Detecting Sarcasm from Speech Using Transformer-Based and Traditional Methods

Gábor Gosztolya[1,2(✉)], Mercedes Kiss-Vetráb[1], Márton Makrai[3,4], and László Balázs[4]

[1] Institute of Informatics, University of Szeged, Szeged, Hungary
ggabor@inf.u-szeged.hu
[2] HUN-REN–SZTE Research Group on Artificial Intelligence, Szeged, Hungary
[3] ELTE Research Centre for Linguistics, Budapest, Hungary
[4] HUN-REN Research Centre for Natural Sciences, Budapest, Hungary

Abstract. The invention of transformer-based architectures and self-supervised learning have revolutionized automatic speech recognition. Such end-to-end models mostly replaced the previous pipelines in paralinguistic speech processing as well (such as emotion recognition, conflict intensity estimation, and sarcasm detection), which workflows build on hand-crafted features and traditional classifiers like Support Vector Machines (SVMs) and Random Forests. In the paralinguistic area, the amount of training data is often scarce; still, it is rarely investigated whether current end-to-end models really outperform the traditional workflows even under such circumstances. In this study we focus on the sarcasm detection task in particular, and compare these two general approaches: we fine-tune two WavLM models, and also train SVMs on two hand-crafted feature sets. The results, surprisingly, show that the transformer-based architectures do not have a cutting edge, delivering roughly the same performance as the traditional pipelines do. Additionally, we show that even better results can be obtained by combining the two techniques, with statistically significant improvements in most metrics. These experimental results indicate that for such low-resource tasks, it might be worth keeping the traditional methods, which can provide the same level of performance with smaller computational footprints.

Keywords: Sarcasm detection · Computational paralinguistics · WavLM · openSMILE

1 Introduction

Detecting sarcasm is a particularly challenging task in natural language processing and speech processing. The difficulty comes partly from the contradiction between the intent of the speaker and the literal meaning of the uttered words [9], so it is often not enough to rely purely on lexical or semantic information. Luckily, sarcasm is frequently accompanied by specific acoustic cues in

A. Karpov and B. Gerazov (Eds.): SPECOM 2026, LNAI 16935, pp. 3–13, 2027.
https://doi.org/10.1007/978-3-032-37870-5_1

prosody, intonation, and speech tempo [14]. Owing to this, acoustic approaches have the potential to effectively detect the sarcastic intent from the speech signal. To build an acoustic-based sarcasm detector, one can straightforwardly follow the recent developments in deep learning-based speech processing, and utilize some transformer-based model. End-to-end architectures such as wav2vec 2.0 [1], HuBERT [11], and WavLM [4] represent the current state-of-the-art in deep learning. Their potential is further enhanced by the availability of publicly accessible transformer-based models that have been pre-trained on massive audio corpora—sometimes comprising millions of hours of speech—and can be readily fine-tuned for downstream tasks.

Before the transformer era, deep learning methods were significantly constrained by the amount of training data. Therefore, the most suitable toolset in the paralinguistic area was a more traditional classifier such as a Support Vector Machine (SVM, [16]) or Random Forests [2], operating over general features like ComParE functionals [17], x-vectors [20], or some other form of embedding [19]. Due to the new architectures and self-supervised pre-training, deep methods became competitive in low-resource settings as well, reinforced by the attractiveness of an end-to-end setup over the more complicated traditional pipelines. Still, the latter might be more robust and efficient when the amount of training data is limited, while offering more lightweight solutions in terms of model size, memory consumption, and prediction speed [6,18].

In this study, we focus on the acoustic modality in sarcasm identification. Our main research goal is to compare a modern, self-supervised deep learning model with a pipeline built from lightweight, hand-crafted features and an SVM classifier. Our experiments are performed on the MUStARD++ dataset [15], containing samples of sarcastic and non-sarcastic speech.

