KAN You Hear the Truth? Audio Deepfake Detection with Kolmogorov–Arnold Networks

1st Hoan My Tran
Univ Rennes, IRISA, CNRS
Lannion, France
hoan.tran@irisa.fr

2nd Damien Lolive
Univ of South Brittany, IRISA, CNRS
Vannes, France
damien.lolive@irisa.fr

3rd David Guennec
Univ Rennes, IRISA, CNRS
Lannion, France
david.guennec@irisa.fr

4th Aghilas Sini
Le Mans University, LIUM
Le Mans, France
aghilas.sini@univ–lemans.fr

5th Arnaud Delhay
Univ Rennes, IRISA, CNRS
Lannion, France
arnaud.delhay@irisa.fr

6th Pierre–François Marteau
Univ of South Brittany, IRISA, CNRS
Vannes, France
pierre–francois.marteau@irisa.fr

***Abstract*—Kolmogorov–Arnold Networks (KAN) have recently emerged as a promising alternative to traditional multilayer perceptrons, offering superior performance and greater interpretability. In this work, we explore the potential of KAN by demonstrating its ability to enhance audio deepfake detection and robustness. To further improvement, we employ multi–level token classification by leveraging speech representations of foundation model. Experimental results across multiple evaluation datasets demonstrate that our approach enhances both specialization and generalization. These findings provide potential insights into the integration of KAN, laying the way for future research in this domain.**

***Index Terms*—anti–spoofing, kolmogorov–arnold networks, self–supervised learning, audio deepfake detection**

I. Introduction

Recent advancements in Text–to–Speech (TTS) and Voice Conversion (VC) technologies have significantly improved the authenticity and naturalness of synthetic speech, making it increasingly challenging to distinguish from genuine human speech, even for State–Of–The–Art (SOTA) machine learning systems [1]. The rapid progress in speech generation has raised serious concerns, particularly regarding its potential for fraudulent activities and identity theft. Consequently, research in anti–spoofing and synthetic speech detection has advanced rapidly, focusing on distinguishing genuine speech from spoofed audio, with a strong emphasis on securing Automatic Speaker Verification (ASV) systems. The ASVspoof challenge series [1]–[3] has emerged as a crucial benchmark for developing and evaluating robust countermeasures.

Advanced deep neural networks, such as Transformer [4], Conformer [5], and architectures based on the traditional MultiLayer Perceptron (MLP), are widely employed for ASV. These models incorporate learnable linear layers combined with fixed nonlinear activation functions (e.g., ReLU). The emergence of Self–Supervised Learning (SSL) has led to the development of speech foundation models [6]–[8], which serve as powerful feature extractors for various downstream speech tasks, including Audio Deepfake Detection (ADD) [9]–[13].

Recently, KAN [14] has emerged as a promising alternative to traditional MLP, offering improved performance and interpretability, particularly for symbolic tasks. Unlike MLP, KAN replaces fixed nonlinear activation functions and learnable linear layers with learnable nonlinear activation functions. However, its applications in other machine learning domains, particularly speech processing [15]–[17], remain largely unexplored. Despite challenges in integration across various fields, KAN has demonstrated promising potential in ADD [18].

In this work, we investigate the integration of KAN and its potential to improve the performance of ADD. Specifically, we utilize the SOTA Conformer–based with Temporal-Channel Modeling (TCM) module [11] for our experiments. The contributions of our work are summarized as follows:

- Leverage the full potential of pre–trained SSL model to extract rich and informative representations for multi–level token classification.
- Enhance the baseline model by integrating Multi–Head Attention Pooling (MHAP) [19] for improved multi–level token enrichment.
- Introduce KAN as an additional layer to better capture relevant feature for effectively detecting ADD.

