Archives of Acoustics Vol. 50, No. 4, pp. 445–454 (2025), https://doi.org/10.24425/aoa.2025.156933

Research Paper

Analysis of Decision Fusion in Speech Detection

Tomasz MAKA, Lukasz SMIETANKA*

Faculty of Computer Science and Information Technology, West Pomeranian University of Technology in Szczecin
Szczecin, Poland

*Corresponding Author e-mail: lsmietanka@zut.edu.pl

Received December 31, 2024; revised June 7, 2025; accepted November 3, 2025; published online November 19, 2025.

This article addresses the issue of detecting speech signal segments in an acoustic signal and analyzes potential decision fusion for a group of voice activity detectors (VADs). We designed ten new VADs using three different types of neural network architectures and three time-frequency signal representations. One of the proposed models has higher classification efficiency than competitive solutions. We used our VAD models to analyse data fusion and improve the final classification decision. For this purpose, we used gradient-free and gradient-based optimizers with different objective functions. The analysis revealed the impact of individual classifiers on the final decisions and the potential gains or losses resulting from VAD fusion. Compared with existing models, the models we proposed achieved higher classification accuracy at the cost of increased memory requirements. The final choice of a specific model depends on the platform constraints on which the VAD system will be deployed.

Keywords: voice activity detection (VAD); deep neural networks; data fusion.

1. Introduction

The problem of detecting speech in an acoustic signal involves identifying segments that contain speech. The detection mechanism for these segments is commonly used in various tasks where the signal serves as an input data source. This includes speech recognition, speaker identification, keyword spotting, and speech coding in telecommunications systems, all of which directly impact the effectiveness of classification. Although many speech detection systems, such as voice activity detector (VAD), have been developed so far, numerous new solutions have emerged recently. This is because VAD systems must operate under real-world conditions and incorporate adaptation mechanisms to handle varying acoustic environments. Additionally, their use in communication systems requires designers to develop models that account for hardware and time constraints. In speech detection, the main challenge arises from the non-stationary nature of speech signals and the diverse acoustic environments in which the signals are captured. Acoustic events and the momentary appearance and disappearance of sound sources influence the variability of the acoustic environment over time. Additionally, different acquisition conditions can introduce various types of noise into the speech signal at varying signal-to-noise ratios. These conditions make it difficult for machine learning models to accurately detect speech within a highly non-stationary signal.

Currently, existing and developed VAD systems are built based on different deep neural network architectures which very often use attention mechanisms (Song *et al.*, 2022; Wang *et al.*, 2022; Zhang *et al.*, 2023; Zhao, Champagne, 2022). Basic issues covered by such systems are connected with noise robustness and low use of energy and hardware resources. For example, Yang *et al.* (2024) introduced the sVAD model, which is based on an attention mechanism and achieves noteworthy robustness to noise. Moreover, as the authors state, it is characterized by low power consumption. Similarly, Zhao and Champagne (2022) described a VAD system built on top of the transformer architecture with an attention mechanism, which supports noise immunity and has moderate computational complexity. Kim *et al.* (2022) presented

ADA-VAD, which uses an adversarial domain adaptation mechanism to determine the properties of noisy signals. As a result, the proposed VAD is highly robust to various types of noise. SG-VAD model was proposed in (Svirsky, Lindenbaum, 2023). It was designed to work in a low-resource environment and comprises two neural networks. The model contains only 7800 parameters, which makes it suitable for running the system on edge devices. Despite a focus on noise robustness and low resource requirements, solutions for more complex scenarios, such a speech detection in multi-talker environments, have also been proposed (Aloradi *et al.*, 2023). Various issues related to the hardware implementation of 21 VADs, including performance criteria, limitations, and effectiveness, are discussed in (Yadav *et al.*, 2023).

In this work, we analyze the potential for decision fusion across ten VAD models by using an optimization process with three objective functions as examples. The paper is organized as follows: Sec. 2 discusses our VAD models, their architectures, and the dataset used in the experiments; in Sec. 3, we describe the fusion models, briefly discuss the optimisation process, and present the results; Sec. 4 concludes the paper.

2. Voice activity detection

This study aims to develop a VAD system capable of identifying speech segments containing speech signals in long audio recordings. Since our acoustic scene analysis system operates with a frame length of one second, the same frame length was used in the developed VAD modules and for comparisons with other VAD systems. To support this application, we created a custom dataset, generated from a variety of publicly available sources[1].

2.1. Dataset

In our dataset, we included three types of source signals: speech, music recordings without singing, and background noise. We randomly selected one-second frames from each group for the training set, converted them in to the appropriate representation, and used them for model training.

Table 1 presents the characteristics of the training and validation sets. The test set was created by randomly selecting fragments from the source data, each lasting between 5 s and 15 s, with segments ranging from 3 to 20 in number. These selected segments were then joined consecutively to form a single test signal. In total, 1000 such test signals were generated in this manner, 49 of which contained no speech segments.

Table 1. Characteristics of the one-second frame sets used in the training and validation process.

Process	Speech	Music	Background noise	Total
Train	2100	1050	1050	4200
Validation	900	450	450	1800
Total	3000	1500	1500	6000

2.2. VAD architectures

Our approach is based on signal frame classification. The input signal is divided into frames, from which one of four representations ($\widetilde{r}$) is derived. A decision module is then utilized, which outputs the probability (p) that the analyzed frame contains a speech signal. In the final stage, thresholding is applied, resulting in a binary value. If the probability exceeds 50 %, a value of 1 is generated at the output; otherwise, the output is 0. The entire process is illustrated in Fig. 1. To determine speech segments in an audio signal, we decided to use popular neural network architectures in conjunction with three time-frequency audio representations. Nine VAD models were designed in total.

