Intoxicated Speech Detection

Tran, Thi Ngoc Yen & Lin, Li

Universität Stuttgart, Institut für maschinelle Sprachverarbeitung

st181852@stud.uni-stuttgart.de, st185873@stud.uni-stuttgart.de

Abstract

Alcohol consumption causes numerous servere social, psychologial, and health problems including an increase in traffic accidents. Successful identification of alcohol consumption is crucial in addressing and preventing drinking and driving, as well as other safety and security issues. Using the Alcohol Language Corpus, the project's task is designed as a binary classification between sober and drunk speech. This project attempts to use different feature kinds from raw speech and different machine learning/deep learning models to detect intoxicated speech. For each model, diverse techniques are applied such as oversampling, permutation importance for random forest, focal loss for convolutional neural network, memory-efficient tricks for pretrained model to achieve the best performance. All codes can be found at our Github page [1].

Index Terms: speech classification, speech features, computational paralinguistics

1. Introduction

A blood alcohol concentration test is normally used to measure alcohol amount in blood. Such methods like that are invasive and time-consuming. They are also normally done only after the accidents have happened and therefore can not act like a prevention method. Speech-based detection is believed to be non-invasive and potential since slurred speech is one noticeable characteristic of drunk people [1]. Moreover, most modern cars nowadays are equipped with speech-input gadgets.

The INTERSPEECH 2011 Speaker State Challenge [2] focused on the important application domain of security and safety: the computational analysis of intoxication and sleepiness in speech. It introcuded the task of binary intoxicated speech classification and the Alcohol Language Corpus (ALC). Intoxicated speech detection is one of the main area of computational paralinguistic speech research. It has many potential real-world applications such as driver safety, law enforcement, healthcare, etc.

There is a variety of works done in intoxication detection. Since the challenge was released in 2011, most of the works back then depended heavily on the paralinguistic feature engineering. Before the advent of deep learning (DL) and end-to-end learning approaches, feature engineering was a standard practice in various speech processing tasks.

The challenge winner is [3]. They achieved an unweighted average recall performance of 70.54% on the test set while the baseline model accuracy is 65.9%. The used hierarchical acoustic features extracted from both openSMILE and Praat, iterative speaker normalization, a set of Gaussian Mixture Model (GMM) supervectors, an Eigencannel supervector, and score-level fusion.

[4] used openSMILE features combined linguistic features. The information-gain ratio was applied to reduce the feature set. Suport Vector Machines (SVMs) were used as classifiers. [5] coped with the class imbalance by Asymmetric simple Partial Least Squares while using SVMs as claasifier combined with the calibration and multiple fusion methods. [6] compared 5 different feature types with different ways of modeling. The main classifiers of most of the works submitted to the challenge were SVMs or supervector GMM.

The real turning point for DL came in 2012 with the introduction of the deep convolutional neural network (CNN), which outperformed traditional computer vision methods. Since then, DL has evolved and demonstrated impressive results in various domains, including speech and natural language processing. Not only CNN but also other architectures like long short-term memory (LSTM), cross-attention, and self-attention have gained widespread adoption in many applications. Additionally, the introduction of self-supervised learning and transfer learning has contributed significantly to the success of deep neural networks.

With the advancement of DL, the task is revisited with deep neural networks such as in [7] [8] [9] to name just a few. [7] proposed a batch level speaker normalization method with a residual neural network (ResNet) and achieved a similar improvement as the baseline method with speaker normalization (67.7%).

This project aims to discover the use of both feature engineering and advanced DL in solving the task. Project workflow is as seen in Figure 1.

The structure of the report is as follows. Section 2 introduces the challenging dataset, feature set and feature extraction tool. A detailed overview of models and methods used are presented in section 3. Section 4 summarises the experimental results and the analysis on speech styles. Finally, section 5 concludes and points out future work directions.