II. Preliminaries

A. Kolmogorov–Arnold Networks

While the MLP is based on the universal approximation theorem [20], KAN is inspired by the Kolmogorov–Arnold representation theorem. This theorem states that any multivariate continuous function $f : [0, 1]^n \to \mathbb{R}$ can be expressed as a finite composition of univariate functions and summations. Specifically, a function $f(x) = f(x_1, \ldots, x_n)$ can be represented as:

$$f(x) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right), \tag{1}$$

where $\phi_{q,p} : [0,1] \to \mathbb{R}$ and $\Phi_q : \mathbb{R} \to \mathbb{R}$ are continuous functions. In other words, for an input x, a KAN with L layers is structured as:

$$\text{KAN}(x) = (\Phi_L \circ \Phi_{L-1} \circ \cdots \circ \Phi_2 \circ \Phi_1)(x), \tag{2}$$

where Φ_l, for $l \in \{1, \ldots, L\}$, denotes a KAN layer. The output dimensions across layers are defined as $[n_1, \ldots, n_L]$. The transformation of the j-th feature in the l-th layer follows:

$$x_{l,j} = \sum_{i=1}^{n_{l-1}} \phi_{l-1,j,i}(x_{l-1,i}), \quad j = 1, \ldots, n_l, \tag{3}$$

.

where ϕ consists of two components including a spline function and a residual activation function, both parameterized by learnable weights w_b and w_s:

$$\phi(x) = w_b \text{SiLU}(x) + w_s \text{Spline}(x). \tag{4}$$

Here, $\text{Spline}(x)$ is a linear combination of B–spline basis functions, given by:

$$\text{Spline}(x) = \sum_i \alpha_i B_i(x), \tag{5}$$

where $B_i(x)$ represents the B–spline basis functions, while the coefficients α_i determine how these components combine to approximate the target function.

B. FastKAN

FastKAN [21], an optimized variant of the KAN model, enhances computational efficiency by replacing third–order B–spline basis functions with Radial Basis Functions (RBFs) utilizing Gaussian kernels. RBFs [22], [23] are real–valued functions that depend on the radial distance from a center point and are commonly used in tasks like function approximation and pattern recognition. The key idea behind RBFs is to construct a function as a weighted sum of radially symmetric functions, each centered at a specific location in the input space. An RBF network is formulated as:

$$f(x) = \sum_{i=1}^{n} w_i \phi(\|x - c_i\|), \tag{6}$$

where w_i are learnable coefficients, and ϕ represents the RBFs, which depends on the distance between x and the center c_i. The Gaussian RBF is defined as:

$$\phi(r) = \exp\left(-\frac{r^2}{2h^2}\right), \tag{7}$$

where r denotes the radial distance, and h controls the function's spread, determining the influence of each center.

III. Proposed Method

Our baseline architecture relies on the XLS-R with Conformer-based classifier and TCM module [11].

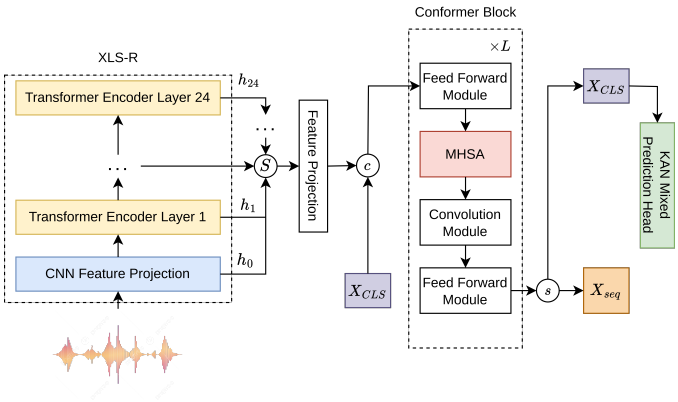

Fig. 1. General architecture of our ADD system. S refers to the stacking operation. s refers to the splitting operation. c refers to concatenation.

A. XLS-R speech foundation model

XLS-R [8], a variant of Wav2Vec2.0 [7], is designed for self-supervised cross-lingual speech representation learning. Trained on over 436,000 hours of multilingual speech data from 128 languages, it extracts high-quality representations directly from raw audio. The feature encoder comprises 7 convolutional layers to reduce complexity while preserving speech features, followed by 24 Transformer encoder layers to capture long-range temporal dependencies.

