KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A++" Grade
Anand Nagar, Krishnankoil, Srivilliputtur (Via), Virudhunagar (Dt) - 626126, Tamil Nadu | info@kalasalingam.ac.in | www.kalasalingam.ac.in

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

A Project Report
on

CURRENCY EXCHANGE RATE FORECASTING

Submitted for partial fulfillment of the requirements for the award of the degree of

MASTER OF SCIENCE IN DATA SCIENCE

BY

R. MOHANKUMAR
(Reg. No. 9922146010)

Under the guidance of

Dr. M. ILAYARAJA, M.C.A., M.Phil., Ph.D.
Assistant Professor
Department of Computer Science and Information Technology
Kalasalingam Academy of Research and Education

KALASALINGAM ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
(ACCREDITED BY NAAC WITH "A++" GRADE)
Anand Nagar
Krishnan Koil- 626126

KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A++" Grade

Anand Nagar, Krishnankoil, Srivilliputtur (Via), Virudhunagar (Dt) - 626126, Tamil Nadu | info@kalasalingam.ac.in | www.kalasalingam.ac.in

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project work entitled **"CURRENCY EXCHANGE RATE FORECASTING"** is a bonafide work carried out by R. MOHANKUMAR (9922146010), in partial fulfillment of the requirements for the award of degree of MASTER OF SCIENCE IN DATA SCIENCE by the KALASALINGAM ACADEMY OF RESEARCH AND EDUCATION, Krishnan Koil, under our guidance and supervision during the year 2023-2024.

Project Guide Head of the Department

Submitted for the project Viva-Voce examination held on ________________

Internal Examiner External Examiner

KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A++" Grade

Anand Nagar, Krishnankoil, Srivilliputtur (Via), Virudhunagar (Dt) - 626126, Tamil Nadu | info@kalasalingam.ac.in | www.kalasalingam.ac.in

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

DECLARATION

This is to certify that the work reported in the present project entitled "**CURRENCY EXCHANAGE RATE FORECASTING"** is a record of work done by us in the Department of Computer Science and Information Technology, Kalasalingam Academy of Research and Education. The report is based on the project work done entirely by us and not copied from any other source.

Place: Krishnan Koil

Date: Signature of the Student,

R. MOHANKUMAR

(Reg. No. 9922146010)

I certify that the declaration made above by the candidate is true

Signature of the Guide
Dr. M. Ilayaraja, M.C.A., M.Phil.,Ph.D.
Assistant Professor
Department of Computer Science & Information Technology
Kalasalingam Academy of Research and Education,
Krishnan Koil

ACKNOWLEDGEMENTS

First of all, we thank the almighty God, who showered blessing on us for succeeding on the project work. Our sincere and heartily thanks go to our parents who are the heart and soul of our life.

We would like to express our wholehearted respects and thanks to our Chancellor **“Ilayavallal” Dr. K. Sridharan** and our honorable Vice President **Dr. S. Shasi Anand**, the Kalasalingam Academy of Research and Education, Krishnan Koil who gave me the golden opportunity to do this project, which also helped me in doing a lot of Research and I came to know about so many new things.

We would like to thanks to our Vice-Chancellor **Dr. S. Narayanan**, the Kalasalingam Academy of Research and Education, Krishnan Koil for having offered the facilities required to complete this project.

We would like to thanks to our Dean School of Computing **Dr. P. Deepalakshmi** for supporting our department

I am also thankful to Head of the department **Dr. K. Bala Subramaniyan** for providing excellent infrastructure and a nice atmosphere for completing this project successfully

I would like to express my sincere gratitude and indebtedness to my project supervisor **Dr. M. Ilayaraja** for his valuable suggestions and interest throughout the course of this project

Finally, I would like to take this opportunity to thank my family for their support through the work. I sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work

Shiash
Focus on Deliverables

13th April 2024

PROJECT COMPLETION CERTIFICATE

This is to certify that **Mr. R. Mohankumar (Reg. No. 9922146010)** Student of **M.Sc.,(Data Science) Kalasalingam Academic Research and Education, Krishnankoil** has successfully completed the industrial project titled **"Currency Exchange Rate Forecasting"** in the **Data Science** platform from **December 2023 to June 2024** in our company. During the period, he had been exposed to different processes and found to be Punctual, Hard Working and Inquisitive.

We wish him every success in life and career.

For Shiash Info Solutions Private Limited

SHIASH INFO SOLUTIONS PRIVATE LIMITED ★ CHENNAI 600 119

Ashwini Kanniyappan

Manager – Human Resources

Shiash Info Solutions Private Limited
#51, Level 3, Tower C, Rattha TEK Meadows, Old Mahabalipuram Road,
Sholinganallur, Chennai – 600 119, Tamil Nadu, India
+91 80158 07428 info@shiash.com

ACKNOWLEDGEMENTS

First of all, we thank the almighty God, who showered blessing on us for succeeding on the project work. Our sincere and heartily thanks go to our parents who are the heart and soul of our life.

We would like to express our wholehearted respects and thanks to our Chancellor **"Ilayavallal" Dr. K. Sridharan** and our honorable Vice President **Dr. S. Shasi Anand**, the Kalasalingam Academy of Research and Education, Krishnan Koil who gave me the golden opportunity to do this project, which also helped me in doing a lot of Research and I came to know about so many new things.

We would like to thanks to our Vice-Chancellor **Dr. S. Narayanan**, the Kalasalingam Academy of Research and Education, Krishnan Koil for having offered the facilities required to complete this project.

We would like to thanks to our Dean School of Computing **Dr. P. Deepalakshmi** for supporting our department

I am also thankful to Head of the department **Dr. K. Bala Subramaniyan** for providing excellent infrastructure and a nice atmosphere for completing this project successfully

I would like to express my sincere gratitude and indebtedness to my project supervisor **Dr. M. Ilayaraja** for his valuable suggestions and interest throughout the course of this project

Finally, I would like to take this opportunity to thank my family for their support through the work. I sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work

ABSTRACT

The impact of fluctuations in currency exchange rates on a nation's economy underscores the critical need for precise forecasting within the realm of the money market. This research delves into the efficacy of various hybrid models, namely **LSTM-ARIMA, LSTM-SVR, LSTM-RF,** and **SVR-RF,** in their capacity to predict FOREX fluctuations. Leveraging datasets containing information on **EUR/INR, NZD/USD**, and **USD/INR**, this study meticulously evaluates the performance of each model. The findings consistently demonstrate that the **LSTM-SVR** hybrid model exhibits superior predictive capabilities compared to its counterparts.

By employing the adaptive learning rate method (**ADAM**) optimization approach, this research endeavors to ascertain the optimal weights for the suggested model. This innovative methodology not only enhances precision but also provides a more nuanced understanding of foreign exchange rate estimation. The commendable performance of the LSTM-SVR hybrid model underscores its potential as a robust forecasting tool for discerning currency exchange rate movements. These insights carry significant implications for policymakers, financial institutions, and investors alike, as they navigate the intricacies of the global currency market with heightened accuracy and confidence.

TABLE OF CONTENTS

	Certificate	**i**
	Declaration	**ii**
	Acknowledgement	**iii**
	Project Completion Certificate	**iv**
	Abstract	**v**
	List Of Tables	**viii**
	List Of Figures	**x**
	List Of Abbreviations	**xii**

Chapter No.	Title	Page No.
1.	**INTRODUCTION**	
	1.1 Overview of the Project	1
	1.2 Challenges	3
	1.3 Motivation of the Proposed Work	5
	1.4 Problem statement	6
2.	**REVIEW OF THE LITERATURE**	7
3.	**SYSTEM ANALYSIS**	
	3.1 Existing System	11
	3.1.1 Disadvantages	11
	3.2 Proposed System	12
	3.2.1 Architecture of The Proposed Model	12
	3.2.2 Proposed System advantages	13
4.	**SYSTEM IMPLEMENTATION**	
	4.1 Needed Specification	14
	4.2 Tools used	14
	4.3 Data Preprocessing	20
	4.4 Algorithms Overview	21
	4.4.1 LSTM-ARIMA	21
	4.4.2 LSTM-SVR	24
	4.4.3 LSTM-RF	27
	4.4.4 SVR-RF	30

	4.5 Training, Testing and Validation	33
	4.6 Visualization	34
	4.7 Model Development Summary	35
5.	**RESULTS AND DISCUSSION**	
	5.1 Experimental Setup	36
	5.1.1 LSTM-ARIMA	36
	5.1.2 LSTM-SVR	40
	5.1.3 LSTM-RF	44
	5.1.4 SVR-RF	49
	5.2 Comparison and the Best Model among the Models	54
6.	**CONCLUSION**	
	6.1 Future Works	56
7.	**REFERENCES**	58
	APPENDICES	
	A. Sample Code	59
	B. Sample Screenshots	65

LIST OF TABLES

No.	Table No.	Table Name	Page No
1.	5.1	LSTM Model Training Parameters	35
2.	5.2	ARIMA Model Training Parameters	36
3	5.3	EUR/INR Evaluation Results With LSTM-ARIMA	37
4	5.4	NZD/USD Evaluation Results With LSTM-ARIMA	37
5	5.5	USD/INR Evaluation Results With LSTM-ARIMA	37
6	5.6	LLSTM Model Training Parameters	40
7	5.7	SVR Model Training Parameters	40
8	5.8	EUR/INR Evaluation Results With LSTM-SVR	41
9	5.9	NZD/USD Evaluation Results With LSTM-SVR	41
10	5.10	USD/INR Evaluation Results With LSTM-SVR	41
11	5.11	LSTM Model Training Parameters	44
12	5.12	RF Model Training Parameters	44
13	5.13	EUR/INR Evaluation Results With LSTM-RF	45
14	5.14	NZD/USD Evaluation Results With LSTM-RF	46
15	5.15	USD/INR Evaluation Results With LSTM-RF	46
16	5.16	SVR Model Training Parameters	49
17	5.17	RF Model Training Parameters	50

18	5.18	EUR/INR Evaluation Results With SVR-RF	51
19	5.19	NZD/USD evaluation results with SVR-RF	51
20	5.20	USD evaluation results with SVR-RF	51

LIST OF FIGURES

No.	Figure No.	Figure Name	Page No.
1	3.1	Workflow diagram for proposed model	12
2	4.1	Keras & TensorFlow	16
3	4.2	NZD/USD rate of dataset	18
4	4.3	EURINR rate of the dataset	18
5	4.4	USDINR rate of the dataset	19
6	4.5	Minmax scaler formula	20
7	4.6	Train test split	20
8	4.7	Data flow diagram	34
9	5.1	EUR/INR actual and predicted price visualization with LSTM-ARIMA	38
10	5.2	NZD/USD actual and predicted price visualization with LSTM-ARIMA	38
11	5.3	USD/INR actual and predicted price visualization with LSTM-ARIMA	39
12	5.4	EUR/INR actual and predicted price visualization with LSTM-SVR	42
13	5.5	NZD/USD actual and predicted price visualization with LSTM-SVR	42
14	5.6	USD/INR actual and predicted price visualization with LSTM-SVR	43
15	5.7	EUR/INR actual and predicted price visualization with LSTM-RF	47
16	5.8	NZD/USD actual and predicted price visualization with LSTM-RF	47
17	5.9	USD/INR actual and predicted price visualization with LSTM-RF	48

18	5.10	EUR/INR actual and predicted price visualization with SVR-RF	52
19	5.11	NZD/USD actual and predicted price visualization with SVR-RF	52
20	5.12	USD/INR actual and predicted price visualization with SVR-RF	53
21	5.13	Evaluation Metrics on EUR/INR	53
22	5.14	Evaluation Metrics on NZD/USD	54
23	5.15	Evaluation Metrics on USD/INR	54

LIST OF ABBREVIATIONS

ML	Machine Learning
GDP	Gross Domestic Product
SVR	Support Vector Regression
LSTM	Long Short-Term Memory
RSI	Relative Strength Index
MSE	Mean Squared Error
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
XAI	Explainable Artificial Intelligence
CNN	Convolution Neural Network
PR	Polynomial Regression
ARIMA	Autoregressive Integrated Moving Average
CART	Classification and Regression Trees
RF	Random Forest
ANN	Artificial Neural Network
GA	Genetic Algorithm
FOREX	Foreign Exchange
RBF	Radial Basis Function

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

In the dynamic and intricate world of finance, accurately predicting currency exchange rates holds immense significance for investors, businesses, and governments. While inherent uncertainties and complexities limit the possibility of perfect foresight, machine learning (ML) algorithms offer powerful tools to navigate these challenges and make informed decisions. This overview delves into the captivating interplay between ML and exchange rate forecasting, exploring its potential, nuances, and ongoing advancements.

1.1.1The Currency Conundrum: Why Forecasting Matters

Exchange rates, representing the relative value of two currencies, fluctuate constantly on the foreign exchange (forex) market. These fluctuations are influenced by a multitude of factors, including:

- Economic Fundamentals: Interest rates, inflation, gross domestic product (GDP), trade balances, and fiscal policies
- Market Psychology: Sentiment, risk aversion, and investor expectations
- Geopolitical Events: Wars, elections, policy changes, and natural disasters
- Technical Factors: Chart patterns, support and resistance levels, and algorithmic trading

Understanding these forces and predicting their impact on exchange rates empowers various stakeholders:

- Investors: Make informed decisions on currency allocation, hedging strategies, and foreign investments.
- Businesses: Manage risks associated with international trade, optimize pricing, and navigate economic fluctuations.
- Governments: Implement monetary policy interventions, stabilize exchange rates, and foster economic growth.

1.1.2Machine Learning: Unveiling Patterns in Financial Chaos

Machine learning shines in its ability to identify subtle patterns and relationships within vast datasets, particularly valuable for the inherently noisy and intricate forex market. Popular ML algorithms utilized for exchange rate forecasting include:

- Linear Regression: Establishes linear relationships between historical data and exchange rates.
- Support Vector Regression (SVR): Maps data points to high-dimensional features, handling non-linearity and outliers.
- Random Forest: Ensembles multiple decision trees, offering robustness and flexibility.
- Neural Networks: Mimic the human brain's structure, learning complex patterns through layered processing.
- Long Short-Term Memory (LSTM) Networks: Ideal for analyzing sequential data, capturing long-term dependencies in exchange rate time series.

1.1.3 Fuelling the Models: Data is the Key

The quality and relevance of training data significantly impact the effectiveness of ML models in exchange rate forecasting. Common data sources include:

- Historical Exchange Rates: Daily, hourly, or even higher-frequency data for the target currency pair.
- Macroeconomic Indicators: Compiled by central banks and international organizations.
- Market Sentiment Data: Social media analysis, surveys, and news sentiment analysis.
- Technical Indicators: Moving averages, Bollinger Bands, and Relative Strength Index (RSI).

Preprocessing steps like cleaning, normalization, and feature engineering are crucial to prepare the data for ML algorithms.

1.1.4 Evaluating Performance: Finding the Right Metric

Measuring the accuracy and quality of exchange rate forecasts is essential. Common metrics include:

- Mean Squared Error (MSE): Average squared difference between predicted and actual rates.
- Mean Absolute Error (MAE): Average absolute difference between predicted and actual rates.

- Root Mean Squared Error (RMSE): Square root of MSE, penalizing larger errors more heavily.
- Directional Accuracy: Percentage of correctly predicted up/down movements.

No single metric is perfect, and the choice depends on the specific forecasting task and risk tolerance.

1.1.5Beyond Accuracy: Embracing Uncertainty

While ML algorithms can lead to impressive results, inherent limitations exist:

- Model Overfitting: Models may learn data noise instead of genuine patterns, leading to poor generalization.
- Data Availability: Historical data may not fully capture future events or sudden shifts in dynamics.
- Interpretability: Complex models like neural networks can be difficult to understand, hindering debugging and trust.

Despite these challenges, combining ML with fundamental analysis, technical analysis, and human expertise can enhance forecasting accuracy and provide valuable insights. Ongoing research explores:

- Ensemble Methods: Combining predictions from multiple models for improved robustness.
- Deep Learning Architectures: Leveraging advancements like generative models and attention mechanisms.
- Explainable AI (XAI): Making models more interpretable for better decision-making.

1.2 CHALLENGES

While machine learning (ML) offers promising tools for currency exchange rate forecasting, it's important to acknowledge the significant challenges involved. Here are some key issues that limit the effectiveness and reliability of ML-based forecasts:

1.2.1 Data Issues:

- Data Availability: Historical data may not capture all relevant factors or future events, impacting model generalizability.
- Data Quality: Missing values, outliers, and noise in data can mislead models and lead to inaccurate predictions.

- Feature Engineering: Selecting and engineering relevant features is crucial, but choosing the wrong ones can negatively impact performance.

1.2.2 Model Challenges:

- Overfitting: Models overly tuned to training data may not perform well on unseen data, requiring careful regularization techniques.
- Interpretability: Complex models like deep neural networks can be difficult to understand, hindering trust and hindering explanation of predictions.
- Algorithmic Bias: Biases in training data or algorithms can lead to discriminatory or unfair predictions.

1.2.3Market Dynamics:

- Non-linearity: Exchange rates exhibit complex non-linear relationships with influencing factors, which simpler models like linear regression may struggle to capture.
- High Dimensionality: Incorporating numerous economic, market, and geopolitical factors leads to high-dimensional feature spaces, posing challenges for model training.
- Market Psychology: Sentiment, risk aversion, and unpredictable events like news announcements can be challenging for models to incorporate reliably.

1.2.4Evaluation and Interpretation:

- Choosing the Right Metrics: Selecting the appropriate metrics (e.g., MAE, RMSE, directional accuracy) depends on the specific forecasting task and risk tolerance.
- Calibration and Uncertainty Quantification: Quantifying prediction uncertainty allows for more informed decision-making, but remains a challenge in many ML models.
- Real-world Applicability: Models often perform well in back testing but may underperform in live trading environments due to factors like transaction costs and slippage.

1.2.5 Additional Challenges:

- Data Security and Privacy: Protecting sensitive financial data used for training ML models is crucial.
- Computational Cost: Training and running complex ML models can be computationally expensive and resource-intensive.
- Regulation and Ethical Considerations: Ensuring fairness, transparency, and responsible use of ML in financial markets requires careful consideration of ethical and regulatory frameworks.

1.3 MOTIVATION OF THE PROPOSED WORK

The motivation behind currency exchange rate forecasting using machine learning techniques stems from several key factors:

- Financial Markets Volatility: Currency exchange rates are highly volatile and subject to various economic, geopolitical, and market factors. Accurate forecasting of these rates is crucial for investors, traders, businesses, and policymakers to make informed decisions and mitigate risks associated with currency fluctuations.
- Profitability and Risk Management: Successful currency trading relies on accurately predicting exchange rate movements. Machine learning models offer the potential to capture complex patterns in currency data, allowing traders to identify profitable trading opportunities and manage risk more effectively.
- Information Integration: Machine learning algorithms can incorporate a wide range of data sources, including economic indicators, news sentiment, market sentiment, and technical indicators. By integrating diverse information, these models can potentially improve the accuracy of exchange rate forecasts compared to traditional econometric models.
- Automation and Scalability: Machine learning techniques enable the automation of the currency exchange rate forecasting process, making it more scalable and efficient. This automation frees up time for analysts and traders to focus on higher-level strategic decision-making tasks.
- Advancements in Technology: Recent advancements in machine learning algorithms, computing power, and data availability have made it increasingly feasible to develop sophisticated models for currency exchange rate forecasting. These advancements have fuelled interest and research in applying machine learning to financial markets, including currency forecasting.
- Competitive Advantage: Accurate and timely exchange rate forecasts can provide a competitive advantage to financial institutions, hedge funds, and other market participants. By leveraging machine learning techniques, these entities can potentially gain insights into market dynamics and outperform competitors in trading and investment activities.