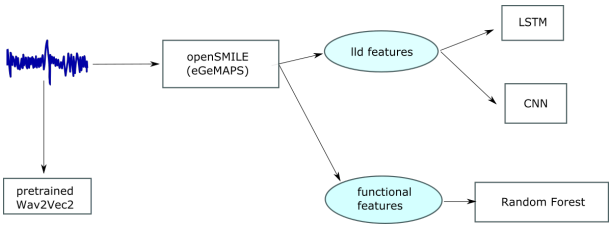

Figure 1: *Project workflow.*

[1] https://github.tik.uni-stuttgart.de/st185873/TeamLab_LiYen.git

Table 1: *Overview of the dataset*

Dataset	NAL	AL	Total (read, spontaneous, command-control speech)
Train	7717	3764	11481 (7544, 1894, 2043)
Dev	945	490	1435 (918, 87, 87)
Eval	950	486	1436 (963, 225, 248)

2. Dataset and feature extraction

2.1. Dataset

ALC is the speech corpus of intoxicated and sober speech recorded from 162 female and male German speakers in automotive environment [10] [11]. The ALC dataset used in the challenge was a gender-balanced dataset of 154 speakers (77 male and 77 female speakers). The dataset used in our project contains speech from complete 162 speakers (85 male and 77 female).

Apart from having many speakers, the dataset also consists of three different styles of speech: read speech, spontaneous speech, and command-control speech. Each style has different characteristics, e.g., speech rate is supposed to be much slower and less variable in read or command-control speech than in spontaneous speech for a specific speaker [3]. Moreover, the duration of each speech also varies, ranging from 0.5 to over 60 seconds.

Read speech involves promps where speaker are asked to read phone numbers, credit card numbers, tax ID numbers, car ID numbers, words, twisted words, and sentences. Spontaneous speech entails talking about an experience or telling a story based on a given picture. Command and control speech involves giving instructions to the car based on the given situation. The analysis of the speech style revealed that read speech was recognized best (67.7%) while spontaneous speech was recognized worst (57.2%) [12].

The overview of the dadaset is presented in Table 1. The proportion of read, spontaneous and command-control speech is about the same in each dataset and each class, around 0.66, 0.17, and 0.17.

2.2. Feature extraction

We use openSMILE [2] toolkit to extract the eGeMAPS [13] features. The Low-level descriptors (LLD) features contain the **minimalistic parameter set** of 18 features:

- Frequency related parameters: *Pitch*, *jitter*, *formant 1, 2, and 3 frequency*, *formant 1*
- Energy/Amplitude related parameters: *Shimmer*, *loudness*, *Harmonics-to-Noise Ratio* (HNR)
- Spectral (balance) parameters: *Alpha ratio*, *Hammarberg index*, *spectral slope 0–500 Hz and 500–1500 Hz*, *formant 1, 2, and 3 relative energy*, *harmonic difference H1–A3*.

and the 7 **extended parameter set** features:

- Spectral (balance/shape/dynamics) parameters: *MFCC 1–4*, *spectral flux*
- Frequency related parameters: *Formant 2–3 bandwidth*

We also added the *MFCC 5-20*. So in total, 41 LLD features.

[2] https://www.audeering.com/research/opensmile/

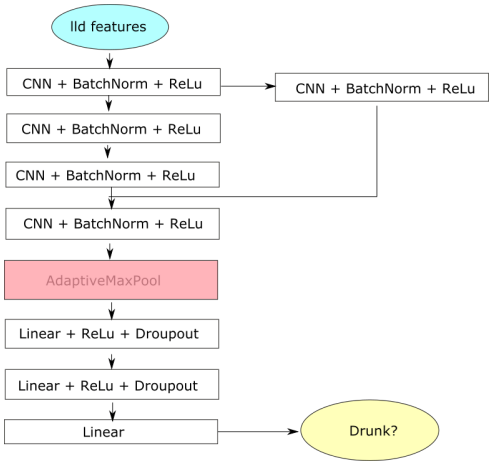

Figure 2: *CNN architecture.*

As for functionals, a total of 152 features were extracted including *arithmetic mean* and *coefficient of variation* of all LLD features; *20-th, 50-th, and 80-th percentile, the range of 20-th to 80-th percentile*; *temporal* and some other features.

