] have found that imbalanced training data potentially have a negative impact on classification performance of CNN. In the following, we introduce Bagging to address the class imbalance by making data distribution even on bootstrap samples.

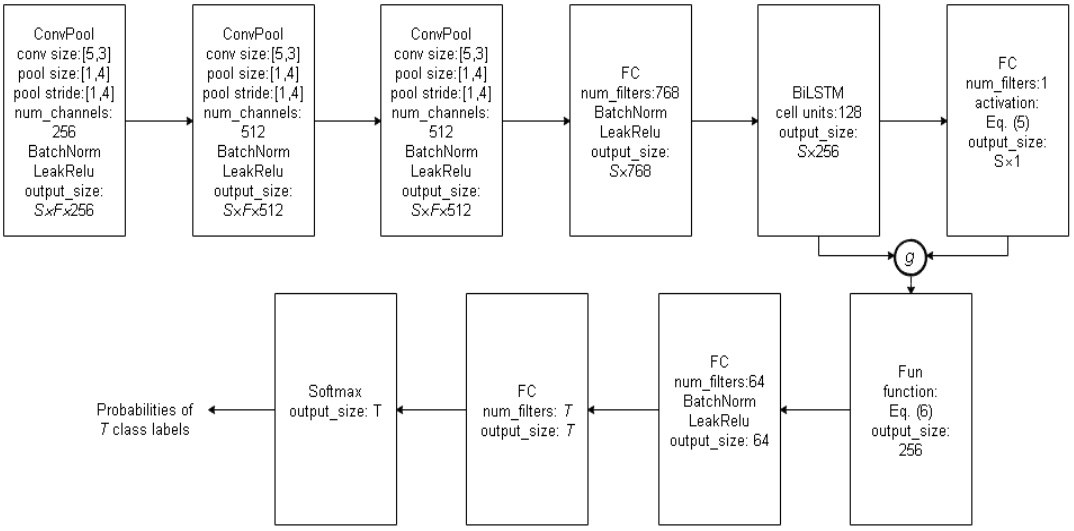


FIGURE 1. An ACRNN with a specific configuration used in our experiments.

C. BAGGING WITH ACRNN

Let X and Y denote the instance space and the set of class labels, respectively, assuming $Y = \{0, 1, \ldots, T-1\}$. Giving A training set $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, D can be divided into $D^* = \{D^0, D^1, \ldots, D^{T-1}\}$, where $y_i = t$ if $(x_i, y_i) \in D^t$. For convenience, suppose $|D^0| \leq |D^1| \leq \cdots \leq |D^{T-1}|$. $|D^t|$ is the number of elements in D^t.

At the k-th round, Bagging subsamples each subset D^t with replacement to generate T subsamples. Then T subsamples are united to construct the k-th bootstrap sample D_k. Next, ACRNN classifier L is trained on the bootstrap sample D_k to obtain the base learner h_k. After the base learners are all ready, the most-voted emotion y is obtained from the outputs of the base learners. The pseudo-code of Bagging is shown in Alg. 1.

D. OBSERVATION DUPLICATION IN BAGGING WITH ACRNN

In Alg. 1, for any $t \in Y$, we have set a constraint $|D_k^t| = |D^0|$. Thus, class labels are evenly distributed in D_k. According to Theorem 1, if $|D^t| \approx |D^0|$, then the probability that an example appears more than once is approximately 0.264.

Theorem 1: At the k-th round, d elements are drawn from D^t to construct D_k^t. For any $x \in X$, if $d \to +\infty$, $|D_k^t| \approx d$, then the probability that draws x at least two times is approximately 0.264 (1 - $\frac{2}{e}$)

Proof of Theorem 1: Let us define some notations as follow.

d_t: D_k^t,

A: an event that draws x at least two times,

$A0$: an event that draws x 0 times,

$A1$: an event that draws x 1 times.

Then the probability that A occurs, P(A), can be obtained by

$$P(A) = 1 - P(A0) - P(A1)$$

Alg. 1 Bagging

Input: Division of training set D^*;
Number of labels T;
ACRNN classifier L;
Number of learning rounds K.
for $k = 1$ to K **do**
  for $t = 0$ to $T - 1$ **do**
    $D_k^t \leftarrow$ observations of drawn from D^t with replacement
  end for
  $D_k \leftarrow D_k^0 \cup D_k^1 \cup \cdots \cup D_k^{T-1}$
  $h_k \leftarrow L(D_k)$
end for
$H(x) \leftarrow \text{argmax}((h_1(x) + h_2(x) + \cdots + h_K(x))/K)$ ▷ argmax($(p_0, p_1, \ldots, p_{T-1})$) returns the position t of the maximal element p_t
Output: hypothesis H

Since

$$P(A0) = (1 - \frac{1}{d_t})^d,$$
$$P(A1) = d\ \frac{1}{d_t}(1 - \frac{1}{d_t})^{d-1} = \frac{d}{d_t - 1}(1 - \frac{1}{d_t})^d$$