1.4 PROBLEM S STATEMENT

- Accuracy: Current forecasting methods often lack the accuracy and reliability needed for effective decision-making in the volatile foreign exchange market.
- Complexity: The dynamics of exchange rates are influenced by a multitude of interacting factors, making their prediction a highly complex task.
- Incorporating diverse data: Traditional models often fail to capture the full range of relevant information, including economic indicators, political events, market sentiment, and non-linear relationships.
- Adaptability: Exchange rates exhibit varying patterns over time, necessitating forecasting models that can adapt to changing market conditions.

CHAPTER 2

REVIEW OF LITRATURE

Durairaj and Krishna Mohan (2022) present an innovative strategy, the Chaos+CNN+PR hybrid model, to address the complexities of financial time series forecasting. Initially, the presence of chaos within the financial time series is evaluated, with Chaos Theory utilized to model it appropriately. Subsequently, the CNN component of the hybrid model generates preliminary predictions based on the modelled time series. The error series derived from these CNN predictions is then employed in Polynomial Regression (PR) to generate error predictions. By integrating these error predictions with the initial CNN predictions, the hybrid model produces final predictions. Evaluation of this approach across various financial time series, including foreign exchange rates, commodity prices, and stock market indices, demonstrates its superiority over traditional models such as ARIMA, Prophet, CNN, CART, RF, Chaos+CNN, Chaos+CART, and Chaos+RF, as indicated by metrics such as MSE, MAPE, Dstat, and Theil's U. Furthermore, the proposed hybrid model's adaptability extends to both financial and non-financial time series, suggesting potential applications beyond its initial scope. Additionally, the regression problem addressed in this study has the potential to be converted into a classification problem, with relevant methodologies from previous research provided for further exploration (Durairaj & Krishna Mohan, 2022).

Sarangi and colleagues (2020) delve into FOREX trend analysis, focusing specifically on the Indian Rupees (INR) versus U.S. Dollars (USD) currency exchange rate, utilizing machine learning techniques. Time series analysis emerges as a crucial tool for deciphering historical data trends and projecting future trajectories, facilitating informed strategic decision-making for organizations. This study employs two distinct approaches: a simplistic Artificial Neural Network (ANN) model and a hybrid model amalgamating ANN with Genetic Algorithm (ANN-GA) optimization for the ANN weight matrix. Comparative analysis of both methodologies is conducted based on Root Mean Square Error (RMSE) values derived from their respective implementations.

Traditional approaches utilizing artificial neural networks with backpropagation algorithms encounter the challenge of local minima, wherein the network gets trapped in error surfaces that do not represent minimal error, leading to instability. This instability primarily stems from the ANN weights. Integration of a genetic algorithm addresses this challenge by optimizing the

ANN weights, thus mitigating the local minima problem and enhancing network accuracy through a population-based approach.

Empirical findings reveal a significant enhancement in predictive performance upon transitioning from the simple ANN with backpropagation learning to the ANN-GA hybrid model. The introduction of the ANN-GA network results in a substantial decrease in RMSE, from 0.39 to 0.018930. This improvement is attributed to the optimization of the weight matrix facilitated by the genetic algorithm, effectively circumventing the issue of local minima. However, determining the appropriate population size poses a challenge in the ANN-GA framework, necessitating a trial-and-error approach for resolution.

In conclusion, while ANN demonstrates efficacy in currency exchange rate forecasting, its potential is significantly augmented through hybridization with a genetic algorithm. The introduction of a hybrid model for optimizing the ANN weight matrix yields efficient and accurate results, underscoring the potential of integrating machine learning techniques for FOREX trend analysis (Sarangi et al., 2020).

Madhumita Panda, Surya Panda, and Pattnaik (2022) venture into the realm of foreign exchange rate forecasting, a pivotal domain for traders seeking insights into currency performance against global counterparts. Their study introduces an innovative approach utilizing a Convolutional Neural Network (CNN) model integrated with a random forest regression layer to forecast future closing prices. The effectiveness of this novel model is assessed across three major currency pairs: AUD/JPY, NZD/USD, and GBP/JPY, leveraging data spanning from January 2, 2001, to May 31, 2020, for AUD/JPY and GBP/JPY, and from January 1, 2003, to May 31, 2020, for NZD/USD. Comparative analysis against Autoregressive Integrated Moving Average (ARIMA), Multi-Layer Perceptron (MLP), and Linear Regression (LR) models demonstrates the superiority of the CNN with Random Forest model in terms of prediction performance metrics such as R^2, MAE, and RMSE.

The experimental results underscore the prowess of the proposed model in forecasting exchange rates across various time horizons, from one to seven months ahead. Noteworthy is the CNN with Random Forest model's ability not only to accurately predict closing prices but also to offer valuable guidance for investors navigating the Forex market. With average R^2 values ranging from 0.9616 to 0.9640 across the three currency pairs, the suggested model emerges as the preferred choice among its counterparts.

In conclusion, the findings of this study bear significant implications for policymakers and investors, providing a robust framework for enhancing decision-making in the foreign exchange market. Furthermore, the authors advocate for future research directions, including the application of the model to additional currency pairs and the development of dynamic datasets for more efficient exchange rate predictions, thus paving the way for further advancements in currency forecasting methodologies (Madhumita Panda, Surya Panda, and Pattnaik, 2022).

Mohammadi (2019) explores the realm of currency exchange rate forecasting, a crucial facet of contemporary financial markets. This Master's thesis aims to investigate the feasibility of predicting future currency prices based on historical data in the FOREX market. Four machine learning models—Backpropagation, Radial Basis Function (RBF), Long Short-Term Memory (LSTM), and Support Vector Regression (SVR)—are scrutinized for forecasting tasks. These models are developed and trained using Python, leveraging three datasets from the Swiss Duckascopy banking group, focusing on currency pairs EUR/USD, USD/JPY, and USD/TRY. Subsequently, the models undergo rigorous training, testing, and comparison to discern their respective strengths and weaknesses. Performance evaluation techniques such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) are employed to determine the optimal model among them.

The experimental findings highlight SVR's superior performance over the other three techniques, with Backpropagation in neural networks exhibiting the lowest performance. This emphasizes the importance of deploying advanced machine learning techniques for precise currency exchange rate forecasting.

S.K. Chandra, M. Sumathi, S.N. Sivanandam (2016) venture into the realm of foreign exchange rate forecasting. LSTM networks have demonstrated proficiency in handling time series data for tasks such as classification, analysis, and prediction. Particularly for short-term forecasting, approaches like ARIMA, LSTM, and Recurrent Neural Networks (RNNs) have been employed. Various window sizes ranging from 5 to 55 days were utilized for prediction, with the 22-day window exhibiting an average accuracy of 71.76 percent, proving to be optimal for short-term validation. Given the volatile and intricate nature of the forex market, investors are continuously seeking more accurate methods. The model proposed in this paper achieves an impressive average accuracy of 93% for one-month ahead prediction, categorized as short-term forecasting. Additionally, for projecting prices 10 and 30 minutes ahead, the GRU-LSTM approach is utilized, evaluated across key currency pairs such as EUR/USD, GBP/USD, USD/CAD, and USD/CHF. Comparisons with other models, including GRU, LSTM, and statistical models based on simple moving averages (SMA), indicate superior performance of the GRU-LSTM model, particularly in terms of R2. Furthermore, our proposed CNN with Random Forest (CNN-RF) model consistently outperforms alternative approaches across all datasets, as assessed by metrics such as MAE, MSE, and RMSE. In a separate experiment, three alternative models leveraging Google Trends and macroeconomic data were employed to predict the value of the Ghanaian Cedis against USD, British Pounds, and Euros for the next 30 days. The LSTM model exhibited adeptness in managing exchange rate data variance compared to the other two models, although it's acknowledged that Google Trends might not always accurately predict changes in the Ghanaian cedi's exchange rate against all other currencies.

J.F. Pfahler (2022) present an innovative strategy, the Support Vector Regression (SVR) method is employed for short-term financial time series forecasting, concentrating on predictions ranging from one to four days ahead. Specifically designed for short-term forecasting, the proposed PCA-ICA-SVR model enables accurate prediction of stock values with minimal error. Notably, this approach has been adopted by various researchers for currency exchange rate prediction as well.

Several machine learning techniques have been explored for multi-currency exchange rate prediction, including Support Vector Regression (SVR), Neural Networks (NN), and Long Short-Term Memory (LSTM) networks, which are a type of deep learning model with hidden layers. These models have been applied to forecast exchange rates between major global currencies.

While initial research suggests promising results, most methods suffer from limitations in long-term prediction accuracy. For instance, a Deep Belief Network (DBN) applied to INR/USD and CNY/USD pairs exhibited decreasing accuracy as the forecast horizon increased. This highlights the ongoing challenge of developing models effective for long-term currency exchange rate prediction.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING METHOD

In today's interconnected global economy, currency exchange rate forecasting plays a pivotal role in decision-making for businesses, investors, financial institutions, and governments alike. The ability to anticipate fluctuations in exchange rates can profoundly impact trade flows, investment decisions, and overall economic stability. To meet this demand for accurate and timely predictions, various methodologies and systems have been developed, spanning from traditional econometric models to cutting-edge machine learning algorithms.

The existing system in currency exchange rate forecasting encompasses a diverse array of approaches, each tailored to capture different aspects of the complex dynamics driving exchange rate movements. These methodologies leverage historical exchange rate data, economic indicators, market sentiment analysis, and computational techniques to generate forecasts across different time horizons, from short-term fluctuations to long-term trends.

3.1.1 DISADVANTAGES

- **Data Limitations:** Currency exchange rate forecasting relies heavily on historical data for model training and validation. However, historical data may not always be representative of future market conditions, especially during periods of economic upheaval or structural shifts in the global economy. Additionally, data quality issues, such as missing or inaccurate data, can compromise the reliability of forecasts.
- **Overfitting and Underfitting:** Machine learning models are susceptible to overfitting, where the model learns noise in the training data rather than capturing underlying patterns. Conversely, underfitting occurs when the model is too simplistic to capture the complexity of exchange rate dynamics, leading to poor forecast accuracy.
- **Performance Evaluation Challenges:** Assessing the performance of forecasting models can be challenging due to the inherent uncertainty in exchange rate movements and the lack of standardized evaluation metrics. Consequently, users may have difficulty discerning the effectiveness of different forecasting approaches and selecting the most suitable model for their needs.

3.2 PROPOSED METHOD

The proposed hybrid method represents a novel approach to currency exchange rate forecasting, aiming to synergistically integrate the predictive capabilities of Timeseries models and machine learning algorithms. By leveraging the complementary strengths of these methodologies, the hybrid method seeks to overcome the limitations inherent in individual approaches, offering enhanced accuracy, robustness, and adaptability in forecasting exchange rates across different time horizons.

At its core, the hybrid method operates on the principle of diversity, recognizing that different forecasting techniques excel in capturing distinct aspects of market dynamics. By fusing econometric models' theoretical rigor with machine learning algorithms' flexibility and predictive power, the hybrid method aims to construct a comprehensive forecasting framework capable of navigating the complexities of global currency markets with greater precision and reliability.

3.2.1 ARCHITECTURE OF THE PROPOSED MODEL

The Architecture of the proposed model is the computational design that defines the structure and/or behaviour of a system. An architecture description is a formal description of a system, organized in a way that supports reasoning about the structural properties of the system. It defines the system components or building blocks and provides a platform which products can be procured, and systems developed, that will work together to implement the overall system.

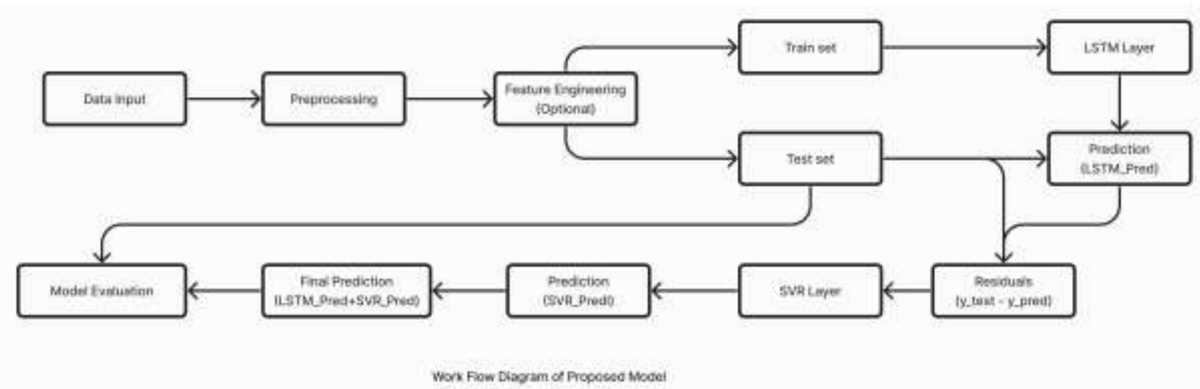


Fig 3.1 workflow diagram for proposed model

The proposed approach for currency exchange rate forecasting leverages a combination of LSTM and SVR models. First, we begin by downloading historical exchange rate data from Google Finance, focusing on the closing price for each date. After preprocessing the data, feature engineering may be employed to identify and incorporate additional relevant factors. The data is then split into training and testing sets. An LSTM model is trained on the training set to capture temporal dependencies in the closing prices. Subsequently, the residuals, the difference between the actual and LSTM-predicted values for the testing set, are calculated.

These residuals are used to train a separate SVR model, potentially capturing non-linear relationships not fully addressed by the LSTM. Finally, the final predictions are obtained by summing the individual predictions from the LSTM and SVR models. The effectiveness of this approach is evaluated by comparing the mean squared error between the actual closing prices in the testing set and the final predictions.

This approach can be build other hybrid models.

3.2.2 ADVANTAGES

- Improved Forecast Accuracy: By combining the strengths of different forecasting methodologies, the hybrid method can often achieve higher forecast accuracy compared to individual approaches. This synergy allows for better capturing of complex patterns and relationships in exchange rate data, leading to more reliable predictions.
- Enhanced Decision Support: The hybrid method provides decision-makers with more informed and reliable forecasts, enabling them to make better-informed decisions regarding currency trading, investment allocation, and risk management. This improved decision support can lead to more efficient resource allocation and better overall financial performance.
- Robustness to Model Limitations: Timeseries models and machine learning algorithms each have their limitations and assumptions. The hybrid method mitigates these limitations by leveraging diverse modelling techniques, thereby enhancing the robustness of forecasts. In situations where one method may struggle, the other may provide valuable insights, resulting in more resilient predictions.

CHAPTER 4

SYSTEM IMPLEMENTATION

This chapter is designed to describe the methods and tools used to forecast the currency exchange rates in the Forex market. Firstly, the used tools are presented with clarifications of how they being used within the research. Then the data cleaning, preprocessing and algorithms are discussed with a brief conclusion and summary of the mentioned issues at the end.

4.1 Needed Specification

4.1.1 Hardware Configuration

Processor: Octo Core Processor

Processor Speed: 3.06 GHz

RAM: 8 GB (minimum)

Hard Disk Drive: 250 GB (minimum)

Monitor: "16" inches

4.1.2 Software Configuration

Operating System: Windows 10 (minimum)

Programming Language: python

Integrated Development Environment (IDE): Visual Studio Code

Data Analysis and Machine Learning Libraries:

- NumPy
- Pandas
- Matplotlib and Seaborn
- Scikit-learn
- TensorFlow
- Statsmodels
- Arima

4.2 Tools Used

As of every research this study used some tools for doing experiments and developing prediction models. Many tools being used for instance the author choose to use python as the programming language of the model development or historical data of different currencies to examine their signals or different libraries for which they make us able to use already implemented algorithms in various massive ways. The following is a brief report of the manners of these tools.

4.2.1 Python

With so many thanks to Guido van Rossum who created python, it is a general-purpose programming language used for different platforms like web, computer GUI, mathematic and huge scientific applications. Python is well known by its simplicity, it actually built on bases of easy programming which is sensitive with blank spaces. Despite that python was built for kids, nowadays its bits so many strong programming languages in the market. It would not be unfair if we say it can do whatever you are able to do with every programming language. Lately, python got popular in the area of artificial intelligence and machine learning because of its handy bunch of libraries which are growing within developers and experts in this field.

Some of the experts in these fields are coming from economic or other technical backgrounds though the most convinced programming language that they can start with is python, with its robustness at the same time. Basically, the above-mentioned issues and the author’s self-interest caused the models to be developed in python. Many libraries being used in this study and these are as follows.

4.2.1.1 NumPy

This open-source library is doing the computing, it supports multi-dimensional arrays and matrices with so many functions which make working easy with arrays and matrices. Since arrays can increase the speed and efficiency in data analysis tasks this library can help a data scientist a lot. In the forecasting of these data, NumPy is used to store data in arrays. Since the prediction models require NumPy arrays as their parameters to operate fast and decrease the training and prediction time.

4.2.1.2 Pandas

This is although an open-source library which provides data structures and data analysis tools. The important note about pandas is its high performance and easy to use especially for manipulating operations in numerical tables and time series data. Though pandas used to store the currency data in data Frame where it then divided in X and Y dimensions and made it ready for scaling and other preprocessing operations.

4.2.1.3 Matplotlib

Matplotlib is a plotting library that is used to create variety of graphs for different purposes. The salience of matplotlib is its easiness in use. A good quality plot can be produced with few lines of code. So matplotlib is used to plot the output historically and also to show the accuracy and difference between the prediction and real output.

4.2.1.4 Scikit-Learn

Scikit-Learn is another python library which is mostly focused on machine learning. The library supports different classification, regression and clustering algorithms namely SVM, Random Forest, Gradient boosting, K-means and DBSCAN. It is based on NumPy and SciPy and interoperates with them fairly. It makes the job of algorithm creation easier for those who implement those algorithms. In this study Scikit-Learn performed important tasks like scaling which is normalizing the data between 0 and 1, it dynamically split data into two, train and test portions. Although, Scikit-Learn is used for calculation of the mean squared error for performance evaluations.

4.2.1.5 Keras & TensorFlow

TensorFlow is also a python library developed by google, for the first it was used internally by google brain team but then it released open source for the public. Basically, it is for mathematical operations and also uses widely for machine learning nowadays. TensorFlow made the implementation of ANNs so easy and convenient for machine learning practitioners. The Data in TensorFlow is called tensor. These tensors can do a variety of tasks such as normalization, vectorization or classification.

