KAN: Kolmogorov–Arnold Networks Final Report

By: Shrey Salaria IMT2021087

Reference Research paper: https://arxiv.org/pdf/2404.19756

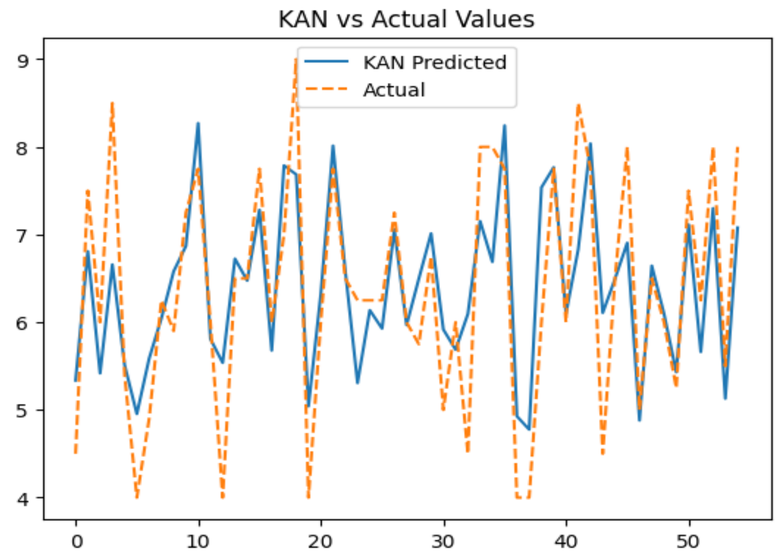
Reason for using KAN:

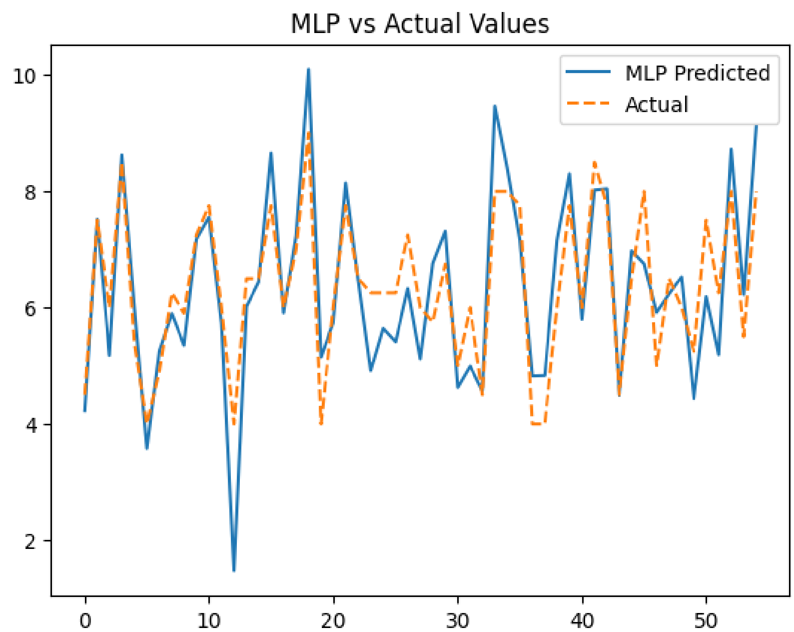
- General MLPs (Multi-Layer Perceptrons), used in Neural Networks (NNs), have several disadvantages, like their “black box” nature. In MLPs, neurons are activated using fixed functions like ReLU or sigmoid, and these activations are passed through linear weight matrices. This approach often leads to lower accuracy despite a higher number of parameters.
- To offer an alternative, a new “KAN” network architecture has been proposed. Instead of using fixed weights, learnable activation functions are placed on the edges or connections between neurons, rather than at the neurons themselves. These functions are parameterized as B-splines in the original implementation, allowing dynamic adaptability, parameter reduction, and potentially better accuracy.

Advantages of KAN over MLP:

Aspect	KANs (Kolmogorov-Arnold Networks)	MLPs (Multi-Layer Perceptrons)
Function Decomposition	Explicit functional decomposition	Implicit approximation via weighted sums
Approximation Process	Exact theorem structure	Generic non-linear layer approximations
Scalability	Limited scalability for high dimensions	Scales well with modern deep learning

Flexibility	Requires precise function construction	Flexible with no prior knowledge required
Ease of Implementation	Complex, often requiring custom setups	Straightforward with widely available tools





Dataset:

The dataset (`repo2001_1.csv`) contains time series data, including economic and financial indicators such as:

- **Date**: The date corresponding to the data entry. It allows tracking of values over time.
- **30yearUSTBill**: The yield or rate of return on a 30-year U.S. Treasury bond. It's a key indicator of long-term interest rates and economic expectations.
- **usdinr**: The exchange rate of the U.S. Dollar (USD) to the Indian Rupee (INR). This indicates the value of one USD in terms of INR, a key metric in forex markets.
- **M3**: Broad money supply, which includes cash, savings deposits, and other near-money assets in an economy. It reflects liquidity and is an economic indicator.
- **FedRate**: The Federal Reserve's interest rate or the federal funds rate. It's a critical rate that influences economic activity and monetary policy in the U.S.
- **forwardpremium**: The premium (or discount) on a forward exchange contract, typically calculated as the difference between the spot exchange rate and the forward rate.
- **wticrude**: The price of West Texas Intermediate (WTI) crude oil, a benchmark for oil pricing. It reflects trends in global energy markets.
- **infl**: Inflation rate, a measure of how much the general price level of goods and services in an economy has increased over a period of time.
- **repo**: The repo rate, often set by central banks, is the rate at which the central bank lends money to commercial banks. It influences interest rates and liquidity in the economy.

Data processing:

The script processes the dataset by the following methods.

- **Converting Dates**: The Date column is converted into a datetime format to facilitate time-based indexing.
- **Visualization**: The repo rates are plotted over time to identify trends, which is a key step in exploratory data analysis (EDA).
- **Rolling Window Feature Engineering**: A rolling window of size 5 is used to capture temporal dependencies and create input-output pairs for supervised learning.

Time Series Forecasting:

The aim is to forecast key variables such as repo. The logic behind the code includes:

- **Preprocessing**: Ensuring the time series data is clean and indexed by date.
- **Feature Engineering**: Creating rolling window datasets (here, size 5) that capture temporal dependencies by grouping sequences of historical data into input-output pairs.
- **Rolling window Method**: Use the first 5 column values to compare the predicted 6th value with the actual 6th value. Repeat this process by shifting the window forward.
- **Forecasting**: Predict future values based on historical patterns and compare it with actual values to judge the effectiveness of the model.
- **Kernel Attention**: Identifying critical moments in the dataset where sharp changes or trends occur, allowing the model to focus on these points.

Forecasting Methods used:

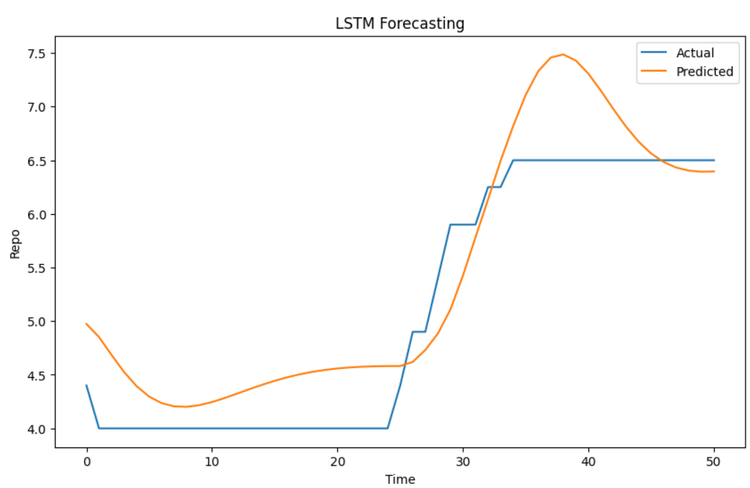
For forecasting, I have taken repo column as the time_series and split the data into training and testing sets. The following observations were made.

ARIMA model forecasting: MSE = 2.56 (Not optimal)

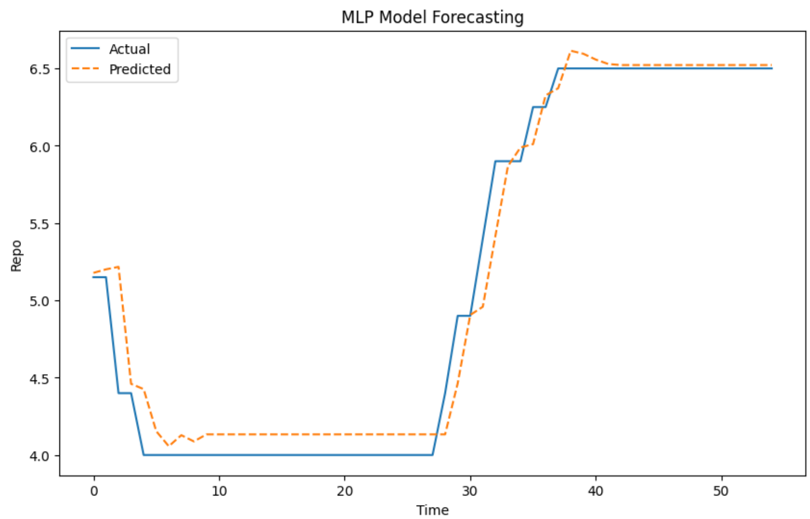
Auto-ARIMA model forecasting: MSE = 1.52 (Improvement but not optimal enough)