Hence,

$$P(A) = 1 - (1 - \frac{1}{d_t})^d - \frac{d}{d_t - 1}(1 - \frac{1}{d_t})^d$$

Thus, if $d \to +\infty$, $|D_k^t| \approx d$, we have

$$P(A) \approx 1 - \frac{1}{e} - \frac{1}{e} = 1 - \frac{2}{e}$$

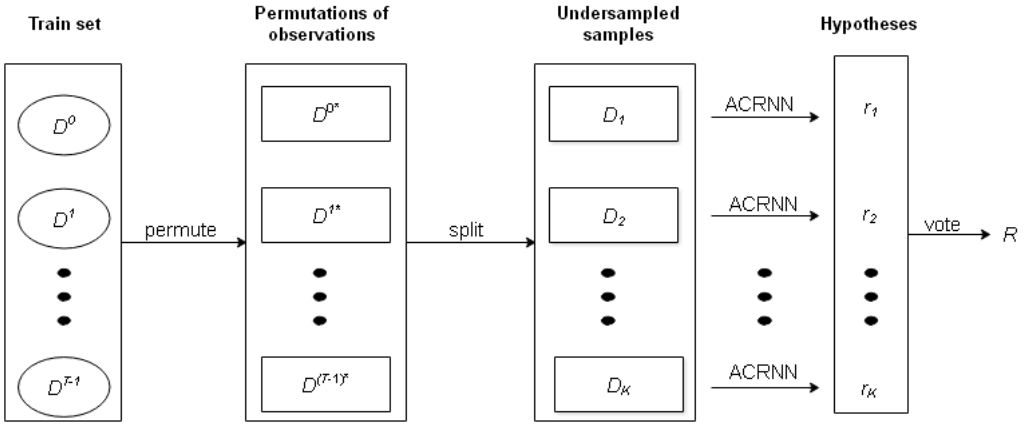


FIGURE 2. The flowchart of Redagging.

Hence, the observation duplication problem occurs on bootstrap samples with a certain probability. Overfitting may happen owing to excessive duplicate examples in a bootstrap sample. In Section V (Experiments), experimental results will prove the impact of observation duplication. Additionally, heterogeneity is also critical. The homogeneous base learner cannot yield a good ensemble either. In the following section, our two methods take both the observation duplication and heterogeneity problems into consideration.

IV. OUR METHODS

Firstly, Redagging is devised and proved to deal with the observation duplication problem based on Bagging. Based on Redagging, Augagging is proposed to combine oversampling and Redagging by majority voting to deal with the lack of whole picture problem.

A. REDAGGING

Our method Redagging includes three main steps. Firstly, Redagging randomly generates a number of different permutations of $D_t(D^{t,1}, D^{t,2}, \ldots, D^{t,I_t})$ by the Mason rotation algorithm [30] and unions them to get D^{t*} according to (7), (8). Then it distributesD^{t*} to K bootstrap samples $D_1, D_2, \ldots, D_K$ in order, where D_k is defined in (9). Finally, using ACRNN as the base classifier, Redagging trains on bootstrap sample D_k to yield the base learner h_k and combines an ensemble H according to (10). Because Redagging randomly generates permutations and then divides the heterogeneous permutations into bootstrap samples, it addresses both the heterogeneity and observation duplication problems. The data flow is shown in Fig. 2. Again, K is the number of bootstrap samples and $|D^{T-1}|$ represents the number of examples of the majority class.

$$I_t = \left\lceil \frac{K|D^0|}{|D^t|} \right\rceil \tag{7}$$

$$D^{t*} = D^{t,1} \cup D^{t,2} \cup, \ldots, \cup D^{t,I_t} \tag{8}$$

$$D_k = D_k^0 \cup D_k^1 \cup, \ldots, \cup D_k^{T-1} \tag{9}$$

where D_k^t represents the subset of examples labelled with t at the kth round.

$$R(x) = \arg\max((r_1(x) + r_2(x) + \cdots + r_K(x))) \tag{10}$$

Theorem 2: For any $t \in Y$, Redagging distributes D^{t*} to K bootstrap samples in order. If $|D^t|\%|D^0| = 0$, for any $x \in D_k^t$, the number of occurrences of x in D_k^t is 1. % returns the remainder left over when one operand is divided by a second operand.

Proof of Theorem 2: Let $\Lambda = \frac{|D^t|}{|D^0|}$. Given D^t, we have permutations

$$\begin{aligned} D^{t,1} &= D_1^t \cup D_2^t \cup, \ldots, \cup D_\Lambda^t \\ D^{t,2} &= D_{\Lambda+1}^t \cup D_{\Lambda+2}^t \cup, \ldots, \cup D_{2\Lambda}^t \\ &\cdots \\ D^{t,I_t} &= D_{(I_t-1)\Lambda+1}^t \cup D_{(I_t-1)\Lambda+2}^t \cup, \ldots, \cup D_{I_t\Lambda}^t \end{aligned}$$

If $|D^t|\%|D^0| = 0$ is satisfied, for $\forall k \in \{1, 2, \ldots, K\}, \exists! i \in \{1, 2, \ldots, I_t\}, D_k^t \subseteq D^{t,i}$. Since $D^{t,i}$ is a permutation of D^t, for any x in D_k^t, x appears only once in D_k^t.

Actually, observation duplication only occurs in a small fraction of bootstrap samples even if $|D^t|\%|D^0| = 0$ is not satisfied. Compared with Bagging, Redagging constructs bootstrap samples with less observation duplications. Additionally, various random seeds introduce the heterogeneity of the bootstrap samples. Thus, Redagging base learners are heterogeneous and address overfitting for observation duplication. Moreover, we will make further analysis of the performance impact of observation duplication in Section V (Experiments).