After passing the audio waveform through the feature encoder, we obtain N hidden states over T frames with D feature dimension, including the projection from the last convolutional layer. These hidden states $\mathbf{H} = (h_0, \ldots, h_{N-1}) \in \mathbb{R}^{N \times T \times D}$, are projected to a lower dimension D', yielding $\mathbf{H}' = \text{Proj}(\mathbf{H}) \in \mathbb{R}^{N \times T \times D'}$. The hidden states are then concatenated along the D'-dimensional space, forming $C = N \times D'$, and processed as:

$$X_{\text{SSL}} = \text{SeLU}([\mathbf{H'}_0, \ldots, \mathbf{H'}_{N-1}]) \in \mathbb{R}^{T \times C}. \tag{8}$$

B. Conformer with temporal-channel multi-level token

As with the original design [10], a learnable classification token $X_{\text{CLS}} \in \mathbb{R}^{\text{C}}$ is prepended to form the $X_{\text{seq}} = [X_{\text{CLS}}, X_{\text{SSL}}] \in \mathbb{R}^{(T+1) \times \text{C}}$ before being fed into the Conformer model consisting of L blocks. Since X_{SSL} has been processed with multi-level SSL representations, we refer to the resulting token as a multi-level token classification. To improve the Multi-Head Self-Attention (MHSA) mechanism for capturing temporal and channel dependencies, *Truong et al.* [11] proposed replacing the standard MHSA in each conformer block with a TCM module [11]. This TCM module includes the head token generation, MHSA, and classification token enrichment.

To generate head tokens, X_{seq} is reshaped into M segments of $d = C/M$, pooled, concatenated and projected back to C dimension. To distinguish head tokens from input tokens, a learnable embedding is concatenated with the input sequence, forming a new temporal-channel token $X_{\text{TC}} \in \mathbb{R}^{(T+M+1) \times \text{C}}$ before passing to MHSA. In our approach, we retain the overall architecture but introduce modifications to the classification

token enrichment step. Specifically, instead of averaging the temporal token $X_{\text{TT}} \in \mathbb{R}^{T\times C}$ and head token $X_{\text{HT}} \in \mathbb{R}^{M\times C}$ as in the original method, we replace these with MHAP means μ_{MHAP} [19].

Given an input $G = (G_1, \ldots, G_T)$ with $G_t \in \mathbb{R}^C$, MHAP splits G into k heads. We define $G_t = [G_{t,1}, \ldots, G_{t,k}]$ with $G_{t,j} \in \mathbb{R}^{C/k}$. Each j-th head is computed as:

$$\mathbf{c}_j = \sum_{t=1}^{T} G_{t,j}^{\top} \frac{\exp\left(G_{t,j}^{\top}\mathbf{u}_j\right)}{\sum_{l=1}^{T}\exp\left(G_{l,j}^{\top}\mathbf{u}_j\right)}, \quad j = 1, \ldots, k. \tag{9}$$

The final representation is the concatenation of all heads $\mathbf{c} = [\mathbf{c}_1, \ldots, \mathbf{c}_k]$. We enrich the multi-level X_{CLS} token for classification as follows:

$$X_{\text{CLS}} = X_{\text{CLS}} + \mu_{\text{MHAP}}(X_{\text{TT}}) + \mu_{\text{MHAP}}(X_{\text{HT}}). \tag{10}$$

C. Integration of KAN

Finally, X_{CLS} is extracted from the Conformer model's output X_{seq} and passed to a classifier to determine if the input speech is genuine or spoofed. There, we investigate replacing the traditional linear layer with KAN as the prediction head. Additionally, we explore KAN in a manner similar to MLP by stacking multiple KAN linear layers. In our final experiment, we combine both the traditional linear layer and the KAN linear layer, with the latter operating in a low-dimensional space to capture relevant features before classification. For all experiments, we employ FastKAN with various configurations, as illustrated in Figure 2.

IV. Experimental Setup

A. Dataset and performance metrics

We use the ASVspoof 2019 Logical Access (19LA) training set for model development and the development set for performance evaluation. Both subsets consist of genuine speech and spoofed samples, the latter generated via TTS and VC attacks, though they exhibit a significant class imbalance (about 10% bona fide and 90% spoofed). Model performance is assessed on 19LA [3], ASVspoof 21 Logical Access (21LA), and DeepFake (21DF) evaluation sets [2]. For out–of–domain evaluation, we use the In-The-Wild (ITW) dataset [24], where spoofed speech is sourced from YouTube. Performance is measured using the Equal Error Rate (EER) [25] where a lower EER indicates a reliable biometric security system.