Audio samples were converted in to three two-dimensional representations, which include spectrogram (spect), CQT-spectrogram (cqt), and mel-spectrogram (mel). The spectrogram uses a linear frequency scale, whereas the CQT-spectrogram uses a constant-Q transform (Schörkhuber, Klapuri, 2010), and in the mel-spectrogram, the frequency scale is mapped into mel scale (Rabiner, Schafer, 2010).

[1]The list of data sources is available at: https://github.com/staticvoice/ovad/blob/main/FusionDataSources.md

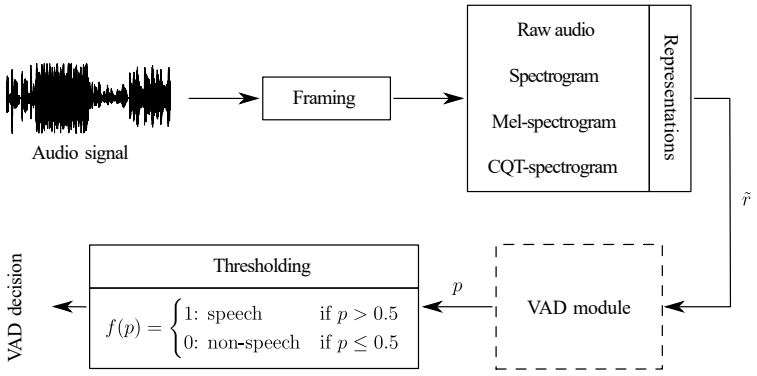

Fig. 1. General architecture for detecting speech-containing segments in acoustic signals.

All data were calculated using the Librosa Library (McFee *et al.*, 2015), and we used the following configuration of these representations:

- spectrogram (spect): n_fft: 1024, win_length: 512, $n\_features$: 513, hop_length: 512;
- CQT-spectrogram (cqt): n_bins: 90, bins_per_octave: 12, $n\_features$: 90, hop_length: 512;
- mel-spectrogram (mel): n_mels: 128, n_fft: 1024, $n\_features$: 128, hop_length: 512.

The following three neural networks architectures were used in the design of our VADmodels[2]:

1) BiLSTM (Ma *et al.*, 2022)
A simple model built with three recurrent layers, a single linear layer and a dropout layer. The first utilized architecture is a BiLSTM (Fig. 2). It is a simple model consisting of three recurrent layers: two unidirectional layers (LSTM layers) separated by a bidirectional layer (BiLSTM layer). The hidden size of the first unidirectional layer and the subsecuent BiLSTM layer is determined by the number of features ($n\_features$) in the input representation. In turn, the hidden size of the second LSTM layer is equal to $2 \cdot n\_features$. Additionally, to mitigate the phenomenon of model overfitting during the training process, the bidirectional recurrent layer is preceded by a dropout layer with a rate of 0.2. The entire model concludes with a fully connected (FC) layer with a sigmoid activation function.

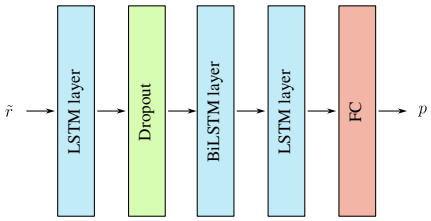

Fig. 2. VAD decision module based on the BiLSTM architecture.

2) ResNet50 (He *et al.*, 2016)
The next model, ResNet50 (Fig. 3), is a slight modification of the original architecture with the same name, differing only in changes to the first convolutional layer and the final FC layer. The first difference arises due to the type of data provided to the network's input. In the original architecture, the input consists of RGB images with three channels. In contrast, the variant used in this study takes spectrograms as input, which are single-channel images. This necessitates the use of a single input channel in the first convolutional layer instead of three. The second modification involves adapting the final FC layer of the model for binary classification. The primary component, which is the sequence of residual blocks (Fig. 6a), remains unchanged. Similarly, the layers responsible for dimensionality reduction (MaxPool, AvgPool) and the layer that converts the data into a one-dimensional vector (flattening) also remain unaltered.

[2]All proposed models can be found at: https://github.com/staticvoice/ovad/models/

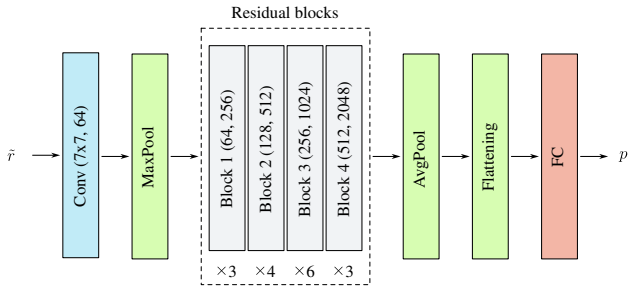

Fig. 3. VAD decision module based on the ResNet architecture.

3) ViT (Dosovitskiy *et al.*, 2021)
The third utilized architecture is the classic vision transformer (ViT), see Fig. 4. In this model, the input data is first divided into patches with a size of 16×16. Each patch is then mapped (via linear projection) to a 128-dimensional vector, which is supplemented with positional information within the sequence. Subsequently, the entire input is processed through a sequence of 12 transformer blocks (Fig. 6b). Each block consists of eight attention heads (MHA), normalization layers (norm), and linear layers (MLP). The architecture concludes with an MLP head composed of fully connected linear layers with a sigmoid activation function. Additionally, due to the need to divide the input data into patches, each input spectrogram was scaled to the following dimensions: cqt and mel to 128×128, and spect to 128×512.