2 The Mustard++ Database

In our experiments, we used the MUStARD++ database [15]. This is a multimodal corpus, consisting of video and acoustic information extended with manual transcripts. The recordings contain excerpts of dialogues from various TV shows, mostly *The Big Bang Theory* and *Friends*. The database is primarily used to develop and evaluate techniques to automatically detect sarcasm and emotions. For these tasks, besides the sarcasm labels, there are annotations for sarcasm type and detailed emotional attributes. However, since the focus of our study is sarcasm detection, we did not utilize these labels in our experiments, and adhered to the binary sarcasm annotations.

To use a speaker-independent split, which is essential in automatic speech processing experiments, we followed the work of Dong et al. [5]. They used the recordings taken from the *Friends* TV show (354 clips in total) as the test set, and the remaining clips (from the TV shows *The Big Bang Theory*, *Silicon Valley*, and some other shows; 848 clips in total) for model training in a 5-fold cross-validation (CV) manner. This leads to a training set of only 68 min and a test set of 25 min.

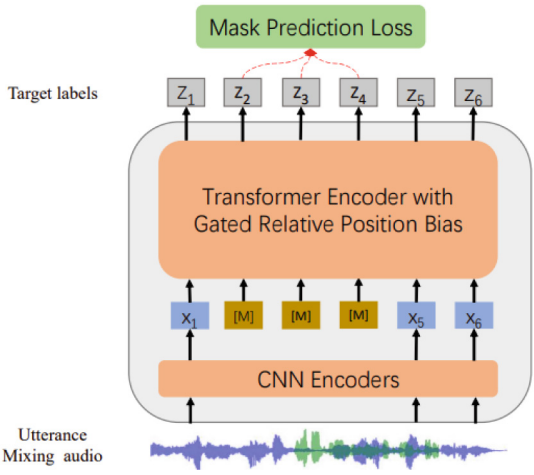

Fig. 1. A demonstration of the WavLM architecture, from the work of Chen et al. [4].

Note that this split slightly altered the originally balanced distribution of the sarcasm class labels. While the full corpus contains 1202 clips, 601 sarcastic and 601 non-sarcastic, in this setup the training set contains 450 sarcastic and 398 non-sarcastic clips, while the test set is composed of 151 and 203 recordings, sarcastic and non-sarcastic, respectively.

3 Models

To address our research question, we experimented with two general approaches: transformer-based acoustic models, and hand-crafted acoustic features with an SVM classifier.

3.1 Transformer-Based Acoustic Models

As the transformer-based encoder, we used WavLM [4], which has a CNN feature preprocessing block. The input waveform is first encoded using seven convolutional layers; then the output of the last layer is passed to the transformer encoder. The next transformer block contains several architectural improvements over previous approaches (e.g. that of HuBERT [11]), like an improved self-attention mechanism and a new relative position embedding. The architecture of the network is illustrated in Fig. 1. The publicly available models were pre-trained with a self-supervised methodology to learn a universal speech representation from a huge amount of unlabeled speech data. During this masked pre-training phase, further tricks were used like the simulation of noisy speech with multiple speakers and various background noise [4].

In our experiments we used two model architectures, called WavLM Base and WavLM Large. Both variants share the convolutional block, but the first one has 12 hidden layers with 768 neurons each in the transformer block (totaling 94M weights), while the latter has 24 transformer layers with 1024 neurons each (containing 317M weights overall). The WavLM Base model was pre-trained in 400k steps on LibriSpeech (960 h) using the label generated by clustering the sixth transformer layer's output from the first iteration HuBERT Base model. The WavLM Large models were pre-trained for 700k steps on 94k hours of diverse data using the labels generated by clustering the ninth transformer layer's output of the released second iteration HuBERT Base model [4].