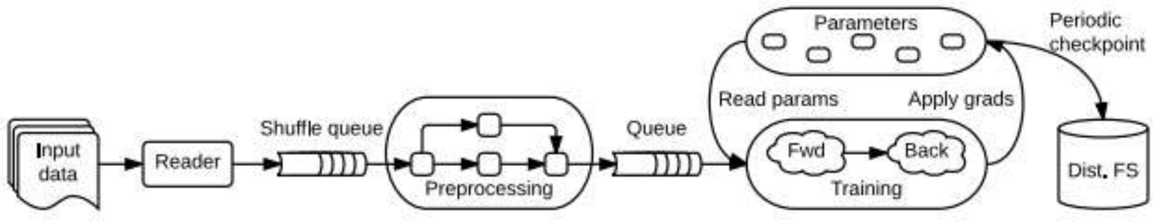


Fig 4.1 Keras & TensorFlow

Keras is also an open-source python library which is developed by a google engineer names François Chollet. It is a special purpose library for neural networks and deep neural networks for building deep learning models. Its creation is basically on the top of TensorFlow, Microsoft Cognitive Toolkit or Theano. It is user friendly and fast for implementing experiments in neural network. The usage is growing and developers use Keras for time saving and capable functionalities. Not to be forgotten that it is extensible that you can add any functionality that you think is not included in the package. Therefore, for developing models and experiments for this study, Keras is used for building the neural network models namely LSTM, RBF, and Backpropagation. Keras has so many built in functions that are used for developing the study models.

4.2.2 Jupyter Notebook

Jupyter Notebook is a web application that provides creation and modification of live codes, visualizations, equations and plaintexts. This open-source notebook supports multiple programming languages. It uses for numerical simulation, information visualization, machine learning, statistical modelling, and many more functionalities. The author used this notebook for writing readable codes for developing models and visualizing the results and accuracy errors.

4.2.3 Computer

For this research, the computer that is used for training and testing the models has the following specification: Lenovo Thinkbook,16 GB RAM, AMD Ryzen7 Octa Core Processor, AMD Radeon graphic 4 GB.

4.2.4 Datasets

Three datasets downloaded from Google Finance database, As mentioned in the past chapters since three pairs of currency tested, their historical data needed. Through the searching in many global active currency exchanger banks and organizations Google Finance provides complete and clean data of our target currency pairs. The range of the data for 1_{st}pairs of currency which is NZU/USD are from 01- 01-2002 to 31-12-2023 with 7964 observations .and, the range of the data for 2_{st} pairs of currency which is USD/INR are from 01- 01-2002 to 31-12-2023 with 7512 observations. The range of the data for 3_{st}pairs of currency which is EUR/INR are from 01- 01-2002 to 31-12-2023 with 7716 observations because of unavailability of the data. The data is daily which means the currency exchange rate between two currencies in a daily based; one record for one day. These three datasets have the following attributes which are used for the forecasting task.

- Date
- Close

The first five attributes used as inputs of the training algorithms and the last attribute which is close is used for output and forecasting each day close price.

To explore our datasets, each dataset is plotted to show the movement of each currency against US dollar over periods of time. These plots can show the volatility and changeability of the prices. The first plot is from NZD/USD dataset.

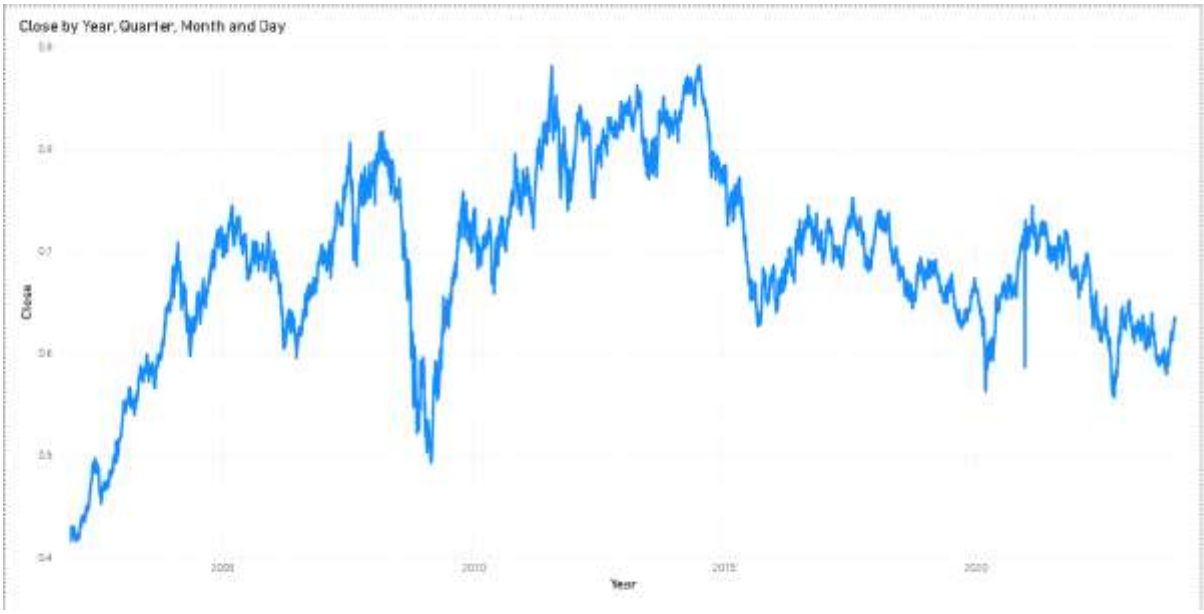


FIG 4.2 NZD/USD rate of dataset

The second dataset is from EURINR within the same date.

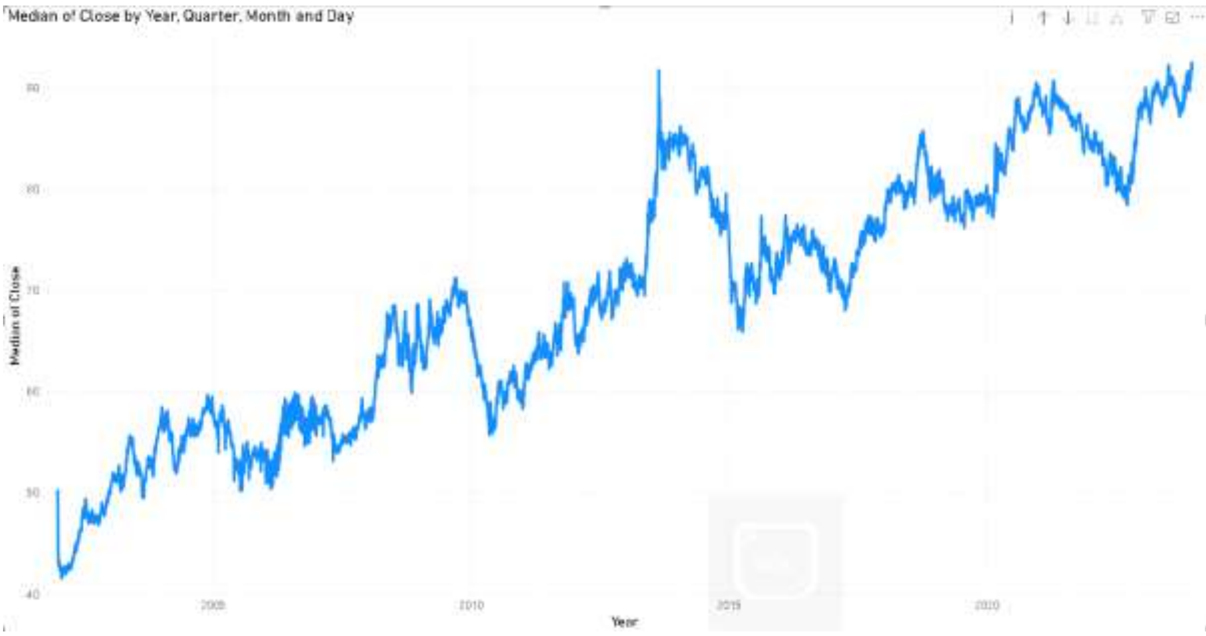


FIG 4.3 EURINR rate of the dataset

The Third dataset is from USDINR within the same date.

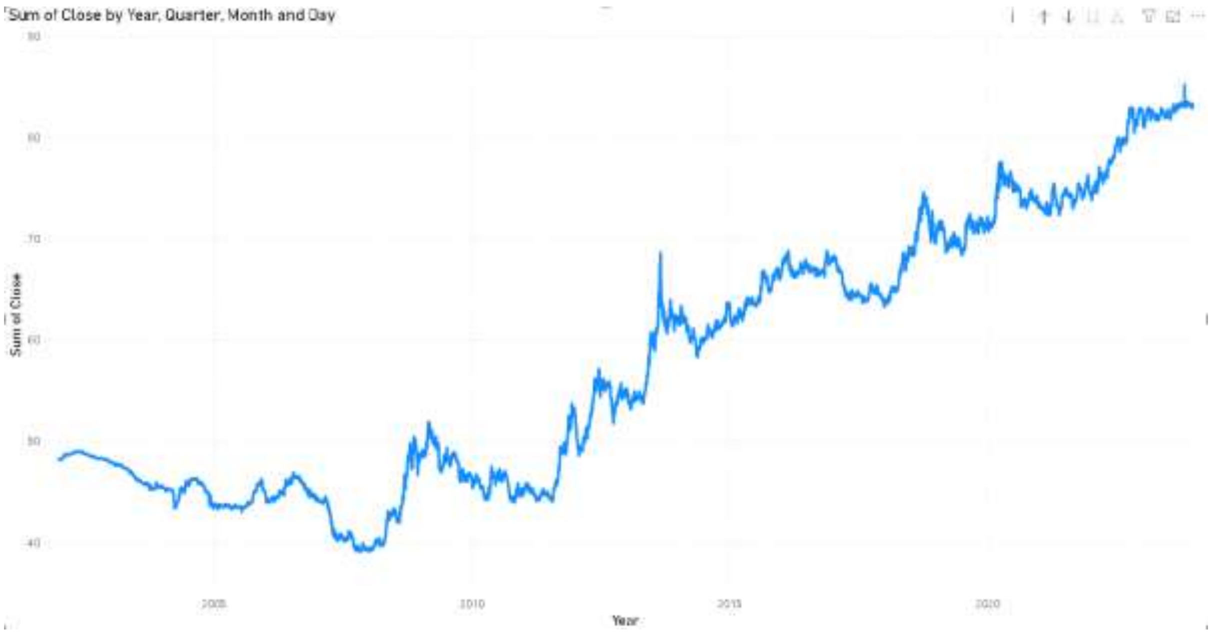


FIG 4.4 USDINR rate of the dataset

4.3 Data Preprocessing

For training an algorithm in machine learning we need to clean and prepare the data to be fit and can be used for training. This preprocessing has many steps which some of them are different from one approach to another. At the meantime, some of these steps are common between all. The preprocessing actions conducted for this research are as follows

. • The first column of data was timestamp that includes both the date and time which the observation is taken. The timestamp is splitted in two columns of time and Date, then the time is removed since this forecasting is daily based not in a timely manner.

• Then for making sure of the non-existence of the null values, all columns checked which the “Volume” had ‘0’ values for Saturday and Sunday of the week. Because the shared market value of the market is not clear for these two days of the week. Since those two days were also important for analysis the ‘0’ value cells filled with their above observations which is one day before that day.

• For the feeding purpose in neural network, the data should be in floating points in tensors (Chollet, 2017). Usually, the good practice is to normalize/scale the data within a specific range which is mostly from 0 to 1 because data must not be in widely different ranges, it basically could affect the training negatively. In this study, the data of all attributes normalized or scaled within this range. The “MinMaxScalr” object is used from ScikitLearn library. It basically takes the maximum value of a column as 1 and the minimum value of the column as 0 and fit other values based on that. The formula for that is as follows:

$$X_{new} = \frac{X_i - \min(X)}{\max(x) - \min(X)}$$

Fig 4.5 Minmax Scaler Formula

This formula is at the backend which the library provides the facility to normalize the data much easier.

• Data is splitted into two parts, training and testing. Training portion is 80% and the testing portion is 20% which is a common practice. It starts from 70% training and 30% testing and goes to a much greater training portion. For this purpose, the “train_test_split” object of Scikit-Learn is used to randomly divide data based on K-Fold Cross validation technique.

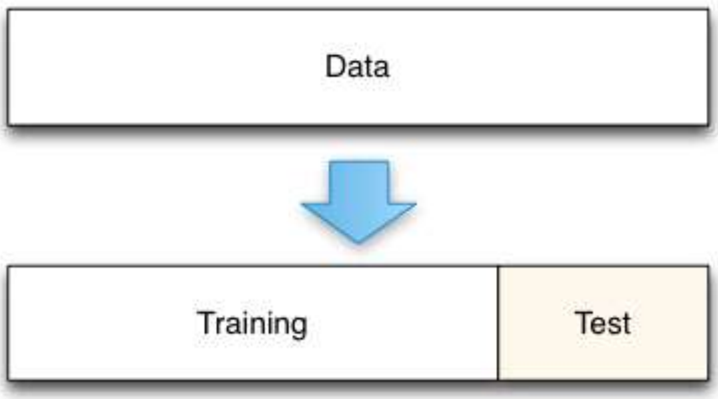


Fig 4.6 Train Test Split

4.4 Algorithms Overview

In the realm of predictive modelling and forecasting, various algorithms are employed to analyse and predict trends in data. Among these algorithms, LSTM (Long Short-Term Memory), SVR (Support Vector Regression), ARIMA (Autoregressive Integrated Moving Average), and RF (Random Forest) are prominent choices due to their effectiveness in handling different types of data and capturing complex patterns.

Below is an overview of these algorithm combinations:

1. LSTM-ARIMA
2. LSTM-SVR
3. LSTM-RF
4. SVR-RF

4.4.1 LSTM-ARIMA

4.4.1.1 LSTM

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to overcome the limitations of traditional RNNs in capturing and learning long-term dependencies in sequential data. LSTM networks are widely used in various fields, including natural language processing, time series analysis, speech recognition, and more, due to their ability to effectively model sequential data with long-range dependencies. Here's an overview of LSTM:

1. Architecture:
   - LSTM networks consist of multiple memory cells, each with three main components: a cell state, an input gate, and an output gate.
   - The cell state serves as the "memory" of the network, allowing it to retain information over long sequences.
   - The input gate controls the flow of information into the cell state, regulating which information is stored or discarded.
   - The output gate determines which parts of the cell state are used to produce the output of the LSTM cell.
2. Long-Term Dependencies:
   - One of the key features of LSTM networks is their ability to capture long-term dependencies in sequential data.
   - Traditional RNNs often suffer from the vanishing gradient problem, where gradients diminish exponentially over time, making it difficult to learn dependencies over long sequences.

- LSTM networks address this issue by incorporating specialized memory cells and gating mechanisms that allow them to maintain information over extended periods, enabling them to learn and exploit long-term dependencies more effectively.

3. Training:

- LSTM networks are typically trained using backpropagation through time (BPTT), an extension of the backpropagation algorithm that propagates gradients through the network over multiple time steps.

- During training, the network learns to adjust its parameters (weights and biases) to minimize the difference between the predicted outputs and the ground truth labels.

4.4.1.2 ARIMA:

Autoregressive Integrated Moving Average (ARIMA) is a widely used time series forecasting model that is particularly effective in capturing and predicting the patterns and trends in univariate time series data. ARIMA models are popular in various fields, including finance, economics, epidemiology, and environmental science. Here's an overview of ARIMA:

1. Components:

- Autoregressive (AR) component: ARIMA models consider the relationship between a current observation and its past observations, also known as lagged values. The AR component represents this relationship, where the current value of the time series is a linear combination of its past values.

- Integrated (I) component: ARIMA models require the data to be stationary, meaning that its statistical properties such as mean and variance remain constant over time. The integrated component of ARIMA (denoted by the "I") transforms the non-stationary data into stationary data by differencing it with its lagged values.

- Moving Average (MA) component: The MA component captures the relationship between the current observation and the residual errors of previous observations. It represents the weighted sum of past error terms.

2. Parameters:

    - ARIMA models are defined by three main parameters: p, d, and q.

        - p (AR order): The number of lagged observations included in the autoregressive component.

        - d (Integrated order): The number of differencing operations required to make the data stationary.

        - q (MA order): The number of lagged forecast errors included in the moving average component.

3. Model Selection:

    - Selecting the appropriate values for the p, d, and q parameters is crucial for building an accurate ARIMA model.

    - This selection process often involves analysing the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots of the time series data to determine the values of p and q, and using statistical tests such as the Augmented Dickey-Fuller (ADF) test to determine the value of d.

4. Training and Forecasting:

    - Once the parameters are determined, the ARIMA model is trained using historical time series data.

    - After training, the model can be used to generate forecasts for future time periods based on the patterns and trends learned from the historical data.

LSTM-ARIMA is a hybrid forecasting model that combines the strengths of two powerful time series forecasting techniques: Long Short-Term Memory (LSTM) and Autoregressive Integrated Moving Average (ARIMA), with a focus on incorporating residuals. In this approach, the LSTM neural network is used to capture the nonlinear patterns and long-term dependencies present in the data, while ARIMA is employed to model the remaining linear components and short-term dynamics.

The residuals, representing the difference between the observed and predicted values from the ARIMA model, are then fed back into the LSTM model to refine its predictions further. by leveraging both LSTM and ARIMA in conjunction with residuals, this hybrid model aims to enhance forecasting accuracy by effectively capturing both linear and nonlinear patterns in the time series data, making it well-suited for applications where traditional linear models may fall short.

4.4.2 LSTM-SVR:

4.4.2.1 LSTM

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to overcome the limitations of traditional RNNs in capturing and learning long-term dependencies in sequential data. LSTM networks are widely used in various fields, including natural language processing, time series analysis, speech recognition, and more, due to their ability to effectively model sequential data with long-range dependencies. Here's an overview of LSTM:

1. Architecture:
    - LSTM networks consist of multiple memory cells, each with three main components: a cell state, an input gate, and an output gate.
    - The cell state serves as the "memory" of the network, allowing it to retain information over long sequences.
    - The input gate controls the flow of information into the cell state, regulating which information is stored or discarded.
    - The output gate determines which parts of the cell state are used to produce the output of the LSTM cell.
2. Long-Term Dependencies:
    - One of the key features of LSTM networks is their ability to capture long-term dependencies in sequential data.
    - Traditional RNNs often suffer from the vanishing gradient problem, where gradients diminish exponentially over time, making it difficult to learn dependencies over long sequences.
    - LSTM networks address this issue by incorporating specialized memory cells and gating mechanisms that allow them to maintain information over extended

periods, enabling them to learn and exploit long-term dependencies more effectively.

3. Training:

    - LSTM networks are typically trained using backpropagation through time (BPTT), an extension of the backpropagation algorithm that propagates gradients through the network over multiple time steps.

    - During training, the network learns to adjust its parameters (weights and biases) to minimize the difference between the predicted outputs and the ground truth labels.

4.4.2.2 SVR

Support Vector Regression (SVR) is a supervised machine learning algorithm used for regression tasks, particularly effective for handling complex datasets with non-linear relationships. SVR is an extension of Support Vector Machines (SVMs) and is widely used in various fields such as finance, economics, engineering, and biology. Here's an overview of SVR:

1. Principles:

    - SVR aims to find the optimal hyperplane that best fits the training data in a high-dimensional space while minimizing the error between the predicted and actual values.