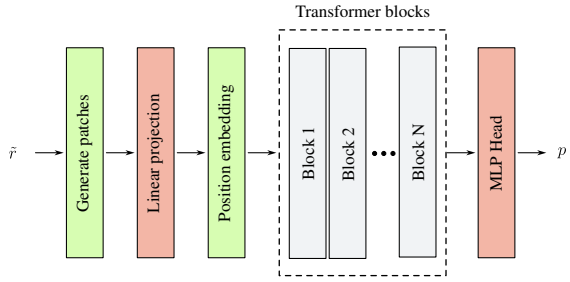

Fig. 4. VAD decision module based on the ViT architecture.

4) AugViT (Smietanka, Maka, 2023)
The final architecture used is AugViT (Fig. 5). This model is based on the standard sequence of transformer blocks, but it is preceded by a block that incorporates additional augmentation. Unlike the three previous models, the input to this

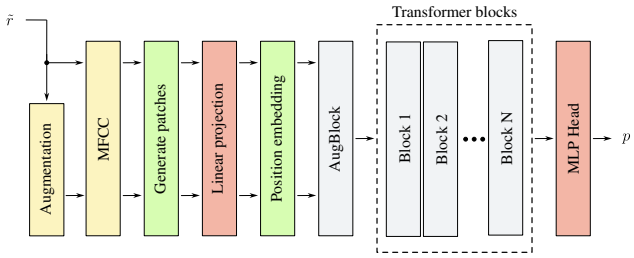

Fig. 5. VAD decision module based on the AugViT architecture.

architecture is raw audio. A random augmentation is applied to a copy of this raw signal. In the next stage, MFCC coefficients are computed separately for both the original and augmented signals. Subsequently, both MFCC representations are divided independently into patches (each patch corresponds to a single MFCC column). These patches are linearly projected into 8-dimensional vectors and supplemented with positional information within the sequence. Next, these sequences are passed to the AugBlock (Fig. 6c). Compared to the original transformer block (Fig. 6b), this block consists of two attention heads: one processes data from the original audio signal, while the other processes data from the augmented signal. The subsequent stages follow the standard ViT structure: a sequence of eight transformer blocks (each with two attention heads) followed by an MLP head.

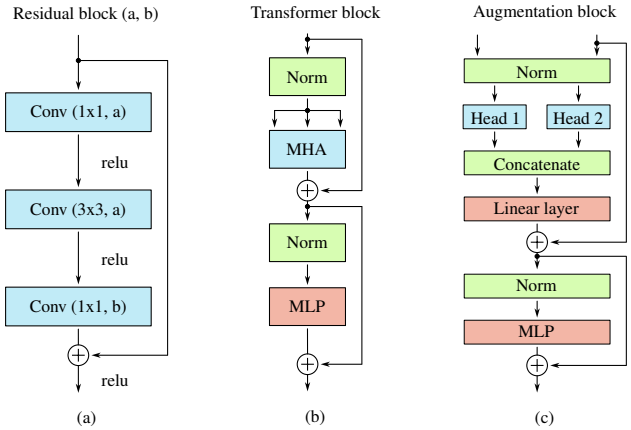

Fig. 6. Auxiliary blocks used in the VAD modules: residual (a), transformer (b), and augmentation (c) blocks.

The following parameters characterized the training procedure of each of these models:

- number of epochs in the training stage: 100 or less if, after 20 epochs, there is no improvement in classification (F_1-score not increase on the validation set);
- for further stages, the checkpoint model that obtained the highest F_1-score on the validation set was selected;
- the batch_size is equal to 16;
- Adam optimizer was selected with a learning_rate equal to 0.001;
- selected loss function: binary cross entropy (BCELoss).

2.3. Evaluation

Each of our speech detectors was tested on the entire test set. Additionally, the same data was used to carry out tests with two popular VADs: Silero (Team, 2024) and Brouhaha (Lavechin *et al.*, 2023). The results of the tests are presented as F_1-score distribution, as shown in Fig. 7. All the proposed VADs exhibit comparable classification efficacy, with the ResNet50-cqt model achieving the highest accuracy on the test set. For a detailed comparison of our best model with the Silero and Brouhaha VADs, we computed the confusion matrices, which are presented in Fig. 8.

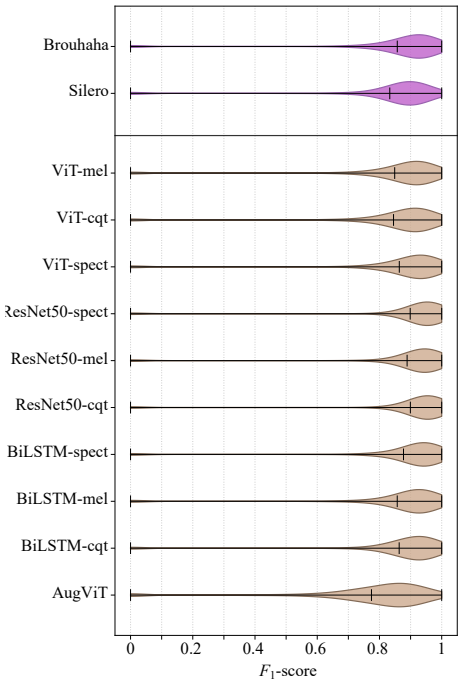

Fig. 7. Comparison of F_1-score distributions for prediction of speech segments in our proposed models and two competitive models obtained using test signals.

