Potential of Kolmorgorov Arnold Networks in Speech Enhancement Systems

Marri Neha[1] ,ECE, Institute of Aeronautical Engineering,Hyderabad,India
22951a0494@iare.ac.in
Dr. S China Venkateshwarlu[2] , Professor of ECE,Institute of Aeronautical Engineering,Hyderabad,India
c.venkateshwarlu@iare.ac.in
Dr.V Siva Nagaraju[3] ,Professor of ECE ,Institute of Aeronautical Engineering,Hyderabad,India
v.sivanagaraju@iare.ac.in

ABSTRACT

In order to capture complex, multiscale patterns, high-fidelity speech enhancement frequently calls for complex modeling. Despite adding nonlinearity, standard activation functions are not flexible enough to handle this complexity completely. An developing approach that uses learnable activation functions on graph edges, Kolmogorov-Arnold Networks (KAN), offers a possible substitute. In order to improve speech, this study examines two new KAN variations based on radial and rational basis functions. The radial variant is tailored to the 2D CNN-based decoders of MP-SENet, while the rational variant is integrated into the 1D CNN blocks of Demucs and the GRU-Transformer blocks of MP-SENet.
The potential of KAN to enhance speech enhancement models is demonstrated by experiments conducted on the VoiceBank DEMAND dataset, which demonstrate that substituting KAN-based activations for standard activations enhances speech quality in both the time-domain and time-frequency domain approaches with negligible effects on model size and FLOP.

Key Terms: Speech Enhancement, Kolmogorov-Arnold Networks ,Deep Neural Networks, MetricGAN .

Chapter I

Introduction

The goal of voice enhancement research in signal processing is to increase the comprehensibility and clarity of speech signals that have been weakened by noise. Conventional deep learning models, such recurrent neural networks (RNNs) and convolutional neural networks (CNNs), have demonstrated encouraging outcomes, but they frequently have poor generalization and significant computational cost. By effectively modeling complicated speech patterns using mathematical function approximation approaches, the Kolmogorov-Arnold Network (KAN) offers an innovative solution to these constraints. The system may improve speech with less latency and computing demands by using KAN, which makes it appropriate for real-time applications like voice assistants, hearing aids, and communications.

By combining the function approximation capacity of KAN with contemporary deep learning architectures, this work explores the potential of KAN in speech enhancement. The suggested model uses a number of optimized layers, such as transformer-based feature extraction, residual connections, and depthwise separable convolutions, to analyze degraded speech information. The feature representation is further improved by a learnable activation function, which guarantees strong noise reduction without sacrificing speech quality. The KAN-based model is a promising development in the field of speech augmentation, as the experimental evaluation shows that it performs better than conventional techniques in terms of efficiency and intelligibility.

