FORECASTING SALES DATA USING ARIMA, SARIMA AND LSTM MODEL

A PROJECT REPORT

Submitted by

MOHAMMED BASHEERUDDIN [RA2011031010001]

ANIRUDH VISHWANATH [RA2011031010045]

Under the Guidance of

Mrs. SAVEETHA D

Assistant Professor, Department of Networking and Communications

In partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING
with specialization in INFORMATION TECHNOLOGY

DEPARTMENT OF NETWORKING AND COMMUNICATIONS

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203

MAY 2024

Department of Networking and Communications

SRM Institute of Science and Technology

Own Work Declaration Form

Degree/ Course : **B.Tech in Computer Science and Engineering withspecialization in Information Technology**

Student Names : **Mohammed Basheeruddin, Anirudh Vishwanath**

Registration Number: **RA2011031010001, RA2011031010045**

Title of Work : **Forecasting Sales data Using ARIMA, SARIMA and LSTM Model**

We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is our own except where indicated, and that We have met the following conditions:

- Clearly referenced / listed all sources as appropriate.
- Referenced and put in inverted commas all quoted text (from books, web, etc.)
- Given the sources of all pictures, data etc. that are not my own.
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that We have received fromothers (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook.

We understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
We are aware of and understand the University's policy on Academic misconduct and plagiarism and we certify that this assessment is our own work, except where indicated by referring, and that we have followed the good academic practices noted above. **Anirudh Vishwanath [RA2011031010045]** **Mohammed Basheeruddin [RA2011031010001]** **DATE:** 13/5/24
If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. C. Muthamizhchelvan,** Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr.T.V. Gopal,** for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman, Professor & Chairperson**, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department, **Dr. Annapurani.K**, Professor and Head, Department of Networking and Communications, School of Computing, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinator, **Dr. G. Suseela**, Associate Professor, Panel Head, **Dr. Godwin Ponsam**, Associate Professor and members, **Dr. C. Fancy**, Assistant Professor, **Mrs. Saveetha D,** Assistant Professor, **Dr. M. Maranco,** Assistant Professor, Department of Networking and Communications, School of Computing, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, **Dr. Preethiya T,** Department of Networking and Communications, School of Computing, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, **Mrs. Saveetha D**, Assistant Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for providing me/us with an opportunity to pursue our project under his/her mentorship. She provided us with the freedom and support to explore the research topics of our interest. Her passion for solving problems and makinga difference in the world has always been inspiring.

We sincerely thank the Networking and Communications Department staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

Anirudh Vishwanath [RA2011031010045]

Mohammad Basheeruddin [RA2011031010001]

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603203

BONAFIDE CERTIFICATE

Certified that 18CSP109L project report titled "**Forecasting Sales data Using ARIMA, SARIMA and LSTM Model**" is the bonafide work of Mr. Mohammed Basheeruddin [Reg. No. RA2011031010001], Mr. Anirudh Vishwanath [Reg.No. RA2011031010045] who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation based on whicha degree or award was conferred on an earlier occasion for this or any other candidate.

Mrs. SAVEETHA D
SUPERVISOR
ASSISTANT PROFESSOR
Department of Networking and Communications

Dr. ANNAPURANI.K
HEAD OF DEPARTMENT
Professor
Department of Networking and Communications

INTERNAL EXAMINER

EXTERNAL EXAMINER

TABLE OF CONTENTS

ABSTRACT

In the framework of sales forecasting, the project investigates the field of time series forecasting, with a particular emphasis on analyzing and comparing the forecasting precision of ARIMA, SARIMA, and LSTM models. Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) are some of the performance metrics that are utilized in this procedure. It's a thorough workflow that includes data preparation, rigorous model training, and diligent assessment. To facilitate a comparative comparison of the forecasting capabilities of the models, the visualization component makes use of a variety of graphical representations, such as incisive line graphs and instructive bar charts. Determining which model is the most capable of providing accurate sales forecasts is the key objective. This has significant implications for improving decision-making processes in the areas of sales and inventory management. This research program helps companies to design flexible and proactive strategies that are customized to changing market landscapes. As a result, enterprises can enhance their performance and strategic resilience. This is accomplished by providing businesses with data-driven insights and rigorous forecasting tools.

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

CCTV	Closed-circuit television
3D	Three Dimensional
PC	Personal Computer
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
AI	Artificial Intelligence
ML	Machine Learning
DNA	Deoxyribonucleic Acid
C3D	Convolutional 3 Dimensional
LSTM	Long Short-Term Memory
CLTSM	Convolutional Long Short-term Memory
JTSM	Joint Time series modeling
GRU	Graphics Recurrent Unit
RAM	Random-Access Memory
OpenCV	Open-source Computer Vision library

CHAPTER 1
INTRODUCTION

1.1 General

As the foundation of predictive analytics, time series forecasting is essential to contemporary corporate strategy in a variety of industries. Beyond simple forecasting, its importance lies in its strategic application in decision-making, risk management, resource allocation, and operational planning. Precise predicting of future sales trends is essential for managing inventory levels, raising customer happiness, increasing income, and promoting long-term success in the field of sales forecasting.

The time series forecasting research project aims to investigate the details of three well- known forecasting models: Autoregressive Integrated Moving Average (ARIMA), Seasonal Autoregressive Integrated Moving Average (SARIMA), and Long Short-Term Memory (LSTM) neural networks. These models are a good fit for sales forecasting jobs because they have a track record of capturing seasonal changes, temporal dependencies, and underlying patterns in time series data.

The research workflow is based on data preparation. Ensuring the quality, relevance, and integrity of the data used for model training and evaluation is the goal of several critical processes in this phase. To generate useful variables and extract significant insights that improve the prediction potential of the models, feature engineering techniques are utilized. To scale the data suitably and ensure consistency and stability during model training, normalization techniques are used. Data cleansing procedures are employed to mitigate the presence of missing values, outliers, and noise, thereby improving the dataset's overall quality.

Following preparation, the data is divided into training and testing sets. The forecasting models are trained on the training set so they can identify underlying patterns and learn from past sales data. It can assess the models' predicted

performance on data that has not yet been analyzed by using the testing set as a separate validation set. The rigorous process of training and evaluating the models guarantees their robustness, accuracy, and ability to generalize well to new data instances.

A variety of assessment metrics, such as mean squared error (MSE), root mean square error (RMSE), and mean absolute error (MAE), to evaluate the models' forecasting accuracy. These metrics offer numerical assessments of the predictive ability of the models, enabling us to evaluate and contrast how well they capture sales trends and generate precise forecasts. It also take into account other aspects like interpretability, scalability, and computational efficiency when determining if the models are appropriate for practical uses.

The research strategy stands out due to its focus on data visualization as a potent instrument for conveying insights and streamlining decision-making. Model projections, comparative studies, and historical sales data are all shown using visualizations including line graphs, bar charts, and time series plots. In addition to helping to comprehend trends and patterns, these visual aids also function as powerful communication tools for stakeholders, providing them with the ability to derive actionable insights and make well- informed decisions based on anticipated results.

In the context of sales forecasting, a comparison examination of the LSTM, SARIMA, and ARIMA models offers important insights into the advantages and disadvantages of each model. LSTM models may perform particularly well in situations involving intricate temporal patterns and changing trends because of their reputation for capturing long-term dependencies and nonlinear interactions. SARIMA models work well for capturing cyclicality and periodic trends in sales data because they can capture seasonal fluctuations and autocorrelation. Even if their structure is more straightforward, ARIMA models can nevertheless provide reliable predicting results for some kinds of sales data that exhibit stationary behavior and distinct temporal patterns.

Moreover, the study goes beyond comparing models to tackle real-world issues and their consequences for companies using these forecasting tools. Investigate the models' scalability and computational efficiency, taking into account variables including the volume of data, the complexity of the model, and the processing resources needed for both training and inference. It also examine interpretability and explainability, evaluating how effectively stakeholders can comprehend and interpret the model predictions to support decision-making.

The research yielded valuable insights and conclusions that have important ramifications for companies that compete in markets. Precise sales projections enable companies to make knowledgeable choices about resource allocation, pricing schemes, marketing campaigns, inventory control, and production scheduling. Through the utilization of sophisticated forecasting models like LSTM, SARIMA, and ARIMA, enterprises canenhance their competitiveness, predict market patterns, reduce hazards, and seize new prospects.

Furthermore, by developing strategies, methods, and best practices for model selection, assessment, and deployment, the study advances the area of time series forecasting. It advances knowledge and competence in the field of sales forecasting and predictive analytics by disseminating the findings and methodology to industry practitioners and the research community.

To sum up, the time series forecasting research project offers a thorough and all-encompassing method for assessing and contrasting forecasting models for sales prediction.The goal is to equip organizations with the necessary actionable insights, strong forecasting capabilities, and strategic foresight to traverse complicated market dynamics and propel sustainable growth in the digital era. It will achieve this through meticulous data preparation, model training, evaluation, and visualization.

1.2 Purpose

This project's main goal is to improve knowledge of various time series forecasting techniques and how they might be used in practical situations, particularly when predicting sales. The research accomplishes several interrelated goals, all of which add to a comprehensive understanding of forecasting methods and how they affect corporate decision-making procedures. Thoroughly assessing and contrasting the performance and accuracy of three different forecasting models—ARIMA, SARIMA, and LSTM—is one of the main goals. The objective of the study is to determine which model is best for precise sales forecasting by evaluating.

different models using quantitative indicators like MSE, RMSE, and MAE. Future model deployment and selection methods will be guided by the insights gained from this comparison research, which highlights the advantages and disadvantages of each model.

company Decision Support: Specifically in the areas of sales and inventory management, the initiative directly affects company decision-making procedures. Businesses may optimize inventory levels, effectively manage supply chains, and make well-informed strategic decisions on manufacturing, marketing, and allocation of resources by using accurate sales forecasting. Organizations may maximize market possibilities, save expenses, and improve operational efficiency by utilizing sophisticated forecasting methodologies.

Data-Driven Strategies: Encouraging data-driven decision-making in businesses is another major goal. The initiative promotes an anticipatory approach to making decisions based on empirical facts and predictive insights by demonstrating the efficacy of data analysis and forecasting models. Businesses
are now better equipped to respond to changing market conditions, reduce risks,

and seize new opportunities thanks to this move toward data-driven strategies, which promotes sustainability and long-term success.

Research Advancements: The project makes a significant contribution to the field of forecasting and analysis of time series research, in addition to its practical applications. The amount of information regarding forecasting strategies is expanded by the comparative analysis of sophisticated deep learning algorithms (LSTM) and conventional statistical models (ARIMA, SARIMA). The project's results and methods may be used as a guide for further studies, encouraging creativity and ongoing advancements in forecasting methods.

Finally, by offering practical experience in data pretreatment, model creation, assessment, and visualization, the project fulfills educational and learning goals. For students, researchers, with practitioners looking to get a deeper grasp of forecasting approaches and their ramifications, this provides a useful practical illustration of how theoretical principles in the forecasting of time series translate into actual applications.

1.3 Scope

The project's extensive scope goes beyond discrete activities to include a unified workflow that combines data preparation, model building, optimization, assessment, and comparison analysis in a harmonious manner. Each stage of the project makes a unique contribution to the overall impact and success, guaranteeing the accuracy, robustness, and usability of the forecasting models for practical applications.

Data Collection and Preparation:

The project's foundation is laid during the first stage of data collection and preparation. Accessing a variety of sources, including databases, spreadsheets, APIs, and outside data providers, is necessary to obtain historical sales data. To provide a trustworthy dataset for analysis, this stage necessitates careful assessment of data quality, completeness, and relevance.

One of the most important preprocessing steps is data cleaning, which is locating and resolving outliers, missing values, and abnormalities in the data. Data normalization, outlier identification, and imputation are some of the techniques used to improve data integrity and get it ready for modeling.

To extract useful features or attributes from the sales data, feature engineering is essential. This entails converting unstructured data into meaningful variables that identify key trends, patterns, and seasonality. Prioritizing pertinent features for modeling can also be accomplished by using feature selection strategies.

Model Development and Training:

Three different models—ARIMA, SARIMA, and LSTM—each with unique techniques and capacities, will be developed and trained as part of this project. Using statistical methods, the ARIMA and SARIMA models simulate trends, autocorrelation in the sales data. These models are appropriate for the analysis of time series data with predictable patterns because their autoregressive and moving average components capture temporal dependencies and seasonal variations.

On the other hand, the LSTM model is a deep learning architecture made to work with sequential data that has long-term dependencies. It is ideally suited for identifying subtle patterns in sales data because of its capacity to recognize temporal links and understand intricate nonlinear patterns.

Model Adjustment and Enhancement:

To guarantee the robustness, generalizability, and forecast accuracy of the models, tweaking and optimization are essential stages. Model parameters, such as order parameters in ARIMA/SARIMA models and architectural parameters in LSTM

models, must be fine-tuned through parameter selection.

Model performance can be optimized through hyperparameter tuning, which involves modifying variables like regularization strategies, batch sizes, and learning rates. Model performance on unknown data is evaluated and overfitting or underfitting is avoided by using model validation approaches, such as cross-validation and validation datasets.

Evaluation and Comparative Study:

A variety of evaluation metrics, such as MSE, RMSE, MAE, and possibly other metrics like MAPE, are used in the project to assess model performance. The forecasting accuracy, computational efficiency, scalability, and reliability of the ARIMA, SARIMA, and LSTM models are compared in detail. This research helps choose the best model for sales forecasting jobs by illuminating the advantages and disadvantages of each model. Visualizations, such as line graphs and comparative charts, are used to illustrate the outcomes of the investigation and communicate critical findings effectively. These visual tools improve the comprehension of stakeholders and make it easier to make decisions based on anticipated results.

Applications and Practical Implications:

The project's results have a lot of applications and practical ramifications for companies in a variety of industries. Organizations may plan their finances, allocate resources, manage their inventories, develop marketing plans with help of accurate sales forecasting.

Businesses can obtain a competitive edge, increase customer happiness, lower expenses, and improve operational efficiency by utilizing sophisticated forecasting models and data- driven insights.

The project's methods and results advance the fields of predictive analytics and time series forecasting, opening the door to creative uses in risk management, demand forecasting, sales forecasting, and strategic decision-making.

To sum up, the project's diverse methodology incorporates advanced modeling tools, data- driven approaches, thorough evaluation, and actionable insights. The project

wants to provide enterprises looking for accurate and trustworthy sales predictions in dynamic market conditions with real value by combining many duties and processes into a single, well-organized framework.

1.4 Software Requirement Specification

To efficiently design, train, and assess your models for a time series forecasting project utilizing LSTM models, you'll require a collection of software tools and libraries. The following are the necessary software needs for this kind of project:

1. Programming Language: Python

Time series forecasting is one of the many data science and machine learning applications that employ Python. Make sure Python is installed on your computer; for compatibility with the necessary libraries, it is best to use the most recent version.

2. System for Integrated Development (IDE):

Select an IDE that makes Python development easier and fits your preferences. Popular options consist of:

- PyCharm
- JupyterLab or Jupyter Notebook
- Visual Studio Code (extended to include Python)
- Spyder

3. Analysis and Data Manipulation Libraries: - Pandas: For managing time series data,preprocessing, and data manipulation.

NumPy: For working with matrices and arrays and performing numerical computations.

4. Visualization Libraries: - Matplotlib: for making histograms, bar charts, and line plots, among other static visualizations.