To compare the prediction speed and accuracy of the selected models, we predicted an audio signal of 138 seconds in length, containing four speech segments (30.07 % of the audio file) among nine other segments. The predictions were performed on a machine equipped with an i5-13600K CPU, an RTX 4070Ti GPU, and 32 GB of RAM. The results, including the models' memory requirements, are presented in Table 2. In the case of the ResNet50 architecture, there is no difference in the number of parameters or model size due to the first layers' independence from the complexity of the input data. The first layer in

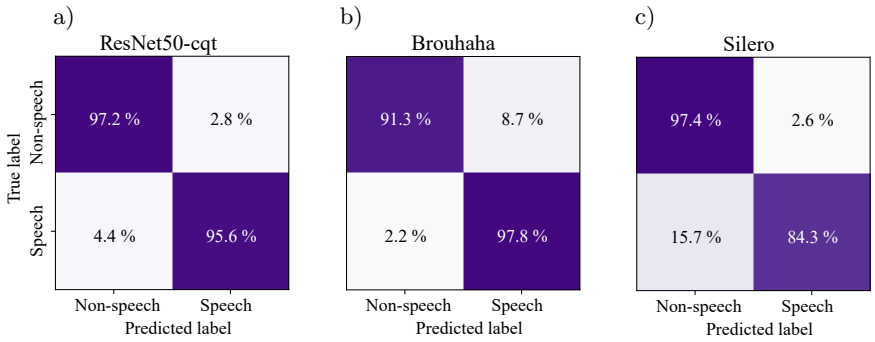

Fig. 8. Confusion matrices of our best model ResNet50-cqt (a), Brouhaha (b), and Silero (c) VADs.

Table 2. Comparison of models in predicting speech segments for an example test signal. The prediction times measured using both the CPU and GPU are presented, along with the model sizes, the number of parameters for each model, and the achieved prediction performance expressed as the F_1-score.

Model	CPU [s]	GPU [s]	F_1-score	Parameters	Size [MB]
Silero	0.6118 (±0.1102)	0.7076 (±0.1248)	0.889	462 594	2.1
Brouhaha	2.1283 (±0.0484)	0.3879 (±0.0076)	0.941	3 930 599	45
ViT-spect	2.0145 (±0.2207)	0.4116 (±0.1235)	0.895	2 446 721	9.4
ViT-mel	11.3619 (±0.4708)	0.6790 (±0.0048)	0.838	2 422 145	9.3
ViT-cqt	2.8021 (±0.2333)	2.0526 (±0.0031)	0.886	2 422 145	9.3
ResNet50-cqt	4.4269 (±0.0410)	2.0888 (±0.0321)	0.950	23 503 809	90
ResNet50-mel	4.2817 (±0.0345)	0.7287 (±0.0023)	0.925	23 503 809	90
ResNet50-spect	9.2796 (±0.0017)	0.4187 (±0.0089)	0.937	23 503 809	90
BiLSTM-cqt	2.1067 (±0.0024)	1.9555 (±0.0027)	0.817	457 381	1.8
BiLSTM-mel	0.8850 (±0.0071)	0.5373 (±0.0072)	0.865	922 881	3.5
BiLSTM-spect	7.5258 (±0.1073)	0.8784 (±0.0562)	0.897	14 759 011	56.3
AugViT	0.3921 (±0.0005)	0.5100 (±0.0720)	0.886	10 681	0.4

this architecture is a convolutional layer (Conv2d) with a fixed number of filters across all audio representations. For the ViT architecture, there is a slight difference in model size when using the spectrogram compared to other audio representations. This variation is due to differences in the number of patches into which the input can be divided. However, this number has minimal influence on the overall number of parameters. For instance, both cqt and mel spectrograms have the same number of parameters because both were interpolated to a size of 128×128, whereas the standard spectrogram was interpolated to 512×128. In contrast, for the BiLSTM models, the size of the initial LSTM layer depends on the number of rows (i.e., frequency bins) in the input representation. This, in turn, affects the total number of intermediate states in subsequent layers.

3. Data fusion

Fusing classifier outputs can be implemented in various ways (Kittler *et al.*, 1998). The basic classifier fusion techniques include so-called voting techniques: hard voting and soft voting. In the case of the first voting technique, a given class is determined as the one selected by the majority of classifiers. The second method involves averaging the probabilities and comparing them against a predefined threshold (Rokach, 2005). All of the 10 classifiers described in Subsec. 2.2 were used to fuse their individual decisions to improve speech signal detection on the test set. Because we obtained vectors with probabilities from the classifiers' outputs, we decided to use them to determine the final decision. For this purpose, for each vector, the probability of each classifier, we assigned $\alpha_n \in (0,1)$ coefficients to scale the entire vector, and thus the degree of its impact in merging the decisions of all classifiers. To determine what values α_n coefficients should be assigned to the individual vectors; we used optimization procedures and proposed the following three models for fusion. The first model is a linear combination of the probabilities from individual VAD modules:

$$\widehat{f}_1(k) = \sum_{n=1}^{N} p_n(k) \cdot \alpha_n. \tag{1}$$

The second model is also a linear combination of decisions, but only from those modules where the probability of speech presence in a given frame exceeds 60 %:

$$\widehat{f}_2(k) = \sum_{n=1}^{N} p_n(k) \cdot \widetilde{\alpha}_n; \qquad \widetilde{\alpha}_n = \begin{cases} \alpha_n & \text{if } \alpha_n > r, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where $r = 0.6$, $N = 10$, $\epsilon = 10^{-8}$, $p_n(k)$ is the probability of the k-th frame of the n-th classifier, and α_n is the model coefficient for the n-th classifier.

