End-to-End Audio Deepfake Detection Using Kolmogorov-Arnold Network

Arth J. Shah, Aniket Pandey, Vishnu Vardhan and Hemant A. Patil
Speech Research Lab, DA-IICT Gandhinagar, Gujarat, India-382007.
{202101154, 202411001, 202411071, hemant_patil}@daiict.ac.in

***Paper Submitted to InterSpeech 2025**

#1 Aim

- Classification of real *vs* fake signal to infer the authenticity of the audio signal, thereby enhancing security.
- End-to-End (E2E) model, cost and time efficient.

#2 Dataset Used

- The datasets being used in this study are FoR [1] for training an testing the proposed model, In-the-Wild [2] for testing.
- The term *"FoR"* refers to "*Fake or Real*" containing samples of certain preprocessing set on the original audio. FoR-2seconds dataset is used to study 2 second audio files. In-The-Wild dataset is used as the test-portion to study for evaluating the model on cross-database scenarios.

	Training		Testing		Validation	
	Fake	Real	Fake	Real	Fake	Real
FoR 2Seconds	6978	6978	544	544	1413	1413
In-The-Wild	-		11816	19963	-	

#3 KAN Architecture

- **Kolmogorov-Arnold Representation Theorem (KART) [3, 4]**: If f is a multivariate continuous function on a bounded domain, then f can be written as a finite composition of continuous functions of a single variable and the binary operation of addition.
  More specifically, for a smooth $f : [0, 1] \to \mathbb{R}$,

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

  where $\Phi_{q,p} : [0, 1] \to \mathbb{R}, \quad \Phi_q : \mathbb{R} \to \mathbb{R}$.
- The Base Weight transformation is given by:

$$Z_{\text{base}} = XW_{\text{base}}^T,$$

  where:
  - $W_{\text{base}} \in \mathbb{R}^{\text{out\_features} \times \text{in\_features}}$ is the learnable weight matrix (I).
  - $X \in \mathbb{R}^{\text{batch} \times \text{in\_features}}$ (input feature vector).
  - $Z_{\text{base}} \in \mathbb{R}^{\text{batch} \times \text{out\_features}}$ is the output before applying activation.
  - batch = Number of input audio signals
- $\text{PReLU}(x) = max(0, x_i) + a_i min(0, x_i)$
  $W_{\text{base}}^{\text{act}} = \text{PReLU}(W_{\text{base}})$, where
  each weight element in W_{base} is individually activated before multiplying with input [5].
  Base transformation: $Z_{\text{base}} = XW_{\text{base}}^{\text{act}^T}$
- B-spline basis functions are computed based on a **grid** that defines knot points for the splines.
  Total knots in the grid = $(\text{grid\_size} + (2 \times \text{spline\_order}) + 1)$.
- For each input feature (in_features features), a set of B-spline basis functions is computed using [6]:

$$B_{i,k}(x) = \frac{x - t_i}{t_{i+k} - t_i} B_{i,k-1}(x) + \frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(x),$$

  where $B_{i,k}(x)$ is the k order basis function, t_i are the knot positions in the grid.
  Number of B-spline basis functions = (grid_size + spline_order).
  For each sample in the batch,
  Spline Output = $(\sum_{i=1}^{in\_features} \sum_{j=1}^{bf} B_{i,j} \cdot W_{o,i,j})$,
  where:
  - bf equals to total number of B-spline basis functions.
  - $B_{i,j}$ represents the B-spline basis function activations (shape: $(\text{batch}, \text{in\_features}, bf)$).
  - $W_{o,i,j}$ represents the learnable spline weight coefficients (shape: $(\text{out\_features}, \text{in\_features}, bf)$).
- **Final Output** = (Base_output) + (Spline_output), where:
  - Base_output = base weight × activation
  - Spline_output = spline weight × B-Spline basis.

#4 Classifiers Used

We employ CNN-based classifier combined with KAN, with similar structure as in Figure 1.

#5 Proposed Methodology

- In the proposed model, we replace the lower dense layers of CNN with KANLinear layers.
- Along with the learnable weights, now the fixed activation function of CNN also becomes learnable thereby both weights and activation function become learnable.