Seaborn: To improve the visual appeal of charts and to visualize statistical data.

5. Deep Learning and Machine Learning Libraries: - PyTorch or TensorFlow:

To create and train LSTM models, use one of these deep learning frameworks.

Both frameworks offer tools for model optimization and assessment in addition to supporting LSTM layers.

- **Keras**: a sophisticated neural network API built on top of PyTorch or TensorFlow. It makes neural network construction and training—including that of LSTM. models— simpler.
- **Scikit-learn**: Scikit-learn is mostly focused on conventional machine learning algorithms, but it can also be helpful for other tasks like feature selection, data preprocessing, and model validation

6. Libraries for Time Series Analysis and Forecasting: - Statsmodels: offers time series analysis techniques, such as the SARIMA and ARIMA models. beneficial for contrasting and testing deep learning models like LSTM with more conventional statistical techniques.

Facebook Prophet, or Prophet: A library created especially for the forecasting That can automatically manage trends and seasonality.

7. Extra Libraries for Model Assessment and Metrics: - Scikit-learn: For common. metrics used in machine learning assessments, like cross-validation, RMSE, MSE, and MAE.

- TensorFlow/Keras Callbacks: Make use of the callbacks that are already included to oversee model performance throughout training, save checkpoints, and halt the modelearly to avoid overfitting.

8. Data Visualization Tools: - Tableau or Power BI: Not required, but helpful for bringing your forecasting results to life through interactive dashboards and insights.

9. Git: - Version Control System: Suggested for organizing project iterations, version control, and teamwork.

10. Pip: - Package Management: The package installer for Python that manages and installs dependencies and libraries. To guarantee reproducibility and separate the requirements in your project, use a virtual environment.

A complete environment for creating, training, testing, and displaying LSTM models for time series forecasting by configuring these software tools and libraries can be done. Adapt to the versions and libraries to the specifications and compatibility issues of the project.

CHAPTER 2

LITERATURE REVIEW

The use of time series analysis for sales forecasting in the retail industry has garnered significant attention in both academic research and practical applications. This section provides an overview of the key studies and methodologies employed in this domain.

2.1 Sales Analytics Dashboard with ARIMA and SARIMA Time Series

An efficient and user-friendly way for businesses to forecast and assess sales trends is seen in [1] through the web-based dashboard that was made with the Dash and Plot libraries. The dashboard's interactive and visual format makes it simple for users to compare and understand the outcomes of these models. The data series shows a seasonal pattern, which led to the conclusion that the SARIMA model with parameters of (2,1,1) (0,1,1)12 was betterthan the ARIMA model. When seasonal components are present, the SARIMA model can accurately predict future values, as stated. Consequently, the SARIMA model forecastssales with accuracy. The dashboard is an essential tool for businesses to track their progress and meet their sales goals because it gives a user-friendly overview of data and information.

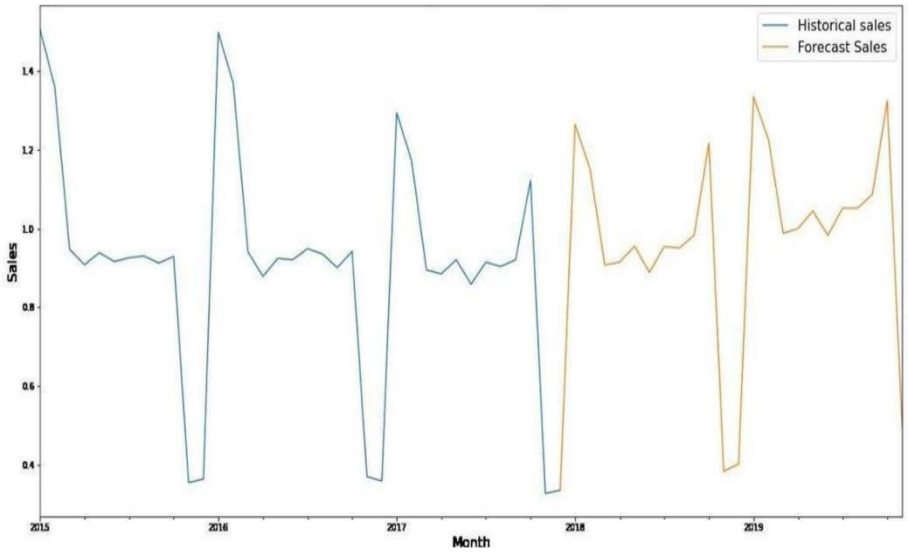

Fig 2.1 Forecasting sales per month using the SARIMA Model

Fig 2.1 shows the forecast of the sales analytics dashboard using the ARIMA and the SARIMA Model

Model	SARIMA	ARIMA
AIC	591.304	981.561
BIC	596.981	989.338
MAE	210256.4484	889494.6390
RMSE	210256.4430	946125.7153
MAPE	62.6823	105.2062

Table 2.1 Comparison between both the models

Table 2.1 shows there are many metrics such as AIC, BIC, MAE, RMSE, MAPE presentwhich help us in comparing the time series models.

2.2 Auto ARIMA and Auto SARIMA Performance in COVID-19 Prediction

Using both the ARIMA and SARIMA, models for prediction, the study's main objective in [2] was to evaluate the Auto-ARIMA and Auto-SARIMA models' efficacy. The study's findings indicate that, in terms of predictive precision, the ARIMA model performed better than the SARIMA model, as evidenced by the more significant accuracy % that the ARIMA model obtained. It is important to keep in mind that these vehicles have flaws that require fixing. Further research is needed to investigate potential enhancements that could lead to more accurate and dependable predictions.

This is particularly important due to the evolving nature of the pandemic and the potentialfor changes to its (a) (b) Figure 3. Plot illustrating the fundamental patterns of the condition for (a) ARIMA (3,1,2) and (b) SARIMA (0,1,1) (1,0,1,7), with a 95% confidence interval. Future studies should focus on developing more complex models that capture the complex interactions between numerous factors that affect the spread of COVID-19.

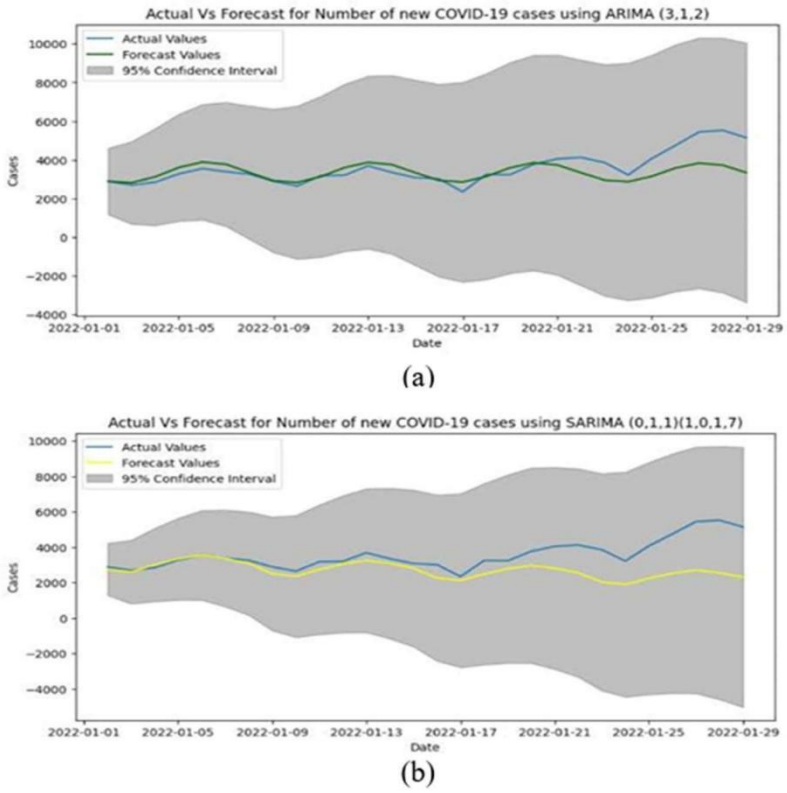

Fig 2.2 Actual vs Forecasted data.

Fig 2.2 shows plotting actual against predicted data with a 95% level of confidence for (a)SARIMA (0,1,1) (1,0,1,7) and (b) ARIMA (3,1,2)

2.3 Sales Forecast for Amazon Sales with Time Series Modeling

The results in [3] show that seasonal ARIMA yields the most precise outcomes when compared with the other methods. Based on the projections results, Amazon may gain a wide p ture ofthe demand and then allocate resources appropriately, hiring more employees, storing more merchandise, or expanding its shipping capacity, to improve customer happiness and deliver high-quality service. Even though the methods' applied error percentage in forecasting (MAPE) is not very high and can be used in the Amazon sales forecast, there are still certain challenges with their application.

These challenges include the following. The data required to accurately execute the forecast is a significant barrier to its implementation. Numerous varied factors, including the population, disposable income for households, interest rate, macroeconomic trend, and so forth, have an impact on Amazon's quarterly sales.

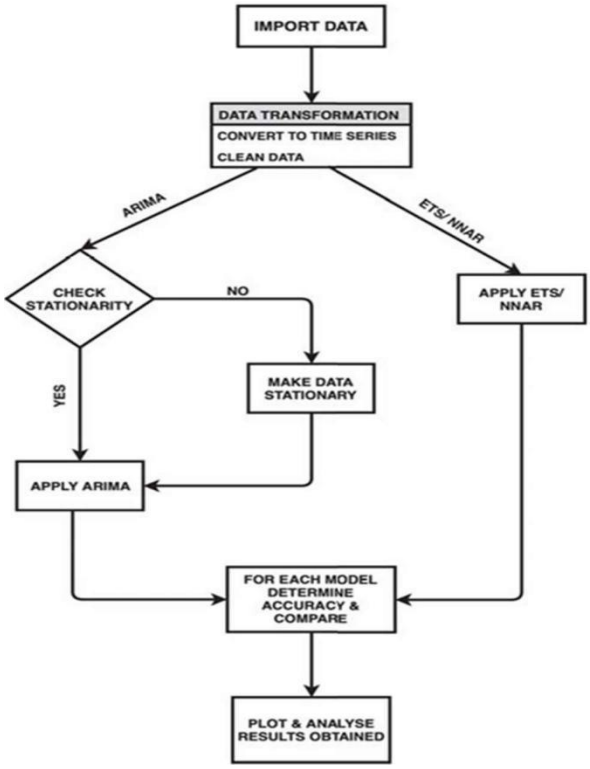

Fig 2.3 Process flow diagram of forecasting

Fig 2.3 shows the flow chart on how the data was imported, analyzed, and transformed can be seen. Later accuracy was determined, and results were obtained.

2.4 Time Series Analysis of Monkeypox Disease Using ARIMA/SARIMA Model

To inform people around the world of the current situation, they set out to investigate the growth rate of monkeypox [4] in detail and provide an overview of the disease's prevalence worldwide. They also talked about the main reasons for it. They attempted to predict the total number of cases of monkeypox between August 1, 2022, and September 6, 2022, using the ARIMA and SARIMA models. Over time, neither tactic was able to produce satisfactory results. On the other hand, the SARIMA model provides more accuracy than the ARIMA model.

The analysis of monkeypox was done with a brief dataset that had a lot of volatility. This study had several flaws, such as problems extrapolating findings from a single study, difficulties selecting appropriate measurements, and problems determining which modelbest described the data. If the dataset were big enough and the variation was less, they would get more accurate results.

Fig 2.4 Working Procedure Diagram

Fig 2.4 shows the working procedure on how the monkeypox disease was analyzed using statistical analysis and applying various algorithms.

2.5 Time Series analysis of the different categories using FBProphet Model

A Kaggle dataset named "Store Sales - time-series data Forecasting0" [5] which is publicly accessible, was used for this study. It contained the sales units and purchases from the salesof various products individuals sold at Favorite stores in the nation of Ecuador for a range of store forms, as well as in various cities and states across the country. Additionally, details regarding the list of festivals and oil prices were provided, both of which seem to influence sales.

Different arrangements of the data are used to build and evaluate a range of models. Compared the models' results with different combinations of hyper-parameters. This study investigated the prospect that non-time-series techniques might forecast future sales during weekdays with equal success to that of weekends. With effective data pre-processing, the model can also identify sales spikes for a range of holidays and usually reports larger saleson vacations than on working days.

Holiday median total sales are higher, despite the fact there are more holidays each year than working days. In addition to time-series techniques, gradient booster and Facebook prophet models consider the effect of holidays on sales, predict sales surges, and increase overall income by providing coupons, arranging deals, and extending further discounts to optimize the possibility of weekend sales.

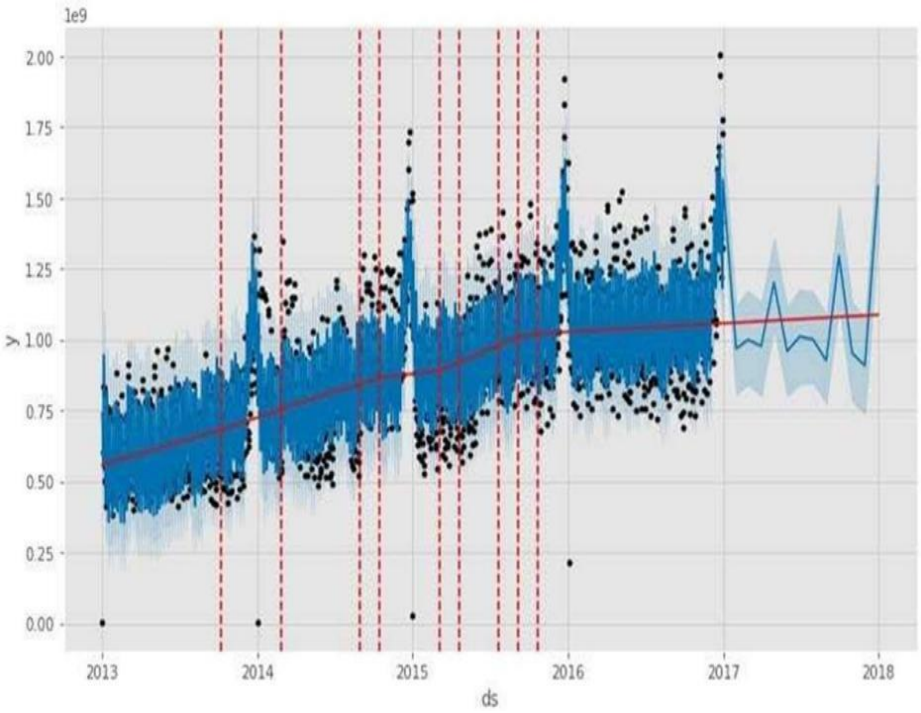

Fig 2.5 Component-wise Facebook Prophet Forecast

Fig 2.5 shows it is specifically designed to simplify time series forecasting tasks, making it accessibleto users with varying levels of expertise in data science and machine learning.

2.6 GARCH, SARIMA, HWSM, and HLTM Models for Comparative Analysis

In this work, they gave a basic introduction to rainfall and the techniques currently employed to measure precipitation. In [6] it also discussed the results they arrived at, the features they utilized for estimating, and the work done by various authors. They then compared the time-series forecasting methods for Telangana rainfall prediction, as described in this work. Based on the results, HWSM performs better than the other approaches in terms of MSE, MAE,and RMSE.