Another straightforward method to address observation duplication is Sampling Without Replacement (SWOR), which samples without replacement on the level of the individual bootstrap samples. But SWOR changes the probability distribution. The probability that each example appears in K samples is not the same. As a result, some examples may never appear if K is small. Considering the computation cost and model complexity, a high K is not practical

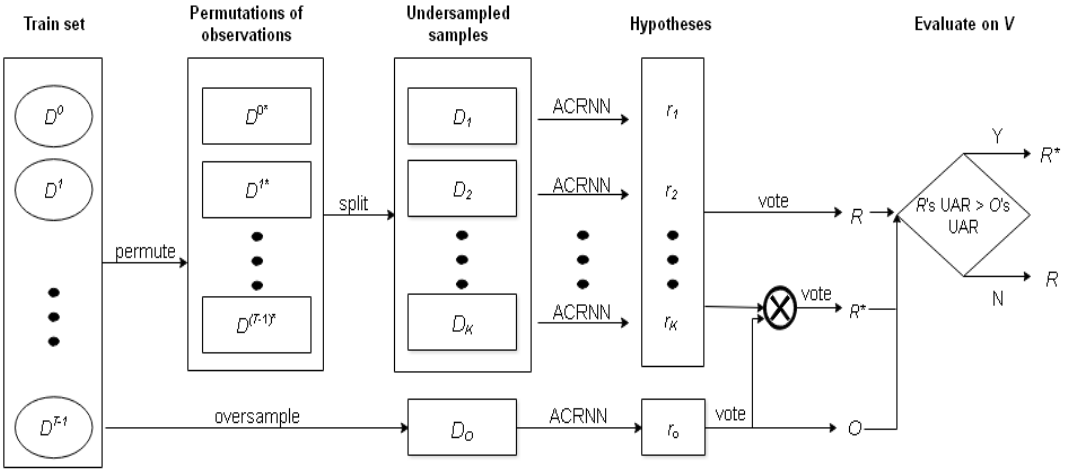


FIGURE 3. The flowchart of Augagging.

in many real-world applications. In contrast, Redagging not only reduce the probability of observation duplication but also maintains the same probability of observations. Therefore, Redagging is superior to SWOR by design.

Redagging addresses the observation duplication problem, whereas the Redagging base learners lack of the whole picture captured by the entire data. In the following, Augagging is proposed to address the observation duplication and lack of whole picture problems by combining oversampling and Redagging by majority voting.

B. AUGAGGING

From resampling perspective, a bootstrap sample is generated using an undersampled scheme. Thus, the knowledge of a Redagging base learner may be incomplete. Instead, oversampling (OS) trains a learner on the oversampled data and acquires the complete knowledge of the source train set. Thus, the combination of Redagging and Oversampling can yield a stronger learner than Redagging ensemble if the OS learner does not introduce overfitting. Augagging just employs the combination of Redagging and Oversampling to address the lack of whole picture problem in Redagging. The data flow of Augagging is shown in Fig. 3.

Different from Redagging, Augagging constructs an extra OS learner r_o learnt from the oversampled training set D_o according to (12). Then $r_1, r_2, \ldots, r_K$, and r_o together construct the undersampling-oversampling ensemble R^* according to (11). If the OS classifier O based on (12) performs better than the Redagging ensemble R on the validation set V in term of unweighted average recall (UAR), R^* participates in majority voting; otherwise R does.

$$R^*(x) = \arg\max((r_1(x)+r_2(x)+\ldots+r_K(x)+r_o(x))) \quad (11)$$

$$O(x) = \arg\max((r_o(x))) \quad (12)$$

The Redagging base learners are trained on heterogeneous samples, whereas the OS learner can see the ''whole picture'' of the original data. When oversampling does not cause overfitting, Augagging combines both the Redagging base learners and OS learner by majority voting. Hence, theoretically Augagging is superior to Redagging.

V. EXPERIMENTS

In this section, we conduct speaker-independent experiments to investigate the performance of comparable methods on both the Interactive Emotional Dyadic Motion Capture database (IEMOCAP) [31] and the Berlin Emotional Database Emo-DB [32].

The experiments run on a H3C G4900 server. The server is configured with a Tesla V100 independent 32G GPU graphics card, and installed Python 3.7.0, CUDA 10.01 acceleration platform and cuDNN 7.4.2.24 deep learning acceleration platform.

The log-Mels are extracted by the Python speech features toolkit [33] with the window size of 25 ms, a 10 ms shift, and N set to 2. The training log-Mels, validation log-Mels and test log-Mels are all normalized by the global mean and the standard deviation of the training set. The NumPy [34] array is used to store features and perform matrix operations. Each layer of ACRNN is implemented by Keras [35]. The parameter of the model is optimized by cross entropy objective function [10] with a mini-batch of 150 examples, using the Adam optimizer [36]. The initial learning rate is set to 10^{-3}.

On the IEMOCAP and Emo-DB databases, ACRNN is the benchmark method. Oversampling randomly clones the examples of the minority class until class distribution is even. Bagging, SWOR, Redagging and Augagging use ACRNN as the base classifier. With different parameter initializations, we can obtain a wide range of results. Thus, we repeat each evaluation for 10 times with different random seeds and report the average to get more reliable results.