2.3. Data imbalanced

The sober and drunk speech data is not balanced as shown in Table 1. Ratio of all sample data between Non-alcoholic and Alcoholic is close to 0.66 : 0.33.

If trained and tested with unbalanced data, the model may have a bias towards the majority class. Therefore, in addition to using accuracy, we analyse the model with two other evaluation metrics: precision and recall.

The simplest way to deal with data imbalance is the resampling method, which includes undersampling and oversampling [14]. Undersampling means randomly deleting half of the data in the sober class to balance the amount of data with the drunk class. However, this results in the loss of one-third of the sample size, which may lead to the loss of important information. Oversampling means doubling the data samples of the drunk class by replication, which may increase the possibility of overfitting as well as increase the training time. After considering the situation of our dataset and referring to other people's experiments [14], we choose the method of oversampling.

3. Methods

We experimented a number of methods and models to make use of all features extracted. As shown in Figure 1, LLD features are passed to a CNN and LSTM while functionals features are input to random forest model. The raw waveform is passed to pretrained Wav2Vec2, which includes a feature extractor and an audio classification model.

3.1. CNN

The CNN model is believed to handle position-independent features well, making it suitable for classification tasks involving paralinguistic features where word order is not crucial. In fact, CNN is widely used for many speech classification tasks, such as emotion recognition [15] or accent classification [16].

The CNN architecture is shown in Figure 2. The adaptive max pooling layer helps to put the length-varied input back to

uniform size before passing to the linear layer, which requires input to be the same size. Dropout for both linear layer is 0.6.

To cope with the imbalanced dataset, we tried the focal loss [17], which deals with class imbalance by modifying the standard cross-entropy loss, reducing the impact of the loss assigned to well-classified examples. The authors in [17] stated that in practice they used 2 hyperparameters γ and α and found $\gamma = 2$ to perform best for their task. We used only one hyperparameter γ and found $\gamma = 1.5$ to perform best for our task.

We trained the model for 25 epochs, learning rate 1e-4, batchsize 16, and optimizer AdamW. This setting is the same for both cross entropy and focal loss.

3.2. BiLSTM

Similarly to CNN, LSTM is also commonly used for speech classification tasks such as in [18] or [19]. In this experiment, we experimented with a bidirectional LSTM (BiLSTM) model.

Firstly, data preprocessing is performed on the time series data, including max-min normalization for each feature data in the samples. After that, the tail of each sample is padded with zeros so that all samples have the same time length. During the training process, only the non-padding values are trained.

In the BiLSTM model, the speech time series data is trained twice from forward and backward [20]. In the final model, the number of LSTM hidden layers in each direction is set to 2, and the number of neurons in each layer is 64. Then the output of the hidden layer of the last state in the forward direction, and the output of the hidden layer of the first state in the backward direction are taken as inputs to the fully connected layer (the number of nodes is 128), and the outputs are two values of the two classes. The loss function used is cross entropy and the model is trained for 180 epochs using the AdamW [3] optimizer.

The hyperparameters in the model are continuously tuned to improve the model performance. The remaining hyperparameter settings for the best model are: the batch size is 64; the parameters of the hidden layer are uniformly initialized from -0.1 to 0.1; dropout for the second LSTM layer is 0.5, and that for the fully-connected layer is 0.4; the learning rate drops 1.5 or twice every 8 or 10 or 15 epochs, from 4e-3 to 6.4-6.

For data imbalance, the overampling dataset was used for training as a comparison, using the same parameters and running 250 epochs.

3.3. Random forest

Random Forest is a Bootstrap Aggregation ensemble learning method [21]. Compared to neural networks, random forests are faster to train, can make selections of features, and are also less prone to overfitting problems.

The experiment is divided into three parts: with the original unbalanced data, with the oversampling data and feature selection with the oversampling data. Among them, the number of decision trees in the forest is 141, the criterion is entropy.