In the case of the third model, the result of the first model is used, with its decision trajectory dynamics altered by applying a logarithmic function:

$$\widehat{f}_3(k) = \log\left[\widehat{f}_1(k) + \epsilon\right]. \tag{3}$$

The process of combining decisions from the set of proposed VAD systems and decision fusion models described by Eqs. (1)–(3) is implemented as the optimization of parameters α_n to maximize the F_1-score. The mechanism for tuning these coefficients is schematically illustrated in Fig. 9. The process of determining the objective function for a single fusion model is carried out in the following steps:

1) for α_n coefficients, determine the resulting function signal, according to the specified model ($\widehat{f}_1$, $\widehat{f}_2$, $\widehat{f}_3$);
2) normalize the obtained signal to the $(0, 1)$ range;
3) apply threshold-based detection with $h = 0.5$;
4) compute the F_1-score value between the detected and target signals which is the final value of the objective function.

The F_1-score is computed as the harmonic mean of precision and recall (Rijsbergen, 1979). Using the true positives (TP), the false positives (FP), and the false negatives (FN) values, the score can be described as follows:

$$F_1\text{-score} = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}}. \tag{4}$$

We used both gradient-free (Opt I) and gradient-based (Opt II) optimization processes to determine the coefficients of the three proposed models. The entire process of optimizing the model coefficients is depicted in Fig. 9. Since every signal in the dataset was automatically generated and labeled, the optimization process was provided with an audio signal and its corresponding valid VAD trajectory. To determine the gain or loss during optimization, we used the following rule, where the value G is expressed as a percentage $G \in (-100, 100)$:

$$G = 100 \cdot \left(1 - \frac{\widehat{F}_1\text{-score}}{\widetilde{F}_1\text{-score}}\right), \tag{5}$$

where $\widehat{F}_1$-score is the best score obtained for whole set of VAD modules, and $\widetilde{F}_1$-score is the best score for the fused architecture.

3.1. Gradient-free optimization

For this type of optimization we used the random annealing algorithm (Blanke, 2020), which uses a hill-climbing technique with a variable step in time, similarly as in the simulated annealing method. We decided to use this algorithm after conducting a series of experiments with signals generated in the same way as those from our test set. This algorithm achieved the best results for each of the proposed models. The optimization procedure was performed separately for each objective function and for all signals in the test set. The procedure was carried out for each signal by maximizing the F_1-score over 1000 iterations. The coefficients α_n were searched within the range of 0 to 1, and the step size was equal to 0.2.

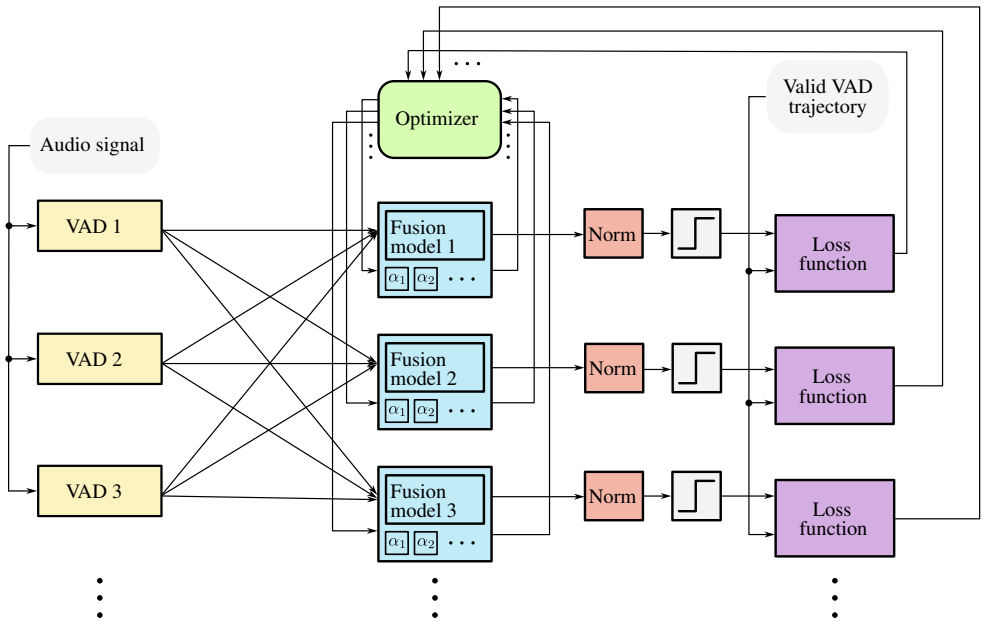

Fig. 9. General framework for optimizing decision fusion models obtained from a set of VAD modules.