Since IEMOCAP and Emo-DB both contain 10 speakers, we employ a 10-fold cross-validation technique in our

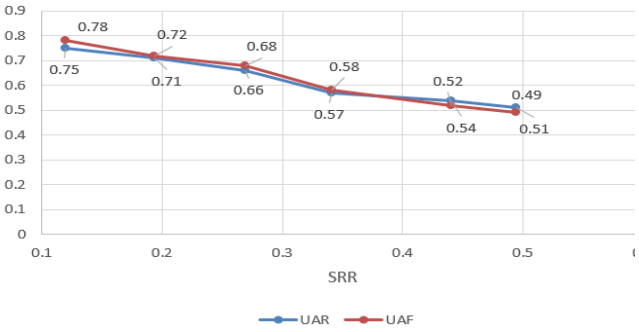


FIGURE 4. UARs and UAFs of Bagging-Redagging aggregation using ACRNN as the base classifier with various SRRs.

evaluations. Specifically, for each evaluation, 8 speakers are selected as the training data and one speaker is select as the validation data, while the remaining one speaker is used as the test data. That is, we have 10 samples in all for each database.

We save the optimal model which achieves the highest UAR. But when evaluating imbalance learning methods, (unweighted average F1-score) UAF is more comprehensive and objective in that F_1-score is a weighted average of recall and precision. Thus, we measure the performance of imbalance learning methods in terms of both UAR and UAF.

A. PERFORMANCE IMPACT OF OBSERVATION DUPLICATION

To measure observation duplication, we define Samples Repetitive Rate (SRR) as defined in (13).

$$SRR = \frac{1}{K}\sum_{k=1}^{K}\frac{\sum_{(x_i,y_i)\in D_k, g((x_i,y_i),D_k)>1} g((x_i,y_i),D_k)}{|D_k|} \quad (13)$$

where $g(a, D_k)$ is the number of occurrences of a in D_k.

The right term of (13) represents the degree of observation duplication of bootstrap samples. A higher SRR means more observation duplications in samples. Then using ACRNN as the base classifier, we conduct experiments on an Emo-DB data set to measure the impact of observation duplication.

To produce various SRRs on the data set, we choose some classes to apply Redagging and the rest classes to apply Bagging to construct different training data. That is, we first choose all five classes of emotions to apply Redagging and obtain the lowest SRR (SRR = 0.12). And then, we choose the first four classes to apply Redagging and the last one to apply Bagging and obtain the second lowest SRR (SRR = 0.19). We repeat the Bagging-Redagging aggregation three more times. Finally, we choose all the five classes to apply Bagging and obtained the highest SRR (SRR = 0.49). Our experimental results under the four different SRRs are shown in Fig. 4.

From Fig. 4, we can see that when SRR = 0.12, we obtained the highest UAR (0.75) and UAF (0.78). With the increment of the SRR value, the accuracy of speech emotion recognition in terms of UAR and UAF decreases consistently. When SRR = 0.49, we obtained the lowest UAR and UAF. This indicates that Redagging can perform better than Bagging by reducing observation duplications. In the following subsections, we compare Redagging and Augagging with other imbalance learning methods.

TABLE 1. Sample distribution of the ten IEMOCAP samples.

No.	TR (# of H,A,S,N)	VS (# of H,A,S,N)	TS (# of H,A,S,N)
I1	381,385,839,1375	28,60,74,169	22,44,103,150
I2	381,385,839,1375	22,44,103,150	28,60,74,169
I3	345,449,854,1330	40,13,97,150	46,27,65,214
I4	345,449,854,1330	46,27,65,214	40,13,97,150
I5	341,344,697,1385	55,38,181,143	35,107,138,166
I6	341,344,697,1385	35,107,138,166	55,38,181,143
I7	377,345,875,1424	16,83,71,76	38,61,70,194
I8	377,345,875,1424	38,61,70,194	16,83,71,76
I9	280,433,799,1262	72,35,113,217	79,22,104,215
I10	280,433,799,1262	79,22,104,215	72,35,113,217

TABLE 2. Average UARs and UAFs of imbalance learning methods using ACRNN as the base classifier on the ten IEMOCAP samples.

Metric	Base	OS	Bag-ging	SWOR	Red-agging	Aug-agging
UAR	0.60	0.61	0.61	0.62	0.64	0.65
UAF	0.54	0.53	0.55	0.56	0.58	0.59

B. EXPERIMENTS ON THE DATABASE IEMOCAP

IEMOCAP consists of five sessions, each session being completed by a pair of speakers (female and male) in recitation lines and improvisation scenarios. The average sample length is 4.5 seconds and the sampling rate is 16 kHz. We conduct experiments on four emotion samples: happy, angry, sad, and neutral. Each task employs 10-fold cross-validation technology. That is, we have 10 samples in all. The sample distribution of the ten samples are described in Table 1, where H = happy, A = angry, S = sad, N = neutral.

We conduct experiments on the ten IEMOCAP samples. $K = 5$ when Bagging, Redagging, or Augagging is evaluated. Then the ACRNN models of the benchmark method, Oversampling (OS), Bagging base learners, Sampling without replacement (SWOR) base learners, Redagging base learners, and Augagging base learners are all trained in 10 epochs. The experimental results are shown in Table 2. When ACRNN is used as the base classifier, Augagging achieves the highest average UAR (0.65), followed by Redagging (0.64) and SWOR (0.62). The average UARs of Oversampling (0.61) and Bagging (0.61) are both higher than the base method (0.60). In terms of UAF, Augagging (0.59) is the best, Redagging (0.58) and SWOR (0.56) are the second best, followed by Bagging (0.55). Oversampling (0.53) even performs worse than the benchmark method (0.54).