There are two main parameters for important feature selection in random forests: decrease in impurity (MDI) and permutation importance (also known as mean decrease accuracy (MDA))[4]. Since almost every functional feature has multiple unique values and MDI is affected by high-cardinality features,

[3] https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html

[4] https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html

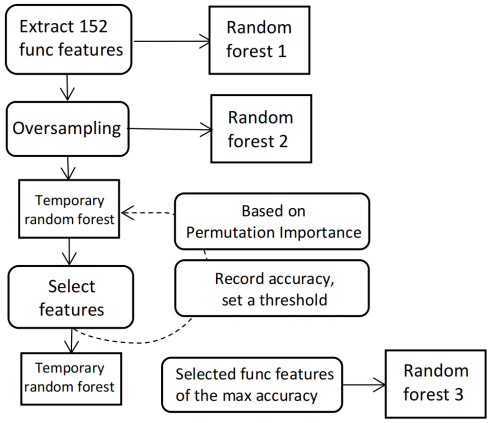

Figure 3: *Random forest steps.*

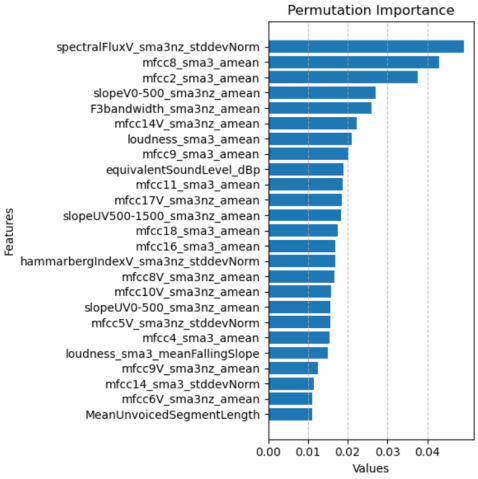

Figure 4: *25 selected features*

the important feature selection in this experiment is based on permutation importance.

The selection steps are shown in Figure 3. We observed that the selection of important features does not improve the accuracy of the model on the test set much, and the accuracy decreases after removing some features. However, the accuracy may increase by continuing to remove features. So here the threshold is set to 0.1, which means if the accuracy is 10% less than the maximum accuracy, the loop is stopped. Where feature selection is done by removing features with permutation importance less than 0, and if all the features have scores greater than 0 and the loop does not exit, the one with the smallest score is removed. Finally, 25 important features are selected when the accuracy is maximum, as shown in Figure 4.

3.4. Pretrained Wav2Vec2

Wav2Vec2 [22] is a framework for self-supervised learning of representations from raw audio data. It learns latent representation of speech mainly by listening to unlabelled data. Even though originally pretrained on speech recognition task, Wav2Vec2 is widely used as a transfer learning framework for any speech classification tasks. The model consists of a feature extractor and an auto model for audio classification. We froze

the feature extractor and fine-tuned the classifier on the train dataset.

The pretrained model used is the multi-lingual versions of Wav2Vec2, XLSR (cross-lingual speech representations) [23]. More specific, it is wav2vec2-large-xlsr-53 [5], a large model pretrained in 53 languages including German.

Some tricks are applied to reduce computer cost such as using mixed precision, gradient accumulation with batchsize 1 but gradient acculumation steps 32. In addition, we feed the model with only 10% of max input length. The model was fine-tuned for 10 epochs with learning rate 1e-4. It was fine-tuned only once, no tuning is done.

4. Results

The models' performances are assessed on the evaluation set. Unweighted accuracy, recall, and precision for each class are used as evaluation metrics. An analysis of speech style recognition rates is also carried out.