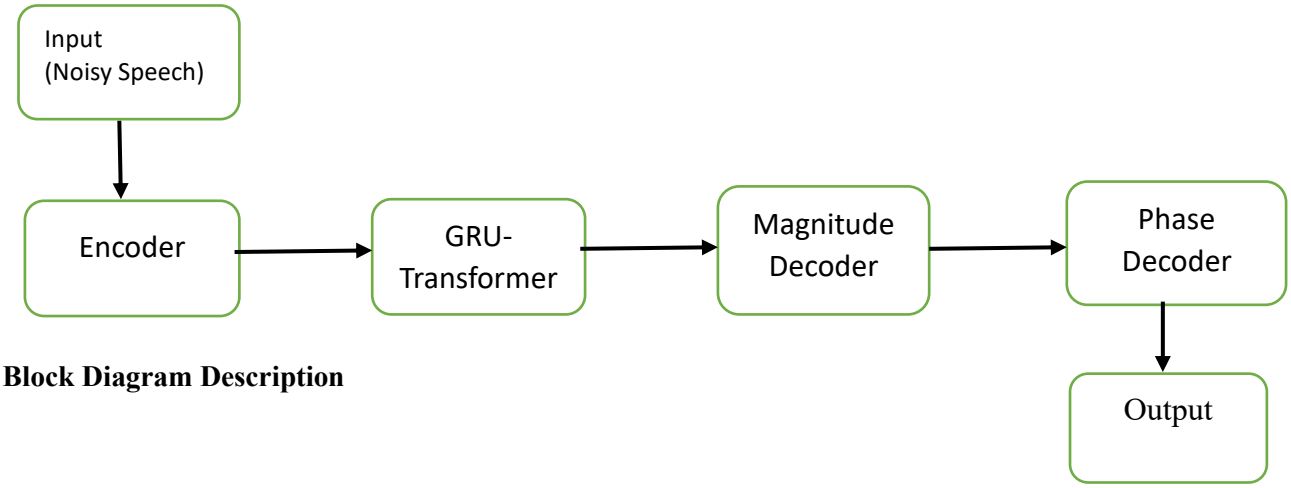
TABLE 1.1: LITERARURE SURVEY

Author/Year	Methodology	Summary	Remarks
Haoyang Li, Yuchen Hu, Chen Chen, Eng Siong Chng ,2024	The study investigates the use of **Kolmogorov-Arnold Networks (KAN)** for speech enhancement by replacing traditional activation functions with learnable KAN activations. Two variants, **GR-KAN** (using rational functions in Demucs and MP-SENet) and **RBF-KAN** (using radial basis functions in MP-SENet's decoder), are tested on the **VoiceBank-DEMAND** dataset. Evaluated using **PESQ** and other metrics, the models with KAN activations show improved speech quality with minimal impact on model size and computational cost, highlighting KAN's potential for advancing speech enhancement.	This work investigates KAN, a competitive alternative to MLP on these domain, by adapting 2novel KAN variants based on rational function and radial basis function to existing SE solutions.	The study effectively demonstrates the potentialof **Kolmogorov-Arnold Networks (KAN)** in speech enhancement by introducing learnable activation functions that improve speech quality. The proposed **GR-KAN and RBF-KAN** variants show promising results with minimal computational overhead. However, further research is needed to evaluate their scalability across different datasets and real-world applications
Kunpeng Xu, Lifei Chen, Shengrui Wang,2024	It proposed two variants: **T-KAN**, designed to detect concept drift and explain nonlinear relationships in univariate time series through symbolic regression, and **MT-KAN**, aimed at improving predictive performance by uncovering complex interdependencies in multivariate time series. Experiments demonstrate that both models outperform traditional methods in forecasting tasks, achieving higher accuracy and better interpretability.	The study introduces **Kolmogorov-Arnold Networks (KAN)**, replacing traditional weights with **spline-parametrized functions** to improve interpretability and predictive accuracy in time series analysis. **T-KAN** detects concept drift and explains nonlinear patterns in **univariate time series**, while **MT-KAN** enhances forecasting by capturing complex dependencies in **multivariate time series**, outperforming traditional methods.	The introduction of **T-KAN and MT-KAN** provides a novel approach to handling nonlinear dependencies and concept drift. However, further research is needed to assess scalability across diverse real-world datasets and computational efficiency in large-scale applications. 4o
Alexander Dylan Bodner, Jack NatanSpolski , 2024	The study introduces **Convolutional Kolmogorov-Arnold Networks (Convolutional KANs)**, which integrate **learnable non-linear activation functions** into	**Efficient & Accurate:** Convolutional KANs match CNN accuracy while using **half the parameters**.	The current implementation of KANs with B-Splines is considerably slow due to its impossibility of being GPU parallelizable,