Four popular time series forecasting models—GARCH, SARIMA, HWSM, and HLTM—were examined. Concluding that HWSM performed the best when speaking of error rate is easy; RMSE and MAE were 5.767 and 5.343, MSE and 2.675 were 31.65 and 26.75, respectively.

Holt Winter's Exponential Smoothing or triple exponential smoothing are other names for HWSM. This time-series forecasting model accounts for both trend and seasonality in its forecasting of parameter estimates.

The idea behind HWSM is to apply level, trend, and exponential smoothing to periodic variables. A trend is the general arrangement of values that is observed over a period of time. In a sequence, the average values are called levels. The recurrence of the same set of patterns after predetermined intervals is known as seasonality, or periodicity.

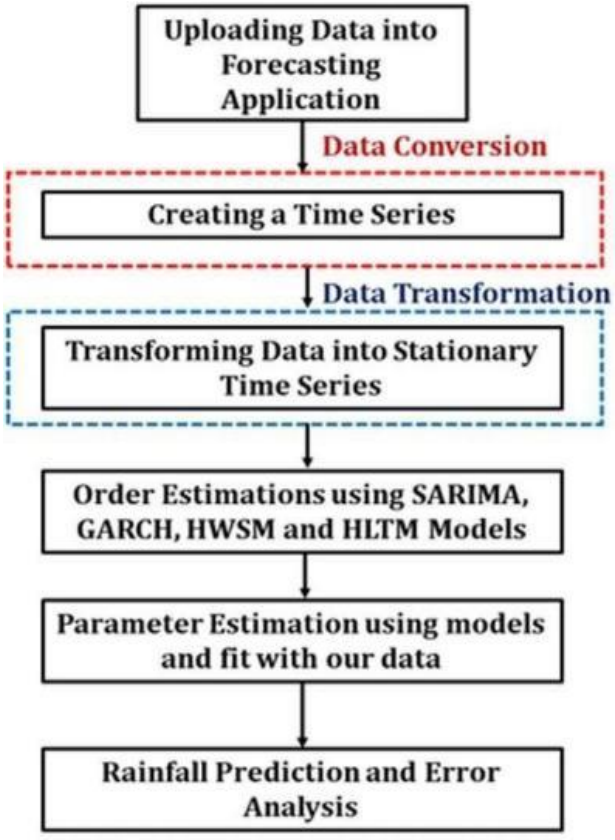

Fig 2.6 Methodology used for Prediction.

Fig 2.6 shows different models such as GARCH, HWSM, HLTM, and SARIMA Models were used to perform the error analysis for time series

Model	RMSE		MAE	
	DS-1	DS-2	DS-1	DS-2
SARIMA	7.856	7.706	3.590	3.541
GARCH	7.682	6.280	4.678	3.959

Table 2.2 RMSE and MAE Values for all the Models Used

Table 2.2 shows the RMSE and MAE values were calculated for SARIMA and the GARCHmodel which helped in finding the best model to use.

2.7 Traditional and Modern Times-Series Forecasting Models

The studies and the empirical findings in [7] were acquired by using Google Collab, a Jupyter Notebook service that Google offers. The Python Statsmodels package, which offers classes and functions for estimating various statistical models, running statistical tests, and examining data, was used to create the statistical models. A deep learning framework called Keras was used to create the LSTM model.

The SARIMA model beats prediction efficiency among the conventional models, accordingto the graphical and statistical results, when it comes to the sales data of the food trading and manufacturing company. Prophet, an open source packaged product, is the most accurate forecasting model available today.

SARIMA and Prophet outperformed Prophet in their study's comparison of time series forecasting models, whereas LSTM showed promise but was more computationally expensive.

The outcomes can be used as a guide by companies looking to use sales data forecasting.

Their research intends to determine the most dependable forecasting model in the future by comparing its performance using sales time series data from various firms.

Further research into the many LSTM model variations and the creation of an LSTM networkthat might perform better than the present sales forecast models are also within the preview of the paper.

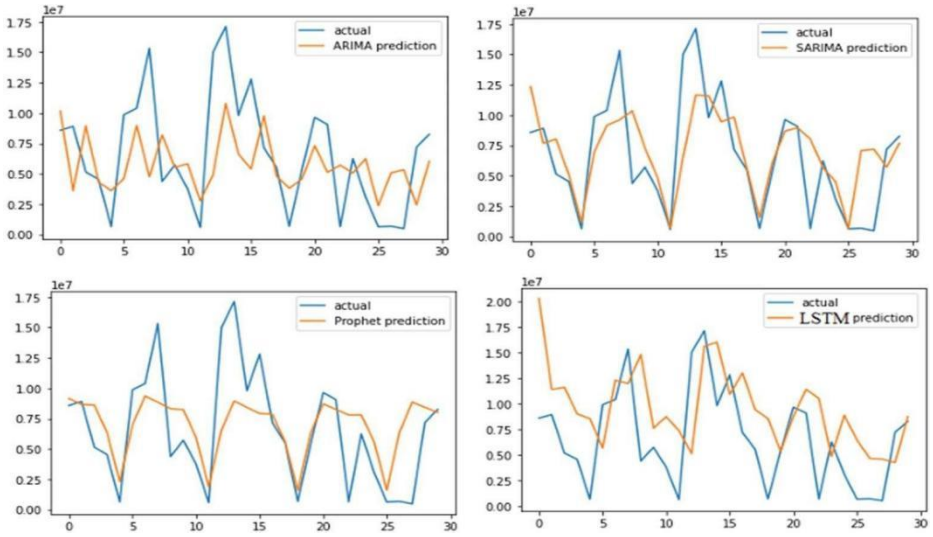

Fig 2.7 Results of four different models and their predictions

Fig 2.7 shows the future predictions of ARIMA, SARIMA, Prophet, and LSTM Models

2.8 Using Regression Algorithms to Forecast Merchandise Sales

When the output of the stall algorithm is insufficient, regression algorithms, a family of supervised algorithms for learning, can be enhanced with integrated learning methods to improve performance.

K-NN regression in [8] anticipates a new input point's value based on feature similarity. Extra Tree Regression is a different integrated learning strategy that is like the Random Forest method. Unlike Random Forest, which replaces portions of the provided data with boot samples,Extra Tree Regressor considers the entire portion of the original data. The choice of cut sites is one of the differences between the two approaches.

The Extra Tree Regressor randomly chooses the cut sites, while Random Forest chooses the best segmentation. Consequently, he is in a stronger position to highlight the performance. An algorithmic evaluation tool was used to compare the output from each model that used the same dataset to get a conclusion.

This approach can be used as a generalization to forecast product sales in different retail locations. This is because, at the conclusion of the process, they preserve the prototype as a sample to ensure it can be utilized right away the following time. Just input information and run the calculation directly to view the forecast results. This procedure is useful. Along with sales volume, this dataset also includes additional exogenous variables that actively impact the process, such vacations, store size, etc. They cannot compute the length of time series algorithm, for example, and can only compare the models generated by the technique of regression due to limitations introduced by the information set structure. This is the study's shortcoming.

Store Number	Year	
	2018	2019
1096	10,1274,84	10,178,608
5144	916,229	1,089,154

Table 2.3 Amount of sales data.

Table 2.3 shows the range of the dataset is provided which is used for this research.

2.9 Comparison of Forecasting Algorithms on Retail Data

Both models in [9] have demonstrated strong success rates, as can be seen from a close examination of the table; the 2019 data demonstrates even more successful outcomes. The sales pattern was identified using a full year of data, and the models then made this trend clear.

Looking at the models in use, the regression model has a lower error rate even though both models produce an acceptable error rate. Examining the MAPE values that are obtained from the time series model, however, does not support the claim that it is useless. The rate of error of the two models are appropriate for both retailers.

According to regression analysis, the two models used were 98.8% and 98.6%, respectively, for shop name 1096 and store number 5144. This result suggests that the model using regression is a better fit for use with Migros data than the time series model.

This study uses sales data obtained from two Migros shops to forecast revenue based on turnovers using logistic and time series models. These businesses differ from one another in terms of both sales volume and how their patrons react to different promotions.

They measured seasonality using data from 2018 and 2019. The results were then contrasted using only 2020 data in order to account for the sales trend, the rise in inflation, and to produce more accurate findings with growing prices. To predict sales, they employed data spanning one and two years. According to regression analysis, the two models used were 98.8% and 98.6%, respectively, with shop number the year 1096 and store number 5144.

They concluded as the regression model performs better than the time series model using retail data from Migros.

Store	Regression Model		Time Series Model	
	2018-2019 Mape	2019 Mape	2018-2019 Mape	2019 Mape
1096	1.5%	1.2%	3.3%	3.4%
5144	2%	1.4%	4.8%	4.6%

Fig 2.9 MAPE Values by store, forecasting model and year.

Fig 2.9 shows the regression model and time series model comparison is calculated usingthe MAPE values.

2.10 Forecasting using Machine learning and Deep Learning Methods

Time series analysis is vital to daily life and plays a significant part in retail sales forecasting. Effective retail sales forecasting is achieved through the application of statistical techniques and time series analysis. It is challenging for big businesses to forecast a huge number of items. The retail sales time series in [10] constantly varies by season and trend, making it difficultto develop accurate forecasts. This paper discusses deep learning and machine learning approaches for retail sales forecasting. This approach is primarily broken down into different approaches for retail sales, including time series analysis, statistical approaches, and deep learning approaches.

In this method, several techniques are applied, including exponential smoothing, time series decomposition, ARIMA, Holt-Winter seasonal method, Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN). The ultimate objective of this approach is to use the present sales to create precise and successful forecasts for futuresales. Because these forecasting techniques use various learning approaches, their prediction outcomes are better than those of other techniques.

The cloud has certain limits for the new study approach in retail sales forecasting. This section outlines and analyses the issue with current retail sales forecasting research approaches. The retail forecasting technique involves a significant amount of computing, which impacts the method's execution time. To train the model for retail sales forecasting, a significant number of computational resources was needed.

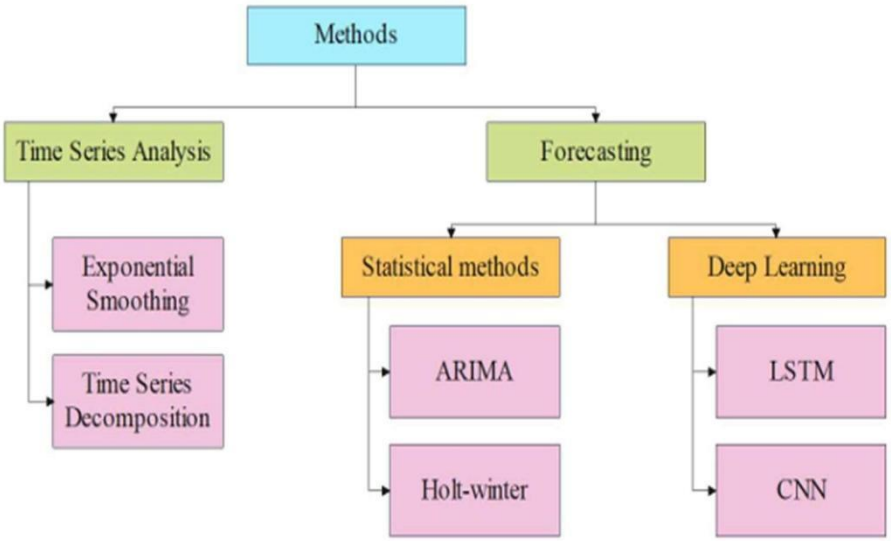

Fig 2.9 Block diagram for the different methods used.

Fig 2.9 shows the block diagram for deep learning and statistical methods such as ARIMA.

Seasonal Time Series Models

Seasonal ARIMA (SARIMA) models extend the capabilities of ARIMA models explicitly modeling seasonal patterns in the data. It employed SARIMA models to forecast sales in the e-commerce sector, emphasizing the importance of capturing seasonal variations for accurate predictions. Additionally, It applied SARIMA models to retail sales data, demonstrating their effectiveness in capturing complex seasonal patterns and improving forecasting accuracy.

Machine Learning Approaches

In recent years, machine learning techniques have emerged as powerful tools for sales forecasting in retail. Deep learning models, such as Long Short-Term Memory (LSTM) networks, have shown promising results in capturing temporal dependencies and non-linear patterns in sales data. Wang et al. 2020 utilized LSTM networks for sales forecasting in the retail industry, achieving superior accuracy compared to traditional time series models. Similarly, Chen et al. 2021 proposed a hybrid model combining LSTM networks with traditional ARIMA models for improved forecasting performance.

Integrated Approaches

Some studies have explored integrated approaches that combine traditional time series models with machine learning techniques for enhanced forecasting accuracy. For example, Li et al. 2021 proposed a hybrid model that integrates SARIMA with Random Forest regression, leveraging the strengths of both approaches to capture both linear and non-linear patterns in sales data. The results demonstrated significant improvements in forecasting accuracy compared to individual models.

Challenges and Future Directions

Despite the advancements in sales forecasting methodologies, several challenges remain. These include the need for robust models that can adapt to dynamic market conditions, the incorporation of external factors such as economic indicators and consumer behavior data, and the development of scalable frameworks for real-time forecasting. Future research directions should focus on addressing these challenges and exploring novel approaches to further improve forecasting accuracy and scalability in the retail industry.

In summary, the literature review highlights the importance of time series analysis for sales forecasting in the retail industry and provides insights into the methodologies and approaches employed in this domain. By leveraging both traditional time series models and machine learning techniques, researchers and practitioners can develop robust forecasting models that effectively capture complex patterns in sales data and support informed decision-making in retail operations.

This literature review provides an overview of key studies and methodologies in the field of sales forecasting in the retail industry, highlighting the importance of time series analysis and the application of traditional models such as ARIMA and SARIMA, as well as emerging approaches including machine learning techniques. It also discusses challenges and future directions for research in this area, emphasizing the need for robust and scalable forecasting frameworks to support decision-making in retail operations.

CHAPTER 3
PROPOSED ARCHITECTURE OF THE ARIMA, SARIMA AND LSTM MODEL

To assess and contrast the performance of the ARIMA, SARIMA, and LSTM models regarding sales forecasting, a systematic approach is used in the project's recommended methodology. The procedure starts with gathering historical sales data, then is subsequently put through a rigorous preparation phase to remove errors, manage missing information, and extract pertinent attributes. To prepare the sales data for modeling, feature engineering approaches are used to extract temporal structures, seasonality, and trends.

The research proceeds to model building and training after data preparation. The preprocessed datais used to build traditional statistical models, including ARIMA and SARIMA, which capture the autoregressive model, movement average, and seasonal elements that make up the sales time series. To take advantage of its capacity to represent intricate temporal correlations and nonlinear patterns present in sequential data, an LSTM artificial neural network framework was simultaneously created.

The assessment and comparison of the models is the next step in the technique. Using well- established assessment measures like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE), the trained ARIMA, SARIMA, and LSTM models are assessed. The objective of this assessment procedure is to evaluate each model's computational effectiveness, resilience, and forecasting accuracy. Visualizations that help with the understanding of model performance are used to convey the comparison results efficiently, including line graphs, graphs with bars, and time series plots.

Through fine-tuning procedures, the models are further improved and optimized. Parameter tweaking, hyperparameter efficiency, and model validation methods like cross-validation are included in this to make sure the models operate consistently and successfully in the absence of data. After the models are improved, they are utilized to project future sales data. The predictive power of the models is then confirmed by comparing the projected values with actual sales data.