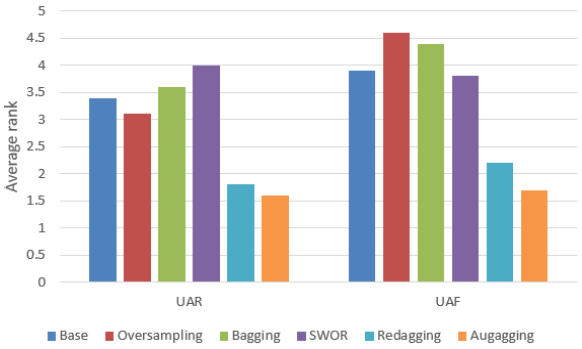


FIGURE 5. The average ranks of the five imbalance learning methods using ACRNN as the base classifier on the ten IEMOCAP samples (the lower, the better).

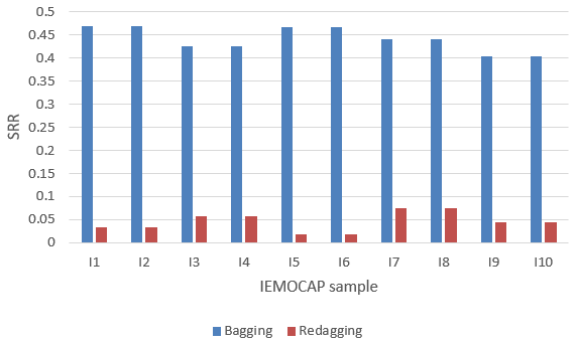


FIGURE 6. SRR results of Bagging and Redagging on the ten IEMOCAP samples.

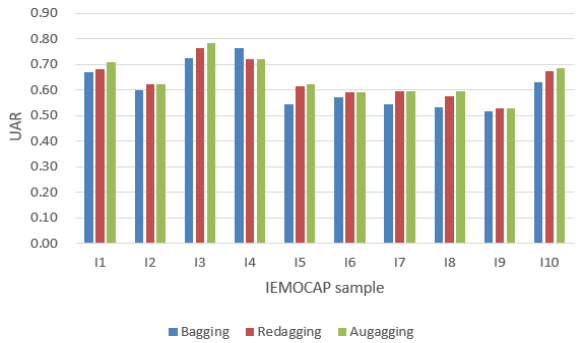


FIGURE 7. A comparison of Bagging, Redagging, and Augagging using ACRNN as the base classifier on the 10 IEMOCAP samples, in terms of UAR.

TABLE 3. Sample distribution of the ten Emo-DB samples.

No.	TR (# of A,B,H,S,N)	VS (# of A,B,H,S,N)	TS (# of A,B,H,S,N)
E1	(101,66,53,46,58)	(12,10,11,9,10)	(14,5,7,7,11)
E2	(102,67,56,49,60)	(13,4,4,4,9)	(12,11,10,9,10)
E3	(104,69,63,55,66)	(10,8,4,3,4)	(13,4,4,4,9)
E4	(106,65,59,52,66)	(11,8,8,7,9)	(10,8,4,3,4)
E5	(104,68,61,51,66)	(12,5,2,4,4)	(11,8,8,7,9)
E6	(103,66,59,53,66)	(12,10,10,5,9)	(12,5,2,4,4)
E7	(99,63,53,47,63)	(16,8,8,10,7)	(12,10,10,5,9)
E8	(98,64,57,48,61)	(13,9,6,4,11)	(16,8,8,10,7)
E9	(100,58,54,49,63)	(14,14,11,9,5)	(13,9,6,4,11)
E10	(99,62,53,46,63)	(14,5,7,7,11)	(14,14,11,9,5)

Additionally, we rank these imbalance learning methods on each sample. The average rank of each method is shown in Fig. 5. From Fig. 5, we can see that Augagging maintains the leading position (the lower, the better) in terms of UAR and UAF. Redagging is the runner-up. SWOR comes higher than ACRNN in terms of UAF but degenerate ACRNN in terms of UAR. The average ranks of Oversampling and Bagging even falls behind the average rank of the benchmark method in terms of UAF. On the whole, Fig. 5 indicates that on the ten EMOCAP samples, Augagging and Redagging can improves performance of ACRNN, but Oversampling, Bagging and SWOR can't.

Furthermore, we conduct a comparison analysis on the ten IEMOCAP samples in terms of SRR and UAR. The SRR results and UAR results are shown in Fig. 6 and Fig 7, respectively. In Fig. 6, the average SRR of Redagging is about 0.05 while that of the SRRs of Bagging reaches 0.44 on average. This indicates that Bagging introduces more observation duplications than Redagging. Moreover, as shown in Fig. 7, Redagging achieves higher UAR than Bagging on 9 out of 10 samples. That indicates that Redagging performs better than Bagging with high probability. Moreover, Augagging outperforms not only Bagging on 9 out of 10 samples but also Redagging on 5 samples (I1, I3, I5, I8, I10). Augagging constructs an ensemble combining both the Redagging base learners and the OS learner. The experimental results on the ten IEMOCAP samples demonstrate the superiority of Augagging.

