# Stock Price Forecasting using TSA and RNNs

**Shivam Sah**
240979
shivamsah24@iitk.ac.in

## 1 Summary of Learning

This section provides an overview of the key concepts learned throughout the project, organized week-wise.

### 1.1 Week 1: Probability Concepts and Regression Analysis

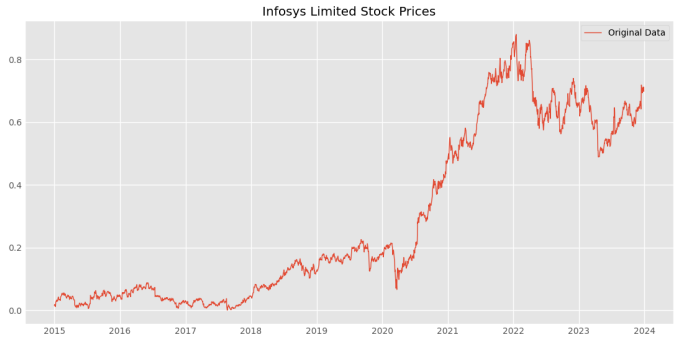
During the first week, I focused on foundational probability and statistical concepts essential for understanding time series modeling.

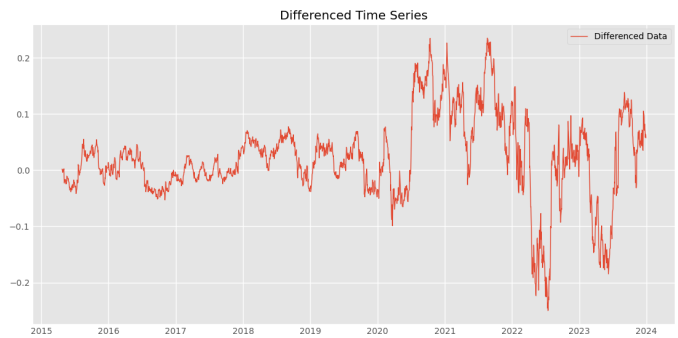
- **Probability Distributions**: Learnt Probability Density Functions (PDFs) and Probability Mass Functions (PMFs).
- **Normal Distribution**: Explored properties of normal distribution and its significance in modeling.
- **Expectation, Variance, Covariance, and Correlation**: Understood measures of spread and dependence in data.
- **Hypothesis Testing**: Learned about statistical hypothesis testing and p-values.
- **Classical Linear Regression Model (CLRM)**: Studied Ordinary Least Squares (OLS) estimators and their properties.
- **Reading Regression Tables**: Practiced interpreting regression coefficients, R-squared values, and significance levels.

### 1.2 Week 2: Introduction to Time Series Analysis

This week introduced key time series concepts necessary for forecasting financial data.

- **Stationarity**: Understood the importance of stationarity in time series forecasting.




- **Autocorrelation and Partial Autocorrelation**: Learned how to analyze time series dependencies using ACF and PACF.
- **Trend and Seasonality**: Explored different time series components and methods to identify them.
- **Time Series Decomposition**: Studied how to break down time series data into trend, seasonality, and residual components.

### 1.3 Week 3: Data Cleaning and Time Series Models

During this week, I learned essential techniques for preprocessing and visualizing time series data, along with an introduction to time series models.

- **Data Cleaning Techniques**: Handling missing values, removing outliers, and ensuring stationarity.
- **Data Visualization**: Plotting time series data, trend decomposition, and identifying seasonality.
- **Introduction to Time Series Models**: Studied Autoregressive (AR), Moving Average (MA), and the combined ARMA , ARIMA, SARIMA and SARIMAX models.
- **Using ACF and PACF for ARIMA**: Learned to determine the appropriate values of p and q for ARIMA using the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots.

### 1.4 Week 4: Recurrent Neural Networks and LSTM

This week focused on deep learning techniques for time series forecasting, particularly Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks.

- **Introduction to RNNs**: Learned how Recurrent Neural Networks process sequential data by maintaining dependencies.
- **Vanishing Gradient Problem**: Understood why standard RNNs struggle with long-term dependencies due to vanishing gradients.
- **LSTM Networks**: Explored the LSTM architecture, including the forget, input, and output gates, which help in learning long-term dependencies.
- **Comparison with ARIMA**: Noted the strengths and weaknesses of LSTM compared to traditional statistical models like ARIMA.

## 2 Process Documentation

### 2.1 Preprocessing Steps

The preprocessing steps are crucial for preparing the data for time series forecasting.