3.2 SVM Classifier with Hand-Crafted Features

To verify the performance of a traditional setup, we employed Support Vector machines on well-known, hand-crafted utterance-level attributes.

eGeMAPS Features. The first feature set was the v02 variant of eGeMAPS (extended Geneva Minimalistic Acoustic Parameter Set). It contains frequency-related frame-level features (jitter, pitch, formant frequencies), energy and amplitude-related parameters (e.g.: shimmer, loudness, HNR), spectral (harmonic differences, formants' relative energy, spectral slope), and cepstral (MFCC 1–4 $\Delta/\Delta\Delta$) characteristics. As functionals, different combinations of the arithmetic mean and the coefficient of variation were applied to the frame-level attributes, giving a total of 88 utterance-level features [8]. This feature set was calculated with the `openSMILE` tool [7], using the `eGeMAPSv02` profile.

ComParE Functionals. This feature set is based on a brute-force combination of frame-level audio features with functionals and the subsequent removal of zero-information features. It includes energy, spectral (MFCC 1–40, rolloff, flux, centroid, bandwidth, zero-crossing rate), voicing-related (F0, probability, period), and signal quality measures (jitter, shimmer, HNR), plus their first- and second-order derivatives (i.e. $\Delta/\Delta\Delta$). These frame-wise values are then summarized using several utterance-level functionals, including extremes (max/min), moments (mean, standard deviation, skewness, kurtosis), percentiles (p1-p99) and temporal measures (amplitude/loudness peaks, voicing-related durations), resulting in 6373 utterance-level features overall [17]. This feature set was also calculated by the `openSMILE` tool [7], with the `ComParE_2016` profile.

4 Experimental Setup

4.1 Evaluation

We applied evaluation metrics that are standard in computational paralinguistics. Besides classification accuracy (i.e. the ratio of correctly identified recordings), to handle the slightly unbalanced class distribution, we also utilized

Unweighted Average Recall (UAR, calculated as the mean of the class-wise recall values) and Macro-F1 (the mean of the class-wise F-measure scores). Lastly, we used the Area-Under-the-ROC-Curve (AUC) score of the positive (sarcastic) class.

Based on our previous experience, we find UAR and AUC the most reliable evaluation metrics. However, since Macro-F1 is a widespread metric in recent paralinguistic studies (see e.g. [10,21]), we felt it necessary to include these scores as well.

4.2 Model Training

Following the work of Dong et al., we used five-fold cross-validation over the 848 recordings of the training set for hyper-parameter setting, and the clips belonging to the TV show *Friends* as the test set [5]. Unfortunately, since the nature of the two approaches tested is completely different, the training procedures had large differences as well.

Training the WavLM Models. The WavLM architecture was developed with speech recognition in mind, thus it has one output for each frame. To suit the utterance-level classification of the sarcasm detection task, we added an average pooling layer over the time axis after the last layer of the transformer block, resulting in 768 and 1024 activations for WavLM Base and WavLM Large respectively. The classification layer consists of two neurons corresponding to sarcastic and non-sarcastic.

We used a learning rate of 1e-4 for the last layer, and 1e-5 for the weights of the transformer block. Following standard practice, the weights of the convolutional block were frozen. The learning rate was scheduled via the AdamW method; training was carried out for at most 25 epochs. For checkpoint selection, we experimented with maximizing classification accuracy, the mean of the recall scores, and Macro-F1 (i.e. the unweighted mean of the F1 values for the two classes). These values were calculated after each epoch on the development set of the actual fold. The weights corresponding to the epoch of the highest metric value were used during inference on the test set. We report the scores measured on the full training set (i.e. after concatenating the network outputs for the five DNN models on the corresponding development sets), and the mean performance values on the test set.

DNN training is a stochastic procedure. Even if most of the weights come from the same pre-training process (as in our case), fine-tuning is stochastic due to the random initialization of the weights of the output layer, and the randomized order of the examples. Owing to this, we trained five WavLM configurations that differed only in the random seed chosen. We report the mean metric values measured for the five predictions.

Training of SVM Models. For the SVM model, to make parameter setting simpler, we used a linear kernel. The complexity (i.e. C) hyperparameter was

Table 1. The performance scores of the WavLM models.