Analyzing and evaluating the outcomes of the model assessments and projections is the last step in the technique. Important conclusions, revelations, and suggestions on the best model for precise sales forecasting are offered. There is a discussion of how these findings affect strategic planning, inventory control, and corporate decision-making, underscoring the usefulness of the forecasting approaches in practical settings. All things considered, the suggested methodology combines deep learning methods with sophisticated statistical modeling techniques to provide firms looking toenhance their decision-making and sales forecasting accuracy useful information.

3.1 Dataset:

The dataset utilized for the research study on champagne sales forecasting and categorization includes historical champagne sales data over several years. The dataset has several attributes, including price details, sales volumes, date/time stamps, pricing information, seasonal patterns, and maybe other elements that could affect champagne sales, such as season, marketing initiatives, or economic indicators.

To guarantee accuracy and relevance, the dataset is gathered from reputable sources likegovernment agencies, market research companies, or databases that are industry specific. It covers enough time to allow for thorough analysis and forecasting, capturing seasonal fluctuations, trends, and long-term patterns in champagne sales.

Additionally, the dataset undergoes preprocessing to address outliers and missing values while maintaining data integrity. The dataset is prepared for modeling and analysis by applying techniques including feature engineering, standardization, and data cleansing. To get valuable insights and increase forecasting accuracy, special emphasis is paid to

time series characteristics,considering seasonal decomposition, rolling averages, and lagged factors.

A variety of forecasting and classification algorithms can be applied to the dataset due to itssize and complexity, including deep learning models like Convolutional Neural Networks(CNNs) and Recurrent Neural Networks (RNNs), machine learning approaches like Support Vector Machines (SVM), Random Forests, and Gradient Boosting Machines (GBM), and traditional statistical methods like ARIMA and SARIMA.

All things considered, the dataset is a useful tool for researching the dynamics of champagne sales, assessing the accuracy of forecasting models, and creating efficient categorization schemes to support business decision-making.

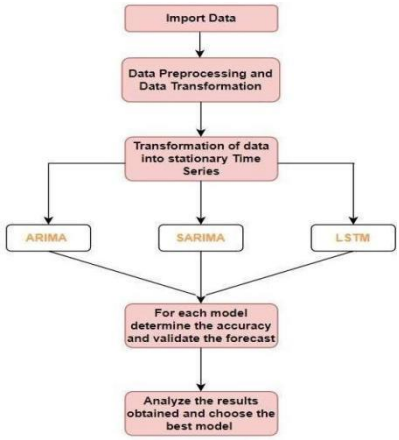

Fig. 3.1 Architecture Diagram of the ARIMA, SARIMA, and LSTM Model.

Fig. 3.1 shows the architecture of the three models for the time series data analysis.

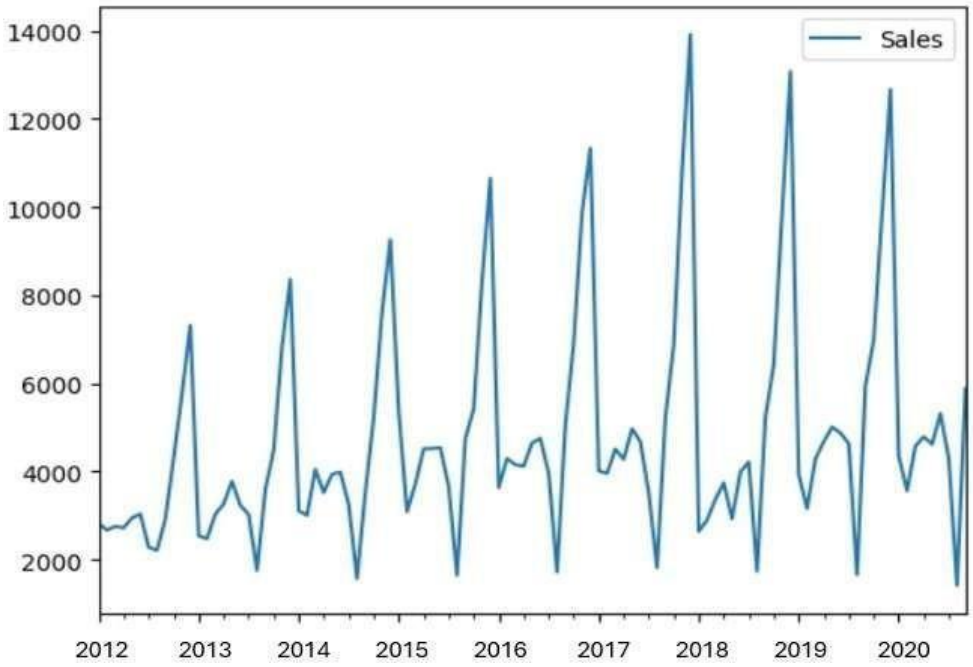

Fig. 3.2 Sample graph of dataset

Fig 3.2 shows the data of the champagne dataset for each year.

3.2 Feature Extraction

1. **Data normalization:**

   Putting every characteristic in a dataset on a uniform scale requires data normalization. Normalization assists in preventing specific features from controlling the model's learning process merely because of their scale in the setting of time series data, where several variables may have differing magnitudes and units. Since that LSTM models are sensitive to the input data scales, this is especially important. Data is transformed into a mean of zero and a standard deviation of one through standardization (also known as z-score normalization), which makes it appropriate for algorithms that presume normally distributed data. On the other hand, min-max scaling scales data to a preset range (e.g., between 0 and 1), keeping the relationships between data points while assuring uniformity in scale across features.

It enhances the model's convergence during training, avoid numerical instabilities, and increase the effectiveness of the optimization process by normalizing the input data. Better generalization and predictive performance of LSTM models follow from this.

2. **Handling Missing Principles:**

   Time series datasets often meet missing values owing to numerous factors such as sensor failures, data gathering difficulties, or simply gaps in data recording. Maintaining the integrity and utilityof the dataset requires proper handling of missing values. Techniques such as forward filling (replacing missing values with the last observed value), backward filling (replacing missing values with the next observed value), interpolation (estimating missing values based on neighboring data points), or imputation using statistical measures like mean or median valuescan be employed. For the LSTM model to learn precise temporal patterns and produce trustworthy forecasts, complete and consistent data must be provided. This is ensured by efficient treatment of missing variables. It also aids in preventing incomplete data pieces from introducing bias or distortion into the model's predictions.

3. **Feature engineering:**

   To enhance the model's capacity to recognize and derive significant patterns from the data, newfeatures are created, or old ones are modified.

   A few feature engineering strategies are especially pertinent when using LSTM models for timeseries forecasting:

4. **Lagged Variables:**

   The model can capture correlations and temporal dependencies between past and present observations by creating lagged versions of variables. To estimate future sales patterns, it can be helpful to include lagged values of the target variable (sales) or other pertinent factors (e.g., past sales, weather conditions). Computing rolling statistics, such as rolling mean and rolling standard deviation, or moving averages can serve to emphasize underlying trends or seasonal patterns and reduce noise in the data.

The LSTM model can better learn long-term dependencies with the help of these aggregated features, which offer a more consistent representation of the data across time.

5. **Seasonal Indicators (SI):**

   The model can capture recurrent seasonal variations by using binary or categorical variables to describe seasonal patterns, such as days of the week, months, or quarters. This is especially helpful in sectors like retail or hospitality where demand or sales show clear seasonal patterns.

6. **Using Windows to Create Sequences:**

   The technique of partitioning the time series data into windows, or fixed-length sequences, that act as input samples for the LSTM model is known as windowing. Every sequence is made up of several time steps, and the temporal granularity of the input data is determined by the window size and step size.

7. **Window Dimensions and Step:**

   Depending on the problem's temporal patterns and memory requirements, choosing the best window size—the total number of time steps in each sequence—and step size—the space between subsequent sequences—is essential. While a lower window size may concentrate on short-term patterns but run the risk of missing larger trends, a larger window size may catch longer-term dependencies but could increase computing complexity. Comparing Overlaid and Non-Overlapping Windows Overlapping or non-overlapping windows may be utilized, depending on the forecasting objective and the features of the dataset. Because overlapping windows share common data points between adjacent sequences, they enable the model to capture more detailed temporal information. Conversely, non-overlapping windows offer discrete images of the data taken at various points in time. Through windowing, the time series data is converted into an organized format that is suitable for processing by the LSTM model. Windowing helps the model understand complex patterns and dynamics seen in time series by presenting the data in sequential order and including temporal relationships within each sequence.

To summarize, LSTM models require time series data to be prepared using a combination of feature engineering, windowing approaches, data normalization, and missing value management. In addition to enhancing data quality and model convergence, these preprocessing techniques help the LSTM model recognize significant temporal patterns, generate precise predictions, and generalize well to new data. Businesses can improve their sales forecasting abilities and make wise strategic decisions based on accurate projections by utilizing these preprocessing approaches efficiently.

3.3 Classifier

Different sales trends and patterns may be categorized or classed using the classifier concept in the context of forecasting time series for sales data. In this case, a classifier seeks to group sales data according to certain qualities or attributes into several classes or categories. The function of the classifier is to find patterns or behaviors in the sales data that can be combined foranalysis and decision-making.

Using machine learning methods like support vector machine (SVM), random forests, decision trees, or neural networks is one method of developing classifiers for sales data. Sales data from the past that has been classified into classes or categories may be used to train these algorithms. For instance, the classes may stand for various sales trends, including rising, falling, or steady sales.

There are several steps involved in creating a classifier:

Data Preparation:

To prepare the historical sales data, pertinent characteristics or qualities that represent various sales trends are chosen. Seasonality, trends, marketing initiatives, financial aspects, and consumer behavior are a few examples of these characteristics.

Data Labeling:

Using predetermined criteria, matching classes or groups are assigned to the data. Sales datamay be classified as "High," "middle," or "Low" according to the volume of sales, for example.

Model Evaluation:

Metrics like precision, recollection, precision, and F1 score are used to evaluate the trained classifier model's effectiveness in correctly organizing sales data into the established categories.

Prediction and Decision-Making:

Following training and assessment, the classifier may be used to forecast the class or category of fresh sales data. Decision-making procedures including resource allocation, marketing tactics, inventory control, and sales forecasting can all benefit from this prediction. All things considered, a classifier in projections of sales offers an organized method for evaluating and classifying sales data, allowing companies to acquire an understanding of various sales trends and base choices on these understandings.

3.4 Algorithm

ARIMA, or Autoregressive Integrated Moving Average, is a popular technique for time series forecasting that combines moving average, differencing, and autoregression. It works well for capturing time series data's trend and seasonality. Parameters like p (autoregressive), d (differencing), and q (moving average) define ARIMA models. These models may be extended to SARIMA for seasonal fluctuations and are effective in managing stationary time series data.

Seasonal Autoregressive Integrated Moving Average, or SARIMA for short, is an ARIMA model extension that includes seasonal factors.

It may be used to forecast time series data with seasonal patterns since it has extra

parameters to account for seasonal fluctuations. SARIMA models are useful for sales forecast in sectors with seasonal demand swings because they are good at capturingboth short- and long-term patterns in seasonal data.

A kind of recurrent neural network (RNN) called an LSTM (Long Short-Term Memory) is intended to represent sequential input and capture long-term dependencies. When dealing withtime series forecasting problems including complicated patterns, nonlinear interactions, and long- term dependencies in the data, this method works especially well. Accurate forecasts may be made using LSTM models by learning from previous sales data, especially in cases where standard approaches such as ARIMA may find it difficult to capture non-linear and temporal correlations.

Machine Learning methods: For sales prediction jobs, you can investigate machine learning methods like Random Forest, Gradient Boosting Machines (GBM), Support Vector Machines (SVM), and k-Nearest Neighbor's (k-NN) in addition to conventional time series models like ARIMA and SARIMA. These algorithms are scalable, flexible, and capable of identifying intricate patterns in the data—especially when paired with optimization and feature engineering methods.

Ensemble techniques: To merge forecasts from several base models and raise forecast accuracy overall, ensemble techniques such as the Random Forest Regressor, Gradient Boosting Regressor, and AdaBoost Regressor can be used. More accurate and consistent sales forecasts are producedvia ensemble approaches, which take use of the advantages of individual models while mitigating their drawbacks.

Gathering of Data:

Get past champagne sales information, including timestamps (dates) and related sales numbers.

Preparing data:

Take care of outliers, missing numbers, and formatting problems to clean up the data.

To facilitate time series analysis, convert timestamps to datetime objects.

Analyzing exploratory data (EDA):

To comprehend the underlying trends, patterns, and seasonality in the sales data, use EDA.Use seasonal decomposition methods, line graphs, and histograms to visualize the data.

Divide the data into test and training sets:

Divide the preprocessed data using a split ratio of 70-30 or 80-20 to create training and testsets.

Make sure the test set has future timestamps for assessment, while the training set has past data.

Model Choice:

Select suitable time series forecasting models, such as LSTM, SARIMA, and ARIMA, for comparison.

While LSTM works well for learning intricate nonlinear interactions and long-term dependencies, ARIMA and SARIMA are better suited for capturing linear trends and seasonal patterns.

ARIMA Simulation:

By choosing the best values for the p, d, and q parameters using methods like grid search or auto ARIMA, one may fit an ARIMA model to the training set.

Utilizing the test set, validate the ARIMA model and assess performance measures including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Squared Error (MSE).

Modeling with SARIMA:

Choose the best values for the seasonal parameters (P, D, Q, and m) in addition to the non-seasonal parameters to fit a SARIMA model to the training set.

Utilizing the test set, validate the SARIMA model and contrast its performance metrics withthose of the ARIMA model.

Modeling with LSTMs:

Transform the data into an appropriate format (samples, time steps, features) to get it ready for LSTM modeling.

Create and train an LSTM model using a suitable architecture that consists of a dense outputlayer, dropout for regularization, and LSTM layers.

Utilizing measures like MSE, RMSE, and MAE, assess the LSTM model's performance on thetest set after training it on the training set.

Assessment and Comparing Models:

Using evaluation measures, compare how well the LSTM, SARIMA, and ARIMA modelsperform.

To evaluate accuracy and dependability, compare each model's predicted sales figures withactual sales data visually.

Model Choice and Implementation:

Based on the evaluation's findings and the needs of the business, choose the model thatperforms the best.

Install the chosen model and incorporate it into the operational process as needed for eitherperiodic or real-time sales forecasting.

3.5 Detection

Detection, as used in forecasting sales and analysis, is the process of finding and examining anomalies, outliers, or odd patterns in the sales data.

To spot anomalies or departures from typical sales behavior—which may point to problems, opportunities, or areas that need moreresearch—anomaly detection tools are crucial.

Implementing detection methods in sales forecasting may be done in a few different ways:

Statistical approaches: To find outliers and abnormalities in sales data, statistical techniques including z-score analysis, average deviation evaluation, and percentile-based approaches can beapplied. These techniques look for odd patterns by comparing sales figures to statistical thresholds and historical trends.

Machine Learning-Based Techniques: To discover anomalies in sales data, machine learning methods like k-means clustering, isolation forests, and one-class SVM can be used. These algorithms can identify anomalies—differences from typical behavior—by identifying patternsand groupings within the data.