B. Implementation details

In this work, we explore the integration of FastKAN-based classifier in different configurations as described in Figure 2. Firstly, we replace the traditional linear classification layer with a FastKAN linear layer. Next, we examine the impact of using multiple FastKAN linear layers for classification. Lastly, we investigate a hybrid approach that combines traditional linear layers with the FastKAN linear layer. Each experiment is conducted with different grid sizes $\in \{2, 4, 8\}$.

We use the pretrained XLS-R model from Huggingface and employ 4 Conformer encoder blocks with 4 attention heads as this configuration yielded the best results in the original method [11]. Audio inputs are dynamically padded to match the longest sample in each batch of size 5. Training is conducted using the Adam optimizer with a learning rate of 3×10^{-3} and a weight decay of 1×10^{-4}. To address class imbalance, we apply weighted cross-entropy loss, assigning a weight of 0.9 to bona fide samples and 0.1 to spoofed samples. Models are fine-tuned for three epochs, selecting the best-performing checkpoint on the development set for evaluation. All experiments are conducted on a single A100 GPU.

To enhance model robustness, we apply RawBoost [26] to the training data, incorporating various noise augmentation techniques. These include linear and nonlinear convolutive noise, impulsive signal-dependent additive noise, stationary signal-independent additive noise, and randomly colored noise. All augmentation strategies are combined during training to improve generalization.

TABLE I
FastKAN configurations with different grid sizes evaluated across 19LA, 21LA, 21DF, and ITW datasets.

Category	Grids	19LA	21LA	21DF	ITW
		EER%	EER%	EER%	EER%
Traditional linear	–	0.26	3.95	1.81	5.87
FastKAN linear	2	0.52	6.59	2.45	5.44
	4	**0.09**	4.15	1.87	6.35
	8	0.37	3.63	1.74	6.22
FastKAN multilayer linear	2	1.94	3.62	2.21	5.66
	4	0.18	3.95	2.17	5.67
	8	0.89	3.60	2.96	6.64
Mixted traditional and FastKAN linear	2	0.11	**2.29**	**1.49**	5.31
	4	0.23	4.78	1.86	7.06
	8	0.23	5.66	1.57	**5.10**

V. Results and analysis

A. FastKAN linear as a replacement of traditional linear

To investigate the impact of FastKAN, Table I presents the performance across different grid sizes. Replacing the baseline with FastKAN linear improves expressiveness, with larger grid sizes enhancing performance in–domain. FastKAN linear (8 grids) achieves a lower EER than the traditional linear and other grids on 21LA (3.63%) and 21DF (1.74%). However, on ITW, performance degrades as the grid size increases, indicating potential overfitting and reduced generalization.

B. FastKAN multilayer linear classifier

The results indicate that increasing the number of grids degrades performance for both in–domain and out–of–domain data, while smaller grids yield slightly better results. Specifically, FastKAN linear (2 grids) achieves 3.62% EER on 21LA, comparable to the traditional linear (3.95%), but larger grids (8 grids) lead to worse performance on 21DF (2.96% vs. 1.81%) and ITW (6.64% vs. 5.87%). This suggests that using FastKAN as a classifier does not enhance performance, as stacking multiple layers hinders both specialization and generalization.

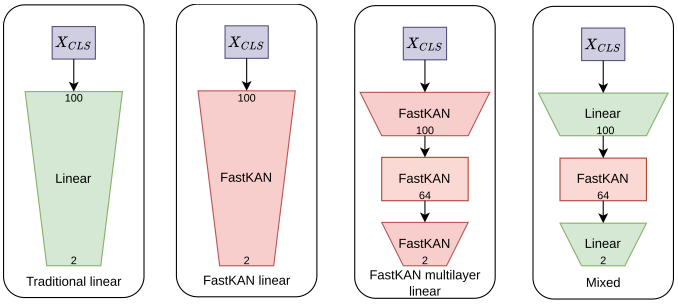

Fig. 2. Different classifier configurations evaluated.