Subset	Model	Checkpoint selection	Evaluation Metrics			
			Acc.	UAR	M. F1	AUC
Cross-validation	WavLM Base	Accuracy	67.1%	66.6%	66.6%	0.699
		UAR	67.3%	66.8%	66.8%	0.699
		Macro-F1	67.2%	66.8%	66.8%	0.698
	WavLM Large	Accuracy	71.1%	70.8%	70.8%	0.746
		UAR	70.8%	70.6%	70.6%	0.744
		Macro-F1	71.0%	70.8%	70.8%	0.745
Test	WavLM Base	Accuracy	68.2%	68.8%	67.9%	0.738
		UAR	68.4%	69.2%	68.1%	0.752
		Macro-F1	68.4%	68.9%	67.9%	0.746
	WavLM Large	Accuracy	72.6%	74.4%	72.6%	0.809
		UAR	73.3%	74.8%	73.3%	0.812
		Macro-F1	73.0%	74.7%	72.9%	0.810

chosen from the values 10^{-5}, 10^{-4}, ... 10^{0}, 10^{1}, based on the maximal AUC score of the positive class in the five-fold cross-validation process. After choosing the best hyperparameter, we trained a final SVM model on the whole training set, which was then evaluated on the test set.

Since all the components of this pipeline are deterministic, there was no need to repeat these experiments with different random seeds.

5 Results

Table 1 shows the results obtained using the **WavLM** models. First, we can observe that the metric used for checkpoint selection does not really affect the performance score. Indeed, in most cases, the procedure selected exactly the same checkpoints (i.e. epochs) regardless of the actual metric. Overall, it seems that UAR and Macro-F1 led to slightly better scores than maximal classification accuracy, although the difference seems insignificant.

Regarding the two model variations, WavLM Large performed noticeably better than WavLM Base. The accuracy, UAR and Macro-F1 values were higher with the latter architecture by 2–5% (absolute) in almost all the cases. There is an even larger difference in the AUC scores, where the Large model brought a 0.050 (absolute) improvement over the Base variant. This, in our view, indicates that the larger architecture is more useful even when the training data is quite limited in amount (in our case, only about 80% of the 68 min training set due to 5-fold cross-validation).

Table 2. The performance scores of the SVM-based pipelines.

Subset	Features	Evaluation Metrics			
		Acc.	UAR	M. F1	AUC
Cross-validation	eGeMAPS	62.0%	61.4%	61.3%	0.665
	ComParE functionals	68.0%	67.6%	67.6%	0.729
Test	eGeMAPS	73.2%	71.4%	71.8%	0.810
	ComParE functionals	74.3%	74.1%	73.9%	0.808

Table 2 shows the results obtained for the SVM-based pipelines. Here the performance difference between the two feature sets is perhaps the most apparent observation: ComParE functionals markedly outperformed eGeMAPS, especially in cross-validation. This is understandable, though, given that eGeMAPS consists only of 88 attributes. On the test set, however, the difference becomes smaller, and when we focus on the AUC metric, it completely vanishes.

Comparing the metrics obtained by the SVM to those of WavLM (i.e. those depicted in Table 1), we see that the deep models are more effective in cross-validation, which holds especially for the WavLM Large architecture. However, on the test set, the scores are quite similar. In particular, WavLM Large produced the highest UAR values, while the best classification accuracy and Macro-F1 scores were obtained by the SVM-based pipeline utilizing the *ComParE functionals* attributes. Regarding the AUC scores, practically every model delivered the same performance.