4.1. Classification results

Table 2: *Classification results on the evaluation set (%)*

Models	**Accuracy**	**Recall**		**Precision**	
		sober	drunk	sober	drunk
For lld features					
CNN (CE loss)	77.72	86.95	59.67	80.82	70.04
CNN (F loss)	76.74	78.52	**73.25**	85.16	63.57
BiLSTM	71.59	85.09	46.08	74.88	62.06
BiLSTM*	68.25	64.96	**74.45**	82.76	52.93
For functional features					
Random Forest	73.12	96.70	28.57	71.89	82.08
Random Forest*	76.39	91.91	47.08	76.64	75.48
RF(select features)*	**78.97**	90.63	56.94	79.91	76.28
For raw waveform					
Pretrained Wav2Vec2	**83.84**	95.79	60.49	82.58	88.02

Without * refers to the model being trained with the original data.
With * refers to the model being trained with the oversampling data.

Low recall plus high precision means that the model does not detect all the data in the class, but the model predicts more credible results for the class. High recall plus low precision means that the model can detect all the data in the class well, but the predicted results are less credible[6]. The rounding of recall and precision for different classes can be based on practical needs.

The use of oversampling data made the accuracy of BiLSTM decrease, but the accuracy of random forest increase. From the BiLSTM and Random forest models, we can see the consistent conclusions: The drunk class recall of the model trained on data imbalance is low. With using oversampling data, the recall of the drunk class is significantly improved, but the recall of the sober class is decreased. At the same time, the use of oversampling improves the precision of the sober class and reduces the precision of the drunk.

[5] https://huggingface.co/facebook/wav2vec2-large-xlsr-53

[6] https://medium.com/@klintcho/explaining-precision-and-recall-c770eb9c69e9

Feature selection is performed on random forest, the number of features is reduced from 152 to 25, accuracy is improved, and recall and precision are also improved by a small margin.

Focal loss does help to improve recall for drunk class significantly and balanced the recall for both classes. However, the tuning of the hyperparameter plays a crucial role. Different γ lead to very varied results.

The pretrained Wav2Vec2 performs best in general and is believed to perform even better since it has not even been tuned due to compute cost issues. Better fine-tuning methods rather than naive fine-tuning are discussed in section 5.

4.2. Speech style analysis

Table 3: *Speech style recognition rate (%)*

Models	**read**	**spontaneous**	**command_ control**
For lld features			
CNN (CE loss)	78.59	71.11	89.65
CNN (F loss)	77.15	73.33	78.22
BiLSTM	72.10	69.60	71.43
BiLSTM*	70.64	60.79	65.87
For functional features			
Random Forest	73.88	69.60	73.41
Random Forest*	76.80	74.45	76.59
RF(select features)*	78.68	75.77	82.94
For raw waveform			
Pretrained Wav2vec	84.32	81.33	84.27

Without * refers to the model being trained with the original data.
With * refers to the model being trained with the oversampling data.

As illustrated in Table 3, the spontaneous style has the lowest recognition rate, while the other two styles are relatively similar. After feature selection for random forest, compared with the read style, the accuracy of the command-control style improved more, probably because the number of data in this style is relatively small.

5. Conclusions

We used four models for the binary classification task on the ALC dataset. Oversampling and focal loss were used to deal with the problem of unbalanced data. It is proved that they both helps increase the minority class's recall. Moreover, feature selection was performed on random forests, and the evaluation performance was instead improved after reducing the number of features.

In terms of improvements in future work, the BiLSTM model can be used with more complex structures, such as adding attention structures; and with more prevention of overfitting. Random forests could try other methods of feature selection, not just simply based on permutation importance.

The pretrained model is demonstrated to be the most robust one, no hyperparameter tuning, no feature engineering, and end-to-end. However, its bottleneck is the computer cost due to the huge ammount of parameters. Besides memory-efficient tricks mentioned above, one can use head or adapter fine-tuning combined with other loss functions [24] to more efficiently and effectively fine-tune the model.

6. References

[1] A. B. S, S. R. Shetty, S. Srinivas, V. Mantri, and V. R. Badri Prasad, "Intoxication detection using audio," in *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*, 2023, pp. 1–6.

[2] B. Schuller, A. Batliner, S. Steidl, F. Schiel, and J. Krajewski, "The interspeech 2011 speaker state challenge," in *Proc. INTERSPEECH 2011, Florence, Italy*, 2011.