C. EXPERIMENTS ON THE DATABASE EMO-DB

Emo-DB consists of 535 sentences from 10 professional actors, covering 7 emotions (neutral, fear, joy, anger, sadness, disgust and boredom). Specifically, the Emo-DB data set has five classes of emotions (containing 101 angry utterances, 66 bored utterances, 53 joyful utterances, 46 sad utterances, and 58 neutral utterances). The original audio is sampled at 44.1 kHz and later downsampled to 16 kHz. The number of anxious examples and the number of disgusting examples are no more than two on some validation sets so that the experiments results fluctuate on the validation set. Thus, we only conduct experiments on five emotions (anger, boredom, joy, sadness and neutral). Each task employs the 10-fold cross-validation technology. The sample distribution of the 10 samples are described in Table 3, where A = anger, B = boredom, H = joy, S = sadness, and N = neutral.

We conduct experiments on the ten Emo-DB samples. $K = 4$ when Bagging, Redagging, or Augagging is evaluated. Then the ACRNN models of the benchmark method, Oversampling (OS), Bagging base learners, Redagging base learners, and Augagging base learners are all trained in 10 epochs. When ACRNN is used as the base classifier, the experimental

TABLE 4. Average UARs and UAFs of imbalance learning methods using ACRNN as the base classifier on the ten Emo-DB samples.

Metric	Base	OS	Bag-ging	SWOR	Red-agging	Aug-agging
UAR	0.66	0.68	0.68	0.68	0.69	0.71
UAF	0.61	0.62	0.64	0.63	0.65	0.66

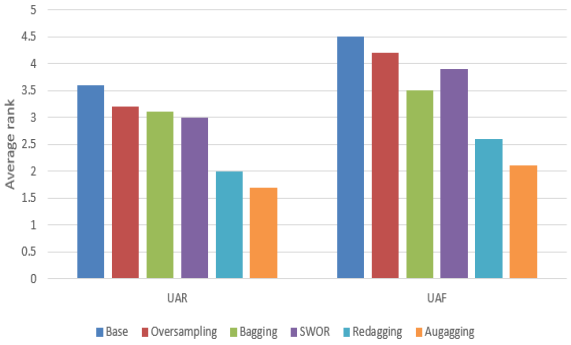


FIGURE 8. The average ranks of the six imbalance learning methods using ACRNN as the base classifier on the ten Emo-DB samples (the lower, the better).

results are shown in Table 4. From Table 4, in terms of UAR, Augagging achieves the average highest UAR (0.71), followed by Redagging (0.69). The average UARs of SWOR, Bagging and Oversampling are the same (0.68), which are higher than that of the base method (0.66). In terms of UAF, Augagging (0.66) is the best, followed by Redagging (0.65), followed by Bagging (0.64), followed by SWOR (0.63). Oversampling (0.62) performs better than the base method (0.61).

Additionally, we rank these imbalance learning methods on each sample. The average rank of each method is shown in Fig. 8. Fig. 8 shows the average rank of Augagging is the best (the lower, the better) in terms of both UAR and UAF. Redagging ranks the second. SWOR ranks above Bagging in terms of UAR but does behind Bagging in terms of UAF. The average ranks of Bagging and OS (oversampling) are better than the rank of the base method. On the whole, Fig. 8 indicates that on the ten Emo-DB samples, these five imbalance learning methods all improves performance of ACRNN, whereas Augagging and Redagging perform the best.

Furthermore, we conduct a comparison analysis of Bagging, Redagging, and Augagging on the ten Emo-DB samples in terms of SRR and UAR. The SRR results and UAR results are shown in Fig. 9 and Fig 10, respectively. In Fig. 9, the mean of the SRRs of Redagging is 0.14 while that of the SRRs of Bagging reaches 0.52. This indicates that Bagging introduces much more observation duplications than Redagging in bootstrap samples. Moreover, as shown in Fig. 10, Redagging achieves higher average UARs than Bagging on 8 out of 10 samples. Augagging outperforms not only Bagging on 9 out of 10 samples but also Redagging on 4 samples (I3, I4, I6, I10), These two figures indicate that when using ACRNN as the base classifier, Redagging and Augagging

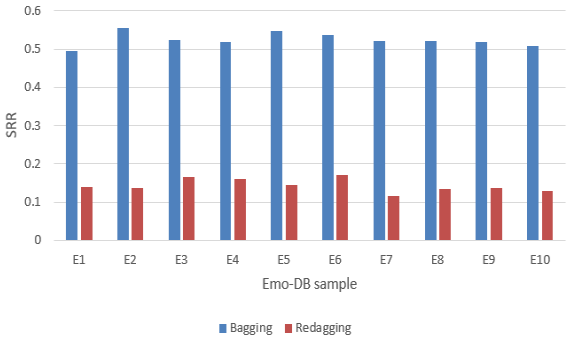


FIGURE 9. SRR results of Bagging and Redagging on the ten Emo-DB samples.