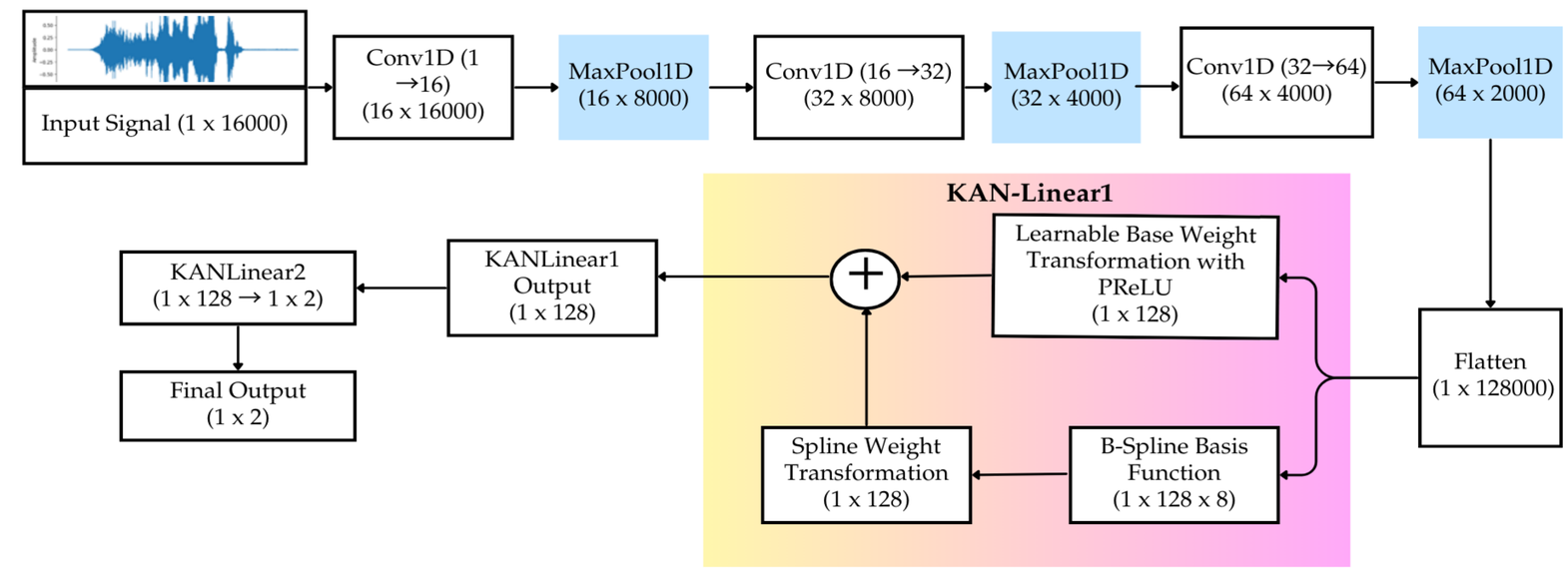

Fig 1: Functional Block Diagram of Proposed Methodology for End-to-End ADD using KAN.

#6 Motivation

- **Interpretability**: KANLinear replaces dense layers, making feature transformations explicit rather than black-boxed.
- **Efficiency**: Reduces the number of *learnable* parameters using structured, low-rank decomposition, lowering memory, and computational costs.
- **Better Generalization**: Captures smooth functional mappings, preventing overfitting, and improving performance on unseen data.
- **Robustness**: More resistant to adversarial attacks and noise compared to traditional dense layers.
- **Biological Inspiration**: Aligns with neuroscientific principles, reflecting structured processing in human cognition.
- **Efficient Learning**: Requires less labeled data for effective generalization.
- **Limitations**: Higher training *time* and *space* complexity as it learns both weights and activation functions.
- **No Need for Normalization**: Unlike CNNs, KAN-based models do not require normalization layers to prevent overfitting.

#7 Experimental Results

Fig 2: Representation of epochwise training on FoR-2-Second dataset.

- The fig. 2 represents the variation in accuracy w.r.t. variation in number of epoch. It can be observed that even in the least epoch scenario, proposed method outperforms classical CNN model, representing significance of need to built FLNs.

	CNN	Real	Fake	KAN+CNN	Real	Fake
Actual	**Real**	1888	792	**Real**	2094	586
	Fake	377	2305	**Fake**	403	2279
	Predicted					

- When tested on CNN, more real samples are predicted as fake, which happens due to unavailability of model to capture key characteristics of real audio from original speech signal.
- Performance on cross dataset scenario: 51.35 % accuracy with CNN classifier, and 54.47 % accuracy with proposed KAN plus CNN classifier.

#8 Summary and Conclusion

- The proposed model paths a Fully Learnable Network, CNN and KAN.
- Analyzed and compared the behavior of CNN with proposed E2E method resulting in better accuracy.
- Future works include optimizing the proposed model by learning the activation functions in each convolution blocks.
- Deeper understanding of theoretical results of KAN with respect to ADD task, and machine learning, in general, remains an open research problem.

#9 Acknowledgments

- The authors sincerely thank the Ministry of Electronics and Information Technology (MeitY), New Delhi, Govt. of India, for sponsoring a consortium project titled 'BHASHINI', (Grant ID: 11(1)2022-HCC.(TDIL)).

#10 Selected References

[1] R. Reimao and V. Tzerpos, "For: A dataset for synthetic speech detection," in 2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD), 2019, Timis,oara, Romania, pp. 1–10.

[2] N. M. Muller, P. Czempin, F. Dieckmann, A. Froghyar, and K. Bottinger, "Does audio deepfake detection generalize?" ¨ arXiv-preprint arXiv:2203.16263, 2022, $\{LastAccessedDate : 2^{nd} February, 2025\}$.

[3] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljaci˘ c, T. Y. Hou, and M. Tegmark, "KAN: Kolmogorov-Arnold Networks," 2024.

[4] V. I. Arnold, "On the representation of functions of several variables as a superposition of functions of a smaller number of variables," Collected works: Representations of functions, celestial mechanics, and KAM theory, 1957–1965, pp. 25–46, 2009.

[5] T. Jiang and J. Cheng, "Target recognition based on cnn with leakyrelu and prelu activation functions," in International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), 2019, Beijing, China, pp. 718–722.

[6] A. Chaudhuri, "B-splines," arXiv preprint arXiv:2108.06617, 2021, $\{LastAccessedDate : 2^{nd} February, 2025\}$.