6 Prediction Fusion

In our last experiment, we fused the two approaches. From the several model configurations tested, we chose the WavLM Large architecture (with UAR as the metric employed for checkpoint selections). From the two SVM-based workflows we chose the one that utilized the ComParE functionals features. Combination was performed by taking the weighted mean of the posterior estimates and then choosing the class (i.e. sarcastic or non-sarcastic) with the highest combined posterior value. Weights were tuned via a simple grid search in the $[0, 1]$ interval, using 0.05 increments, relying on the predictions obtained in cross-validation for the whole training set. We chose the weight vector that led to the highest AUC score. The weights were tuned independently for each random seed for the WavLM models (as the SVM-based models were deterministic); in the following we will refer to the weight of the WavLM Large model as λ.

Figure 2 shows the average AUC values in cross-validation and on the test set as a function of λ; $\lambda = 0$ means the SVM + ComParE functionals pipeline, while $\lambda = 1$ refers to the predictions of the WavLM Large networks. The error bars show the minimum and maximum values for the five random seeds. It is clear that, when we increase λ, the difference between the best and worst models (in terms of AUC score) grows as well. This is obviously due to the stochasticity

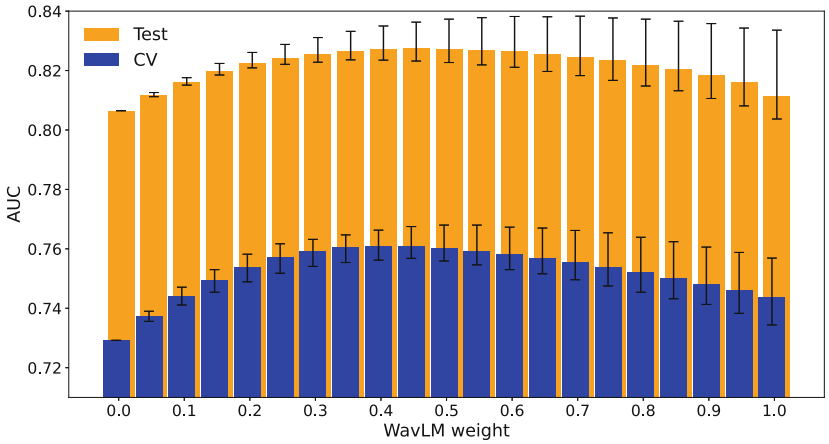

Fig. 2. The AUC values of the combined models obtained in cross-validation (CV) and on the test set.

present in DNN training (while the traditional pipeline was deterministic). The best region for combination is $0.3 \leq \lambda \leq 0.5$, resulting in AUC scores of around 0.760 in cross-validation and over 0.820 on the test set.

Table 3. The performance of the combined models; λ refers to the weight of the WavLM Large model, while the SVM + ComParE functionals approach has a weight of $1 - \lambda$.

Subset	Approach	Evaluation Metrics			
		Acc.	UAR	M. F1	AUC
Cross-validation	WavLM Large (UAR)	71.0%	70.8%	70.8%	0.745
	SVM + ComParE functionals	68.0%	67.6%	67.6%	0.729
	Combination ($\lambda = 0.40$)	70.3%	69.9%	70.0%	0.761
Test	WavLM Large (UAR)	73.3%	74.8%	73.3%	0.812
	SVM + ComParE functionals	74.3%	74.1%	73.9%	0.808
	Combination ($\lambda = 0.40$)	75.5%	76.4%	75.4%	0.827

Table 3 shows the summary of the metric scores of the two basic approaches (averaged over the five random seeds) and the best combined model (chosen based on the AUC score in cross-validation). The results validate our approach to optimize for AUC: although the other metric values slightly decreased in cross-validation, on the test set we observe an (absolute) increase between 1.2% (classification accuracy) and 2.3% (UAR). Using the Mann-Whitney U test [12] (also known as the Wilcoxon rank-sum test), these changes proved to be statistically significant with $p < 0.01$. On the other hand, the improvement in the AUC metric (from 0.812 to 0.827) was not statistically significant ($p = 0.087$).

Table 4. The performance scores of the individual combined models; λ refers to the weight of the WavLM Large model, while the SVM + ComParE functionals approach has a weight of $1 - \lambda$.