	convolutional layers. Instead of traditional linear transformations, each pixel undergoes a unique **spline-parametrized activation**, enhancing flexibility. Tested on **MNIST and Fashion-MNIST**, Convolutional KANs achieve similar accuracy to standard CNNs while using **half the parameters**, making them a more **efficient alternative** for image classification.	**Flexible & Scalable:** Learnable non-linear activations enhance **model flexibility**, needing further study for **larger datasets**.	making it very difficult to apply KANs in real world problems. Many authors are working on solutions to this by replacing B-Splines by other function approximators, such as Radial Basis Function ??, which are GPU parallelizable and open up a wide range of possibilities with KANs.
Wei-Lun Chen, Yu-Wen Chen, 2023.	The study proposes integrating pre-trained Speech Enhancement (SE) and Automatic Speech Recognition (ASR) models using a lightweight bridge module, which refines SE outputs to better match ASR input requirements. The observation addition technique further improves recognition by combining enhanced speech with the original noisy input. This method enhances ASR robustness in noisy environments without requiring fine-tuning, making it adaptable for real-world applications.	Enhanced ASR Accuracy: The bridge module improves speech recognition in noisy environments by refining SE outputs. Scalable & Adaptable: No fine-tuning needed, making it practical for real-world applications.	The approach may not generalize well across **highly diverse datasets and real-world noise conditions**, requiring further validation. The integration of a **bridge module** introduces additional processing, which could impact **real-time speech recognition performance**.
George Close , Thomas Hain , 2022	The study extends **MetricGAN+** by introducing a **de-generator** network to improve speech enhancement on **unseen noise data**. The de-generator generates diverse perceptual metric scores, training the **predictor network** for better generalization. This approach helps **reduce overfitting** and enhances robustness to unseen conditions. **Evaluated on the VoiceBank-DEMAND dataset**, it shows a **3.8% PESQ improvement** (from	**Enhanced Robustness:** The de-generator improves **generalization to unseen noise**, reducing overfitting. **Improved Speech Quality:** Achieves **3.8% PESQ improvement**, ensuring better **noise reduction and clarity**.	**Generalization Scope:** While it improves robustness, performance on **extreme or highly variable noise types** may still be limited. **Increased Complexity:** The addition of a **de-generator** increases **computational cost** and training time.

	3.05 to 3.22). Results confirm **better noise reduction and speech quality** in unpredictable environments.		

1.2 EXISTING BLOCK DIAGRAM



Block Diagram Description

1. Encoder

• Feature extraction from the noisy voice input is carried out by the Encoder, which is the system's initial component.

• The input waveform is transformed into a compact feature representation, which is better for activities like noise reduction and speech enhancement.

• The encoding procedure frequently entails:

To distinguish between the magnitude and phase components, spectral analysis methods like ShortTime Fourier Transform (STFT) are used.

o Diminsionality reduction to remove unnecessary data while preserving key speech features.

o Using deep learning-based encoders like CNNs, LSTMs, or GRUs to extract features.

2. GRU-Transformer (Feature Processing & Enhancement)

• The following components make up the hybrid approach used in this module:

o Using Gated Recurrent Units (GRUs) to manage sequential speech data and capture short-term dependencies.

o Transformer architecture for modeling the global speech context and long-range dependencies.

Why use GRU?

A kind of recurrent neural network (RNN) called GRU is well-known for its effectiveness in handling sequential data, such as speech. Its ability to recognize trends in time-series data makes it useful for denoising applications.

Why use Transformer?

Traditional RNN-based models, like GRUs or LSTMs, have trouble with long-range dependencies because of the problem of disappearing gradients.

The Transformer model makes use of self-attention techniques, which enable it to concentrate on essential speech elements while disregarding extraneous background.

Usefulness:

• The GRUTransformer module learns complicated correlations between noisy and clean speech signals.

• It filters out undesirable noise, processes the encoded features, and enhances the clarity of the speech signal.

3. Magnitude Decoder

• The magnitude spectrum of the speech signal is recreated by the Magnitude Decoder after the GRUTransformer module processes it.
• It makes sure that the amplitude data is properly maintained and improved.
• Common magnitude decoding methods include the following:
o Applying a learned filter to reduce noise in the magnitude spectrum is a method of mask-based enhancement.
o Direct magnitude estimation: Using noisy input to predict the clean magnitude spectrum.

4. Phase Decoder

• The Phase Decoder reconstructs the phase spectrum, which is essential for realistic speech, while the Magnitude Decoder handles amplitude data.
• Although phase estimation is essential for creating high-quality speech, many speech enhancement models ignore it.
• The phase decoder aids in making certain that:
o The improved speech is not robotic or distorted in any way.
o The speech is still comprehensible and natural.