C. FastKAN linear mixted prediction head

In our final experiments, we combined FastKAN linear layers with standard Linear layers, using FastKAN to model high–level features in a low–dimensional space (64), as illustrated in Figure 2. The results in Table I demonstrate that incorporating FastKAN linear with a small grid size enhances the model's ability to capture relevant feature details. Notably, FastKAN linear (2 grids) achieves the lowest EERs on 21LA (2.29%) and 21DF (1.49%), outperforming the traditional linear (3.95% and 1.81%, respectively). Additionally, the mixed approach maintains strong performance on ITW, with FastKAN linear (8 grids) achieving the best EER (5.10%), indicating improved generalization. These findings suggest that combining FastKAN linear with traditional layers balances expressiveness and robustness across different datasets. Since this configuration yields the best results, we adopt it for the remainder of our work.

D. In–domain attacks analysis

Figure 3 shows the performance (EER%) across various TTS–based (A07 to A16) and VC–based (A17 to A19) attacks, evaluated under different conditions (C1–C7) [2] on 21LA, with a pooled performance summary at the end. The model performs well with TTS–based attacks, particularly for A13, which achieves the lowest pooled EER of 0.52%. This suggests the model is effective at detecting these types of deepfake in most conditions. However, VC–based attacks like A18 and A16 present more challenges, with higher pooled EERs of 2.26% and 1.35%, respectively, indicating that the model struggles to detect these attacks effectively. Notably, A10 and A11 from the TTS group show higher EER (4.25% and 2.71%), yielding to more difficulty in distinguishing deepfake in these cases. The overall pooled performance of 2.29% suggests that while the model is successful at detecting certain types of attacks, it requires further refinement to handle more complex TTS and VC–based deepfakes.

Figure 4 summarizes our model's performance (EER %) on 21DF across different conditions (C1–C9) and vocoder types [2]. The results show that our model performs best on neural non–AutoRegressive (AR) vocoders, achieving a pooled EER of 0.50%, indicating strong robustness and generalization. In contrast, detecting deepfakes generated with neural AR vocoders remains the most challenging

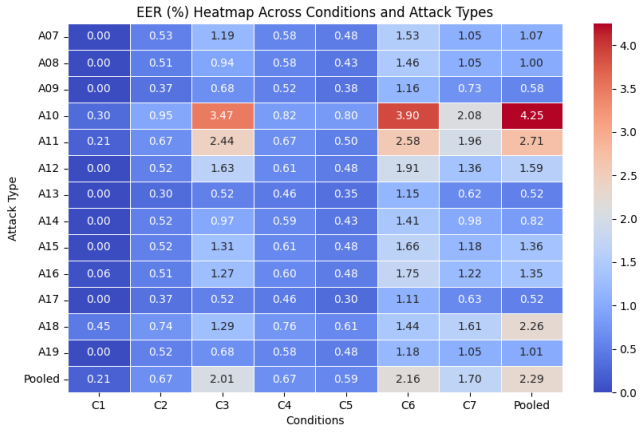

Fig. 3. Heatmap of performance (EER %) of our system with evaluated on 21LA evaluation set. A07 to A16 denotes TTS–based attacks, and A17 to A19 denotes VC–based attacks.

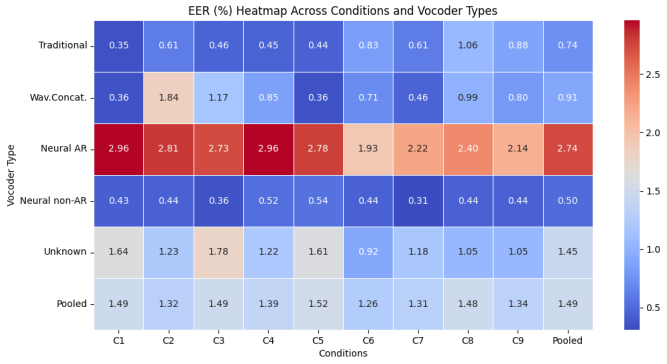

Fig. 4. Heatmap of performance (EER %) of our system with evaluated on 21DF evaluation set. "Wav.Concat." denotes waveform concatenation and AR denotes autoregressive.

with the highest pooled EER of 2.74%. Performance on traditional vocoder (0.74%) and Waveform Concatenation ("Wav.Concat.") (0.91%) vocoders is moderate, while Unknown vocoders result in an EER of 1.45%, suggesting that unseen vocoder types still present difficulties.

TABLE II
OVERALL PERFORMANCE COMPARISON WITH THE SOTA SYSTEMS ACROSS MULTIPLE DATASETS SUCH AS 19LA, 21LA, 21DF, AND ITW EVALUATION SETS.