Subset	Seed Index	λ	Evaluation Metrics			
			Acc.	UAR	M. F1	AUC
Cross–validation	#1	0.40	70.3%	70.0%	70.1%	0.762
	#2	0.35	70.6%	70.3%	70.4%	0.764
	#3	0.50	69.6%	69.1%	69.2%	0.757
	#4	0.35	70.2%	69.9%	69.9%	0.758
	#5	0.50	70.4%	69.9%	70.0%	0.768
	Mean	0.42	70.2%	69.9%	69.9%	0.762
Test	#1	0.40	74.9%	75.9%	74.8%	0.824
	#2	0.35	74.8%	75.9%	74.7%	0.826
	#3	0.50	77.2%	78.1%	77.1%	0.837
	#4	0.35	74.9%	75.7%	74.8%	0.825
	#5	0.50	75.4%	76.4%	75.3%	0.826
	Mean	0.42	75.4%	76.4%	75.3%	0.828

Table 4 shows the metrics obtained for the five random seeds. In this table, the weight λ was tuned independently for the five models. The best scores fell in the range $0.35 \leq \lambda \leq 0.50$, indicating that model performance affects the combination weights as well. The cross-validation results are fairly similar to each other, yielding some slight improvement in the AUC values, while the other metric scores are somewhat lower than those reported in Table 1. This is understandable, though, as we optimized for the AUC values.

On the test set, the average performance improved over both original model types: the values reported are 1.5% higher for classification accuracy, UAR and Macro-F1, and similarly, the AUC score (averaged out over the five random seeds) is also higher (0.828 vs. the original 0.808. . .0.812 values). Overall, these represent some small, but robust improvements over the performance scores of the individual models.

7 Conclusion

We performed sarcasm detection experiments on the MUStARD++ dataset, which contains short excerpts from popular TV shows like The Big Bang Theory and Friends. Although the corpus also contains the video and the manual transcripts of the clips, we relied solely on the acoustic information by utilizing both traditional (standard features + SVM) and transformer-based (WavLM Base and WavLM Large) workflows. Our results showed that, perhaps surprisingly, the former approaches turned out to be competitive. As expected, the best results were obtained by combining the two pipelines.

In our opinion, the main reason for the competitiveness of the traditional workflows is the relatively small training set. In the applied data split (as, for some mysterious reason, MUStARD++ does not seem to have a standard train-dev-test division), the training set consisted of 848 clips, with a total duration of only 68 min (which was even reduced by the applied 5-fold cross-validation process). Under such circumstances, fine-tuning a WavLM Large model did not consistently outperform the decade-old ComParE functional attribute set fed to a linear SVM. This, along with the high computational demand of the transformer-based WavLM Large model (with its 300M+ weights), indicates that there are still circumstances where it is worth to keep the more traditional approaches.

Of course, our investigations have their limitations. First, the experiments were conducted exclusively on a single dataset (i.e. the MUStARD++ corpus), limiting the generalization ability of our findings. Second, we focused solely on the acoustic information, disregarding the lexical and visual modalities, that are otherwise available both in the MUStARD++ corpus and in other datasets (such as IEMOCAP [3] and MELD [13]). Given the increasing role of large language models across practically all research areas (and consequently also in computational paralinguistics), we aim to extend our experiments by incorporating LLMs, and integrating textual and/or visual representations in a joint framework.