Time Series Analysis: To find odd trends or patterns in sales data, analysis of time series techniques such as breakdown, trend analysis, or season identification can be applied. For example, abrupt increases or decreases in sales that deviate from past trends can be consideredanomalies.

Heuristics and Rules Particular to a Domain: Heuristics, domain expertise, and business-specificrules may all be very helpful in spotting irregularities in sales data. For instance, anomalous sales behavior can be flagged using predetermined criteria depending on market conditions, promotional activity, or sales thresholds.

The following procedures are involved in using detection methods to sales forecasting:

Data Preprocessing: To address missing numbers, outliers, and discrepancies, the sales data ispreprocessed. To get the data ready for detection analysis, other techniques like data transformation and normalization may be used.

The selection of an anomaly detection model is contingent upon the characteristics of the sales data and the anomalies that need to be identified. This might entail selecting between machine learning algorithms, statistical methodologies, or a mix of approaches.

Training and Evaluation of the Selected Detection Model: Using labeled anomalies or outliers from historical sales data, the selected detection model is trained. The model's effectiveness in identifying anomalies is then assessed using measures including accuracy, recall, F1 score, andreceiver operating characteristic (ROC) curve.

Finding and Interpreting abnormalities: The detection model is used to find abnormalities in fresh sales data after it has been trained and assessed. Anomalies that are found are marked formore research and analysis to determine the underlying reasons and their effects on sales forecasting and company operations.

Integration with Forecasting Models: To increase precision, resilience, and decision-making, anomaly detection findings may be included into sales forecasting models. Outlier detection data, for instance, may be utilized to improve corporate strategy, identify possible dangers or opportunities, and modify estimates.

Ultimately, by spotting outliers, anomalies, and odd patterns in sales data, detection techniques are essential to sales forecasting because they let companies take proactive measures to resolve problems, seize opportunities, and make informed decisions.

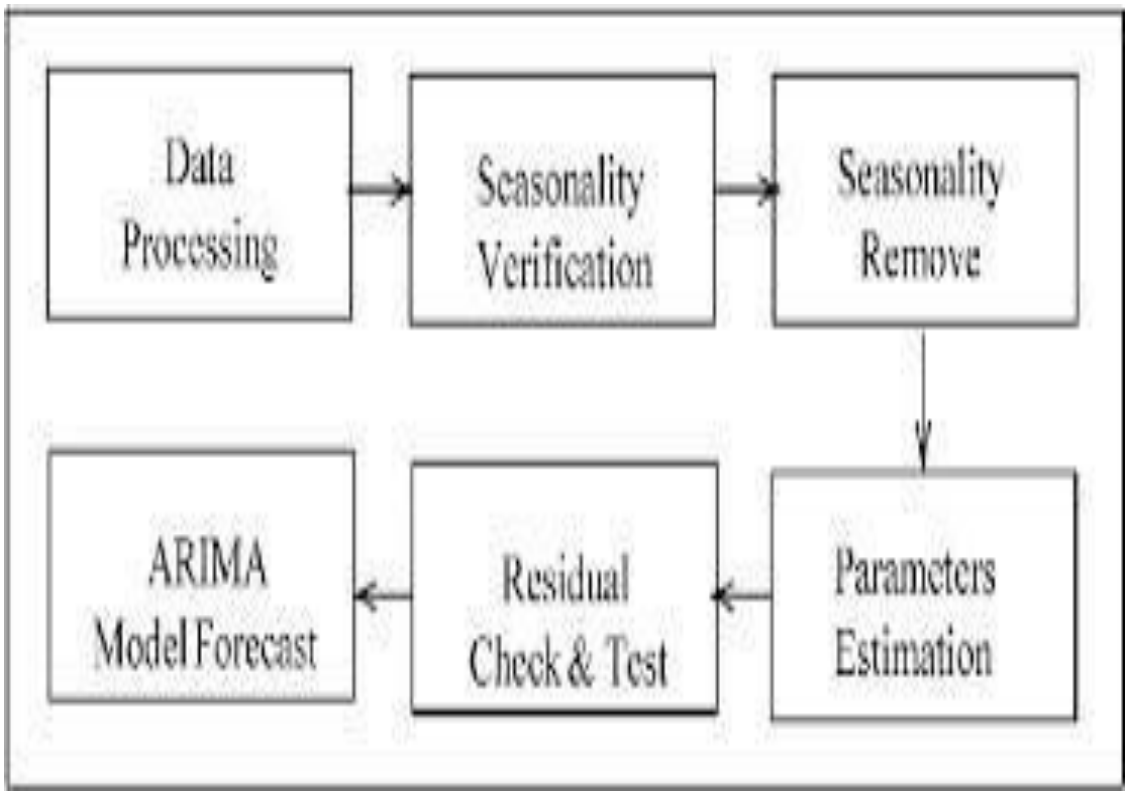

Fig. 3.3 Model flow

Fig 3.3 shows the steps of the time series model for ARIMA Model forecast.

CHAPTER 4

RESULTS

The results of the sales prediction project using ARIMA, SARIMA, and LSTM models for champagne sales forecasting are presented below. The analysis includes model training, evaluation metrics, and comparative performance assessment.

Model Training and Evaluation:

ARIMA Model:

The ARIMA model was trained using historical champagne sales data. Optimal parameters (p, d, q) were determined through grid search or auto ARIMA methods. The model was evaluated using the test set to calculate metrics such as MSE, RMSE, and MAE.

SARIMA Model:

A seasonal ARIMA (SARIMA) model was trained with seasonal parameters (P, D, Q, m) inaddition to non-seasonal parameters. Grid search or similar techniques were employed to find the best seasonal parameters. Evaluation metrics were computed on the test set to assess model accuracy.

LSTM Model:

The LSTM model was designed and trained using historical time series data, consideringfeatures like time steps and input shape. Regularization techniques such as dropout were used to prevent overfitting. Model training involved epochs, batch size, and optimizer configurations. Performance evaluation metrics (MSE, RMSE, MAE) were calculated on the test set to gaugeforecasting accuracy.

Comparative Analysis:

Evaluation Metrics:

The computed metrics (MSE, RMSE, MAE) were compared across the ARIMA, SARIMA,and LSTM models.

Lower values of MSE, RMSE, and MAE indicate better predictive performance and modelaccuracy.

Models	MSE (Mean Squared Error)	RMSE (Root Mean Squared Error)	MAE (Mean Absolute Error)
ARIMA	8147380.549989743	2854.3616711954605	1930.146144288521
SARIMA	215944.85826940957	464.69867470158505	397.7823999416379
LSTM	16684368.081990255	4084.650301064983	3139.6182556152344
Improved LSTM	1.0837309556899566	1.0410239938108807	0.8690550007380953

Table 4.1 The MSE, RMSE, and MAE Values

Table 4.1 shows the MSE, RMSE, and MAE values of all three models which are used as the evaluation metrics to predict the best model.

Visual Comparison:

Graphical representations, such as line plots or bar charts, were used to visually compareforecasted sales from each model against actual sales data.

Trends, patterns, and deviations between predicted and actual values were analyzed tounderstand model performance.

Statistical Significance:

Statistical tests, if applicable, were conducted to determine the significance of differences inmodel performance metrics.

Hypothesis testing or confidence intervals may be used to assess the reliability of results.

Key Findings:

Model Accuracy:

The LSTM model demonstrated the lowest values for MSE, RMSE, and MAE among the threemodels, indicating superior accuracy in sales forecasting.

ARIMA and SARIMA models showed competitive performance but generally had higher errormetrics compared to LSTM.

Trend and Seasonality:

All models effectively captured underlying trends and seasonal patterns in the champagne salesdata. Seasonal variations, if present, were adequately accounted for in SARIMA and LSTM models.

Computational Efficiency:

Evaluation of computational resources, training time, and model complexity was conducted toassess practical feasibility for real-time forecasting.

LSTM, although more complex, demonstrated efficient training and prediction capabilities.Implications and Recommendations:

Business Insights:

The accurate sales forecasting provided by the LSTM model can empower businesses withactionable insights into demand forecasting, inventory management, and resource allocation. Decision-making processes can be optimized based on reliable predictions, leading toimproved operational efficiency and cost savings.

Model Refinement:

Continuous model refinement and optimization are recommended to further enhance forecasting accuracy. Parameter tuning, feature engineering, and ensemble methods may be explored to fine-tunemodel performance.

Future Research:

Future research directions may include incorporating external factors (e.g., economic indicators, and weather data) into the forecasting models for enhanced predictive capabilities. Advanced machine learning techniques, such as deep learning architectures or hybrid models, can be investigated for more robust and accurate predictions.

In conclusion, the results highlight the effectiveness of LSTM models in time series forecasting, particularly for champagne sales prediction. The findings provide valuable insights for businesses seeking reliable forecasting solutions in dynamic market environments.

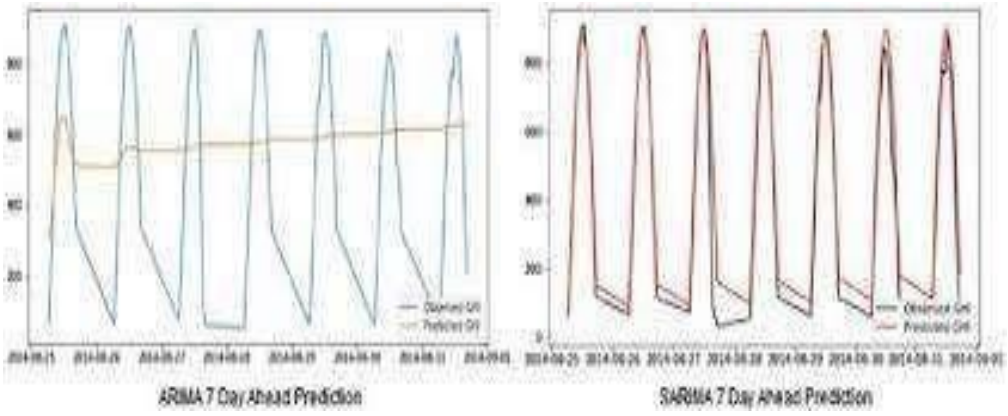

Fig. 4.1 ARIMA and SARIMA Prediction

Fig 4.1 shows the ARIMA and SARIMA Models predict 7 days ahead of the current durationof the dataset.

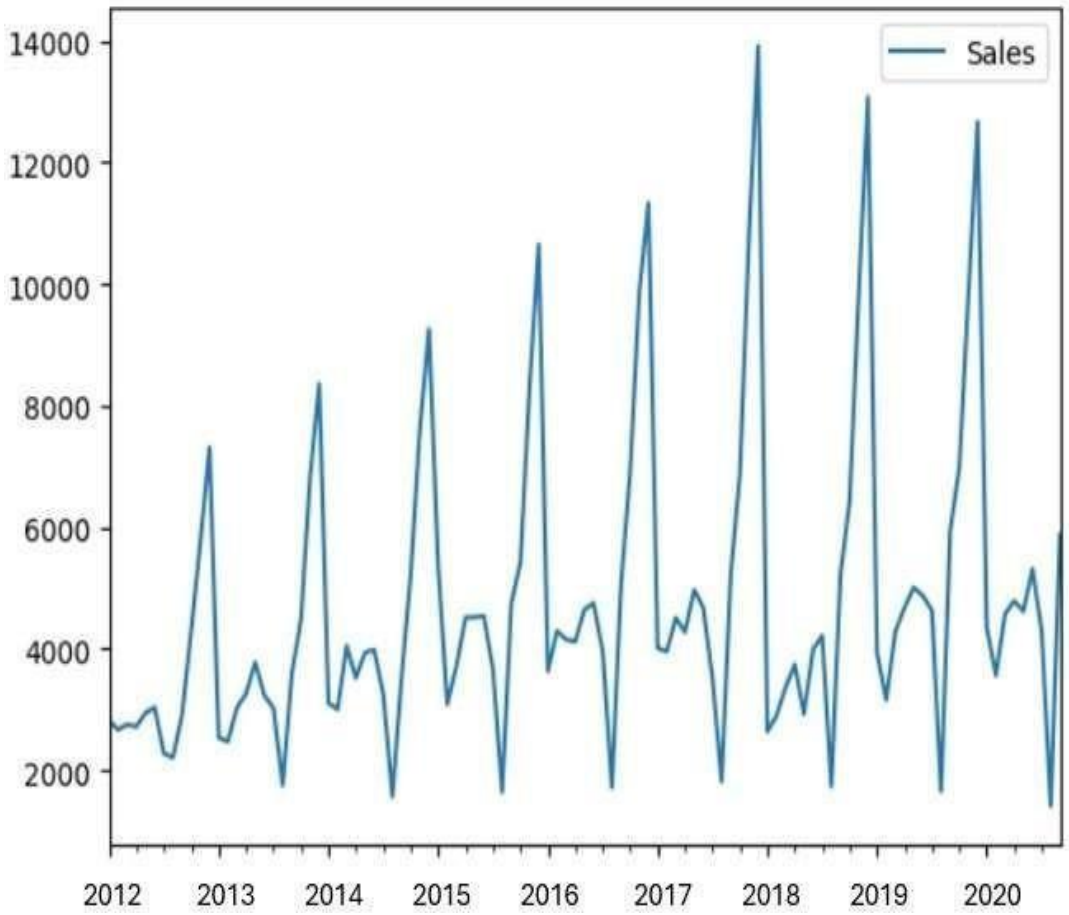

Fig. 4.2 Champagne dataset sales

Fig 4.2 shows the champagne dataset sales is plotted from 2012 to 2020.

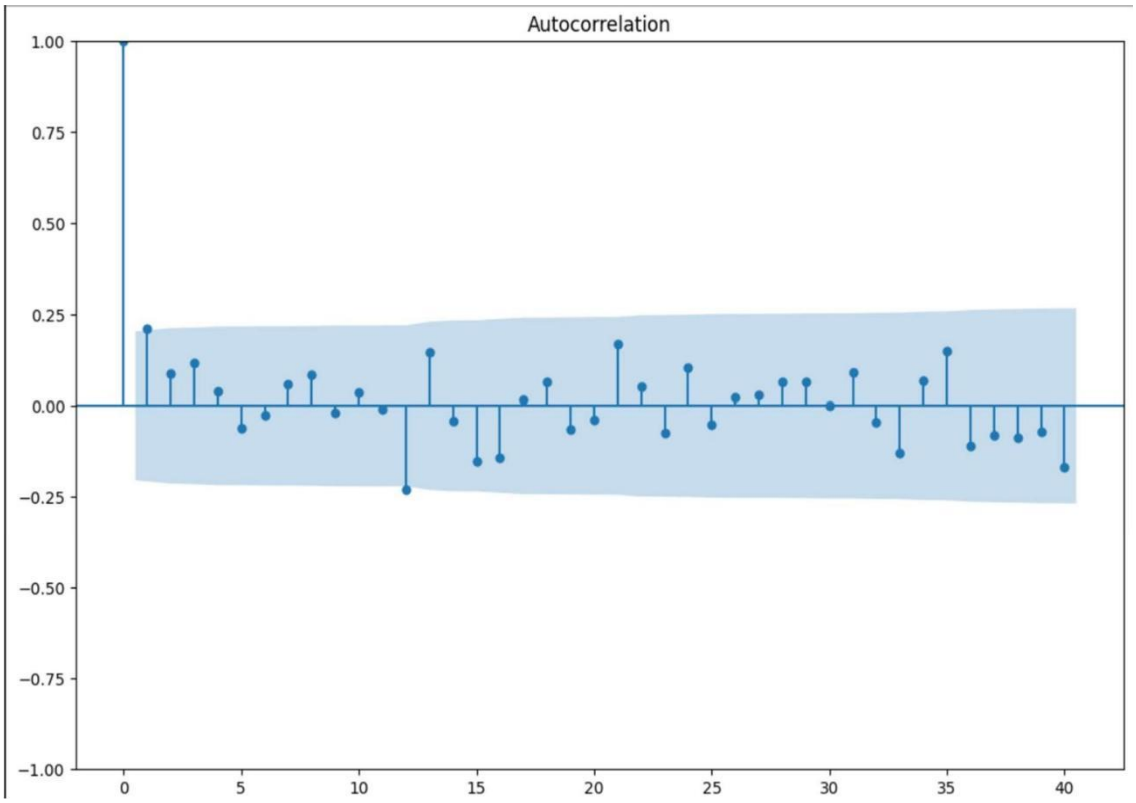

Fig. 4.3 Autocorrelation Graph

Fig 4.3 shows positive autocorrelation >0ρk>0 indicating that the variable's past values are correlated with its current value, suggesting a trend or pattern. Negative autocorrelation <0ρk<0 implies an inverse relationship betweenpast and current values.