3.2. Gradient-based optimization

As a gradient optimiser, we utilized the Adam (adaptive moment estimation) algorithm, an extended version of stochastic gradient descent algorithm. The Adam algorithm is known for its efficiency and robustness, and therefore we use it in the optimization process (Kingma, Ba, 2015). The optimisation procedure was performed as follows. First, we initialized the weight vector W_α with a uniform distribution with values in the $(0, 1)$ range. The variable B_f was initialized with 0; the role of this variable is to hold the highest value of the objective function. The variable B_{W_α} contains the weights for the best F_1-score. We used the binary cross entropy (BCELoss) loss function, a learning rate LR = 0.001, and the number of epochs was equal to 1 000 000. In each epoch, the given fusion model was calculated from the probabilities of ten classifiers and the weight vector W_α. From the resulting signal, the objective function was calculated. If the value of objective function (g) was higher than B_f, then $B_f = g$ and $B_{W_\alpha} = W_\alpha$. Then, an optimization of weight vector W_α using the obtained loss was performed. When, after 1000 epochs, B_f did not increase, the learning rate was reduced: LR = LR $\cdot$ 0.01. Early stopping was applied if, after 10 000 epochs, B_f did not increase. In the end, resulting B_{W_α} weights were the final coefficients of thea given model. In this case, we resigned from limiting the coefficients α_n to the range $(0, 1)$ for comparison purposes with the previous algorithm, as this limitation could have a negative impact on the optimization quality. This caused negative values in the signal after the fusion process, and in this case, it eliminated the $\widehat{f_3}$ model from use.

3.3. Results

To determine the effectiveness of the proposed fusion models, we conducted a series of experiments involving the individual fusion of each signal from the test set. The obtained coefficients were used to determine the new detection trajectory and the corresponding F_1-score, which was then compared to the F_1-score of the best of our VAD model for a given signal. Based on this comparison, gain or loss was determined. Table 3 shows its smallest, largest, and average values. The average gain in the best cases resulting from classifier fusion was less than one percent. Figure 10 depicts the gains and losses obtained on the

Table 3. Fusion results for the test set.

Fusion type	Gain(+) / Loss(−) [%]		
	Minimum	Maximum	Average
Hard voting	−48.03	7.41	−1.53
Soft voting	−41.18	7.41	−1.28
Opt I (model 1)	−10	9.71	+0.67
Opt I (model 2)	−10	10.91	+0.67
Opt I (model 3)	**−10**	**11.76**	**+0.74**
Opt II (model 1)	−15.79	12.59	+0.69
Opt II (model 2)	**−5.26**	**16.08**	**+0.89**

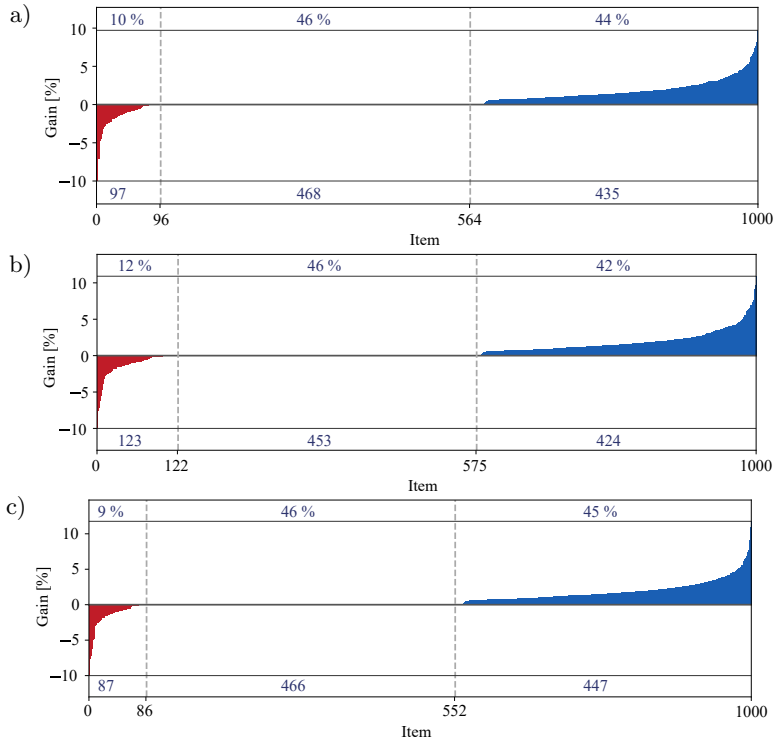

Fig. 10. Gains and losses in the test set from the optimisation process for model 1 (a), model 2 (b), and model 3 (c) using non-gradient optimization (Opt I).

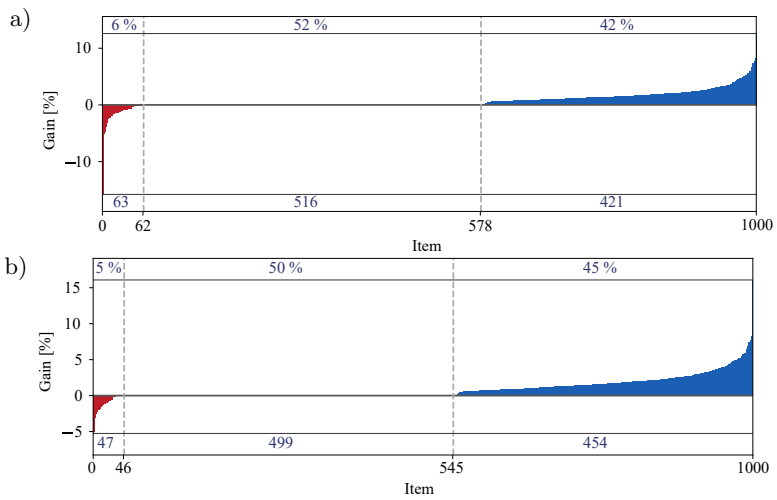

Fig. 11. Gains and losses in the test set from the optimization process for model 1 (a), and model 2 (b) using gradient optimization (Opt II).

test set for non-gradient optimization, whereas Fig. 11 for gradient optimization. In both figures, deterioration in F_1-score is marked in red, and improvement in blue, compared to the best VAD model for individual signals. As shown in Fig. 10, the highest number of cases with improved classification accuracy was achieved with model 3, where only 9 % of test signals experienced a decline in classification performance. For each of the fusion models used, the number of cases with neither improvement nor deterioration in classification was similar, amounting to approximately 46 %.