Exponential Smoothing forecasting: MSE = 1.25

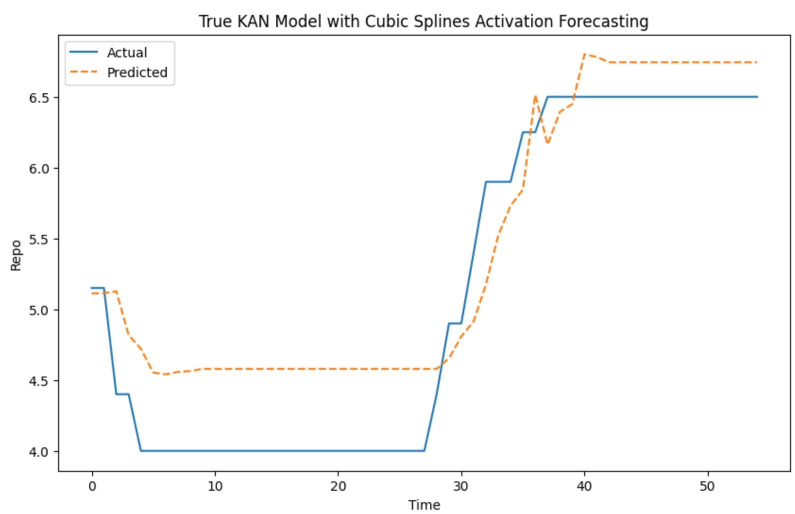
LSTM method forecasting: MSE = 0.24 (Much better)



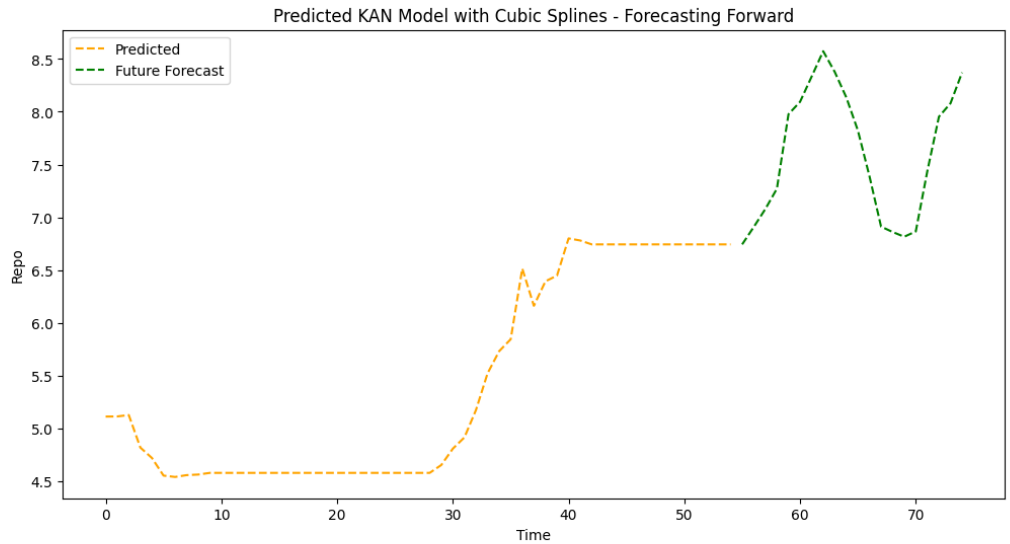
MLP method forecasting: MSE = 0.04 (Most optimised)



KAN method forecasting: MSE = 0.2 (On cubic splines, it outperforms all except MLP. On further optimization, better MSE than MLP can be achieved.)



Future data obtained by KAN forecasting:



Conclusion:

While KAN excels in attention-driven forecasting, its effectiveness depends on several factors as given below.

- **Data Quality**: KAN is sensitive to noisy or incomplete datasets. Like in this case, it is outperformed by MLP.
- **Hyperparameter Tuning**: Careful tuning of parameters like learning rate, attention window size, and kernel design is crucial for optimal performance.
- **Computational Overhead**: Compared to simpler models like MLP, KAN requires more resources and training time.