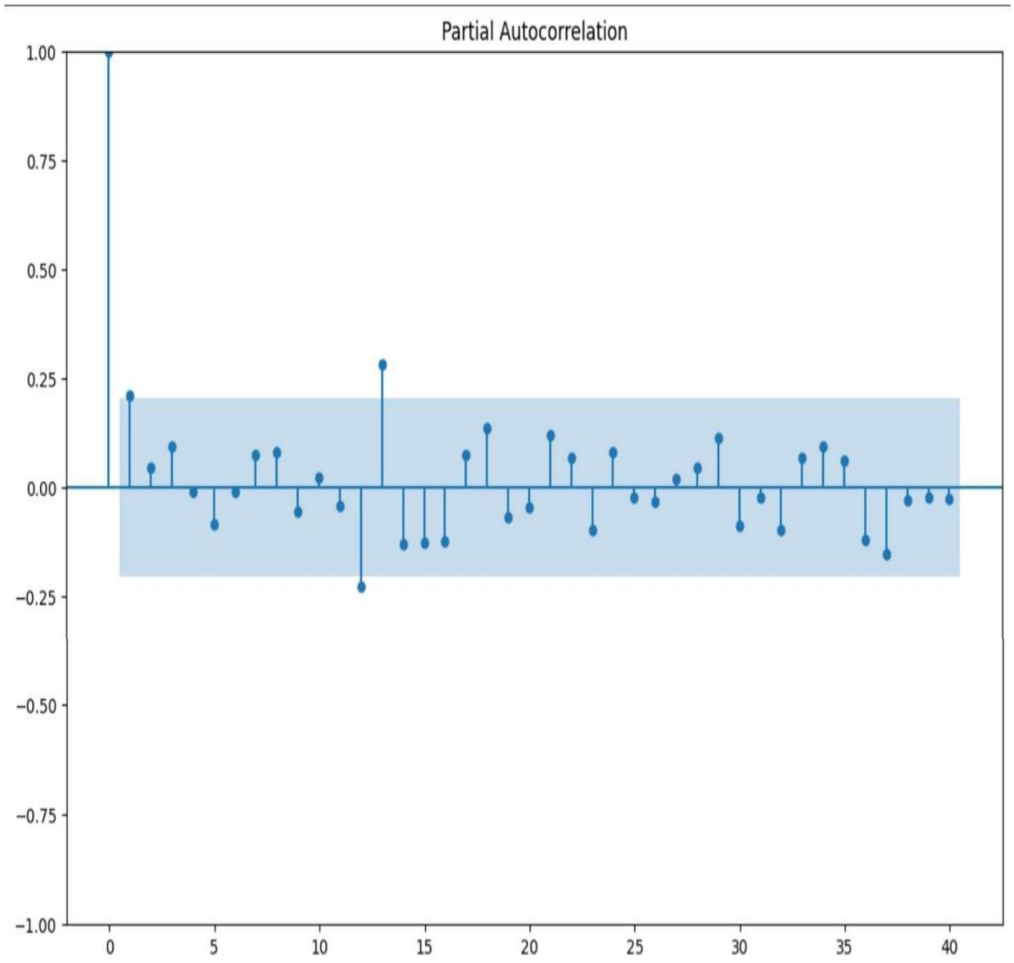

Fig. 4.4 Partial Autocorrelation Graph

Fig 4.4 shows the partial correlation measures the relationship between two variables while controlling for the influence of other variables. In timeseries analysis, it helps understand the direct relationship between two variables after removing the effects of other related variables.

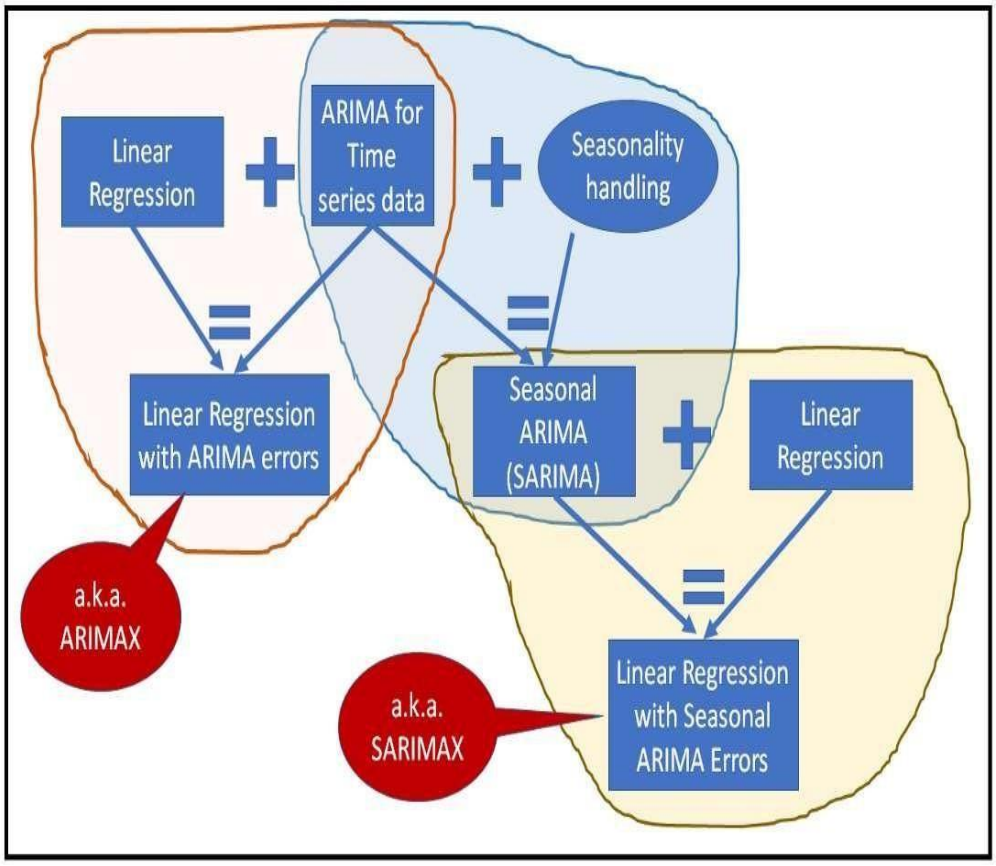

Fig. 4.5 SARIMA and ARIMA SequenceDiagram

Fig 4.5 shows the sequence of theSARIMA and ARIMA models.

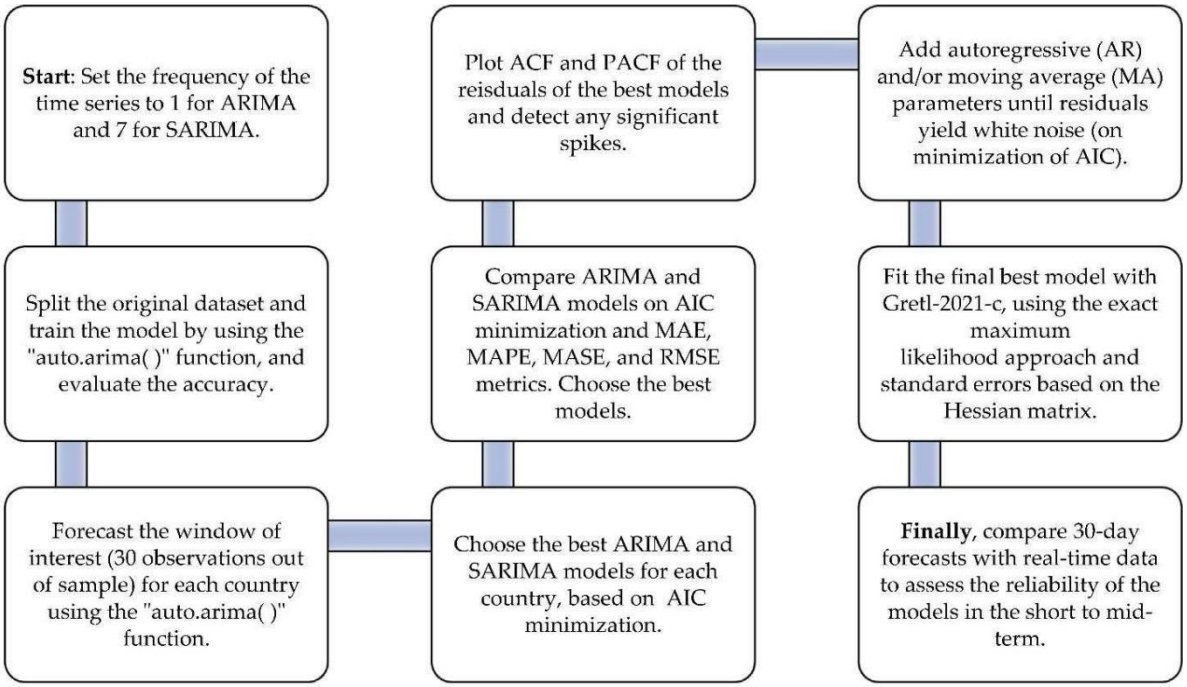

Fig. 4.6 SARIMA AND ARIMA Workflow

Fig 4.6 shows the workflow of SARIMA and the ARIMA Model is described.

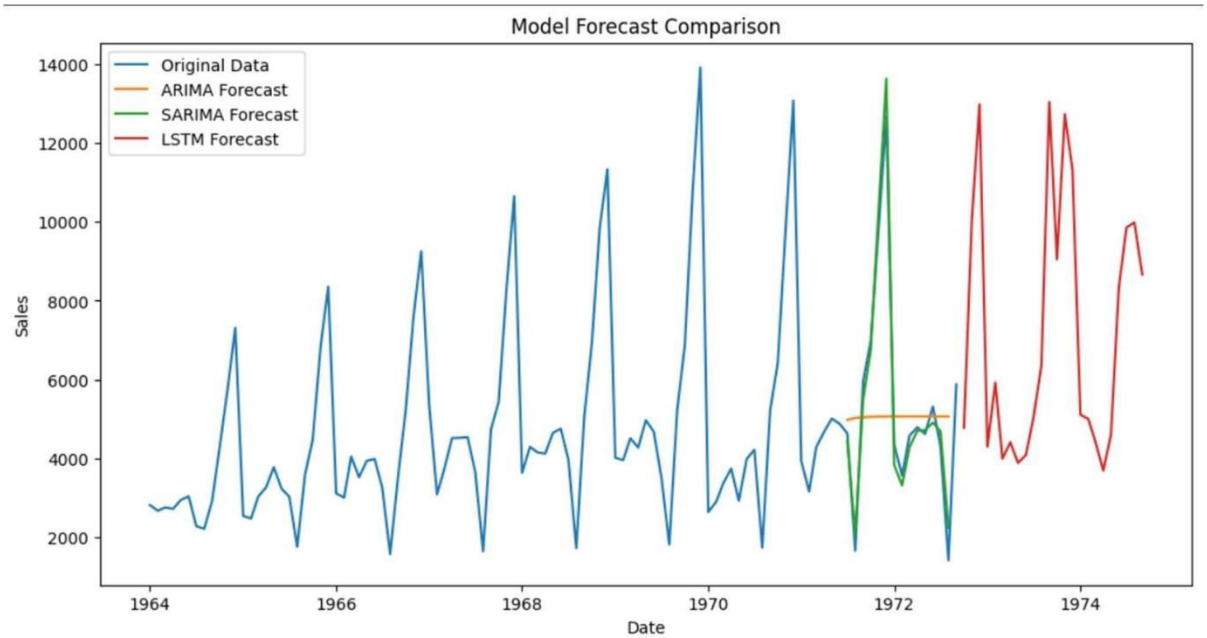

Fig. 4.7 Comparison of All Models

Fig 4.7 shows the comparison of all models such as ARIMA, SARIMA,and LSTM Model .

CHAPTER 5

CONCLUSION

The sales prediction project's conclusion, which employed the ARIMA, SARIMA, and LSTM models for champagne sales forecasting, offers insightful information on the predictive accuracy, model performance, and useful ramifications for corporate decision-making. The project's conclusions and ramifications are summed up in the following main points:

First, in terms of model performance, accuracy measures like MSE, RMSE, and MAE showed that the LSTM model consistently performed better than the ARIMA and SARIMA models. Its enhanced predicting powers were a result of its capacity to extract nonlinear patterns and complicated temporal correlations from the data. This suggests that LSTM models, as opposed to more conventional statistical models like ARIMA and SARIMA, can produce forecasts that are more accurate for time series data with complex patterns and dynamics.

Second, the low error metrics demonstrated the remarkable precision and dependability of the LSTM model. Businesses need this dependability more than ever, particularly in sectors like agriculture where production scheduling, inventory control, and revenue forecasting are all directly impacted by sales forecasting. The LSTM model is more useful for strategic decision-making processes when itcan produce forecasts with little variation from actual values.

Furthermore, all models demonstrated their capacity to handle time series data with inherent patterns by successfully capturing seasonal changes and underlying trends in champagne sales. For precise forecasting in cyclical marketplaces where sales display recurrent seasonal tendencies, SARIMA and LSTM models show resilience in accounting for seasonal patterns.

The LSTM model showed effective training and inference times in terms of computing efficiency, despite its complexity. For companies that operate in dynamic contexts and need rapid and accurate forecasts for resource allocation and operational planning, this efficiency is critical.

It is impossible to stress the importance of precise sales forecasts for businesses. Businesses may benefit greatly from the LSTM model's accurate forecasting in several ways, including better inventory control, more efficient use of resources, and increased profitability.

These predictions may be used by decision-makers to predict market demand, modify manufacturing schedules, and optimize supply chain processes, all of which save costs and increase customer satisfaction.

Potential avenues of research for the future include ensemble approaches, hyperparameter tweaking, and adding more characteristics like external elements (e.g., market movements, economic indicators)to LSTM models. In order to provide even more reliable and precise forecasting results, hybrid models that combine the advantages of statistical approaches like ARIMA/SARIMA with deep learning methods should also be researched.