    - Unlike traditional regression methods that focus on minimizing the error between predictions and actual values, SVR focuses on minimizing the margin violations, i.e., the distance between the predicted values and the hyperplane, subject to a specified tolerance or epsilon (ε).

2. Kernel Trick:

    - SVR can handle non-linear relationships between features and target variables by mapping the input features into a higher-dimensional space using kernel functions.

- Commonly used kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid kernels. These kernels allow SVR to capture complex relationships between variables without explicitly transforming the data into higher dimensions.

3. Parameters:

   - SVR has several key parameters that affect its performance, including C (regularization parameter), epsilon (ε), and the choice of kernel function.
   - The regularization parameter C controls the trade-off between maximizing the margin and minimizing the training error. A smaller value of C results in a wider margin and higher tolerance for errors, while a larger value of C penalizes errors more heavily.
   - The epsilon parameter (ε) determines the width of the margin around the hyperplane within which no penalty is incurred.

4. Training and Prediction:

   - During training, SVR identifies support vectors, which are the data points that lie closest to the hyperplane.
   - The SVR model learns the optimal hyperplane that maximizes the margin while ensuring that the error between the predicted and actual values is within the specified tolerance (ε).
   - Once trained, the SVR model can make predictions for new data points by calculating the distance between the input features and the hyperplane.

LSTM-SVR is a hybrid forecasting model that combines the strengths of two powerful time series forecasting techniques: Long Short-Term Memory (LSTM) and Support Vector Machine (SVR), with a focus on incorporating residuals. In this approach, the LSTM neural network is used to capture the nonlinear patterns and long-term dependencies present in the data, while SVR is employed to model the remaining linear components and short-term dynamics.

The residuals, representing the difference between the observed and predicted values from the SVR model, are then fed back into the LSTM model to refine its predictions further. by leveraging both LSTM and SVR in conjunction with residuals, this hybrid model aims to enhance forecasting accuracy by effectively capturing both linear and nonlinear patterns in the time series data

4.4.3LSTM-RF:

4.4.3.1 LSTM

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to overcome the limitations of traditional RNNs in capturing and learning long-term dependencies in sequential data. LSTM networks are widely used in various fields, including natural language processing, time series analysis, speech recognition, and more, due to their ability to effectively model sequential data with long-range dependencies. Here's an overview of LSTM:

1. Architecture:
    - LSTM networks consist of multiple memory cells, each with three main components: a cell state, an input gate, and an output gate.
    - The cell state serves as the "memory" of the network, allowing it to retain information over long sequences.
    - The input gate controls the flow of information into the cell state, regulating which information is stored or discarded.
    - The output gate determines which parts of the cell state are used to produce the output of the LSTM cell.
2. Long-Term Dependencies:
    - One of the key features of LSTM networks is their ability to capture long-term dependencies in sequential data.
    - Traditional RNNs often suffer from the vanishing gradient problem, where gradients diminish exponentially over time, making it difficult to learn dependencies over long sequences.
    - LSTM networks address this issue by incorporating specialized memory cells and gating mechanisms that allow them to maintain information over extended

periods, enabling them to learn and exploit long-term dependencies more effectively.

3. Training:
    - LSTM networks are typically trained using backpropagation through time (BPTT), an extension of the backpropagation algorithm that propagates gradients through the network over multiple time steps.
    - During training, the network learns to adjust its parameters (weights and biases) to minimize the difference between the predicted outputs and the ground truth labels.

4.4.3.2RF

Random Forest (RF) is a versatile and powerful machine learning algorithm commonly used for both classification and regression tasks. It belongs to the ensemble learning family and is known for its high accuracy, robustness, and ability to handle large and complex datasets. Here's an overview of Random Forest:

1. Ensemble Learning:
    - Random Forest is an ensemble learning method that combines the predictions of multiple individual decision trees to make more accurate and robust predictions.
    - The fundamental idea behind ensemble learning is to train multiple weak learners (in this case, decision trees) and then combine their predictions to produce a stronger model.
2. Decision Trees:
    - Random Forest is built upon the foundation of decision trees, which are hierarchical structures consisting of nodes that represent features, branches that represent decisions, and leaf nodes that represent outcomes or predictions.
    - Each decision tree in the Random Forest is trained independently on a subset of the training data and a random subset of features.

3. Randomization and Diversity:
    - Random Forest introduces randomness in two key ways:

- Random sampling of training data: Each decision tree in the Random Forest is trained on a bootstrap sample (random sample with replacement) of the original training data. This process, known as bagging (bootstrap aggregation), introduces diversity among the trees.
- Random selection of features: At each node of the decision tree, only a random subset of features is considered for splitting. This ensures that different trees in the forest learn from different subsets of features.

4. Aggregation and Prediction:

- In classification tasks, Random Forest aggregates the predictions of individual decision trees through majority voting. The class with the most votes among the trees is chosen as the final prediction.
- In regression tasks, Random Forest aggregates the predictions of individual decision trees by averaging their outputs. The average value is taken as the final prediction.

LSTM-RF is a hybrid forecasting model that combines the strengths of two powerful time series forecasting techniques: Long Short-Term Memory (LSTM) and Random Forest (RF), with a focus on incorporating residuals. In this approach, the LSTM neural network is used to capture the nonlinear patterns and long-term dependencies present in the data, while RF is employed to model the remaining linear components and short-term dynamics.

The residuals, representing the difference between the observed and predicted values from the RF model, are then fed back into the LSTM model to refine its predictions further. by leveraging both LSTM and RF in conjunction with residuals, this hybrid model aims to enhance forecasting accuracy by effectively capturing both linear and nonlinear patterns in the time series data

4.4.4 SVR-RF

4.4.4.1 SVR

Support Vector Regression (SVR) is a supervised machine learning algorithm used for regression tasks, particularly effective for handling complex datasets with non-linear relationships. SVR is an extension of Support Vector Machines (SVMs) and is widely used in various fields such as finance, economics, engineering, and biology. Here's an overview of SVR:

1. Principles:

    - SVR aims to find the optimal hyperplane that best fits the training data in a high-dimensional space while minimizing the error between the predicted and actual values.
    - Unlike traditional regression methods that focus on minimizing the error between predictions and actual values, SVR focuses on minimizing the margin violations, i.e., the distance between the predicted values and the hyperplane, subject to a specified tolerance or epsilon (ε).

2. Kernel Trick:

    - SVR can handle non-linear relationships between features and target variables by mapping the input features into a higher-dimensional space using kernel functions.
    - Commonly used kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid kernels. These kernels allow SVR to capture complex relationships between variables without explicitly transforming the data into higher dimensions.

3. Parameters:

    - SVR has several key parameters that affect its performance, including C (regularization parameter), epsilon (ε), and the choice of kernel function.
    - The regularization parameter C controls the trade-off between maximizing the margin and minimizing the training error. A smaller value of C results in a wider margin and higher tolerance for errors, while a larger value of C penalizes errors more heavily.
    - The epsilon parameter (ε) determines the width of the margin around the hyperplane within which no penalty is incurred.

4. Training and Prediction:

    - During training, SVR identifies support vectors, which are the data points that lie closest to the hyperplane.

- The SVR model learns the optimal hyperplane that maximizes the margin while ensuring that the error between the predicted and actual values is within the specified tolerance (ε).
- Once trained, the SVR model can make predictions for new data points by calculating the distance between the input features and the hyperplane.

4.4.42 RF

Random Forest (RF) is a versatile and powerful machine learning algorithm commonly used for both classification and regression tasks. It belongs to the ensemble learning family and is known for its high accuracy, robustness, and ability to handle large and complex datasets. Here's an overview of Random Forest:

1. Ensemble Learning:
    a. Random Forest is an ensemble learning method that combines the predictions of multiple individual decision trees to make more accurate and robust predictions.
    b. The fundamental idea behind ensemble learning is to train multiple weak learners (in this case, decision trees) and then combine their predictions to produce a stronger model.
2. Decision Trees:
    a. Random Forest is built upon the foundation of decision trees, which are hierarchical structures consisting of nodes that represent features, branches that represent decisions, and leaf nodes that represent outcomes or predictions.
    b. Each decision tree in the Random Forest is trained independently on a subset of the training data and a random subset of features.
3. Randomization and Diversity:
    a. Random Forest introduces randomness in two key ways:
        i. Random sampling of training data: Each decision tree in the Random Forest is trained on a bootstrap sample (random sample with replacement) of the original training data. This process, known as bagging (bootstrap aggregation), introduces diversity among the trees.

ii. Random selection of features: At each node of the decision tree, only a random subset of features is considered for splitting. This ensures that different trees in the forest learn from different subsets of features.

4. Aggregation and Prediction:
   a. In classification tasks, Random Forest aggregates the predictions of individual decision trees through majority voting. The class with the most votes among the trees is chosen as the final prediction.
   b. In regression tasks, Random Forest aggregates the predictions of individual decision trees by averaging their outputs. The average value is taken as the final prediction.

SVR-RF is a hybrid forecasting model that combines the strengths of two powerful time series forecasting techniques: Support Vector Machine (SVR) and Random Forest (RF), with a focus on incorporating residuals. In this approach, the SVR is used to capture the nonlinear patterns and long-term dependencies present in the data, while RF is employed to model the remaining linear components and short-term dynamics.

The residuals, representing the difference between the observed and predicted values from the RF model, are then fed back into the SVR model to refine its predictions further. by leveraging both SVR and RF in conjunction with residuals, this hybrid model aims to enhance forecasting accuracy by effectively capturing both linear and nonlinear patterns in the time

4.5 Training, Testing, and Validation

While working on algorithms to monitor the model accuracy the data should separate into training, testing and validations sets. The training set is used for the learning process which the algorithm learns the signals of the training examples, the correlation between inputs, their weights and the output. As described earlier the dataset should be divided into two parts in cross-validation, though the testing portion is kept aside for testing the accuracy of the model after it is trained. Though in this study based on most common practices the data is divided into 80% for training and 20% for testing.

The first training is conducted in the training part, the models are trained based on the examples in this portion. Many techniques have been used for improving the learning process. Two activation functions “Relu” and “Sigmoid” is tested. Since the “sigmoid” gave a better result, it is used as activation function. meanwhile different number of “epochs” and “batch size” is examined to find the best possible values for these parameters in order to retrieve a good accuracy and results from the training which ends mostly same for testing as well.

Despite those efforts, three performance evaluation techniques are used to calculate and evaluate our models. These three techniques are “RMSE”, “MAE” and “MSE”. The advantage of these techniques is not just finding the accuracy but at the same time, they make us able to compare the algorithms and claim which one provides the most accurate result.

Based on the result of the accuracy evaluation techniques, the algorithms compared with each other which is described in details in Chapter 5.

4.6 Visualization

With a simple definition, visualization is the most convenient way of communicating a message. This has been heard too much that a picture worth thousands of words. Since visualization is also creating images, graphs, and visionary shapes, it makes it easy to make complex topics easy for understanding.

Therefore, in this study, visualization is used for clarification of the learning process and the accuracy improvement in training phase. In addition, both the actual values which is the real output is comparatively visualized with predicted output to show the difference in very precise and clear sight. Matplotlib from python libraries is used for doing this helpful task.

4.7 Model Development Summary

The steps for completing the entire process are described in details. Each step should be done in a specific time and order to develop a complete and reliable model for currency exchange rate forecasting. A flow diagram of the entire process is shown in figure 4.7.

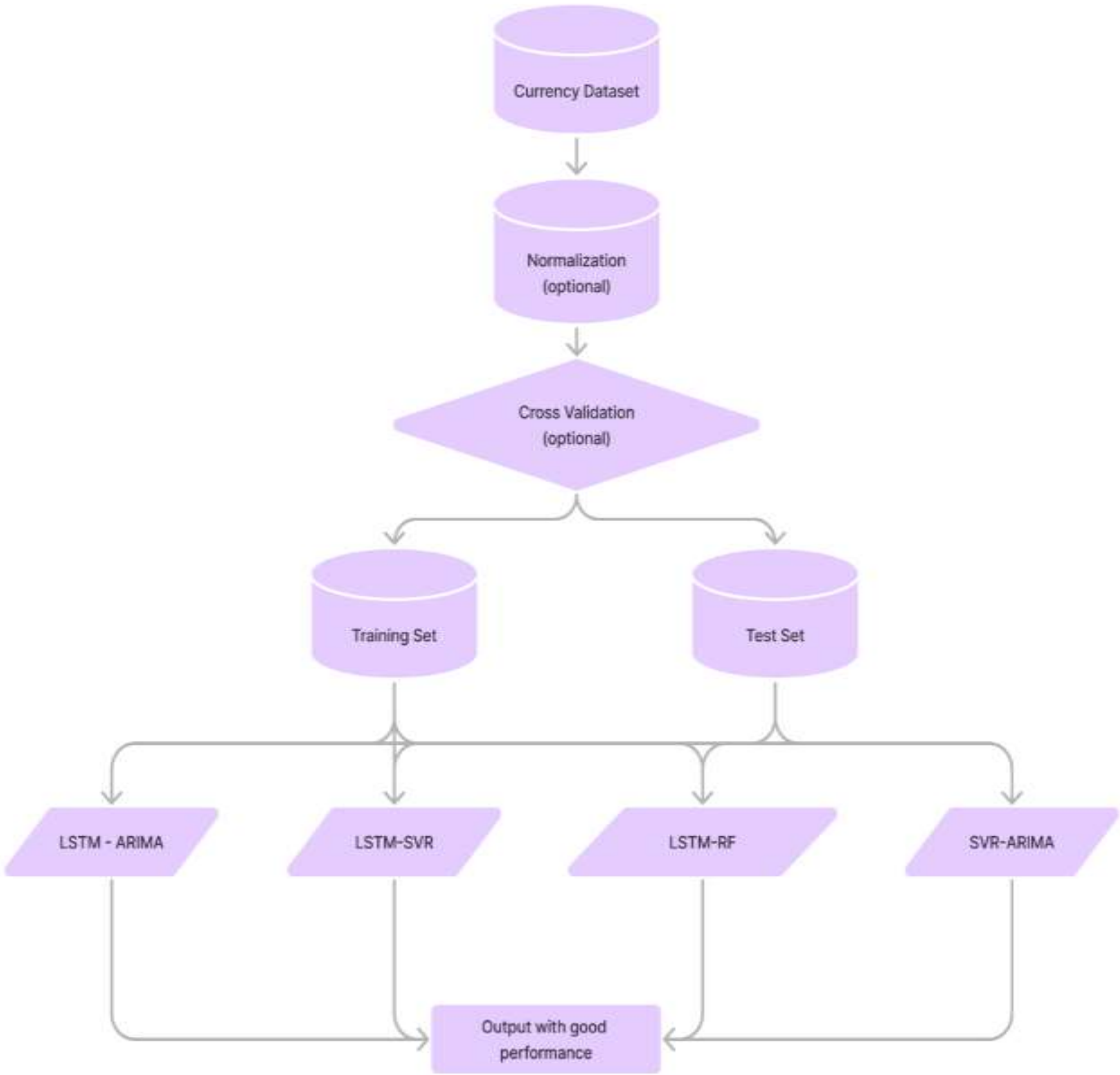


Fig 4.7 flow diagram of the entire process

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Experimental Setup

Few options are available when you are going to implement machine learning algorithms. The most popular tools are MATLAB, Python, and R programming language. Due to the rich availability of libraries in python and easiness of use the researcher chooses to use python in this study. The details of the tools and techniques are presented in Chapter 4. Libraries like Scikit-Learn, Pandas, NumPy, matplotlib, and Keras are used to carry out amazing functionalities and are used to process data, create models and visualize data properly.

This chapter describes the model creation and results of the algorithms being used in this study. The results and evaluation will be presented for 4 hybrid models. Finally, a comparison between the results of each algorithm and at the same time the best model among them is presented as well.

5.1.1 LSTM-ARIMA

LSTM-ARIMA is a hybrid forecasting model that combines the strengths of two powerful time series forecasting techniques: Long Short-Term Memory (LSTM) and Autoregressive Integrated Moving Average (ARIMA), with a focus on incorporating residuals. In this approach, the LSTM neural network is used to capture the nonlinear patterns and long-term dependencies present in the data, while ARIMA is employed to model the remaining linear components and short-term dynamics.

The residuals, representing the difference between the observed and predicted values from the ARIMA model, are then fed back into the LSTM model to refine its predictions further.

Parameter	Value
Number of LSTM layers	2
LSTM units per layer	50
Look-back window	10

Input shape	(10, 1)
Return sequences	True (for the first layer)
Loss function	Mean Squared Error
Optimizer	Adam
Number of epochs	20
Batch size	1
Verbose	0

Table 5.1 LSTM Model Training Parameters

by leveraging both LSTM and ARIMA in conjunction with residuals, this hybrid model aims to enhance forecasting accuracy by effectively capturing both linear and nonlinear patterns in the time series data, making it well-suited for applications where traditional linear models may fall short.

Parameter	**Description**
p (AR order)	Number of lag observations included in the model (Autoregressive term)
d (Integrated order)	Degree of differencing (number of times the series is differenced to achieve stationarity)
q (MA order)	Size of the moving average window (Moving Average term)

Table 5.2 ARIMA Model training parameters

These parameters being used for the training of the models. Different values have been tested and the best ones among them selected. After training the model, the performance evaluation techniques implemented to measure the accuracy of training and testing separately. Needed to mention that since the model is tested with three pairs of currencies, the evaluation results are different from currency to another that each of them has different movements and impact factors in the global market. The results of the NZD/USD, USD/INR and EUR/INR currency pairs are presented in Table 5.3, Table 5.4 and Table 5.5.

Table 5.3 EUR/INR evaluation results with LSTM-ARIMA

EUR/INR	MAE	MSE	RMSE	R_2 VALUE
LSTM	0.012628	0.000233	0.015277	0.963702
ARIMA	0.836306	0.706579	0.840582	-49366803.7
Hybrid	0.011947	0.000208	0.014432	0.967605568

Table 5.4 NZD/USD evaluation results with LSTM-ARIMA

NZD/USD	MAE	MSE	RMSE	R_2 VALUE
LSTM	0.0057653	0.0000862	0.009285	0.94495
ARIMA	0.6572968	0.433622	0.658499	-2982424
Hybrid	0.0042453	0.00006747	0.008214	0.956923

Table 5.5 USD/INR evaluation results with LSTM-ARIMA

USD/INR	MAE	MSE	RMSE	R_2 VALUE
LSTM	0.0202487	0.0006364	0.0252273	0.9497896
ARIMA	0.9943487	1.0044847	1.0022398	-519374098
Hybrid	0.0260441	0.0008961	0.0299359	0.9292986

The model worked well for the prediction purpose; it gives an acceptable result which can be used as a model to predict the future close rate of these currency pairs. To make it more understandable that how much the model is able to forecast these, better sightseeing is visualized in Figure 5.1, Figure 5.2 and Figure 5.3 to show how close or far the prediction and the real values are.