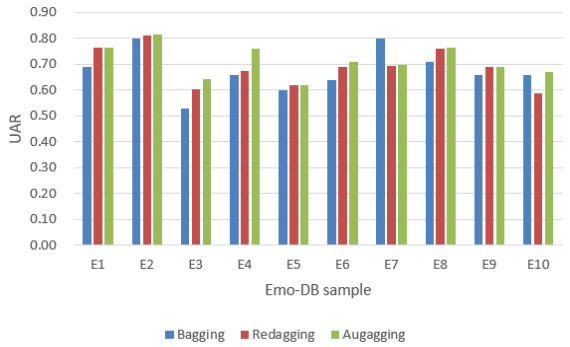


FIGURE 10. A comparison of Bagging, Redagging, and Augagging using ACRNN as the base classifier on the 10 Emo-DB samples, in terms of UAR.

perform better than Bagging by reducing observation duplications of bootstrap samples. Moreover, Augagging combines the Redagging base learners and the OS learner by majority voting. The experimental results on the ten Emo-DB samples demonstrate the superiority of Augagging.

VI. CONCLUSION

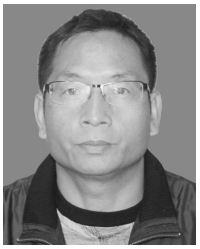
This article studied the observation duplication and the lack of whole picture problems for ensemble learning with ACRNN in imbalanced speech emotion recognition. Redagging is proposed and proved to address the observation duplication problem in bootstrap samples by generating bootstrap samples from permutations of observations. Furthermore, the proposed Augagging deals with the lack of whole picture problem by making the OS learner participate in majority voting. Finally, extensive experiments on IEMOCAP and Emo-DB are given to demonstrate the superiority of the proposed methods. For the future research, our work can be extended to positive Markovian jumping neural networks [37], [38].

REFERENCES

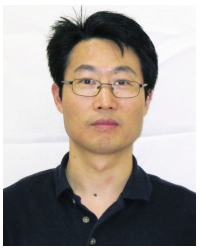
[1] N. Yang, R. Muraleedharan, J. Kohl, I. Demirkol, W. Heinzelman, and M. Sturge-Apple, "Speech-based emotion classification using multiclass SVM with hybrid kernel and thresholding fusion," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Miami, FL, USA, Dec. 2012, pp. 455–460.

[2] M. S. Hossain, "Patient state recognition system for healthcare using speech and facial expressions," *J. Med. Syst.*, vol. 40, no. 12, pp. 1–8, Dec. 2016.

[3] N. Kamaruddin, A. W. A. Rahman, and A. N. R. Shah, "Measuring customer satisfaction through speech using valence-arousal approach," in *Proc. 6th Int. Conf. Inf. Commun. Technol. Muslim World (ICT4M)*, Jakarta, Indonesia, Nov. 2016, pp. 298–303.
[4] M. Bojanić, V. Delić, and A. Karpov, "Call redistribution for a call center based on speech emotion recognition," *Appl. Sci.*, vol. 10, pp. 4653–4670, Jul. 2020.
[5] J. Lee and I. Tashev, "High-level feature representation using recurrent neural network for speech emotion recognition," in *Proc. INTERSPEECH*, Dresden, Germany, Sep. 2015, pp. 1537–1540.
[6] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 6645–6649.
[7] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller, and S. Zafeiriou, "Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Shanghai, China, Mar. 2016, pp. 5200–5204.
[8] X. Chen, A. Ragni, X. Liu, and M. J. F. Gales, "Investigating bidirectional recurrent neural network language models for speech recognition," in *Proc. Interspeech*, Stockholm, Sweden, Aug. 2017, pp. 269–273.
[9] C.-W. Huang and S. S. Narayanan, "Deep convolutional recurrent neural network with attention mechanism for robust speech emotion recognition," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Hong Kong, Jul. 2017, pp. 583–588.
[10] J. He, I. Pedroza, and Q. Liu, "MetaNet: A boosting-inspired deep learning image classification ensemble technique," in *Proc. IPCV*, Las Vegas, NV, USA, Aug. 2019, pp. 51–54.
[11] D. Masko and P. Hensman, "The impact of imbalanced training data for convolutional neural networks," M.S. thesis, KTH Roy. Inst. Technol., Stockholm, Sweden, May 2015.
[12] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Netw.*, vol. 106, pp. 249–259, Oct. 2018.
[13] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.
[14] C. M. Lee, S. Yildirim, M. Bulut, A. Kazemzadeh, C. Busso, Z. Deng, S. Lee, and S. Narayanan, "Emotion recognition based on phoneme classes," in *Proc. ICSLP*, Jeju Island, South Korea, Oct. 2004, pp. 889–892.
[15] M. W. Bhatti, Y. Wang, and L. Guan, "A neural network approach for human emotion recognition in speech," in *Proc. IEEE Int. Symp. Circuits Syst.*, Vancouver, BC, Canada, May 2004, pp. 181–184.
[16] A. Hassan and R. Damper, "Multi-class and hierarchical SVMs for emotion recognition," in *Proc. Interspeech*. Makuhari, Japan, Sep. 2010, pp. 2354–2357.
[17] G. Yuan, T. S. Lim, W. K. Juan, H. M.-H. Ringo, and Q. Li, "A GMM based 2-stage architecture for multi-subject emotion recognition using physiological responses," in *Proc. 1st Augmented Hum. Int. Conf. (AH)*, Megève, France, 2010, pp. 1–6.
[18] M. Chen, X. He, J. Yang, and H. Zhang, "3-D convolutional recurrent neural networks with attention model for speech emotion recognition," *IEEE Signal Process. Lett.*, vol. 25, no. 10, pp. 1440–1444, Oct. 2018.
[19] S. Latif, R. Rana, S. Khalifa, R. Jurdak, and J. Epps, "Direct modelling of speech emotion from raw speech," in *Proc. Interspeech*, Graz, Austria, Sep. 2019, pp. 3920–3924.
[20] S. Kwon, "A CNN-assisted enhanced audio signal processing for speech emotion recognition," *Sensors*, vol. 20, pp. 183–197, Dec. 2020.
[21] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.
[22] C. Etienne, G. Fidanza, A. Petrovskii, L. Devillers, and B. Schmauch, "CNN+LSTM architecture for speech emotion recognition with data augmentation," 2018, *arXiv:1802.05630*. [Online]. Available: http://arxiv.org/abs/1802.05630
[23] C. Zheng, C. Wang, and N. Jia, "An ensemble model for multi-level speech emotion recognition," *Appl. Sci.*, vol. 10, pp. 205–224, Dec. 2019.
[24] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.
[25] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, Jan. 1992.
[26] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 20–29, Jun. 2004.
[27] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
[28] F. Vučković, I. Ugrina, G. Lauc, and Y. Aulchenko, "Normalization and batch correction methods for high-throughput glycomics," in *Proc. Int. Symp. Glycoconjugates (GLYCO)*, Split, Croatia, Sep. 2015, p. 242.
[29] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, Atlanta, GA, USA, Jun. 2013, p. 3.
[30] Y. Li and M. Zhang, "A Software/Hardware parallel uniform random number generation framework," in *Proc. 10th Int. Conf. Commun. Softw. Netw. (ICCSN)*, Chengdu, China, Jul. 2018, pp. 471–474.
[31] C. Busso, M. Bulut, 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, Nov. 2008.
[32] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, and B. Weiss, "A database of German emotional speech," in *Proc. 9th Eur. Conf. Speech Commun. Technol.*, Lisbon, Portugal, Sep. 2005, pp. 1517–1520.
[33] *Python Speech Features*. Accessed: Apr. 15, 2019. [Online]. Available: http://www.github.com/jameslyons/python_speech_features
[34] E. Bisong, *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, Berlin, Germany: Springer, 2019, pp. 91–113.
[35] A. Gulli and S. Pal, *Deep learning With Keras Olton*. Birmingham, U.K.: Packt, 2017.
[36] G. Franchini and L. Zanni, "On the steplenght selection in stochastic gradient methods," in *Proc. Int. Conf. Numer. Computations, Theory Algorithms (NUMTA)*, Crotone, Italy, Jun. 2019, pp. 186–197.
[37] C. Ren, S. He, X. Luan, F. Liu, and H. R. Karimi, "Finite-time L_2-gain asynchronous control for continuous-time positive hidden Markov jump systems via T-S fuzzy model approach," *IEEE Trans. Cybern.*, early access, Jun. 10, 2020.
[38] C. Ren and S. He, "Finite-time stabilization for positive Markovian jumping neural networks," *Appl. Math. Comput.*, vol. 365, pp. 124631–124642, Jan. 2020.