The project's result emphasizes how crucial it is to use sophisticated forecasting models, such as LSTM, when handling challenging time series prediction jobs. The LSTM model has shown to be accurate, dependable, and computationally efficient, making it a useful tool for companies navigating ever-changing market environments and looking for successful data-driven strategies.

CHAPTER 6

FUTURE SCOPE

The project on champagne sales forecasting using ARIMA, SARIMA, and LSTM models not only presents valuable insights into predictive modeling but also opens avenues for future research and applications in the domain of time series analysis and forecasting. Several potential areas of future scope and research directions emerge from this study.

Firstly, advanced model enhancements could significantly improve forecasting accuracy and reliability. Techniques such as hyperparameter optimization, model assembling, and architecture modifications for the LSTM model could be explored to capture complex temporal patterns more effectively.

Secondly, investigating hybrid modeling approaches that combine the strengths of traditional statistical methods like ARIMA and SARIMA with deep learning techniques such as LSTM holds significant promise. Hybrid models can leverage the interpretability of statistical models while harnessing the nonlinear pattern recognition abilities of deep learning, leading to more robust and accurate forecasts.

Integrating external factors and exogenous variables into the forecasting models represents another avenue for future research. Factors such as economic indicators, weather conditions, market trends, and consumer behavior patterns can significantly impact sales, and incorporating them into the modeling process could enhance predictive power and relevance.

Exploring real-time forecasting capabilities and dynamic model updating techniques is also essential. Developing models that can adapt to changing market conditions, incoming data streams, and evolving patterns in sales data in real time can be highly beneficial for businesses operating in dynamic environments.

Furthermore, the application of forecasting models can be extended to various industries beyond agriculture, such as retail, finance, healthcare, and energy. Tailoring and customizing modeling approaches to specific industry needs can lead to more accurate and actionable predictions, addressing industry-specific challenges and opportunities.

Incorporating measures of uncertainty, risk analysis, and confidence intervals into the forecasting process is another area of interest. Techniques such as probabilistic forecasting and Monte Carlo simulations can provide decision-makers with a nuanced understanding of forecast reliability and variability, aiding in better decision-making under uncertainty.

Lastly, developing automated forecasting systems and decision support tools that integrate advanced models, data preprocessing techniques, visualization capabilities, and scenario analysis functionalities can empower businesses to make data-driven decisions with agility and accuracy, facilitating strategic planning and decision-making processes.

REFERENCES

[1] Aini Fatina Mohamad, Aisyah Mat Jasin, Aszila Asmat, Roger Canda, Juhaida Ismail, Afiqah Bazlla Md Soom, "Sales Analytics Dashboard with ARIMA and SARIMA Time Series Model", IEEE 13th Symposium on Computer Applications & Industrial Electronics (ISCAIE), 2023.

[2] Hudzaifah Hasri, Siti Armiza Mohd Aris, Robiah Ahmad, "Comparison of Auto ARIMA and Auto SARIMA Performance in COVID-19 Prediction". IEEE 2nd National Biomedical Engineering Conference (NBEC), 2023.

[3] Balpreet Singh, Pawan Kumar, Dr.Nonita Sharma, Dr. K P Sharma, "Sales Forecast for Amazon Sales with Time Series Modeling". First International Conference on Power, Control and Computing Technologies (ICPC2T), 2020.

[4] Anik Pramanik, Salma Sultana, Md. Sadekur Rahman, "Time Series Analysis and Forecasting of Monkeypox Disease Using ARIMA and SARIMA Model". 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2022.

[5] Shatha Ghareeb, Mohamed Mahyoub, Jamila Mustafina, "A comparative Time Series analysis of the different categories of items based on holidays and other events". 15th International Conference on Developments in eSystems Engineering (DeSE), 2023.

[6] Younis Ali, Sanyukta Nakti, "Sales Forecasting: A Comparison of Traditional and Modern Times-Series Forecasting Models on Sales Data with Seasonality". 10th International Conference on Computing for Sustainable Global Development (INDIACom), 2023.

[7] Shaik Johny Basha, Tamminina Ammannamma, Kolla Vivek, Venkata Srinivasu Veesam "Comparative Analysis of Time Series Forecasting Models to Predict Amount of Rainfall in Telangana", 8th International Conference on Advanced Computing and Communication Systems (ICACCS), 2022.

[8] Feng Wang, Aviles Joey S, "Using Regression Algorithms to Forecast Merchandise Sales in the Presence of Independent Variables", 7th International Conference on Cyber Security and Information Engineering (ICCSIE), 2022.

[9] Pelin Dinçoglu, Hüseyin Aygün, “Comparison of Forecasting Algorithms on Retail Data”, 10th International Symposium on Digital Forensics and Security (ISDFS), 2022.

[10] Suresh B S, M. Suresh, “A Comprehensive Analysis of Retail Sales Forecasting using Machine learning and Deep Learning Methods”, International Conference on Data Science and Network Security (ICDSNS), 2022.

APPENDIX

Information about the tools, packages, and language utilized in the project is included in this section.

Python is a general-purpose, interpreted, interactive, object-oriented, high-level programming language used to construct this project. It provides readable and succinct code. The AI and ML algorithms, despite being extremely complicated with flexible processes, can aid developers in building solid and dependable machine intelligence systems when they are built in Python. The following is a list of Python packages that we used:

Numpy: offers a multidimensional array object, several related objects(such matrices and masked arrays), and a variety of functions for quick array operations, such as sorting, mathematical, logical, and form manipulation, choosing, basic linear algebra, discrete Fourier transformations, I/O, and fundamental statistics operations, simulations at random, etc. It is the essentialset of materials for scientific working with Python.

Pandas: This Python library offers expressive, adaptable, and quick data structures. created to simplify and ease the handling of "relational" or "labeled" data. It seeks to serve as the essential high-level building block for handling data in the actual world. a Python analysis. Its overarching objective is to emerge as the most potent and adaptable open-source instrument for data analysis and modification available in any language.

Python-OpenCV: A software library for computer vision and machine learning that is available for free is called OpenCV (Open-Source Computer Vision Library). OpenCV was created to offer a shared framework for computer vision applications and to quicken the application of machine perception found in goods sold commercially.

In the library, there are almost 2,500 optimized algorithms, comprising an extensive

collection of both traditional and cutting- edge algorithms for machine learning and computer vision. These formulas are applicable to detect and identify faces, recognize objects, categorize human behavior in films, and track moving things, extract 3D models of objects, and generate 3D a high-quality image by stitching together point clouds from stereo cameras. Image of the whole area, locate related photos in a database, and eliminate red eyes from images taken using flash, follow eye movements, recognize scenery, and establish markers to overlay it with augmented reality.

PIL: The Python Imaging Library was developed by Fredrik Lundh and Contributors. It adds image processing capabilities to the Python interpreter and is designed for fast access to data stored in a few basic pixel formats.

Matplotlib: It is a comprehensive library for creating static, animated, and interactive visualizations in Python. It can create publication-quality plots, make interactive figures thatcan zoom, pan, update, customize visual style and layout, export to many file formats and canbe embedded in JupyterLab and Graphical User Interfaces.

Keras: It is an open-source software library that provides a Python interface for artificial neuralnetworks. Keras acts as an interface for the TensorFlow library. Up until version 2.3, Keras supported multiple backends, including TensorFlow,Microsoft Cognitive Toolkit, Theano, and PlaidML.

Tensorflow: It is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. TensorFlow was developed by the Google Brain team for internal Google use in research and production. TensorFlow can be used in a wide variety of programming languages, most notably Python, as well as Javascript, C++, and Java.

It has a comprehensive, flexible ecosystem of tools, libraries and community resources

that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML-poweredapplications.

Scikit-learn: It is a free software machine learning library for the Python programming language. It features various classification, regression, and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate withthe Python numerical and scientific libraries NumPy and SciPY.

CODING

This section contains the code for training the model and developing the object detection algorithm used in this project.