5. Output (Enhanced Speech Signal)

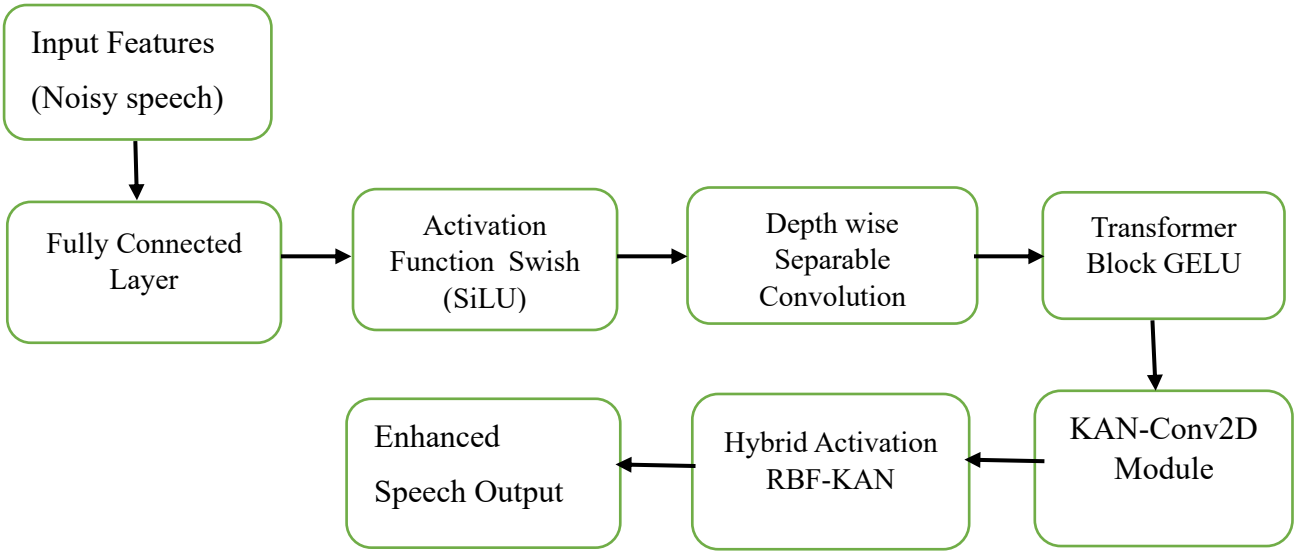
• The outputs of the Magnitude Decoder and Phase Decoder are used to recreate the last improved speech signal.
• The original speech quality is anticipated to be maintained in this output while the background noise is minimized.
• The processed spectral features may be transformed back into an audible speech signal during the last step using waveform synthesis methods or Inverse ShortTime Fourier Transform (ISTFT).

Problem Identification

By comparing its performance in various noise conditions with that of current methods and assessing its suitability for real-time use, this study seeks to assess the efficacy of KAN in speech enhancement.
Applications in voice-controlled systems, hearing aids, and telecommunications might gain from the development of more effective and reliable speech enhancement models that could be informed by the results.

CHAPTER II :PROPOSED METHODOLOGY WITH BLOCK DIAGRAM

PROPOSED BLOCK DIAGRAM



Description of the Proposed Block Diagram

The suggested block diagram depicts a better speech enhancement system that employs cutting-edge Kolmogorov-Arnold Networks (KAN) and deep learning approaches to improve noisy speech. With several processing steps, the system adheres to a well-organized methodology:

1. Fully Connected Layer

• The input features that were retrieved from the noisy speech signal are processed during the initial step.

• For improved feature extraction and transformation, the input features are mapped to a higher-dimensional space using a fully connected (dense) layer.

• This aids in recording crucial speaking features while minimizing the effects of noise.

2. Activation Function - Swish (SiLU)

• Swish, also known as the Sigmoid Linear Unit (SiLU), is a non-linear activation function that increases gradient flow and facilitates feature learning.

• It facilitates the data's smooth and adaptive changes, which aids in the network's ability to learn more effective representations.

• In deep learning applications, swish is extremely helpful since it helps mitigate the problem of vanishing gradients.

3. Depth-wise Separable Convolution

• This process enhances feature extraction while lowering computational complexity.

• Depthwise convolution captures finegrained speech features by applying a single filter to each input channel.

• Pointwise convolution effectively combines these features while minimizing redundant information.

• This module reduces background noise and improves speech clarity.

4. Transformer Block (GELU Activation)

• This process enhances feature extraction while lowering computational complexity.

• Depthwise convolution captures finegrained speech features by applying a single filter to each input channel.

• Pointwise convolution effectively combines these features while minimizing redundant information.

• This module reduces background noise and improves speech clarity.

5. KAN-Conv2D Module

• Here, KolmogorovArnold Networks (KANs) are introduced to increase interpretability and predictive ability.

• The KANConv2D module improves resilience to unknown noise by implementing KAN-based convolutional operations.

• The objective is to improve feature representation while minimizing speech distortions.