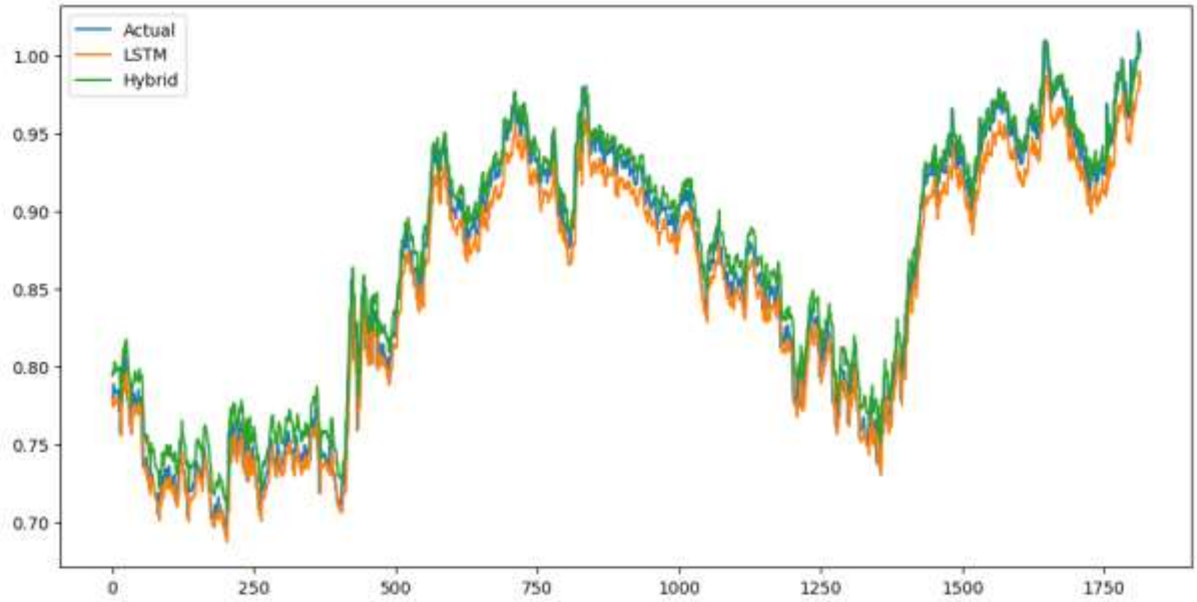


Fig 5.1 EUR/INR actual and predicted price visualization with LSTM-ARIMA

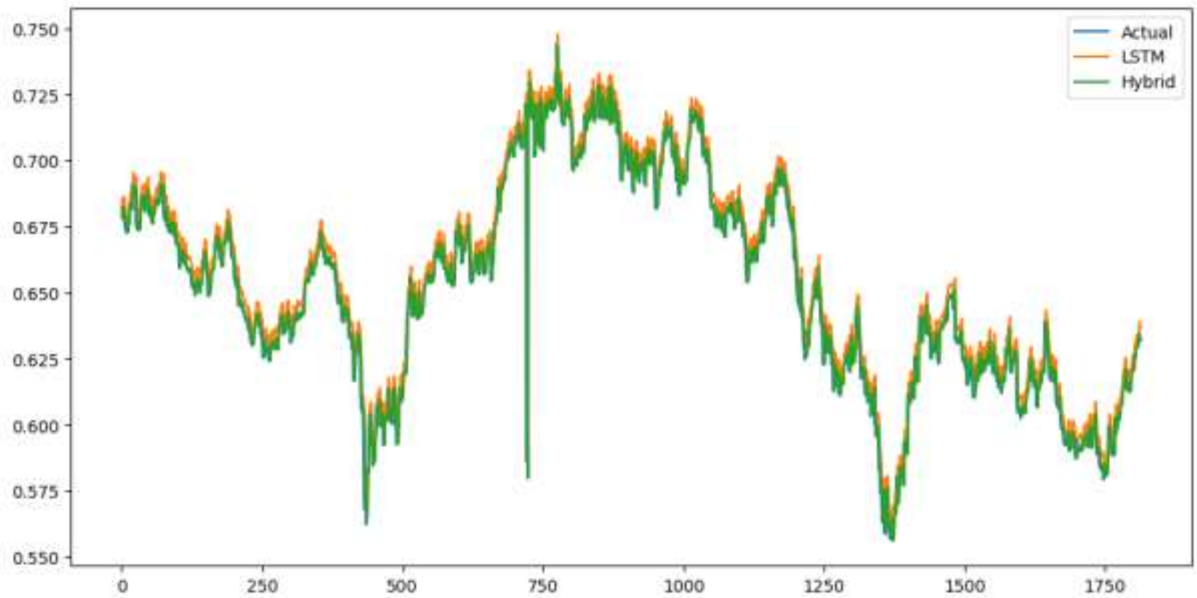


Fig 5.2 NZD/USD actual and predicted price visualization with LSTM-ARIMA

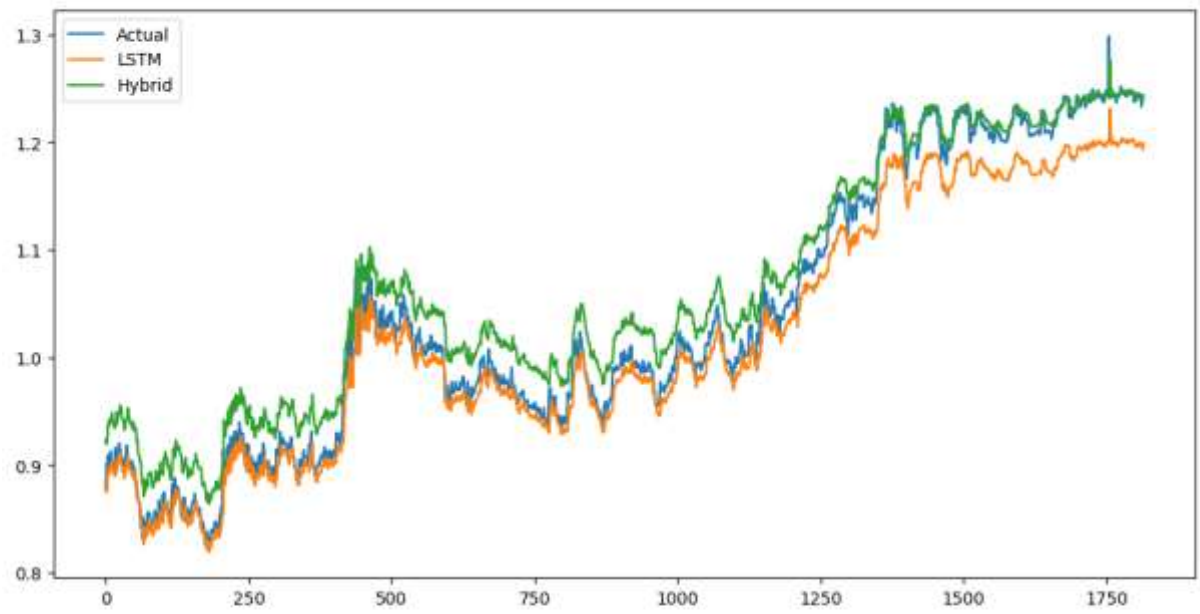


Fig 5.3 USD/INR actual and predicted price visualization with LSTM-ARIMA

5.2.2 LSTM-SVR

1. **LSTM (Long Short-Term Memory):** LSTM is a type of recurrent neural network (RNN) designed to model sequential data by capturing long-term dependencies. In the context of time series forecasting, LSTM can effectively learn patterns and trends from historical data.

2. **SVR (Support Vector Regression):** SVR is a machine learning algorithm used for regression tasks. It works by finding the hyperplane that best fits the data while maximizing the margin between the hyperplane and the data points. SVR is known for its ability to handle non-linear data and robustness to outliers.

3. Residuals: Residuals are the differences between the actual values and the predicted values by the LSTM model. By incorporating residuals into the LSTM-SVR model, we aim to capture and correct the errors or deviations made by the LSTM model.

The LSTM-SVR model with residuals typically follows these steps:

- Train an LSTM model on historical time series data to generate initial predictions.
- Compute the residuals by taking the differences between the actual values and the LSTM-predicted values.
- Train an SVR model using the residuals as additional features, along with other relevant input features.

- Combine the predictions from the LSTM model and the SVR model to obtain the final forecast.

Parameter	Value
Number of LSTM layers	2
LSTM units per layer	50
Look-back window	10
Input shape	(10, 1)
Return sequences	True (for the first layer)
Loss function	Mean Squared Error
Optimizer	Adam
Number of epochs	20
Batch size	1
Verbose	0

Table 5.6 LSTM Model Training Parameters

Parameter	Default Value	Description
kernel	'rbf'	Kernel function used for mapping input space to high-dimensional feature space. Other options include 'linear', 'poly', and 'sigmoid'.
C	1.0	Penalty parameter of the error term. Controls the trade-off between achieving a low training error and a low model complexity.
epsilon	0.1	Epsilon parameter in the epsilon-insensitive loss function. Determines the margin of tolerance where no penalty is given to errors.
gamma	'scale'	Kernel coefficient for 'rbf', 'poly', and 'sigmoid'. If 'scale', uses 1 / (n_features * X.var()), where X is the input data. If 'auto', uses 1 / n_features.
degree	3	Degree of the polynomial kernel function (used when kernel is 'poly').

Table 5.7 SVR Model Training Parameters

These parameters being used for the training of the models. Different values have been tested and the best ones among them selected. After training the model, the performance evaluation techniques implemented to measure the accuracy of training and testing separately. Needed to mention that since the model is tested with three pairs of currencies, the evaluation results are different from currency to another that each of them has different movements and impact factors in the global market. The results of the NZD/USD, USD/INR and EUR/INR currency pairs are presented in Table 5.8, Table 5.9 and Table 5.10.

Table 5.8 EUR/INR evaluation results with LSTM-SVR

EUR/INR	MAE	MSE	RMSE	R_2 VALUE
LSTM	0.011382	0.0001332	0.0115422	0.980579
SVR	0.844820	0.720899	0.8490580	-2.395599
Hybrid	0.001744	0.00000417	0.0020435	0.999391

Table 5.9 NZD/USD evaluation results with LSTM-SVR

NZD/USD	MAE	MSE	RMSE	R_2 VALUE
LSTM	0.0013002	0.00000169	0.0013014	0.999766
SVR	0.510348	0.26769	0.517395	0.0
Hybrid	0.00005968	0.00000000486	0.00006975941	0.9999993

Table 5.10 USD/INR evaluation results with LSTM-SVR

USD/INR	MAE	MSE	RMSE	R_2 VALUE
LSTM	0.011256	0.0001582	0.0125809	0.989050
SVR	1.0233880	1.063144	1.031089	-3539
Hybrid	0.006235	0.00004498	0.0066707	0.9969216

The model worked well for the prediction purpose; it gives an acceptable result which can be used as a model to predict the future close rate of these currency pairs. To make it more understandable that how much the model is able to forecast these, better sightseeing is visualized in Figure 5.4, Figure 5.5 and Figure 5.6 to show how close or far the prediction and the real values are.

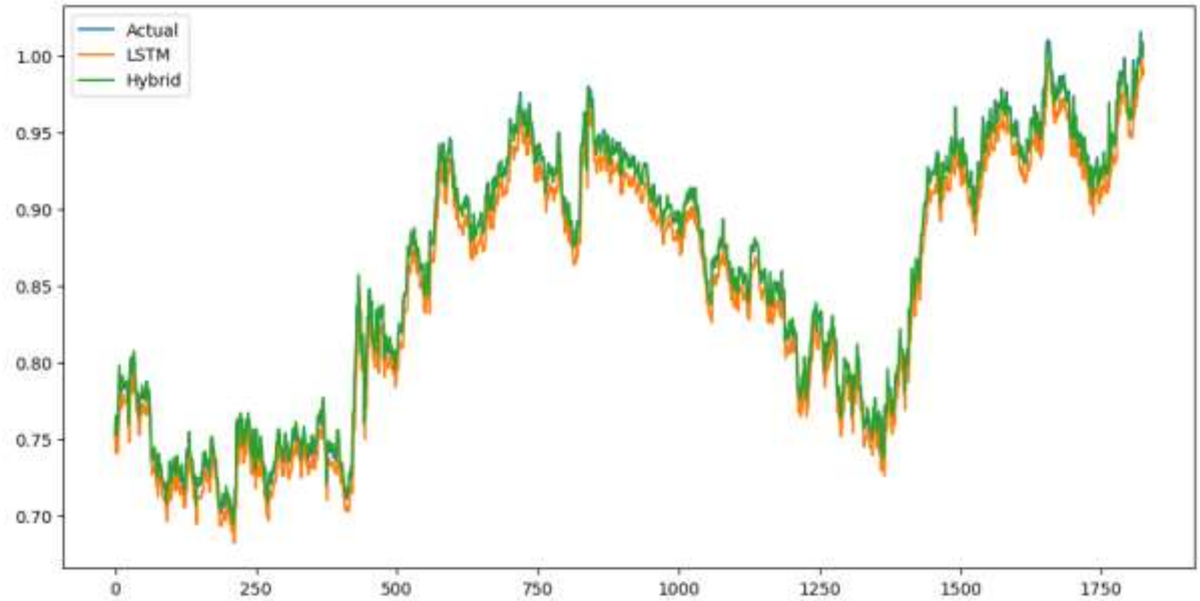


Fig 5.4 EUR/INR actual and predicted price visualization with LSTM-SVR

Fig 5.5 NZD/USD actual and predicted price visualization with LSTM-SVR

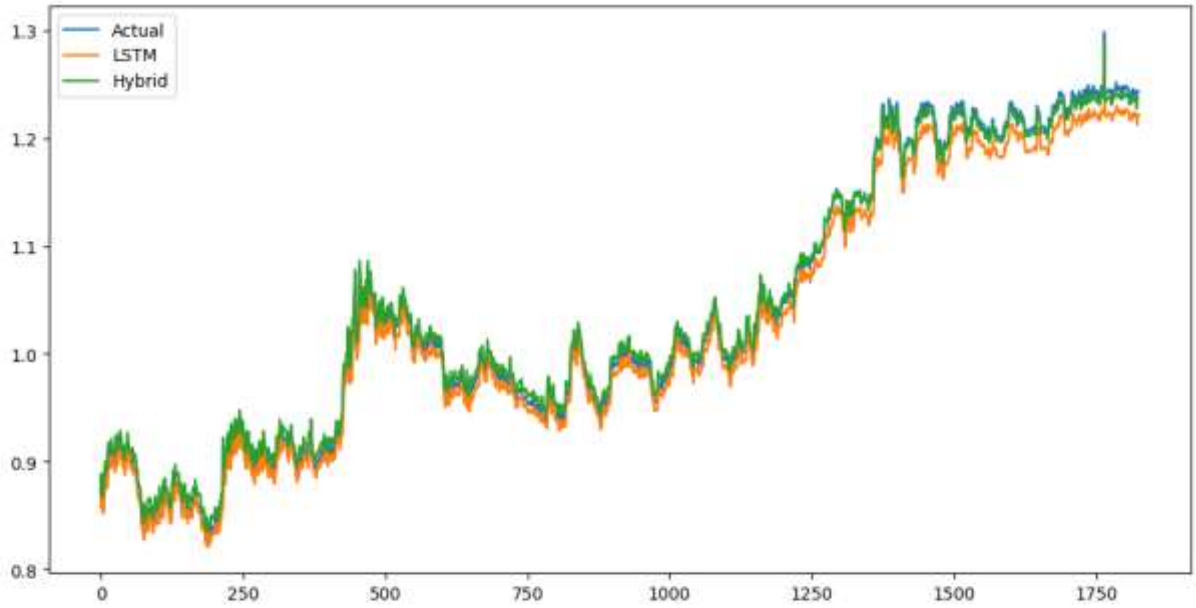


Fig 5.6 USD/INR actual and predicted price visualization with LSTM-SVR

5.2.3 LSTM-RF

The LSTM-RF (Long Short-Term Memory - Random Forest) model with residuals is a hybrid approach to time series forecasting that combines the strengths of LSTM and Random Forest algorithms while incorporating residuals for enhanced prediction accuracy.

Here's an overview of how this model works:

1. **LSTM (Long Short-Term Memory):** LSTM is a type of recurrent neural network (RNN) that is well-suited for processing and making predictions on sequential data, such as time series. LSTM models excel at capturing long-term dependencies and patterns in time series data.
2. **Random Forest (RF):** Random Forest is an ensemble learning algorithm that consists of a collection of decision trees. Each decision tree in the forest is trained on a random subset of the training data and makes predictions independently. The final prediction is typically the average or majority vote of the predictions from individual trees.
3. Residuals: Residuals are the differences between the actual values and the predicted values obtained from the LSTM model. By incorporating residuals into the LSTM-RF model, we aim to capture and correct any errors or deviations made by the LSTM model.

The LSTM-RF model with residuals typically follows these steps:

- Train an LSTM model on historical time series data to generate initial predictions.
- Compute the residuals by taking the differences between the actual values and the LSTM-predicted values.
- Train a Random Forest model using the residuals as additional features, along with other relevant input features.
- Combine the predictions from the LSTM model and the Random Forest model to obtain the final forecast

Parameter	**Value**
Number of LSTM layers	2
LSTM units per layer	50
Look-back window	10
Input shape	(10, 1)
Return sequences	True (for the first layer)
Loss function	Mean Squared Error
Optimizer	Adam
Number of epochs	20
Batch size	1
Verbose	0

Table 5.11 LSTM Model Training Parameters

Parameter	**Default Value**	**Description**
n_estimators	100	The number of trees in the forest.
criterion	'mse'	The function to measure the quality of a split. 'mse' stands for mean squared error, and 'mae' stands for mean absolute error.
max_depth	None	The maximum depth of the tree. If None, the nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

min_samples_split	2	The minimum number of samples required to split an internal node.
min_samples_leaf	1	The minimum number of samples required to be at a leaf node.
min_weight_fraction_leaf	0.0	The minimum weighted fraction of the sum total of weights (of all input samples) required to be at a leaf node.
max_features	'auto'	The number of features to consider when looking for the best split. If 'auto', then max_features=sqrt(n_features).
max_leaf_nodes	None	Grow trees with max_leaf_nodes in best-first fashion.
min_impurity_decrease	0.0	A node will be split if this split induces a decrease of the impurity greater than or equal to this value.
bootstrap	True	Whether bootstrap samples are used when building trees.
oob_score	False	Whether to use out-of-bag samples to estimate the generalization accuracy.
n_jobs	None	The number of jobs to run in parallel for both fit and predict. None means 1 unless in a joblib.parallel_backend context. -1 means using all processors.

Table 5.12 RF Model Training Parameters

These parameters being used for the training of the models. Different values have been tested and the best ones among them selected. After training the model, the performance evaluation techniques implemented to measure the accuracy of training and testing separately. Needed to mention that since the model is tested with three pairs of currencies, the evaluation results are different from currency to another that each of them has different movements and impact factors in the global market. The results of the NZD/USD, USD/INR and EUR/INR currency pairs are presented in Table 5.13, Table 5.14 and Table 5.15.

.