[3] D. Bone, M. P. Black, M. Li, A. Metallinou, S. Lee, and S. Narayanan, "Intoxicated speech detection by fusion of speaker normalized hierarchical features and gmm supervectors," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[4] S. Ultes, A. Schmitt, and W. Minker, "Attention, sobriety checkpoint! can humans determine by means of voice, if someone is drunk… and can automatic classifiers compete?" in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[5] D.-Y. Huang, S. S. Ge, and Z. Zhang, "Speaker state classification based on fusion of asymmetric simpls and support vector machines," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[6] T. Bocklet, K. Riedhammer, and E. Nöth, "Drink and speak: On the automatic classification of alcohol intoxication by acoustic, prosodic and text-based features," in *Twelfth Annual Conference of the International Speech Communication Association*. Citeseer, 2011.

[7] W. Wang, H. Wu, and M. Li, "Deep neural networks with batch speaker normalization for intoxicated speech detection," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2019, pp. 1323–1327.

[8] B. Sertolli, Z. Ren, B. W. Schuller, and N. Cummins, "Representation transfer learning from deep end-to-end speech recognition networks for the classification of health states from speech," *Computer Speech & Language*, vol. 68, p. 101204, 2021.

[9] Z. Zhang, F. Weninger, M. Wöllmer, J. Han, and B. Schuller, "Towards intoxicated speech recognition," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 1555–1559.

[10] F. Schiel, C. Heinrich, S. Barfüsser, and T. Gilg, "Alc: Alcohol language corpus." in *LREC*, 2008.

[11] F. Schiel, C. Heinrich, and S. Barfüsser, "Alcohol language corpus: the first public corpus of alcoholized german speech," *Language resources and evaluation*, vol. 46, pp. 503–521, 2012.

[12] B. Baumeister and F. Schiel, "Human perception of alcoholic intoxication in speech." in *INTERSPEECH*, 2013, pp. 1419–1423.

[13] F. Eyben, K. R. Scherer, B. W. Schuller, J. Sundberg, E. André, C. Busso, L. Y. Devillers, J. Epps, P. Laukka, S. S. Narayanan *et al.*, "The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing," *IEEE transactions on affective computing*, vol. 7, no. 2, pp. 190–202, 2015.

[14] M. Roweida, R. Jumanah, and A. Malak, "Machine learning with oversampling and undersampling techniques: Overview study and experimental results," *2020 11th International Conference on Information and Communication Systems (ICICS)*, 2020.

[15] M. Neumann and N. T. Vu, "Improving speech emotion recognition with unsupervised representation learning on unlabeled speech," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7390–7394.

[16] M. Lesnichaia, V. Mikhailava, N. Bogach, I. Lezhenin, J. Blake, and E. Pyshkin, "Classification of accented english using cnn model trained on amplitude mel-spectrograms," *Proc. Interspeech 2022*, pp. 3669–3673, 2022.

[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[18] Y. Xie, R. Liang, Z. Liang, C. Huang, C. Zou, and B. Schuller, "Speech emotion classification using attention-based lstm," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1675–1685, 2019.

[19] J. Wang, M. Xue, R. Culhane, E. Diao, J. Ding, and V. Tarokh, "Speech emotion recognition with dual-sequence lstm architecture," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6474–6478.

[20] S.-N. Sima, T. Neda, and S. N. Akbar, "The performance of lstm and bilstm in forecasting time series," *2019 IEEE International Conference on Big Data (Big Data)*, 2019.

[21] K. K. Nawas, M. K. Barik, and A. N. Khan, "Speaker recognition using random forest," in *ITM Web of Conferences*, vol. 37. EDP Sciences, 2021, p. 01022.

[22] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[23] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," *arXiv preprint arXiv:2006.13979*, 2020.

[24] F. Lux, C.-Y. Chen, and N. T. Vu, "Combining contrastive and non-contrastive losses for fine-tuning pretrained models in speech analysis," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 876–883.