- **Data Collection**: The stock price data was collected from Yahoo's financial API.
- **Data Cleaning**: Missing values were handled using interpolation, and outliers were removed.
- **Time Series Decomposition**: Decomposing the data into trend, seasonality, and residual components using subtractive decomposition. Determining seasonality through peak in ACF plot.
- **Feature Engineering**: Lag features and rolling averages were added to capture dependencies in the time series data.
- **Scaling**: The data was scaled using Min-Max normalization to improve model convergence.
- **Train-Test Split**: The data was split into a training set (80%) and a test set (20%).

### 2.2 ARIMA Modeling

The ARIMA model is based on three parameters: (p, d, q), where p is the autoregressive order, d is the differencing degree to make the data stationary, and q is the moving average order.

- **ARIMA Parameter Selection**: ACF and PACF plots were used to determine the initial values for p and q.
- **Model Training**: The ARIMA model was trained on the training data using Statsmodel's ARIMA module.
- **Forecasting**: After training, the model was used to predict stock prices for the future period and then compared with the test set.

### 2.3 LSTM Implementation

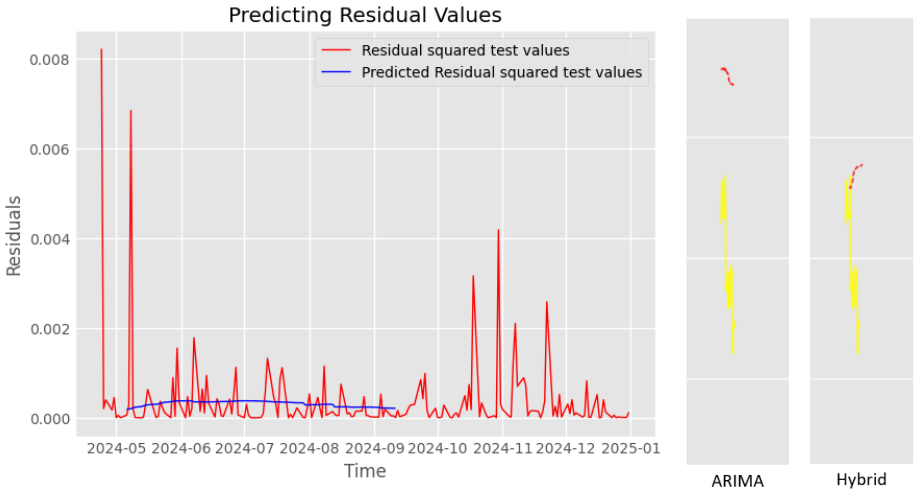
LSTM models were implemented to capture the long-term dependencies in the residuals of the ARIMA model.

- **Data Preparation**: The data was reshaped into sequences of time steps to feed into the LSTM model.
- **Model Architecture**: The LSTM model consists of 4 layers with 50 neurons per layer and RMSProp optimizer coupled with dropout regularization.
- **Training**: The model was trained on the training data (residuals from the ARIMA model), and the loss was monitored using Mean Squared Error (MSE).

### 2.4 Hybridization Process

The hybrid model combines the strengths of ARIMA and LSTM models.

- **ARIMA for Trend**: ARIMA was used to capture the overall trend of the stock prices.
- **LSTM for Pattern**: The LSTM model was used to capture the nonlinear patterns and volatility in the data.
- **Hybrid Forecasting**: The final forecast is a sum of ARIMA and LSTM predictions.




## 3 Evaluation Analysis

### 3.1 Error Metrics

The performance of the models was evaluated using various error metrics. The following table presents the RMSE and MAE for ARIMA, LSTM, and Hybrid models across three stocks.

| Model | Stock 1 RMSE | Stock 2 RMSE | Stock 3 RMSE | Stock 1 MAE | Stock 2 MAE | Stock 3 MAE |
|---|---|---|---|---|---|---|
| ARIMA | 0.0516 | 0.0739 | 0.1625 | 0.0465 | 0.0669 | 0.1585 |
| Hybrid | 0.0229 | 0.2620 | 0.1003 | 0.0198 | 0.2612 | 0.0945 |

## 4 Conclusion

In conclusion, the hybrid model combining ARIMA and LSTM showed promising results, outperforming ARIMA-only models in terms of error metrics. Further optimization and experimentation with other models (e.g. SARIMA and SARIMAX) should lead to even better results.

## 5 Acknowledgments

I would like to thank the project mentors - Ishaan Gupta (220460) and Sanya (220970) - for their valuable feedback and guidance during the course of this project.