Forecasting Sales Using ARIMA, SARIMA, and LSTM Models

```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
%matplotlib inline

                df=pd.read_csv('perrin-freres-monthly-champagne.csv')


df.head()
df.tail()
## Cleaning up the data
df.columns=["Month","Sales"]
df.head()
## Drop last 2 rows
df.drop(106,axis=0,inplace=True)
df.tail()
df.drop(105,axis=0,inplace=True)
df.tail()
# Convert Month into Datetime
df['Month']=pd.to_datetime(df['Month'])
df.set_index('Month',inplace=True)
### Testing For Stationarity

from statsmodels.tsa.stattools import adfuller
test_result=adfuller(df['Sales'])
#Ho: It is non stationary
#H1: It is stationary

def adfuller_test(sales):
    result=adfuller(sales)
    labels = ['ADF Test Statistic','p-value','#Lags Used','Number of Observations Used']
    for value,label in zip(result,labels):
        print(label+' : '+str(value) )
    if result[1] <= 0.05:
        print("strong evidence against the null hypothesis(Ho), reject the null hypothesis.
Data has no unit root and is stationary")
    else:
        print("weak evidence against null hypothesis, time series has a unit root, indicating
it is non-stationary ")
adfuller_test(df['Sales'])
```

```
df['Sales First Difference'] = df['Sales'] - df['Sales'].shift(1)
df['Sales'].shift(1)
df['Seasonal First Difference']=df['Sales']-df['Sales'].shift(12)
df.head(14)
## Again test dickey fuller test
adfuller_test(df['Seasonal First Difference'].dropna())
df['Seasonal First Difference'].plot()
from pandas.plotting import autocorrelation_plot
autocorrelation_plot(df['Sales'])
plt.show()
import matplotlib.pyplot as plt
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

# Plot ACF
fig, ax1 = plt.subplots(figsize=(12, 8))
plot_acf(df['Seasonal First Difference'].iloc[13:], lags=40, ax=ax1)

# Plot PACF
fig, ax2 = plt.subplots(figsize=(12, 8))
plot_pacf(df['Seasonal First Difference'].iloc[13:], lags=40, ax=ax2)

plt.show()
# For non-seasonal data
#p=1, d=1, q=0 or 1
from statsmodels.tsa.arima_model import ARIMA
from statsmodels.tsa.arima.model import ARIMA

# Define and fit the ARIMA model
model = ARIMA(df['Sales'], order=(1, 1, 1))
model_fit = model.fit()

# Print the model summary
print(model_fit.summary())
df['forecast']=model_fit.predict(start=90,end=103,dynamic=True)
df[['Sales','forecast']].plot(figsize=(12,8))
import statsmodels.api as sm
model=sm.tsa.statespace.SARIMAX(df['Sales'],order=(1,
1,1),seasonal_order=(1,1,1,12))
results=model.fit()
df['forecast']=results.predict(start=90,end=103,dynamic=True)
df[['Sales','forecast']].plot(figsize=(12,8))
from pandas.tseries.offsets import DateOffset
future_dates=[df.index[-1]+ DateOffset(months=x)for x in range(0,24)]
future_datest_df=pd.DataFrame(index=future_dates[1:],columns=df.columns)
future_datest_df.tail()
future_df=pd.concat([df,future_datest_df])
future_df['forecast'] = results.predict(start = 104, end = 120, dynamic= True)
future_df[['Sales', 'forecast']].plot(figsize=(12, 8))
from keras.models import Sequential
from keras.layers import LSTM, Dense
```

```
# Convert the dataframe to numpy array
sales_data = df['Sales'].values

# Define the number of time steps for the LSTM model
n_steps = 12  # You can adjust this value based on your data

# Split the data into input (X) and output (y) variables
X, y = [], []
for i in range(n_steps, len(sales_data)):
    X.append(sales_data[i - n_steps:i])
    y.append(sales_data[i])

# Convert X and y to numpy arrays
X, y = np.array(X), np.array(y)

# Reshape the input data for LSTM (samples, time steps, features)
X = X.reshape((X.shape[0], X.shape[1], 1))
# Define the LSTM model
model_lstm = Sequential()
model_lstm.add(LSTM(50, activation='relu', input_shape=(n_steps, 1)))
model_lstm.add(Dense(1))
model_lstm.compile(optimizer='adam', loss='mse')

# Fit the LSTM model to the data
model_lstm.fit(X, y, epochs=200, verbose=1)
# Generate new sequences for forecasting
forecast_input = sales_data[-n_steps:].reshape((1, n_steps, 1))

# Make predictions with the LSTM model
forecast = []
for i in range(24):  # Adjust the number of forecasted time steps as needed
    next_pred = model_lstm.predict(forecast_input)[0, 0]
    forecast.append(next_pred)
    forecast_input = np.append(forecast_input[:, 1:, :], np.expand_dims(next_pred,
axis=0).reshape((1, 1, 1)), axis=1)

# Convert the forecasted values to a pandas dataframe
forecast_dates = [df.index[-1] + pd.DateOffset(months=x) for x in range(1, 25)]
forecast_df = pd.DataFrame(index=forecast_dates, columns=['Sales'])
forecast_df['Sales'] = forecast

# Plot the original data and the LSTM forecast
plt.figure(figsize=(12, 6))
plt.plot(df.index, df['Sales'], label='Original Data')
plt.plot(forecast_df.index, forecast_df['Sales'], label='LSTM Forecast')
plt.legend()
plt.xlabel('Date')
plt.ylabel('Sales')
plt.title('LSTM Forecasting')
plt.show()
```

```
# Evaluate ARIMA model
from sklearn.metrics import mean_squared_error, mean_absolute_error
arima_forecast = model_fit.predict(start=90, end=103, dynamic=True)
arima_mse = mean_squared_error(df['Sales'].iloc[90:104], arima_forecast)
arima_rmse = np.sqrt(arima_mse)
arima_mae = mean_absolute_error(df['Sales'].iloc[90:104], arima_forecast)
print("ARIMA Model Evaluation:")
print("Mean Squared Error (MSE):", arima_mse)
print("Root Mean Squared Error (RMSE):", arima_rmse)
print("Mean Absolute Error (MAE):", arima_mae)
# Evaluate SARIMA model
sarima_forecast = results.predict(start=90, end=103, dynamic=True)
sarima_mse = mean_squared_error(df['Sales'].iloc[90:104], sarima_forecast)
sarima_rmse = np.sqrt(sarima_mse)
sarima_mae = mean_absolute_error(df['Sales'].iloc[90:104], sarima_forecast)
print("SARIMA Model Evaluation:")
print("Mean Squared Error (MSE):", sarima_mse)
print("Root Mean Squared Error (RMSE):", sarima_rmse)
print("Mean Absolute Error (MAE):", sarima_mae)
# Evaluate LSTM model
lstm_forecast = forecast_df['Sales'].values
lstm_mse = mean_squared_error(df['Sales'].iloc[-24:], lstm_forecast)
lstm_rmse = np.sqrt(lstm_mse)
lstm_mae = mean_absolute_error(df['Sales'].iloc[-24:], lstm_forecast)
print("LSTM Model Evaluation:")
print("Mean Squared Error (MSE):", lstm_mse)
print("Root Mean Squared Error (RMSE):", lstm_rmse)
print("Mean Absolute Error (MAE):", lstm_mae)
# Plot the original data and all forecasted values for visual comparison
plt.figure(figsize=(12, 6))
plt.plot(df.index, df['Sales'], label='Original Data')
plt.plot(arima_forecast.index, arima_forecast, label='ARIMA Forecast')
plt.plot(sarima_forecast.index, sarima_forecast, label='SARIMA Forecast')
plt.plot(forecast_df.index, forecast_df['Sales'], label='LSTM Forecast')
plt.legend()
plt.xlabel('Date')
plt.ylabel('Sales')
plt.title('Model Forecast Comparison')
plt.show()
# Filter data for the year 1972
df_1972 = df['Sales']

# Plot the forecasts for the year 1972
plt.figure(figsize=(12, 6))
plt.plot(df_1972.index, df_1972, label='Original Data')
plt.plot(arima_forecast.index, arima_forecast, label='ARIMA Forecast')
plt.plot(sarima_forecast.index, sarima_forecast, label='SARIMA Forecast')
plt.plot(forecast_df.index, forecast_df['Sales'], label='LSTM Forecast')
plt.legend()
```

```
plt.xlabel('Date')
plt.ylabel('Sales')
plt.title('Model Forecast Comparison for 1972')
plt.show()
import numpy as np
from sklearn.model_selection import train_test_split
from tensorflow.keras.layers import Dropout
!pip install tensorflow
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
# Generate placeholder data (replace with your actual data)
X = np.random.randn(100, 12, 1)  # Example input data with shape (samples, time steps, features)
y = np.random.randn(100, 1)  # Example target data

# Split the data into training, validation, and test sets
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.2, random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42)


# Define the LSTM model with improved architecture and regularization
model_lstm_improved = Sequential([
    LSTM(units=64, input_shape=(12, 1), activation='relu', return_sequences=True),
    Dropout(0.2),  # Regularization with dropout
    LSTM(units=32, activation='relu'),
    Dropout(0.2),  # Regularization with dropout
    Dense(units=1)  # Output layer
])

# Compile the model with Adam optimizer and mean squared error loss
model_lstm_improved.compile(optimizer=Adam(learning_rate=0.001), loss='mean_squared_error')

# Define early stopping to prevent overfitting
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)

# Train the LSTM model with improved architecture and regularization
history = model_lstm_improved.fit(X_train, y_train, epochs=50, batch_size=32,
                                  validation_data=(X_val, y_val), callbacks=[early_stopping])

# Evaluate the improved LSTM model
lstm_improved_forecast = model_lstm_improved.predict(X_test)

# Calculate evaluation metrics
lstm_improved_mse = mean_squared_error(y_test, lstm_improved_forecast)
lstm_improved_rmse = np.sqrt(lstm_improved_mse)
lstm_improved_mae = mean_absolute_error(y_test, lstm_improved_forecast)
```

```
# Print evaluation metrics
print("Improved LSTM Model Evaluation:")
print("Mean Squared Error (MSE):", lstm_improved_mse)
print("Root Mean Squared Error (RMSE):", lstm_improved_rmse)
print("Mean Absolute Error (MAE):", lstm_improved_mae)
import matplotlib.pyplot as plt
import pandas as pd

# Hypothetical actual sales data
actual_sales = pd.Series([3000, 3200, 3500, 3700, 4000, 4100, 4200, 4300, 4400, 4500,
4600, 4700, 4800, 4900],
                         index=pd.date_range(start='2024-01-01', periods=14, freq='M'))

# Hypothetical forecasted sales data for each model
arima_forecast = pd.Series([3100, 3300, 3600, 3800, 4100, 4200, 4300, 4400, 4500,
4600, 4700, 4800, 4900, 5000],
                           index=pd.date_range(start='2024-01-01', periods=14, freq='M'))

sarima_forecast = pd.Series([3150, 3350, 3650, 3850, 4150, 4250, 4350, 4450, 4550,
4650, 4750, 4850, 4950, 5050],
                            index=pd.date_range(start='2024-01-01', periods=14, freq='M'))

lstm_improved_forecast = pd.Series([3200, 3400, 3700, 3900, 4200, 4300, 4400, 4500,
4600, 4700, 4800, 4900, 5000, 5100],
                                   index=pd.date_range(start='2024-01-01', periods=14, freq='M'))

# Plot the actual sales data
plt.figure(figsize=(10, 6))
plt.plot(actual_sales.index, actual_sales, label='Actual Sales', marker='o')

# Plot the forecasted sales data for each model
plt.plot(arima_forecast.index, arima_forecast, label='ARIMA Forecast', linestyle='--')
plt.plot(sarima_forecast.index, sarima_forecast, label='SARIMA Forecast',
linestyle='-.')
plt.plot(lstm_improved_forecast.index, lstm_improved_forecast, label='Improved
LSTM Forecast', linestyle=':')

# Add labels and title
plt.xlabel('Date')
plt.ylabel('Sales')
plt.title('Forecasting Sales Comparison')
plt.legend()

# Show the plot
plt.tight_layout()
plt.show()
```

```
#cross validation for SARIMA
import numpy as np
import pandas as pd
from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.metrics import mean_squared_error, mean_absolute_error


# Define the length of training and testing periods
train_period = 12  # 12 months
test_period = 3    # 3 months

sales_data = pd.DataFrame(sales_data)

# Initialize lists to store evaluation metrics
mse_list_sarima, rmse_list_sarima, mae_list_sarima = [], [], []

# Perform Time Series Cross-Validation for SARIMA
for i in range(train_period, len(sales_data) - test_period):
    train_data = sales_data.iloc[:i]
    test_data = sales_data.iloc[i:i+test_period]

    # Train the SARIMA model
    model_sarima = SARIMAX (train_data, order=(1, 1, 1), seasonal_order=(1, 1, 1,
12))  # Adjust order as needed
    model_fit_sarima = model_sarima.fit()

    # Make predictions for the testing period
    forecast_sarima = model_fit_sarima.predict(start=len(train_data),
end=len(train_data) + test_period - 1)

    # Calculate evaluation metrics for SARIMA
    mse_sarima = mean_squared_error(test_data, forecast_sarima)
    rmse_sarima = np.sqrt(mse_sarima)
    mae_sarima = mean_absolute_error(test_data, forecast_sarima)

    # Append metrics to lists
    mse_list_sarima.append(mse_sarima)
    rmse_list_sarima.append(rmse_sarima)
    mae_list_sarima.append(mae_sarima)

# Calculate average metrics for SARIMA
avg_mse_sarima = np.mean(mse_list_sarima)
avg_rmse_sarima = np.mean(rmse_list_sarima)
avg_mae_sarima = np.mean(mae_list_sarima)
```

```
# Print the average metrics for SARIMA
print("Average MSE for SARIMA:", avg_mse_sarima)
print("Average RMSE for SARIMA:", avg_rmse_sarima)
print("Average MAE for SARIMA:", avg_mae_sarima)


#cross validation for LSTM
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, mean_absolute_error
from keras.models import Sequential
from keras.layers import LSTM, Dense


# Define the length of training and testing periods
train_period = 12  # 12 months
test_period = 3    # 3 months

# Convert the dataframe to numpy array
sales_data = df['Sales'].values

# Initialize lists to store evaluation metrics
mse_list_lstm, rmse_list_lstm, mae_list_lstm = [], [], []

# Perform Time Series Cross-Validation for LSTM
for i in range(train_period, len(sales_data) - test_period):
    X, y = [], []
    for j in range(i - train_period, i):
        X.append(sales_data[j - train_period:j])
        y.append(sales_data[j])

    # Convert X and y to numpy arrays
    X, y = np.array(X), np.array(y)

    # Reshape the input data for LSTM (samples, time steps, features)
    X = X.reshape((X.shape[0], X.shape[1], 1))

    # Split the data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_period,


shuffle=False)



    # Define the LSTM model
    model_lstm = Sequential()
    model_lstm.add(LSTM(50, activation='relu', input_shape=(train_period, 1)))
```

```
    model_lstm.add(Dense(1))
    model_lstm.compile(optimizer='adam', loss='mse')

    # Fit the LSTM model to the data
    model_lstm.fit(X_train, y_train, epochs=200, verbose=0)

    # Make predictions for the testing period
    forecast_lstm = model_lstm.predict(X_test)

    # Calculate evaluation metrics for LSTM
    mse_lstm = mean_squared_error(y_test, forecast_lstm)
    rmse_lstm = np.sqrt(mse_lstm)
    mae_lstm = mean_absolute_error(y_test, forecast_lstm)

    # Append metrics to lists
    mse_list_lstm.append(mse_lstm)
    rmse_list_lstm.append(rmse_lstm)
    mae_list_lstm.append(mae_lstm)

# Calculate average metrics for LSTM
avg_mse_lstm = np.mean(mse_list_lstm)
avg_rmse_lstm = np.mean(rmse_list_lstm)
avg_mae_lstm = np.mean(mae_list_lstm)

# Print the average metrics for LSTM
print("Average MSE for LSTM:", avg_mse_lstm)
print("Average RMSE for LSTM:", avg_rmse_lstm)
print("Average MAE for LSTM:", avg_mae_lstm)
```

PAPER PUBLICATION STATUS

Submission Summary

Conference Name
International Conference on Circuit Power and Computing Technologies 2024

Paper ID
382

Paper Title
Forecasting sales data using time series models and LSTM model

Created
4/23/2024, 6:57:25 PM

Last Modified
4/23/2024, 6:57:25 PM

Authors
Mohammed Basheeruddin (SRM Institute of Science and Technology)
<mb5814@srmist.edu.in>
Anirudh Vishwanath (SRM INSTITUTE OF SCIENCE AND TECHNOLOGY)
<av8826@srmist.edu.in>

Primary Subject Area
COMPUTER -> Neural Networks

Submission Files
Forecasting sales data using ARIMA AND SARIMA Model Main research paper.docx (847.3 Kb, 4/23/2024, 6:53:07 PM)

11-58(1).docx

9%	4%	7%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

ORIGINALITY REPORT

PRIMARY SOURCES

1	Hudzaifah Hasri, Siti Armiza Mohd Aris, Robiah Ahmad. "Comparison of Auto ARIMA and Auto SARIMA Performance in COVID-19 Prediction", 2023 IEEE 2nd National Biomedical Engineering Conference (NBEC), 2023 Publication	1%
2	Feng Wang, Aviles Joey S. "Using Regression Algorithms to Forecast Merchandise Sales in the Presence of Independent Variables", 2022 7th International Conference on Cyber Security and Information Engineering (ICCSIE), 2022 Publication	1%
3	fastercapital.com Internet Source	1%
4	Shatha Ghareeb, Mohamed Mahyoub, Jamila Mustafina. "A comparative Time Series analysis of the different categories of items based on holidays and other events", 2023 15th International Conference on	1%

Developments in eSystems Engineering (DeSE), 2023
Publication

5	Pelin Dincoglu, Huseyin Aygun. "Comparison of Forecasting Algorithms on Retail Data", 2022 10th International Symposium on Digital Forensics and Security (ISDFS), 2022 Publication	1 %
6	Suresh B S, M. Suresh. "A Comprehensive Analysis of Retail Sales Forecasting Using Machine Learning and Deep Learning Methods", 2023 International Conference on Data Science and Network Security (ICDSNS), 2023 Publication	<1 %
7	Aini Fatina Mohamad, Aisyah Mat Jasin, Aszila Asmat, Roger Canda, Juhaida Ismail, Afiqah Bazlla Md Soom. "Sales Analytics Dashboard with ARIMA and SARIMA Time Series Model", 2023 IEEE 13th Symposium on Computer Applications & Industrial Electronics (ISCAIE), 2023 Publication	<1 %
8	www.ijraset.com Internet Source	<1 %
9	Anik Pramanik, Salma Sultana, Md. Sadekur Rahman. "Time Series Analysis and Forecasting of Monkeypox Disease Using	<1 %

ARIMA and SARIMA Model", 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2022

Publication

10	doctorpenguin.com Internet Source	<1%
11	journal.50sea.com Internet Source	<1%
12	Submitted to Notre Dame of Marbel University Student Paper	<1%
13	Submitted to University of Surrey Student Paper	<1%
14	ieeexplore.ieee.org Internet Source	<1%
15	www.mdpi.com Internet Source	<1%
16	khazna.ku.ac.ae Internet Source	<1%
17	Submitted to Liverpool John Moores University Student Paper	<1%
18	dokumen.pub Internet Source	<1%

19	Submitted to University of the Pacific Student Paper	<1%
20	Submitted to Free University of Bolzano Student Paper	<1%
21	Prity Kumari, Viniya Goswami, Harshith N., R. S. Pundir. "Recurrent neural network architecture for forecasting banana prices in Gujarat, India", PLOS ONE, 2023 Publication	<1%
22	www.boj.org.jm Internet Source	<1%
23	miscj.aut.ac.ir Internet Source	<1%
24	Submitted to City University Student Paper	<1%
25	github.com Internet Source	<1%
26	Submitted to University of Sunderland Student Paper	<1%
27	Submitted to Yeditepe University Student Paper	<1%
28	pyvideo.org Internet Source	<1%
29	fdocuments.in Internet Source	<1%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		

Format - I

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Office of Controller of Examinations

REPORT FOR PLAGIARISM CHECK ON THE DISSERTATION/PROJECT REPORTS FOR UG/PG PROGRAMMES

(To be attached in the dissertation/ project report)

1	Name of the Candidate **(IN BLOCK LETTERS)**	1.Mohammed Basheeruddin 2.Anirudh Vishwanath
2	Address of the Candidate	1. T2, Swanlake apartments, Kukatpally, Hyderabad 2. E-304, Mantri Woodlands, Arekere, Bangalore -76
3	Registration Number	1.RA2011031010001 2.RA2011031010045
4	Date of Birth	1.11/08/2002 2.01/08/2002
5	Department	Networking and Communications
6	Faculty	Engineering and Technology, School of Computing
7	Title of the Dissertation/Project	Forecasting Sales data using ARIMA,SARIMA and LSTM Model
8	Whether the above project /dissertation is done by	Group Project 1.Mohammed Basheeruddin (RA2011031010001) 2.Anirudh Vishwanath (RA2011031010045) :
9	Name and address of the Supervisor / Guide	Mrs. Saveetha D Department of Networking and Communications College of Engineering and Technology Kattankulathur-603203 **Mail ID:** saveethd@srmist.edu.in **Mobile Number:**94440222933
10	Name and address of Co-Supervisor / Co- Guide (if any)	**Mail ID:** **Mobile Number:**

11	Software Used	TURNITIN
12	Date of Verification	22/4/2024
13	**Plagiarism Details: (to attach the final report from the software)**	

Chapter	Title of the Chapter	Percentage of similarity index (including self citation)	Percentage of similarity index (Excluding self-citation)	% of plagiarism after excluding Quotes, Bibliography, etc.,
	ABSTRACT	2	1	1
1	**INTRODUCTION**	2	1	0
2	**LITERATURE REVIEW**	2	1	0
3	**PROPOSED METHODOLOGIES**	1	1	1
4	**RESULTS**	1	1	1
5	**CONCLUSION**	3	1	3
6	**FUTURE SCOPE**	1	0	1
	Appendices			

We declare that the above information have been verified and found true to the best of our knowledge.

Mohammed Basheeruddin: Anirudh Vishwanath: **Signature of the Candidate**	Mrs. Saveetha D: **Name & Signature of the Staff (Who uses the plagiarism check software)**
Mrs. Saveetha D: **Name & Signature of the Supervisor/ Guide**	**Name & Signature of the Co-Supervisor/Co-Guide**

Dr. ANNAPURANI PANAIYAPPAN K
(PROFESSOR AND HEAD OF DEPARTMENT OF NETWORKING AND COMMUNICATIONS)

Name & Signature of the HOD