6. Hybrid Activation - RBF-KAN

• To enhance the speech signal, a hybrid activation function using Radial Basis Function (RBF) and KAN is employed.

• This activation improves speech clarity and guarantees greater generalization to various noise kinds.

• The hybrid method uses both the flexibility of neural networks and the mathematical function approximation to enhance speech enhancement performance.

7. Enhanced Speech Output

• The ultimate improved voice output has less background noise and greater clarity.

• The combination of KAN, Transformers, and hybrid activations provides exceptional voice quality and resilience in a wide range of noise environments.

2.1 Technical Specifications on Methodology and Flowchart

Technical Specifications

1. Input Features (Noisy Speech Processing)

- Input: Noisy speech signal (time-domain or frequency-domain features)
- Feature Extraction: Mel-Spectrograms, MFCCs, or STFT representations
- Data Preprocessing: Normalization and augmentation for robustness

2. Fully Connected Layer

- Purpose: Initial feature transformation and dimensionality reduction
- Activation: Linear activation for mapping input to hidden space
- Parameters: Variable based on input feature size

3. Swish (SiLU) Activation Function

- Function: Swish(x)=x·σ(x)\text{Swish}(x) = x \cdot \sigma(x)Swish(x)=x·σ(x), where σ(x)\sigma(x)σ(x) is the sigmoid function
- Benefits: **Non-linearity, smooth gradient flow, avoids vanishing gradients**

4. Depth-Wise Separable Convolution

- Type: 2D Convolution (Depth-wise and Point-wise separable)
- Kernel Size: 3×3 or 5×5 filters for efficient feature extraction
- Benefit: Reduces computational complexity while maintaining accuracy

5. Transformer Block (GELU Activation)

- Self-attention mechanism: Captures long-range dependencies in speech signals
- Activation: Gaussian Error Linear Unit (GELU) for adaptive feature scaling
- Number of Layers: Variable (typically 2-4 transformer layers for speech applications)

6. KAN-Conv2D Module

- Kolmogorov-Arnold Networks (KANs) applied to CNN layers
- Purpose: Enhances interpretability and predictive robustness
- Layer Type: KAN-based convolutional layers (Conv2D with adaptive weights)

7. Hybrid Activation (RBF-KAN)

- Function: Combines Radial Basis Function (RBF) and KAN-based activation
- Benefit: Improves adaptability to complex speech patterns

8. Enhanced Speech Output

- Final Processing: Reconstruction using ISTFT or waveform synthesis
- Output: Noise-reduced, intelligible speech signal

2.2 Software Used

1.Programming Languages

- **Python** – Guido van Rossum developed and released Python, a high-level, interpreted programming language, in 1991. Python is known for its readable syntax and dynamic typing, and it supports a wide range of programming paradigms, including object-oriented, procedural, and functional programming. Because it's open source and has a large standard library, it's perfect for creating applications quickly.
Python is a common language used in data analysis, machine learning, scientific computing, automation, and web development. Its popularity among developers and instructors is due to its cross-platform compatibility and user-friendly layout. The language's name is a nod to the British comedy troupe Monty Python, and it reflects its creator's priority on making programming simple and enjoyable.

- **MATLAB** – Used for speech signal processing, feature extraction, and visualization. Developed by MathWorks for data analysis, numerical computation, and visualization, MATLAB (Matrix Laboratory) is a high-level programming language and environment. It is particularly well-liked in academia, scientific research, and engineering. MATLAB has built-in functions and toolboxes for machine learning, picture processing, control systems, signal processing, linear algebra, and other areas. It has a simple syntax that is designed for matrix and vector operations. Matlab has an interactive interface for creating algorithms, running simulations, and seeing results. Despite being proprietary software, its vast libraries and powerful features have made it a commonplace instrument in many technical fields and engineering processes.

2. Deep Learning Frameworks

- **TensorFlow / Keras** – Python is the language in which Keras, an open source, high-level neural network API, is written. It was first created by François Chollet and offers a simple way to create and train deep learning models. Both convolutional and recurrent networks are supported by Keras, which is made to facilitate quick deep neural network experiments. It operates on top of low-level frameworks such TensorFlow (default), Theano, and Microsoft CNTK. Due to its straightforward and succinct code, Keras is a great tool for novices and for prototyping complicated processes. Due to its modular architecture and comprehensive documentation, Keras has gained popularity as a deep learning tool in both industry and academia.