In the case of gradient optimization, the results include only two models. As mentioned in Subsec. 3.2, the possibility of weight coefficients dropping below zero and the use of a logarithm in model 3 made its inclusion in the experiments impossible. Based on the obtained results in this case, it can be observed that the number of instances where classification performance deteriorated due to fusion is almost halved compared to non-gradient optimization. Additionally, the highest gain achieved in this case exceeded 16 %.

Because, in the case of non-gradient optimization, the coefficients α_n directly influenced the significance of the probabilities in each of the VAD models, Fig. 12 shows their distribution for the entire test set in the best case where model 3 was used. Based on the obtained results, it can be concluded that the greatest contribution to the final decision comes from the ResNet50-spect model (α_7), BiLSTM-mel (α_3), and AugViT (α_1).

Interestingly, for the same architectures but different representations, there are significant differences in the distribution of weight coefficients (e.g., α_5, α_6, and α_7) determining the fusion of individual VAD modules. This indicates that the representation of the acoustic signal also plays a significant role in the effectiveness of the VAD module.

Table 4 presents the percentage contribution of individual VAD models to the correct classification of frame groups. Each group represents frames from the test set that were correctly classified by at least one and at most all classifiers. Additionally, the last row of the table shows what percentage of the entire test set each group represents.

A total of 74.1 % of frames were correctly classified by all classifiers. In 15.2 % of cases, frames were

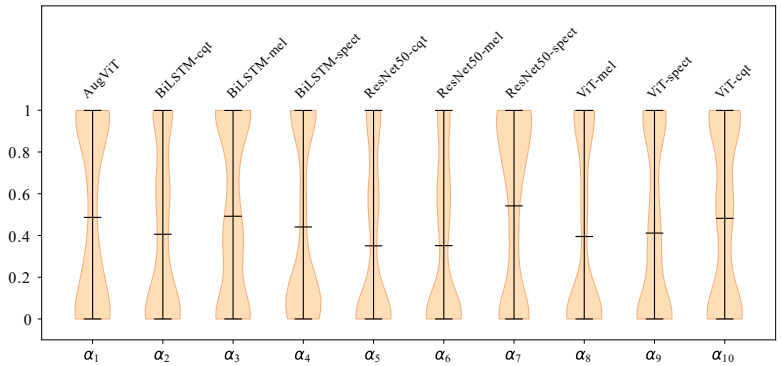

Fig. 12. Variability of the α coefficients in model 3 of the fusion (Opt I), showing the highest average value across 1000 test signals.

Table 4. Percentage contribution of VAD models to the classification of individual frame groups.

VAD	Group									
	1	2	3	4	5	6	7	8	9	10
AugViT	24.3	35.9	44.5	50.9	50.4	56.3	58.2	58.5	53.3	100
BiLSTM-cqt	16.1	23.5	31.1	41.2	45.5	57.4	68	80.5	91.7	100
BiLSTM-mel	6.1	15	24.9	33.2	41.4	49.8	61.6	74.3	95.1	100
BiLSTM-spect	8.5	20.9	35.7	48.1	54.6	67.2	75.7	82.5	95	100
ResNet50-cqt	2.7	7	24.4	36	57.3	72.8	89	95.4	99.2	100
ResNet50-mel	4.4	11.7	19.6	33.7	52.8	66.3	81.8	91.1	97.5	100
ResNet50-spect	11.8	32.6	45.6	54	67	74.9	85.1	92.5	97.7	100
ViT-cqt	10.6	21.1	30.6	36.9	45.9	58.3	64.5	73.9	84.8	100
ViT-mel	6.8	12.2	14.1	28	34.4	41.4	53.7	75	92.3	100
ViT-spect	8.7	20	29.5	38.1	50.7	55.5	62.4	76.3	93.3	100
Number of frames	0.4	0.4	0.4	0.5	0.7	1.1	1.9	4.5	15.2	74.1

correctly classified by any nine models. A smaller part of the set, 4.5 %, was correctly classified by any eight models, with ResNet VAD being the most accurate. On the other hand, 0.4 % of frames were correctly classified by only a single model, with AugViT VAD performing the best. A similar situation is observed for frames correctly classified by 2 to 7 classifiers. In these cases, each model correctly classified only a portion of the frames, but the fusion of their decisions positively affected the final result. The number of frames not correctly classified by any model was 964 (0.8 %).

4. Conclusion

In the case of analyzing the fusion mechanisms, the individual VADs learned on the same data and therefore the fusion influence in such a case was small. All the VADs we proposed were quite efficient, with an average F_1-score above 0.8, which directly impacts the fusion of decisions. This may lead to the conclusion that the chosen network architecture and signal input representation have less impact on the efficiency of VAD performance compared to the quality of the data used to train these models. When examining the resulting trajectory after detection, one can see that there many single frames that are wrongly classified. Thus, applying well-known post-processing techniques (Peinado, Segura, 2006) may improve the accuracy of frame classification. In this work, we attempted to analyze the decision fusion process in ten VAD modules. As the results demonstrate, the decision to implement the fusion process in a practical solution must be based on factors such as computational and memory resource constraints, the characteristics of the source data, and the conditions of signal acquisition. These factors directly impact the effectiveness of VAD models and, consequently, the potential contribution of fusion process in improving the overall classification performance.