Table 5.13 EUR/INR evaluation results with LSTM-RF

EUR/INR	MAE	MSE	RMSE	R_2 VALUE
LSTM	0.009196	0.000135	0.0116421	0.9906715
RF	1.030978	1.077449	1.038002	-38450.65
Hybrid	0.004382	0.0000516	0.007185	0.996732

Table 5.14 NZD/USD evaluation results with LSTM-RF

NZD/USD	MAE	MSE	RMSE	R_2 VALUE
LSTM	0.0066836	0.000164	0.012840	0.977381
RF	0.511961	0.269397	0.519035	-3503546.8
Hybrid	0.005924	0.000155	0.012480	0.978504

Table 5.15 USD/INR evaluation results with LSTM-RF

USD/INR	MAE	MSE	RMSE	R_2 VALUE
LSTM	0.009196	0.0001355	0.0116421	0.990671
RF	1.030978	1.077449	1.0380027	-38450.65
Hybrid	0.004382	0.000051438	0.0071859	0.996732

The model worked well for the prediction purpose; it gives an acceptable result which can be used as a model to predict the future close rate of these currency pairs. To make it more understandable that how much the model is able to forecast these, better sightseeing is visualized in Figure 5.7, Figure 5.8 and Figure 5.9 to show how close or far the prediction and the real values are.

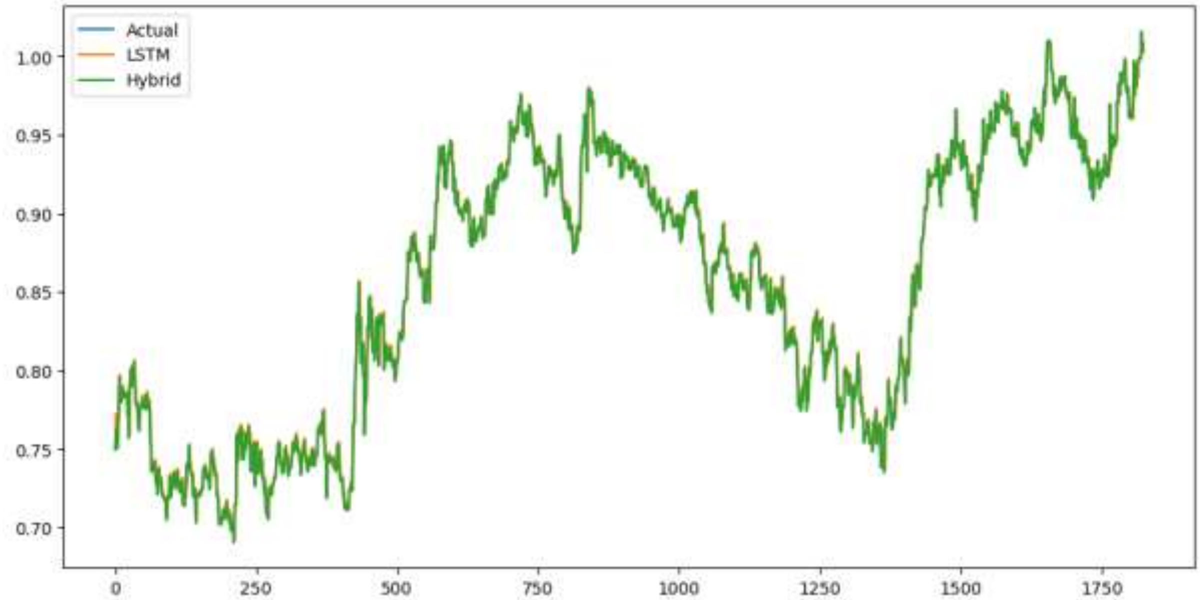


Fig 5.7 EUR/INR actual and predicted price visualization with LSTM-RF

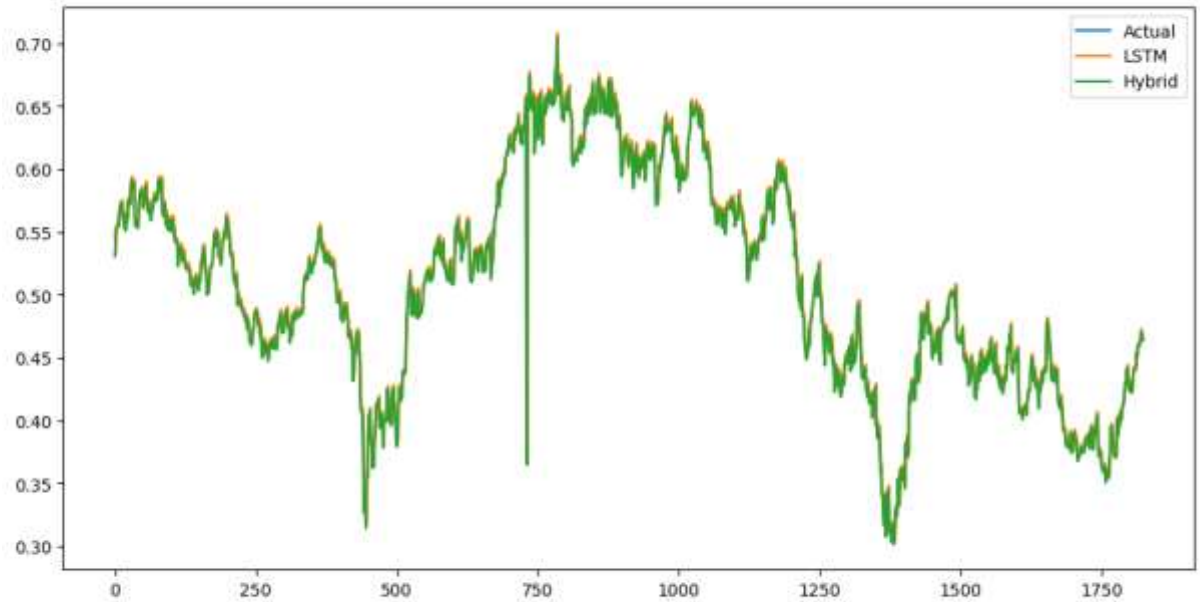


Fig5.8 NZD/USD actual and predicted price visualization with LSTM-RF

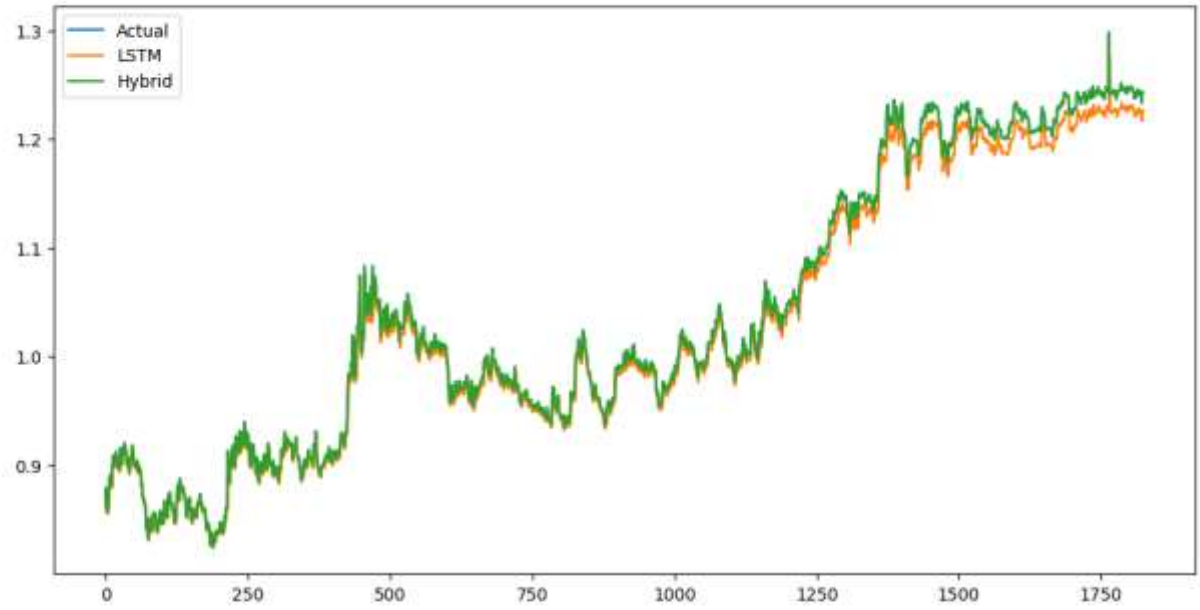


Fig 5.9 USD/INR actual and predicted price visualization with LSTM-RF

5.2.4 SVR-RF

Support Vector Regression with Random Forest (SVR-RF) with residuals is an interesting hybrid approach that combines the strengths of Support Vector Regression (SVR) and Random Forest (RF) algorithms.

1. **Support Vector Regression (SVR)**: SVR is a supervised learning algorithm that is used for regression tasks. It works by finding the hyperplane that best fits the data points. SVR is particularly useful when dealing with high-dimensional data and situations where there are non-linear relationships between the independent and dependent variables. SVR aims to minimize the error, or in other words, maximize the margin, between the predicted values and the actual values.
2. **Random Forest (RF)**: RF is an ensemble learning method that builds multiple decision trees during training and outputs the mean prediction (regression) or the mode of the predictions (classification) of the individual trees. It's known for its robustness and effectiveness in handling complex datasets with high dimensionality.

1. In SVR-RF with residuals, the residuals from the SVR model are used as additional features in the Random Forest model. Residuals represent the difference between the actual target values and the predicted values from the SVR model. By incorporating residuals into the Random Forest model, the aim is to capture any remaining patterns or relationships in the data that were not captured by the SVR model alone. This can potentially lead to improved predictive performance, especially if there are complex interactions or non-linear relationships in the data that are better captured by the ensemble nature of Random Forest.

+ . 3Parameter	Default Value	Description
kernel	'rbf'	Kernel function used for mapping input space to high-dimensional feature space. Other options include 'linear', 'poly', and 'sigmoid'.
C	1.0	Penalty parameter of the error term. Controls the trade-off between achieving a low training error and a low model complexity.
epsilon	0.1	Epsilon parameter in the epsilon-insensitive loss function. Determines the margin of tolerance where no penalty is given to errors.
gamma	'scale'	Kernel coefficient for 'rbf', 'poly', and 'sigmoid'. If 'scale', uses 1 / (n_features * X.var()), where X is the input data. If 'auto', uses 1 / n_features.
degree	3	Degree of the polynomial kernel function (used when kernel is 'poly').

Table 5.16 SVR Model Training Parameters

Parameter	Default Value	Description
n_estimators	100	The number of trees in the forest.
criterion	'mse'	The function to measure the quality of a split. 'mse' stands for mean squared error, and 'mae' stands for mean absolute error.
max_depth	None	The maximum depth of the tree. If None, the nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.
min_samples_split	2	The minimum number of samples required to split an internal node.
min_samples_leaf	1	The minimum number of samples required to be at a leaf node.
min_weight_fraction_leaf	0.0	The minimum weighted fraction of the sum total of weights (of all input samples) required to be at a leaf node.
max_features	'auto'	The number of features to consider when looking for the best split. If 'auto', then max_features=sqrt(n_features).
max_leaf_nodes	None	Grow trees with max_leaf_nodes in best-first fashion.
min_impurity_decrease	0.0	A node will be split if this split induces a decrease of the impurity greater than or equal to this value.
bootstrap	True	Whether bootstrap samples are used when building trees.
oob_score	False	Whether to use out-of-bag samples to estimate the generalization accuracy.
n_jobs	None	The number of jobs to run in parallel for both fit and predict. None means 1 unless in a joblib.parallel_backend context. -1 means using all processors.

Table 5.17 RF Model Training Parameters

These parameters being used for the training of the models. Different values have been tested and the best ones among them selected. After training the model, the performance evaluation techniques implemented to measure the accuracy of training and testing separately. Needed to mention that since the model is tested with three pairs of currencies, the evaluation results are different from currency to another that each of them has different movements and impact factors in the global market. The results of the NZD/USD, USD/INR and EUR/INR currency pairs are presented in Table 5.18, Table 5.19 and Table 5.20.

Table 5.18 EUR/INR evaluation results with SVR-RF

EUR/INR	MAE	MSE	RMSE	R_2 VALUE
LSTM	0.0334221	0.0018321	0.0428041	0.7401928
RF	0.8242595	0.686464	0.828531	-988.3961
Hybrid	0.005548	0.000065595	0.008099	0.990852

Table 5.19 NZD/USD evaluation results with SVR-RF

NSD/USD	MAE	MSE	RMSE	R_2 VALUE
LSTM	0.0278439	0.0011271	0.0335727	0.7077909
RF	0.4874032	0.241501	0.491427	-447.85092
Hybrid	0.0062535	0.00015	0.012436	0.9785149

Table5.20 USD evaluation results with SVR-RF

NSD/USD	MAE	MSE	RMSE	R_2 VALUE
LSTM	0.2112614	0.085067	0.291663	-10.64908
RF	0.82724944	0.691692	0.8316807	-16.135124
Hybrid	0.0058193	0.000075933	0.00871400	0.995185

The model worked well for the prediction purpose; it gives an acceptable result which can be used as a model to predict the future close rate of these currency pairs. To make it more understandable that how much the model is able to forecast these, better sightseeing is visualized in Figure 5.10, Figure 5.11 and Figure 5.12 to show how close or far the prediction and the real values are.

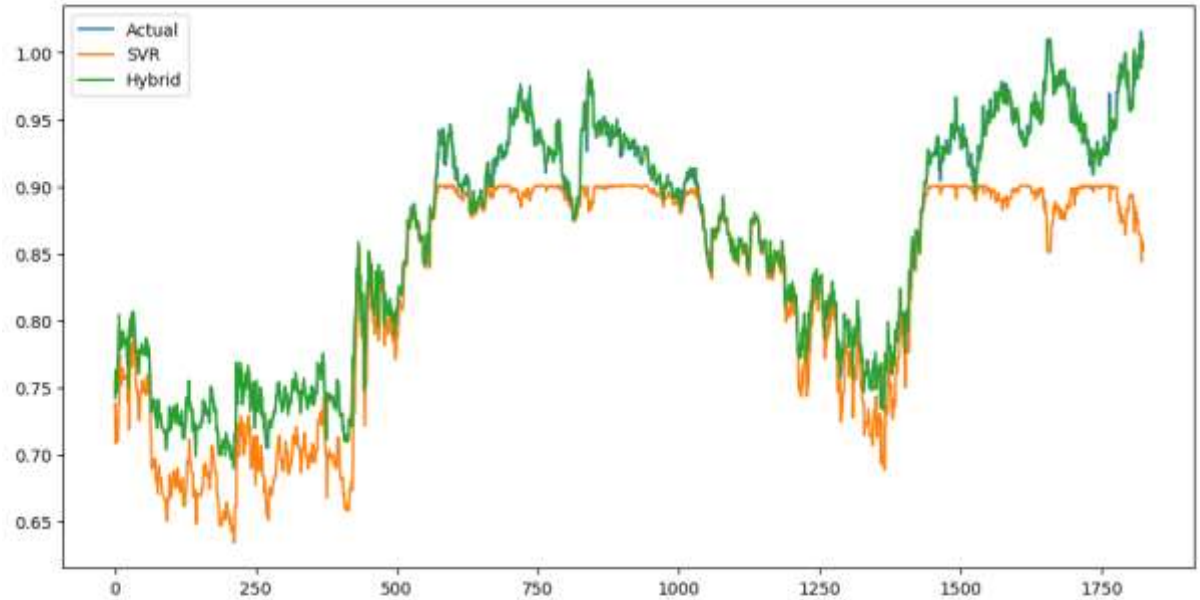


Fig 5.10 EUR/INR actual and predicted price visualization with SVR-RF

Fig 5.11 NZD/USD actual and predicted price visualization with SVR-RF

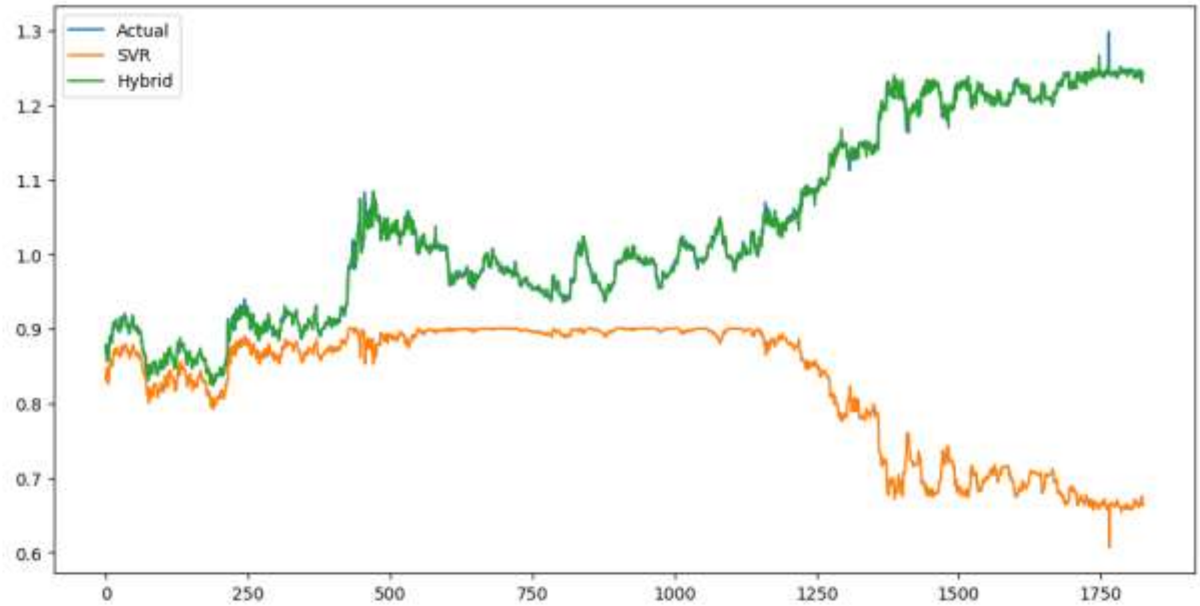


Fig 5.12 USD/INR actual and predicted price visualization with SVR-RF

5.4 Comparison and the Best Model among the Models

The results of tested models for three pairs of currencies presented in details each has good and bad results in different situations. For instance, for EUR/INR and USD/INR and NZD/USD, LSTM-SVR which performed best in all cases. To conclude the discussion better to evaluate and compare their performance in a visionary way.