Model	19LA EER(%)	21LA EER(%)	21DF EER(%)	ITW EER(%)	Params (M)
WavLM+MFA [27]	0.42	5.08	2.56	–	N/A
WavLM+AttM [28]	0.65	3.50	3.19	–	N/A
XLS-R+MoE [29]	0.74	2.96	2.54	12.48	341
XLS-R+AASIST [9]	–	**0.82**	2.85	–	N/A
XLS-R+AASIST2 [30]	0.15	1.61	2.77	–	N/A
XLS-R+Conformer+TCM [11], [31]	–	1.03	2.06	7.79	319
XLS-R+SLS [32]	–	2.87	1.92	7.46	N/A
XLS-R+LSR+LSA [33]	0.12	1.05	1.86	5.54	N/A
XLS-R+DuaBiMamba [31]	–	0.93	1.88	6.71	319
KAN-based Model (Proposed)	**0.11**	2.29	**1.49**	**5.31**	317

E. Comparison with the state–of–the–art systems

As shown in Table II, our proposed KAN–based model with multi–level token classification achieves SOTA performance on multiple datasets. Comparing to the baseline [11] which benefits the checkpoint averaging, our approach leads to a slightly higher EER on 21LA (2.29%) but outperforms on 21DF (1.49%). On ITW, it also sets a new benchmark with 5.31% EER, improving robustness to real–world conditions.

VI. Conclusion

In this work, we investigate the integration of Kolmogorov–Arnold Networks into speech deepfake detection, yielding promising results and significantly improving model performance, particularly for both in–domain and out–of–domain data. Our experiments with different grid sizes show that smaller grid configurations offer better performance by modeling high–level details, enhancing specialization without leading to overfitting. Additionally, combining KAN to traditional linear layers with multi–level token classification contributes to improved generalization and robustness for detecting unseen deepfake samples. Overall, our findings suggest that KAN–based classifier hold much potential for future research aimed at further enhancing robustness across various attacks, with exploring adaptive grids and hybrid architectures. The source code will be made available on https://github.com/hoanmyTran/kan_spoofing_detection.

Acknowledgement

This work was granted access to the HPC/AI resources of IDRIS under the allocation 2023–AD011013889R1 made by GENCI and funded by Côtes d'Armor departmental council and by ANR (Agence Nationale de la Recherche) through Doctoral Contract as part of the ANR–20–THIA–0018 project.

References

[1] X. Wang, H. Delgado, H. Tak, J. weon Jung, H. jin Shim, M. Todisco, I. Kukanov, X. Liu, M. Sahidullah, T. H. Kinnunen, N. Evans, K. A. Lee, and J. Yamagishi, "Asvspoof 5: crowdsourced speech data, deepfakes, and adversarial attacks at scale," in *ASVspoof*, 2024.

[2] X. Liu, X. Wang, M. Sahidullah, J. Patino, H. Delgado, T. Kinnunen, M. Todisco, J. Yamagishi, N. Evans, A. Nautsch, and K. A. Lee, "Asvspoof 2021: Towards spoofed and deepfake speech detection in the wild," *IEEE/ACM Tr. ASLP*, 2023.

[3] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. H. Kinnunen, and K. A. Lee, "ASVspoof 2019: Future horizons in spoofed and fake audio detection," in *Interspeech*, 2019.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in NeurIPS*, 2017.

[5] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Interspeech*, 2020.

[6] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE J. STSP*, 2022.

[7] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in NeurIPS*, 2020.

[8] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli, "Xls-r: Self-supervised cross-lingual speech representation learning at scale," in *Interspeech*, 2022.

[9] H. Tak, M. Todisco, X. Wang, J. weon Jung, J. Yamagishi, and N. Evans, "Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation," in *The SLR Workshop (Odyssey 2022)*, 2022.

[10] E. Rosello, A. Gomez-Alanis, A. M. Gomez, and A. Peinado, "A conformer-based classifier for variable-length utterance processing in anti-spoofing," in *Interspeech*, 2023.

[11] D.-T. Truong, R. Tao, T. Nguyen, H.-T. Luong, K. A. Lee, and E. S. Chng, "Temporal-channel modeling in multi-head self-attention for synthetic speech detection," in *Interspeech*, 2024.