Fundings

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Authors' contributions

Tomasz Maka – implementation of data fusion mechanisms, preparation of the audio dataset; Lukasz Smietanka – model implementation and training, development of the framework for the VAD components. Both authors contributed to the conceptualization of the study as well as to the development and analysis of the results. All authors reviewed and approved the final manuscript.

References

1. Aloradi A., Elminshawi M., Chetupalli S.R., Habets E.A.P (2023), Target-speaker voice activity detection in multi-talker scenarios: An empirical study, [in:] *Speech Communication – 15th ITG Conference*, pp. 250–254, https://doi.org/10.30420/456164049.
2. Blanke S. (2020), Gradient-Free-Optimizers: Simple and reliable optimization with local, global, population-based and sequential techniques in numerical search spaces, https://github.com/SimonBlanke/Gradient-Free-Optimizers (access: 16.06.2024).
3. Dosovitskiy A. *et al.* (2021), An image is worth 16x16 words: Transformers for image recognition at scale, https://arxiv.org/abs/2010.11929.
4. He K., Zhang X., Ren S., Sun J. (2016), Deep residual learning for image recognition, [in:] *2016 IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.

5. Kim T., Chang J., Ko J.H. (2022), ADA-VAD: Unpaired adversarial domain adaptation for noise-robust voice activity detection, [in:] *ICASSP 2022 – 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7327–7331, https://doi.org/10.1109/icassp43922.2022.9746755.
6. Kingma D.P., Ba J. (2015), Adam: A method for stochastic optimization, [in:] *ICLR 2015*.
7. Kittler J., Hatef M., Duin R.P.W., Matas J. (1998), On combining classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(3): 226–239, https://doi.org/10.1109/34.667881.
8. Lavechin M. *et al.* (2023), Brouhaha: Multi-task training for voice activity detection, speech-to-noise ratio, and C50 room acoustics estimation, https://doi.org/10.48550/arXiv.2210.13248.
9. Ma C., Dai G., Zhou J. (2022), Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM_BILSTM method, *IEEE Transactions on Intelligent Transportation Systems*, **23**(6): 5615–5624, https://doi.org/10.1109/tits.2021.3055258.
10. McFee B. *et al.* (2015), librosa: Audio and music signal analysis in python, [in:] *Proceedings of the 14th Python in Science Conference*, https://doi.org/10.25080/Majora-7b98e3ed-003.
11. Peinado A.M., Segura J.C. (2006), *Speech Recognition Over Digital Channels: Robustness and Standards*, Wiley, https://doi.org/10.1002/0470024720.
12. Rabiner L.R., Schafer R.W. (2010), *Theory and Applications of Digital Speech Processing*, Pearson.
13. Rijsbergen C.J.V. (1979), *Information Retrieval*, 2nd ed., Butterworth-Heinemann.
14. Rokach L. (2005), Ensemble methods for classifiers, [in:] *Data Mining and Knowledge Discovery Handbook*, Maimon O., Rokach L. [Eds], pp. 957–980, Springer, https://doi.org/10.1007/0-387-25465-x_45.
15. Schörkhuber C., Klapuri A. (2010), Constant-Q transform toolbox for music processing, [in:] *7th Sound and Music Computing Conference (SMC2010)*, https://doi.org/10.5281/zenodo.849741.
16. Smietanka L., Maka T. (2023), Augmented transformer for speech detection in adverse acoustical conditions, [in:] *2023 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, pp. 14–18, https://doi.org/10.23919/spa59660.2023.10274438.
17. Song S., Desplanques B., Demuynck K., Madhu N. (2022), SoftVAD in iVector-based acoustic scene classification for robustness to foreground speech, [in:] *2022 30th European Signal Processing Conference (EUSIPCO)*, pp. 404–408, https://doi.org/10.23919/eusipco55093.2022.9909938.
18. Svirsky J., Lindenbaum O. (2023), SG-VAD: Stochastic gates based speech activity detection, [in:] *ICASSP 2023 – 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, https://doi.org/10.1109/icassp49357.2023.10096938.
19. Team S. (2024), Silero VAD: Pre-trained enterprise-grade voice activity detector (VAD), number detector and language classifier, https://github.com/snakers4/silero-vad.
20. Wang R., Moazzen I., Zhu W.-P. (2022), A computation-efficient neural network for VAD using multi-channel feature, [in:] *2022 30th European Signal Processing Conference (EUSIPCO)*, pp. 170–174, https://doi.org/10.23919/eusipco55093.2022.9909914.
21. Yadav S., Legaspi P.A.D., Alink M.S.O., Kokkeler A.B.J., Nauta B. (2023), Hardware implementations for voice activity detection: Trends, challenges and outlook, *IEEE Transactions on Circuits and Systems I: Regular Papers*, **70**(3): 1083–1096, https://doi.org/10.1109/tcsi.2022.3225717.
22. Yang Q., Liu Q., Li N., Ge M., Song Z., Li H. (2024), SVAD: A robust, low-power, and light-weight voice activity detection with spiking neural networks, [in:] *ICASSP 2024 – 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 221–225, https://doi.org/10.1109/icassp48485.2024.10446945.
23. Zhang Y., Zou H., Zhu J. (2023), Vsanet: Real-time speech enhancement based on voice activity detection and causal spatial attention, [in:] *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8, https://doi.org/10.1109/asru57964.2023.10389633.
24. Zhao Y., Champagne B. (2022), An efficient transformer-based model for voice activity detection, [in:] *2022 IEEE 32nd International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, https://doi.org/10.1109/mlsp55214.2022.9943501.