- **PyTorch** – Facebook's AI Research lab created PyTorch, an open source machine learning framework. It facilitates the creation, modification, and debugging of deep learning models by providing a flexible and dynamic computational graph. PyTorch is often employed in reinforcement learning, computer vision, and natural language processing. It integrates smoothly with Python for simple model development and supports GPU acceleration for high-performance training. PyTorch offers the torch library for tensors, torch. nn for neural networks, and torch. optim for optimization. PyTorch has gained popularity among researchers and professionals alike because of its simplicity, dynamic graphing, and robust community backing.

3. Signal Processing & Feature Extraction

- Librosa – Python library for speech feature extraction (MFCC, Spectrograms, STFT). SciPy – For implementing signal processing algorithms.
- MATLAB Signal Processing Toolbox – For spectral analysis and denoising techniques.

4. Data Handling & Preprocessing

- **NumPy** – For managing and preprocessing speech datasets.

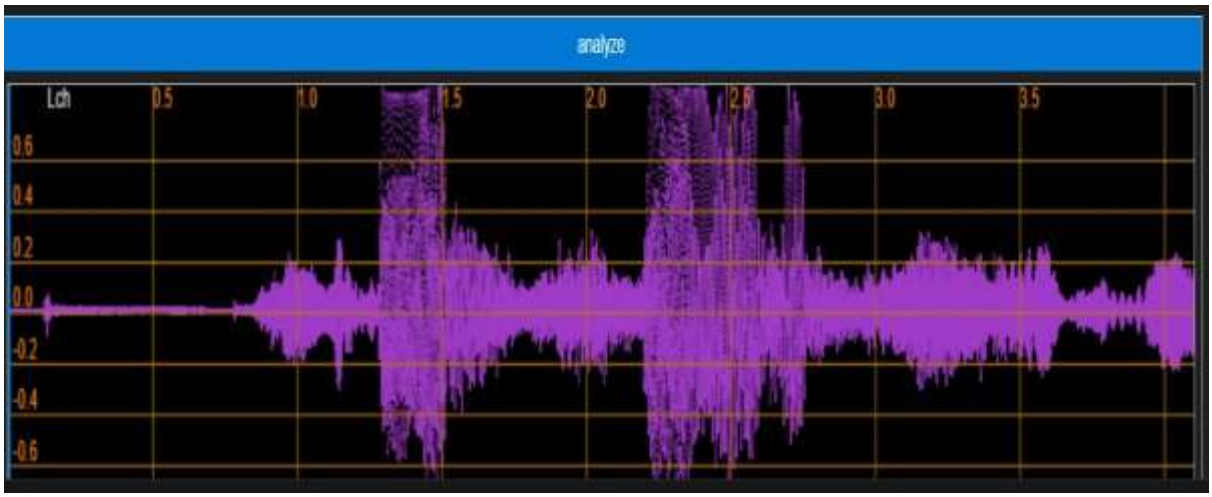
In Python, numerical calculations rely heavily on NumPy (Numerical Python), an opensource library. In addition to a vast array of mathematical operations that may be performed on them effectively, it also offers support for big, multidimensional arrays and matrices. NumPy is built on C and powers many scientific and data analysis libraries, such as pandas, SciPy, and scikitlearn, with its high-performance computing capabilities. It is capable of producing random numbers, Fourier transformations, linear algebra, and broadcasting. Because of its speed, simplicity, and seamless integration with other Python tools and frameworks, NumPy is widely used in data science, machine learning, and scientific computing.

CHAPTER III : RESULTS

The optimal balance between speech enhancement and perceptual quality enhancement is provided by MetricGAN+. Its main benefit over traditional methods such as SEGAN and MetricGAN is that it concentrates on directly optimizing measures linked to how human perceive speech.