XUSHENG AI received the B.S. degree in computer science and technology from Zhengzhou University, Zhengzhou, China, in 1997, and the M.S. and Ph.D. degrees in computer science and technology from Soochow University, Suzhou, in 2003 and 2016, respectively. His research interests include machine learning, speech emotion recognition, and image recognition.



VICTOR S. SHENG (Senior Member, IEEE) received the master's degree in computer science from the University of New Brunswick, Canada, in 2003, and the Ph.D. degree in computer science from Western University, London, ON, Canada, in 2007.

He is currently an Associate Professor of computer science with Texas Tech University and the Founding Director of the Data Analytics Laboratory (DAL). His research interests include data mining, machine learning, crowdsourcing, and related applications in business, industry, medical informatics, and software engineering. He was an Associate Professor with the University of Central Arkansas and an Associate Research Scientist and NSERC Postdoctoral Fellow with the Information Systems, Stern Business School, New York University. He is a Lifetime Member of ACM. He organized several conferences. He is an Editorial Board Member for several journals, a SPC and PC member for many international conferences, and a Reviewer of more than twenty international journals. He received several best paper awards.

WEI FANG received the M.Sc. and Ph.D. degrees in computer science from Soochow University. He is currently an Associate Professor with the Department of Computer Science, Nanjing University of Information Science and Technology, China, and the State Key Laboratory for Novel Software Technology, Nanjing University. His research interests include artificial intelligence, big data, deep learning, and meteorological information processing. He is a PC member for a number of international conferences and a Reviewer for several international journals.

CHARLES X. LING has been a Faculty Member for over 25 years. He is a Professor of computer science and the Distinguished Professor of science with Western University. His research interests include (big) data analytics, machine learning, and data mining applications. He has published over 150 peer-reviewed research articles. Recently, he applies his research to mobile health, and created glucoguide, which won the First Prize in Diabetes Research Day with the Schulich School of Medicine and Dentistry of Western. He is the CEO and Founder of GlucoGuide Corporation.

CHUNHUA LI received the B.S. degree in detection technology and instrument from the Wuhan University of Technology, Wuhan, China, in 2000, the M.S. degree in pattern recognition and intelligent system from the Huazhong University of Science and Technology, Wuhan, in 2003, and the Ph.D. degree in computer science and technology from Soochow University, Suzhou, China, in 2017. Her research interests include machine learning, knowledge engineering, deep learning, and crowdsourcing.