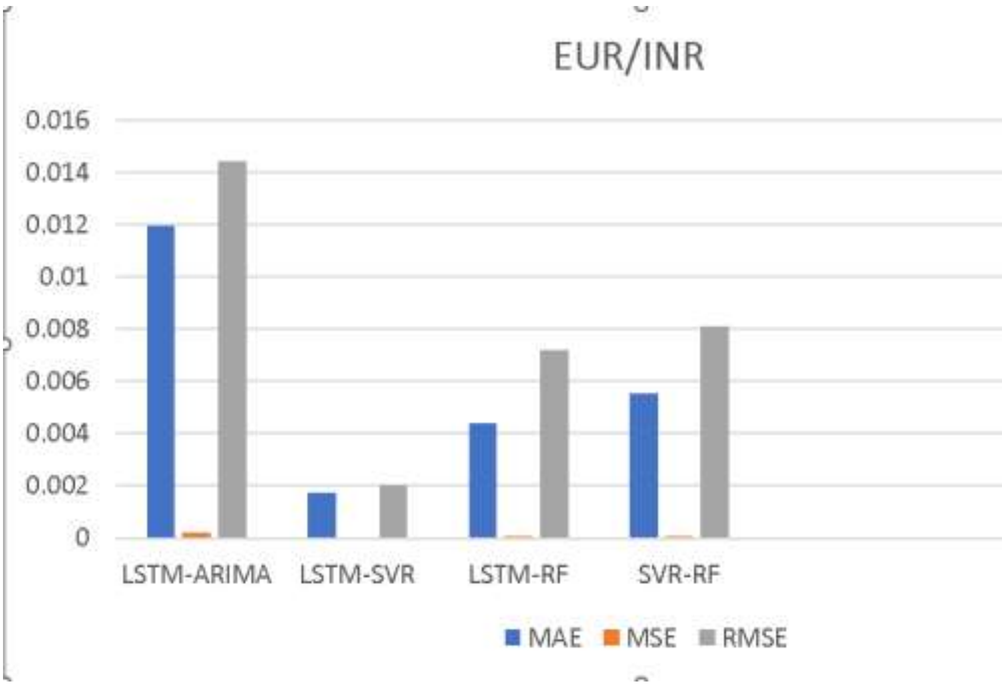


Fig 5.13 Evolution Metrics

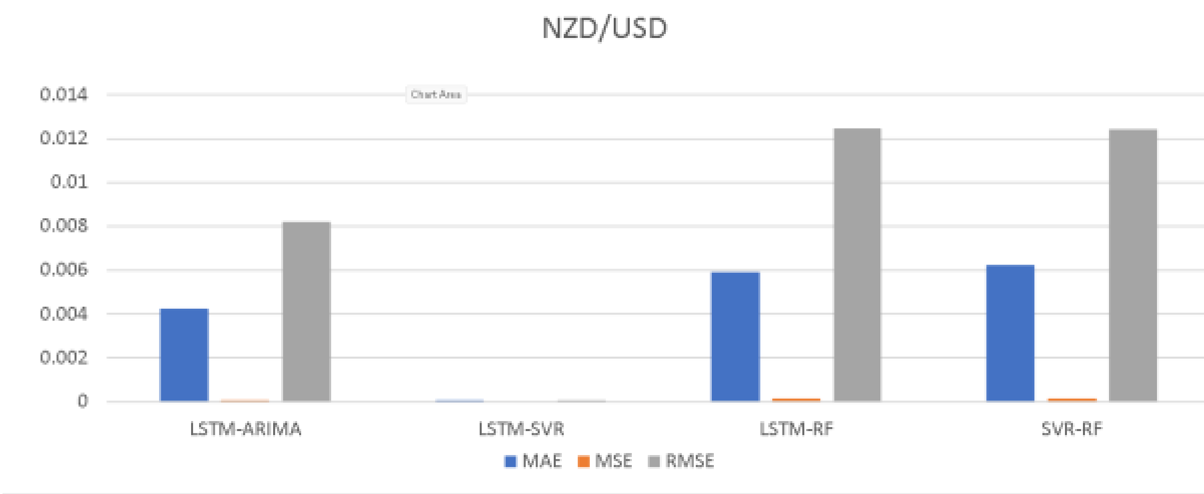


Fig 5.14 evaluation Metrics

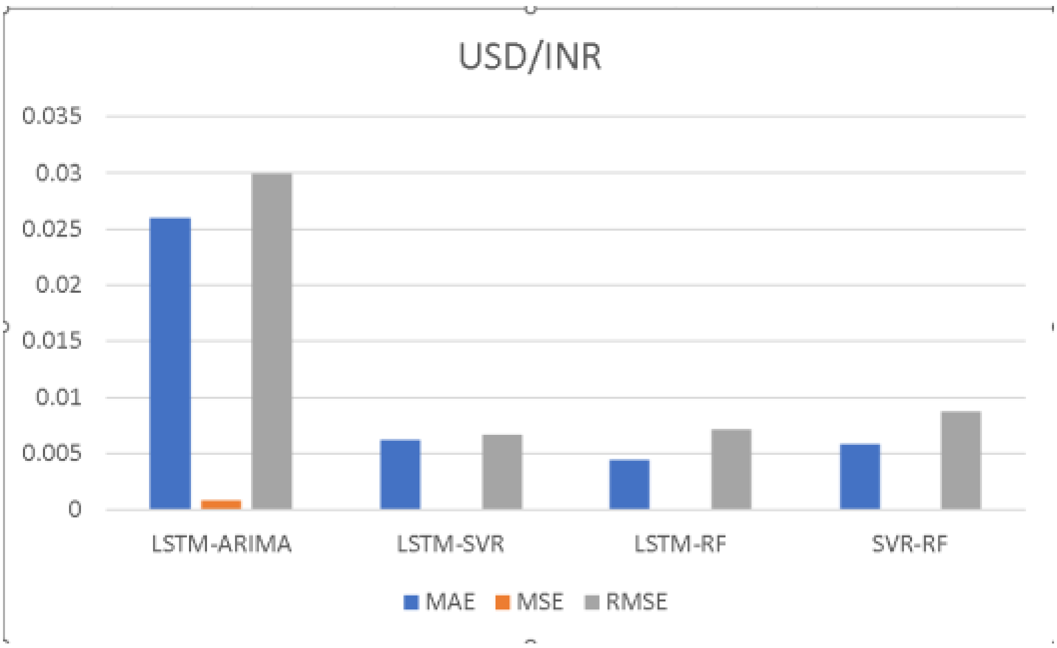


Fig 5.15 evaluation Metrics

As clearly shown in Figure 5.13, 5.14, 5.15 LSTM-SVR performed better than three other models. And the values for MSE, RMSE, and MAE which are error measurement techniques shows that LSTM-SVR had less errors compared to others. At the same time, LSTM-RF is worked better than the other two models and the SVR-RF is the last one.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

Growing economies in parallel to the rapid strong technology improvements brought the needs for forecasting in almost every industry. Whether it is customer-based marketing, car demands, mobile devices or currency exchange rate. Currency exchange rate forecasting became a challenge for human beings. Almost everybody's economy directly or indirectly connected to foreign currency prices. Therefore, currency exchange is vital for humans in this era. Machine learning and generally artificial intelligence technologies going to help humans predict currency rates.

This study developed four models to forecast three pair currency prices in Forex market comparatively. The models are worked well. More suggested Models were selected based on literature review. Which the LSTM-SVR outperformed other three models and LSTM-ARIMA performed least than others. The procedure is described in details, from head to tail. The model's performance was great due to rich data on the training phase. But the fact that different factors still remain which the currency rate has many aspects thus forecasting only time series and historical data is good but not enough.

The volatility property of rates is affected by those factors. The limitation of this study was not considering those factors which likely to be received from financial and political news of the media.

6.1 Future Works

Since the limitation for the study which is mentioned earlier is crucial, resolving those challenges would be a great help for improving the accuracy and reliability of the result of this study. Therefore, the first future work can be a combination of historical data with daily financial and political news analysis with Natural Language Processing (NLP) models.

The second suggestion for accuracy and reliability improvements is to analyse the factors which are mentioned in last chapters and examine whether collecting the data from those factors which affects the currency prices directly can be combined with historical data or not. How would be the accuracy if some futures of those factors combine with historical data in order to train and test the algorithms based on them. According to believe of the author they will have crucial impacts on the result if the extraction and collection of data conduct precisely and accordingly. Reaching to the point of predicting the currency exchange rate does not look impossible. According to the author's believe machine learning will be able to predict currency rate precisely soon. The impact would be, if the technology is handed to the world business leaders, the gap between the societies will increase rapidly as it did somehow. But there is one hope that each technology that humans invent is able to control itself and balance the damages and its usefulness.

REFERENCES

- http://docs.neu.edu.tr/library/6721800683.pdf
- https://www.imperial.ac.uk/media/imperial-college/faculty-of-natural-sciences/department-of-mathematics/math-finance/Nielsen_Laurids-Gert_01424460.pdf
- https://www.mdpi.com/1911-8074/15/1/2
- Exchange Rate Forecasting via a Machine Learning Approach (scirp.org)
- https://cs229.stanford.edu/proj2018/report/76.pdf
- Yu X, Chu Y, Jiang F, Guo Y, Gong D (2018) SVMs classification based two-side cross domain collaborative filtering by inferring intrinsic user and item features. Knowl Based Syst 141:80–91 13336 Neural Computing and Applications (2022) 34:13319–13337 123
- Yu X, Jiang F, Du J, Gong D (2019) A cross-domain collaborative filtering algorithm with expanding user and item features via the latent factor space of auxiliary domains. Pattern Recognit 94:96–109
- Zhang K, Zhong G, Dong J, Wang S, Wang Y (2019) Stock market prediction based on generative adversarial network. Procedia Comput Sci 147:400–406

APPENDICES

A.SAMPLE CODING:

1.LSTM-SVR_EUR/INR

```
# ### Phase1: Collecting Required Packages and Dataset

# %%

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

from tensorflow import keras

from keras.models import Sequential

from keras.layers import Dense, LSTM

from sklearn.preprocessing import MinMaxScaler

from sklearn.preprocessing import MinMaxScaler

from sklearn.model_selection import train_test_split

from sklearn.metrics import mean_squared_error,mean_absolute_error,r2_score

# %%

df = pd.read_csv("C:/Users/Mohan/Documents/___.__/Project/NZDUSD.csv")

# ### Phase2: Preprocessing

# %%

df.head()

# %%

print(df.info())

# %%
```

```
df['Date'] = pd.to_datetime(df['Date'])

df.set_axis(df['Date'], inplace=True)

# %%

df.info()

# %%

data=df[['Close']]

df['Close'].plot(y='Close')

plt.show()

# %%

split_date = pd.Timestamp('2019-01-01 23:58:00')

test = data.loc[split_date:]

train = data.loc[:split_date]

ax = train.plot()

test.plot(ax=ax)

plt.legend(['train', 'test'])

plt.show()

# %%

sc = MinMaxScaler()

train_sc = sc.fit_transform(train)

test_sc = sc.transform(test)

#train_sc=np.log(train)

#test_sc=np.log(test)

# %%

train_sc_df = pd.DataFrame(train_sc, columns=['Y'], index=train.index)
```

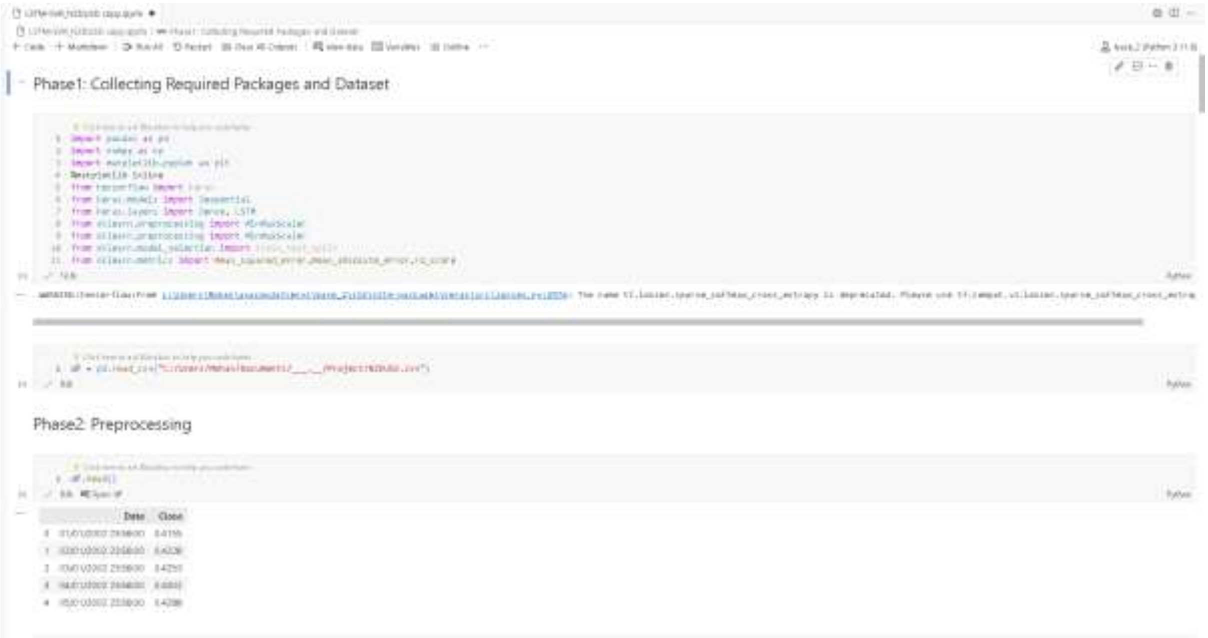
```
test_sc_df = pd.DataFrame(test_sc, columns=['Y'], index=test.index)
for s in range(1,2):
    train_sc_df['X_{}'.format(s)] = train_sc_df['Y'].shift(s)
    test_sc_df['X_{}'.format(s)] = test_sc_df['Y'].shift(s)
X_train = train_sc_df.dropna().drop('Y', axis=1)
y_train = train_sc_df.dropna().drop('X_1', axis=1)
X_test = test_sc_df.dropna().drop('Y', axis=1)
y_test = test_sc_df.dropna().drop('X_1', axis=1)
X_train = X_train.to_numpy()
y_train = y_train.to_numpy()
X_test = X_test.to_numpy()
y_test = y_test.to_numpy()
# ### Phase3: LSTM layer
# %%
look_back = 10
model = Sequential()
model.add(LSTM(units=50, return_sequences=True, input_shape=(look_back, 1)))  #
Hyperparameter: Number of LSTM units
model.add(LSTM(units=50))
model.add(Dense(1))
model.compile(loss="mse", optimizer="adam")
model.fit(X_train, y_train, epochs=100, batch_size=32)
# %%
model.summary()
# %%
```

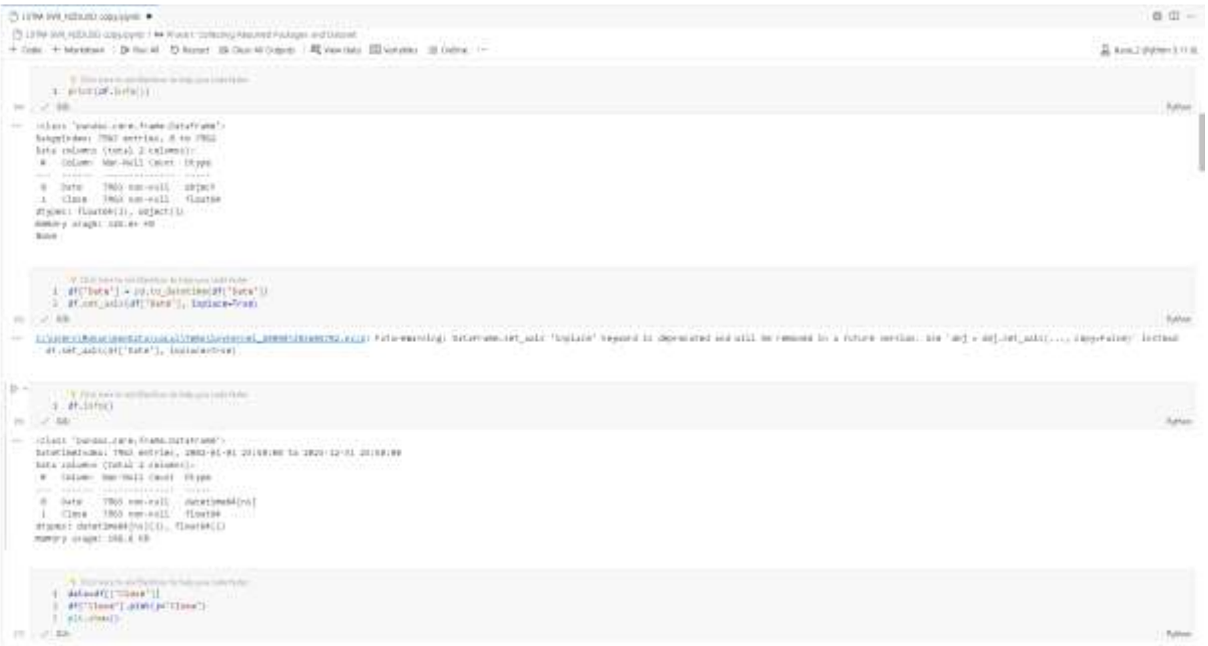
```
lstm_predictions = model.predict(test_sc)

# %%

residuals = test_sc - lstm_predictions

# %%

residuals.ravel()

# %%

residuals= pd.DataFrame(residuals, columns=['Y'])

for s in range(1,2):

    residuals['X_{}'.format(s)] = residuals['Y'].shift(s)

residuals_x = residuals.dropna().drop('Y', axis=1)

residuals_y = residuals.dropna().drop('X_1', axis=1)

# %%

residuals_x.shape,residuals_y.shape

# ### Phase4: Svr layer

# %%

from sklearn.svm import SVR

svr_model = SVR(kernel='poly', C=1e3, epsilon=0.2)

# %%

svr_model.fit(residuals_x, residuals_y)

# %%

svr_predictions = svr_model.predict(residuals_x)

# %%

svr_predictions=svr_predictions.reshape(-1,1)
```

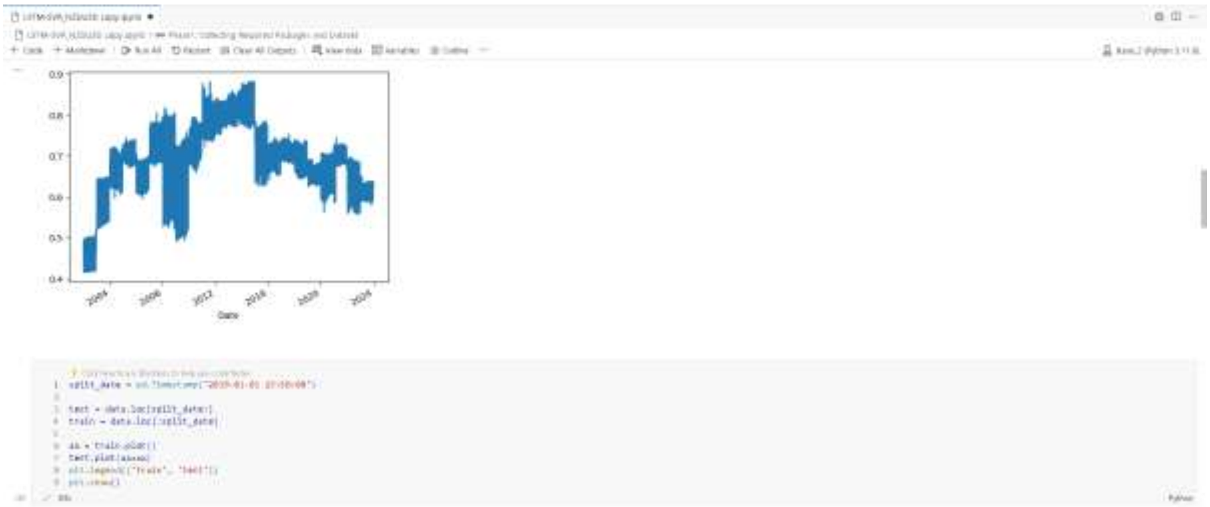
```
# %%
lstm_predictions.shape,svr_predictions.shape
# %%
final_predictions = lstm_predictions[1:] + svr_predictions
# %%
test_sc[1:].shape,final_predictions.shape
test_mse = mean_squared_error(test_sc[1:], final_predictions)
print('Test MSE:', test_mse)
test_sc.shape,final_predictions.shape
mae_lstm = mean_absolute_error(lstm_predictions[1:],y_test)
mae_svr =mean_absolute_error(svr_predictions, y_test)
mae_hybrid =mean_absolute_error(final_predictions,y_test)
mse_lstm = mean_squared_error(lstm_predictions[1:],y_test)
mse_svr =mean_squared_error(svr_predictions, y_test)
mse_hybrid =mean_squared_error(final_predictions,y_test)
rmse_lstm = np.sqrt(mean_squared_error(lstm_predictions[1:],y_test))
rmse_svr =np.sqrt(mean_squared_error(svr_predictions, y_test))
rmse_hybrid =np.sqrt(mean_squared_error(final_predictions,y_test))
r2_lstm = r2_score(lstm_predictions[1:],y_test)
r2_svr =r2_score(svr_predictions, y_test)
r2_hybrid =r2_score(final_predictions,y_test)


# %% [markdown]
# ### Phase5: Results and Conclusions
```