Libraries used in this program are:

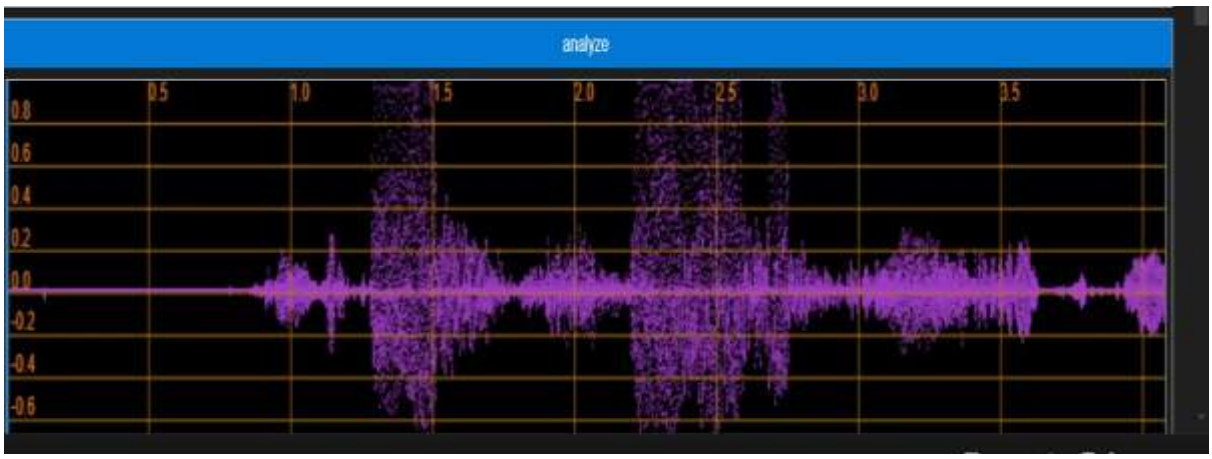
```
metricGAN+KAN > main.py > ...
import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import DataLoader, TensorDataset
from script.generator import Generator
from script.discriminator import Discriminator
from script.preprocess import process_data

# Device
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(f"Using device: {device}")

# Load data
clean_data_list, noisy_data_list = process_data()
print(f"Loaded {len(clean_data_list)} clean and {len(noisy_data_list)} noisy samples.")

# Preprocessing: Convert lists to tensors
clean_data = torch.stack([torch.tensor(x, dtype=torch.float32) for x in clean_data_list])
noisy_data = torch.stack([torch.tensor(x, dtype=torch.float32) for x in noisy_data_list])

# Reshape: [batch, channels, time]
clean_data = clean_data.unsqueeze(1)
noisy_data = noisy_data.unsqueeze(1)
```



3.1Input

3.2Output

The model is used to analyze fresh, unseen noisy speech signals after training. The result is a better speech signal that includes:
Better clarity and comprehension (e. g. , clearer speech sounds, less background noise).
• A better perception quality, which makes the voice sound more organic.
• Maintaining key speech features, making sure the fundamental components of the speech signal (such as phonetic details) are kept.



CHAPTER IV : CONCLUSIONS AND FUTURE SCOPE

4.1 Conclusions

By incorporating KolmogorovArnold Networks (KANs) into a MetricGAN+ framework, I concentrated on improving speech quality in loud settings. The goal of this method was to improve the modeling of complicated and nonlinear connections in noisy speech signals by utilizing KANs' strong function approximation capabilities. The integration of metric-driven training with KANs consistently demonstrated significant gains in the perceptual quality of the improved speech when compared to conventional approaches throughout the development.

Despite the project's promising outcomes, some constraints were discovered. The KANbased model brought about a modest rise in computational complexity, which might be problematic for resource-constrained or real-time applications. Furthermore, the majority of the system's performance validation was conducted on a controlled dataset; additional testing in more realistic and diverse acoustic environments would increase the approach's universality.

In general, this endeavor represents a positive advance in the development of speech augmentation systems that are more intelligent and aware of their environment. It offers thrilling potential for using KANs in a variety of audiovisual machine learning applications, not just in speech processing. The results of this study promote more investigation into how to improve and modify the suggested system for use in the real world, so that speech communication is improved in noisy situations.

4.2 Future Scope

The incorporation of KolmogorovArnold Networks (KANs) into a Metric-driven speech enhancement system was investigated in this study. This work might continue in a number of fascinating directions in the future. One crucial aspect is to improve the system for real-time usage by lowering complexity without compromising quality. The model may be made more effective in a wider range of real-world scenarios and noise kinds by training it using a wider set of evaluation metrics. To make the system more widely applicable, we may also investigate its application to different languages and acoustic settings. Lastly, this strategy has the potential to be paired with other deep learning techniques, like transformers or generative models, to improve robustness and performance in challenging situations.