Acknowledgments. This study was supported in part by ESA PRODEX 4000140862 AstroSpeech. This research was also supported in part by the European Union project RRF-2.3.1-21-2022-00004 within the framework of the Artificial Intelligence National Laboratory (MILAB) Program. This study was also supported in part by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the a 2024-1.2.3-HU-RIZONT funding scheme (project no. 2024-1.2.3-HU-RIZONT-2024-00017).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Baevski, A., Zhou, H., Mohamed, A., Auli, M.: wav2vec 2.0: a framework for self-supervised learning of speech representations. Adv. Neural. Inf. Process. Syst. **33**, 12449–12460 (2020)
2. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
3. Busso, C., et al.: IEMOCAP: interactive emotional dyadic motion capture database. J. Lang. Res. Eval. **42**(4), 335–359 (2008)
4. Chen, S., et al.: WavLM: large-scale self-supervised pre-training for full stack speech processing. IEEE J. Selected Topics Signal Process. **16**(6), 1505–1518 (2022)
5. Dong, Z., Wang, D., Chen, C., Huang, D.Y., Zhang, Z.: MHSDB: a comprehensive benchmark for multimodal humor and sarcasm detection leveraging foundation models. In: Proceedings of ICASSP, Hyderabad, India, p. 10887877 (2025)
6. Elsner, D., Langer, S., Ritz, F., Mueller, R., Illium, S.: Deep neural baselines for computational paralinguistics. In: Proceedings of Interspeech, Graz, Austria, pp. 2388–2392 (2019)

7. Eyben, F., Wöllmer, M., Schuller, B.: Opensmile: the munich versatile and fast open-source audio feature extractor. In: Proceedings of ACM Multimedia, pp. 1459–1462 (2010)
8. Eyben, F., et al.: The Geneva Minimalistic Acoustic Parameter Set (gemaps) for voice research and affective computing. IEEE Trans. Affect. Comput. **7**(2), 190–202 (2016)
9. Ghosh, D., Musi, E., Muresan, S.: Interpreting verbal irony: linguistic strategies and the connection to the type of semantic incongruity. In: Proceedings of SCiL, New York, NY, pp. 82–93 (2020)
10. Helal, N.A., Hassan, A., Badr, N.L., Afify, Y.M.: A contextual-based approach for sarcasm detection. Sci. Rep. **14**(1), 15415 (2024)
11. Hsu, W.N., Bolte, B., Tsai, Y.H.H., Lakhotia, K., Salakhutdinov, R., Mohamed, A.: HuBERT: self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Trans. Audio Speech Lang. Process. **29**, 3451–3460 (2021)
12. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. Ann. Math. Stat. **18**(1), 50–60 (1947)
13. Poria, S., Hazarika, D., Majumder, N., Naik, G., Cambria, E., Mihalcea, R.: MELD: a multimodal multi-party dataset for emotion recognition in conversations. In: Proceedings of ACL, Florence, Italy, pp. 527–536 (2019)
14. Rankin, K.P., et al.: Detecting sarcasm from paralinguistic cues: anatomic and cognitive correlates in neurodegenerative disease. Neuroimage **47**(4), 2005–2015 (2009)
15. Ray, A., Mishra, S., Nunna, A., Bhattacharyya, P.: A multimodal corpus for emotion recognition in sarcasm. In: Proceedings of LREC, Marseille, France, pp. 6992–7003 (2025)
16. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Comput. **13**(7), 1443–1471 (2001)
17. Schuller, B., et al.: The Interspeech 2016 computational paralinguistics challenge: deception, sincerity & native language. In: Proceedings of Interspeech, San Francisco, CA, USA, pp. 2001–2005 (2016)
18. Schuller, B., et al.: Affective and behavioural computing: lessons learnt from the first computational paralinguistics challenge. Comput. Speech Lang. **53**, 156–180 (2019)
19. Shor, J., Venugopalan, S.: TRILLsson: distilled universal paralinguistic speech representations. In: Proceedings of Interspeech, Incheon, Korea, pp. 356–360 (2022)
20. Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S.: X-vectors: robust dnn embeddings for speaker verification. In: Proceedings of ICASSP, Calgary, Alberta, Canada, pp. 5329–5333 (2018)
21. Wang, H., Deng, J., Meng, F., Zheng, R.: Enhancing speech emotion recognition with multi-task learning and dynamic feature fusion. In: Proceedings of Interspeech, Rotterdam, The Netherlands, pp. 4713–4717 (2025)