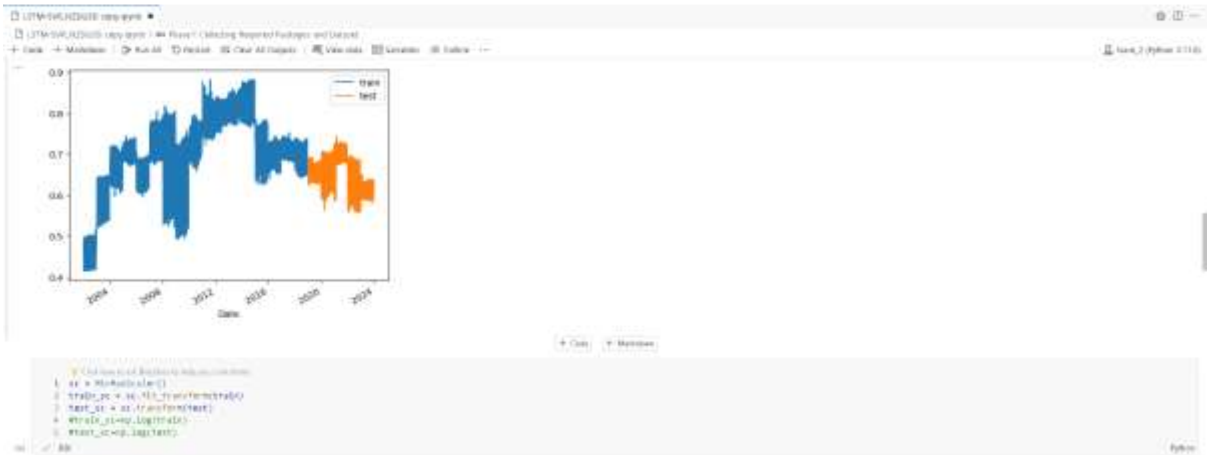
```
# %%
print("Mean Absolute Error (LSTM)  :", mae_lstm)
print("Mean Absolute Error (SVR)    :", mae_svr)
print("Mean Absolute Error (Hybrid):", mae_hybrid)
print("")
print("Mean Squared Error (LSTM)  :", mse_lstm)
print("Mean Squared Error (SVR)    :", mse_svr)
print("Mean Squared Error (Hybrid):", mse_hybrid)
print("")
print("Root Mean Squared Error (LSTM)  :", rmse_lstm)
print("Root Mean Squared Error (SVR)    :", rmse_svr)
print("Root Mean Squared Error (Hybrid):", rmse_hybrid)
print("")
print("r2_score (LSTM)  :", r2_lstm)
print("r2_score (SVR)    :", r2_svr)
print("r2_score (Hybrid):", r2_hybrid)
# Plot forecasts
plt.figure(figsize=(12,6))
plt.plot(y_test, label='Actual')
plt.plot(lstm_predictions, label='LSTM')
#plt.plot(svr_predictions, label='ARIMA')
plt.plot(final_predictions, label='Hybrid')
plt.legend()
plt.show()
```

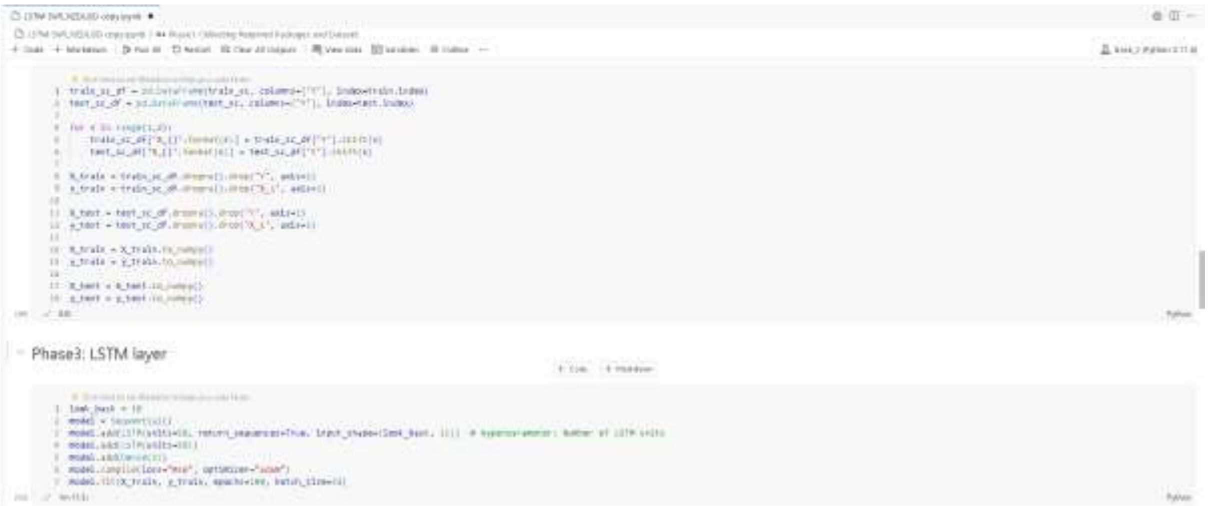
B.SAMPLE SCREENSHOTS

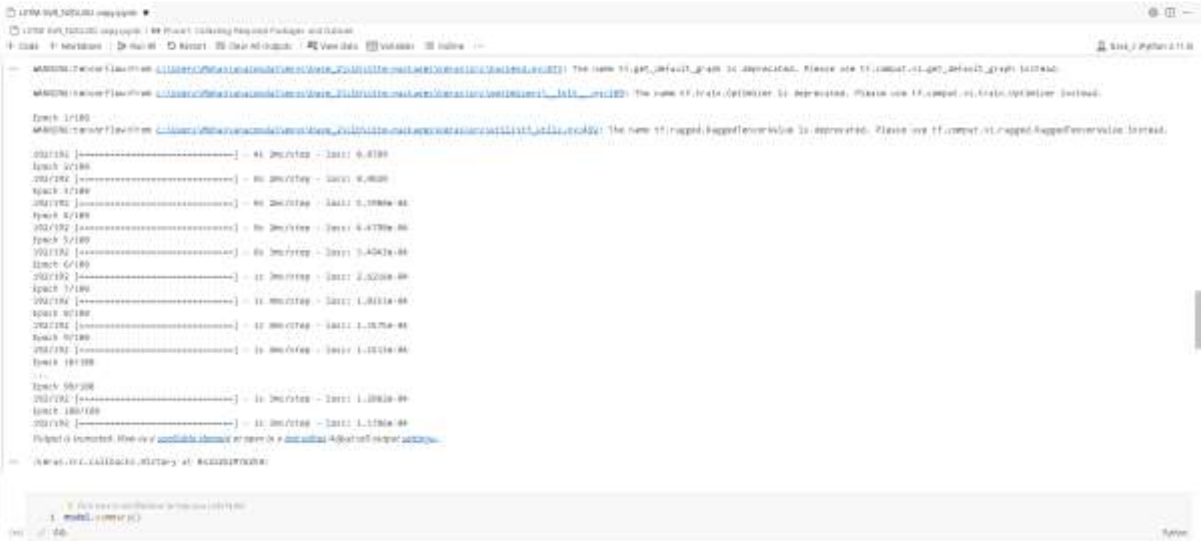




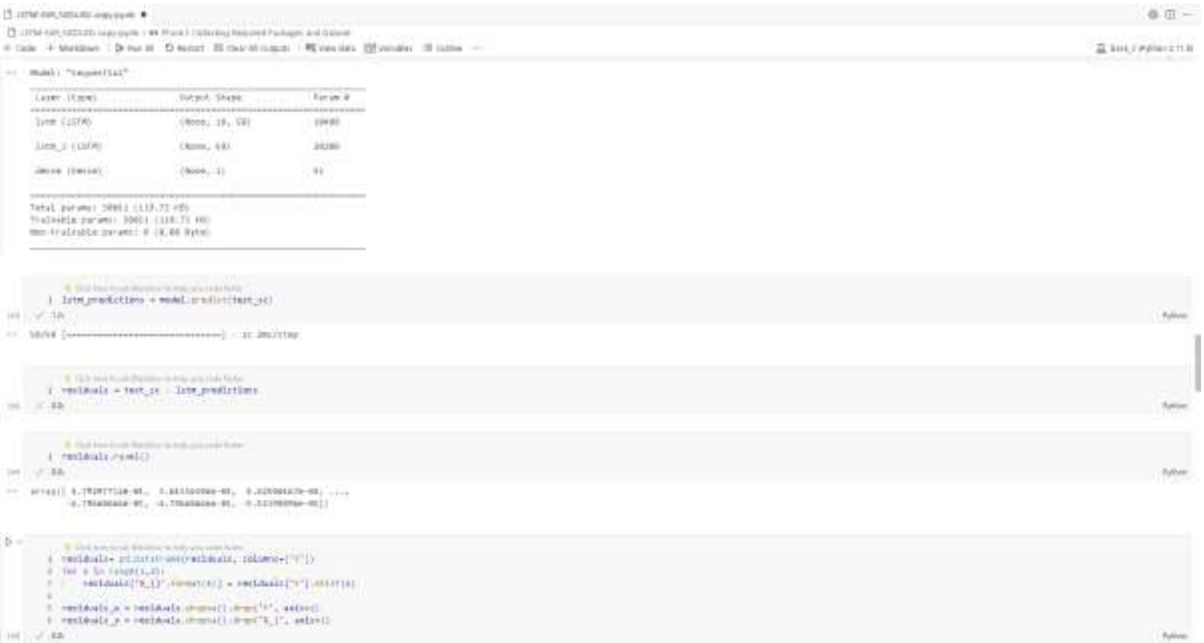

Date


train
test
Date

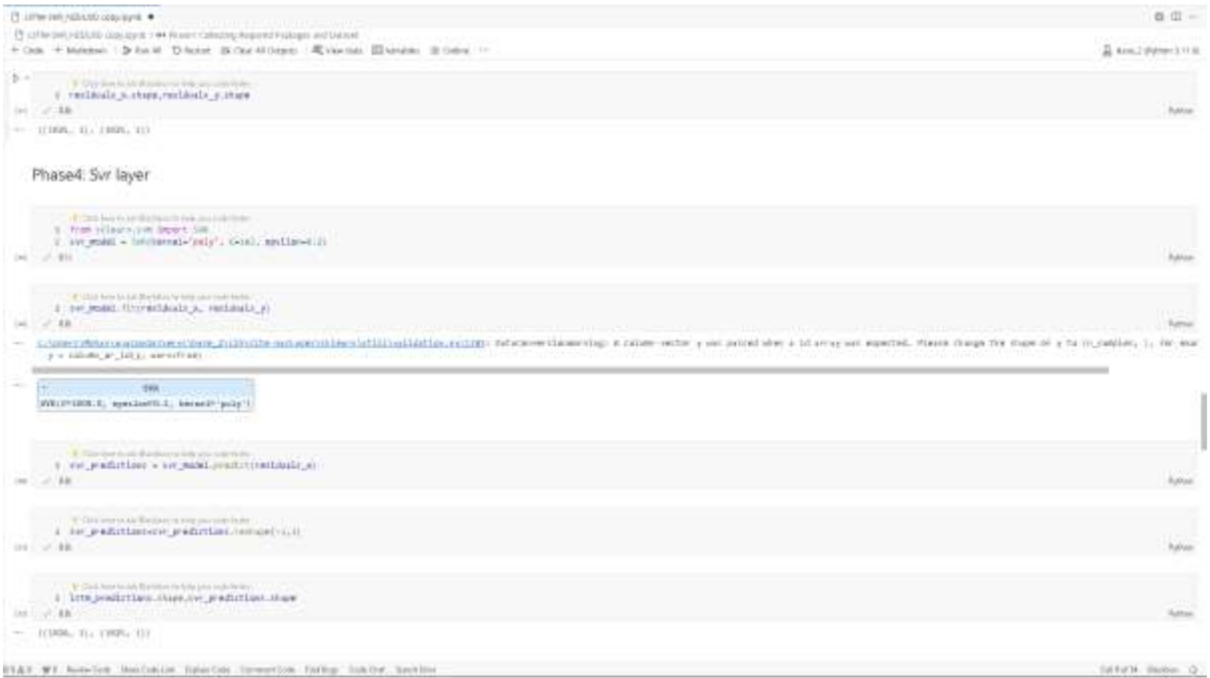

Phase3: LSTM layer

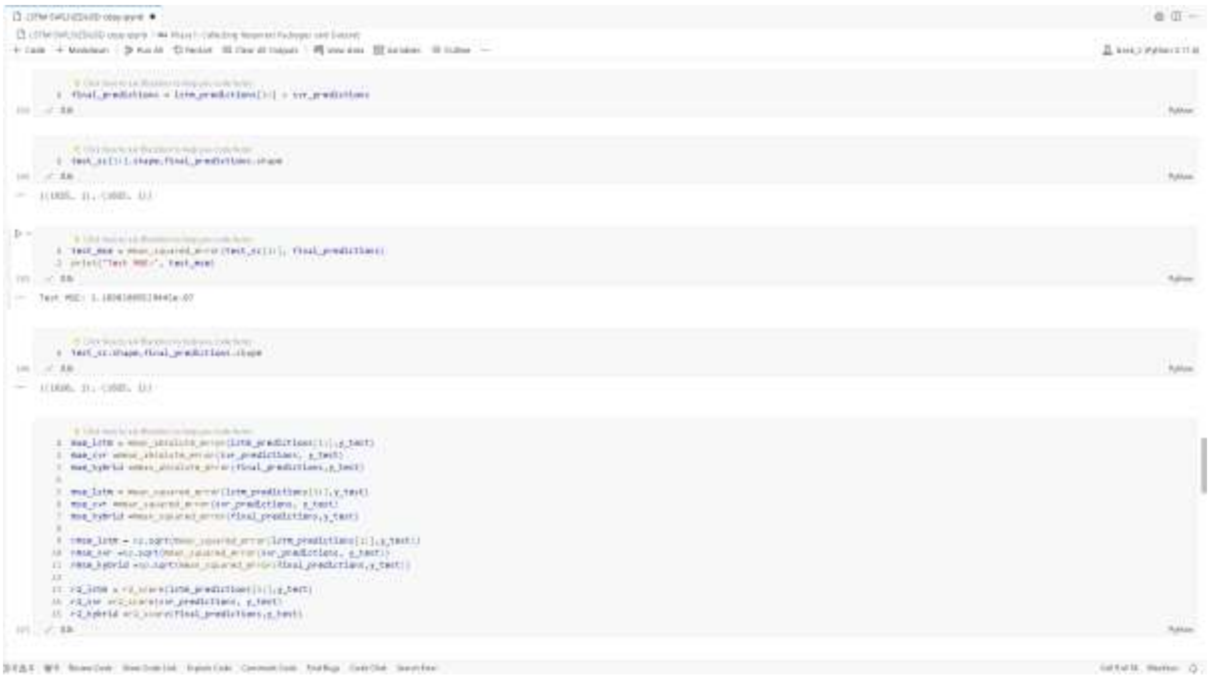


Epoch 1/100
model.summary()



Model: "sequential"
Layer (type)
Output Shape
Param #
lstm_predictions = model.predict(test_sc)
residuals = test_sc - lstm_predictions
residuals.ravel()


residuals_x.shape,residuals_y.shape
Phase4: Svr layer
from sklearn.svm import SVR
svr_model.fit(residuals_x, residuals_y)
SVR
svr_predictions = svr_model.predict(residuals_x)
svr_predictions=svr_predictions.reshape(-1,1)
lstm_predictions.shape,svr_predictions.shape


final_predictions = lstm_predictions[1:] + svr_predictions
print("Test MSE:", test_mse)
mae_lstm = mean_absolute_error(lstm_predictions[1:],y_test)
mae_svr =mean_absolute_error(svr_predictions, y_test)
mae_hybrid =mean_absolute_error(final_predictions,y_test)
mse_lstm = mean_squared_error(lstm_predictions[1:],y_test)
mse_svr =mean_squared_error(svr_predictions, y_test)
mse_hybrid =mean_squared_error(final_predictions,y_test)
r2_lstm = r2_score(lstm_predictions[1:],y_test)
r2_svr =r2_score(svr_predictions, y_test)
r2_hybrid =r2_score(final_predictions,y_test)

Phase5: Results and Conclusions

```
print("Mean Absolute Error (LSTM)  :", mae_lstm)
print("Mean Absolute Error (SVR)   :", mae_svr)
print("Mean Absolute Error (Hybrid):", mae_hybrid)
print("")
print("Mean Squared Error (LSTM)  :", mse_lstm)
print("Mean Squared Error (SVR)   :", mse_svr)
print("Mean Squared Error (Hybrid):", mse_hybrid)
print("")
print("Root Mean Squared Error (LSTM)  :", rmse_lstm)
print("Root Mean Squared Error (SVR)   :", rmse_svr)
print("Root Mean Squared Error (Hybrid):", rmse_hybrid)
print("")
print("r2_score (LSTM)  :", r2_lstm)
print("r2_score (SVR)   :", r2_svr)
print("r2_score (Hybrid):", r2_hybrid)
# Plot forecasts
plt.figure(figsize=[illegible])
plt.plot(y_test, label='Actual')
plt.plot(lstm_predictions, label='LSTM')
#plt.plot(svr_predictions, label='SVR')
plt.plot(final_predictions, label='Hybrid')
plt.legend()
plt.show()
```

```
Mean Absolute Error (LSTM)  : [illegible]
Mean Absolute Error (SVR)   : [illegible]
Mean Absolute Error (Hybrid): [illegible]

Mean Squared Error (LSTM)  : [illegible]
Mean Squared Error (SVR)   : [illegible]
Mean Squared Error (Hybrid): [illegible]

Root Mean Squared Error (LSTM)  : [illegible]
Root Mean Squared Error (SVR)   : [illegible]
Root Mean Squared Error (Hybrid): [illegible]

r2_score (LSTM)  : [illegible]
r2_score (SVR)   : [illegible]
r2_score (Hybrid): [illegible]
```