[12] A. Kulkarni, H. M. Tran, A. Kulkarni, S. Dowerah, D. Lolive, and M. M. Doss, "Exploring generalization to unseen audio data for spoofing: insights from ssl models," in *ASVspoof*, 2024.

[13] H. M. Tran, D. Guennec, P. Martin, A. Sini, D. Lolive, A. Delhay, and P.-F. Marteau, "Spoofed speech detection with a focus on speaker embedding," in *Interspeech*, 2024.

[14] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljacic, T. Y. Hou, and M. Tegmark, "KAN: Kolmogorov–arnold networks," in *The Thirteenth International Conference on Learning Representations*, 2025.

[15] A. Xu, B. Zhang, S. Kong, Y. Huang, Z. Yang, S. Srivastava, and M. Sun, "Effective integration of kan for keyword spotting," in *ICASSP*, 2025.

[16] H. Li, Y. Hu, C. Chen, and E. S. Chng, "An investigation on the potential of kan in speech enhancement," *arXiv preprint arXiv:2412.17778*, 2024.

[17] C. Fan, Y. Gao, Z. Pan, J. Zhang, H. Zhang, J. Zhang, and Z. Lv, "Improved feature extraction network for neuro-oriented target speaker extraction," in *ICASSP*, 2025.

[18] K. Borodin, V. Kudryavtsev, D. Korzh, A. Efimenko, G. Mkrtchian, M. Gorodnichev, and O. Y. Rogov, "Aasist3: Kan-enhanced aasist speech deepfake detection using ssl features and additional regularization for the asvspoof 2024 challenge," in *ASVspoof*, 2024.

[19] M. India, P. Safari, and J. Hernando, "Self multi-head attention for speaker recognition," in *Interspeech*, 2019.

[20] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, 1989.

[21] Z. Li, "Kolmogorov-arnold networks are radial basis function networks," *arXiv preprint arXiv:2405.06721*, 2024.

[22] M. D. Buhmann, "Radial basis functions," *Acta numerica*, vol. 9, pp. 1–38, 2000.

[23] M. J. Orr *et al.*, "Introduction to radial basis function networks," 1996.

[24] N. Müller, P. Czempin, F. Diekmann, A. Froghyar, and K. Böttinger, "Does audio deepfake detection generalize?" in *Interspeech*, 2022.

[25] N. Brümmer and E. De Villiers, "The bosaris toolkit: Theory, algorithms and code for surviving the new dcf," *arXiv preprint arXiv:1304.2865*, 2013.

[26] H. Tak, M. Kamble, J. Patino, M. Todisco, and N. Evans, "Rawboost: A raw data boosting and augmentation method applied to automatic speaker verification anti-spoofing," in *ICASSP*, 2022.

[27] Y. Guo, H. Huang, X. Chen, H. Zhao, and Y. Wang, "Audio deepfake detection with self-supervised wavlm and multi-fusion attentive classifier," in *ICASSP*, 2024.

[28] Z. Pan, T. Liu, H. B. Sailor, and Q. Wang, "Attentive merging of hidden embeddings from pre-trained speech model for anti-spoofing detection," in *Interspeech*, 2024.

[29] Z. Wang, R. Fu, Z. Wen, J. Tao, X. Wang, Y. Xie, X. Qi, S. Shi, Y. Lu, Y. Liu, C. Li, X. Liu, and G. Li, "Mixture of experts fusion for fake audio detection using frozen wav2vec 2.0," in *ICASSP*, 2025.

[30] Y. Zhang, J. Lu, Z. Shang, W. Wang, and P. Zhang, "Improving short utterance anti-spoofing with aasist2," in *ICASSP*, 2024.

[31] Y. Xiao and R. K. Das, "Xlsr-mamba: A dual-column bidirectional state space model for spoofing attack detection," *IEEE Signal Processing Letters*, 2025.

[32] Q. Zhang, S. Wen, and T. Hu, "Audio deepfake detection with self-supervised XLS-r and SLS classifier," in *ACM Multimedia 2024*, 2024.

[33] W. Huang, Y. Gu, Z. Wang, H. Zhu, and Y. Qian, "Generalizable audio deepfake detection via latent space refinement and augmentation," in *ICASSP*, 2025.