FORECASTING HOUSE PROPERTY SALES PRICES USING LSTM-BASED MULTIVARIATE TIME SERIES ANALYSIS

Dr. David Raj Micheal

Division of Mathematics

School of Advanced Sciences

Vellore Institute of Technology Chennai

Tamil Nadu -600127

davidraj.micheal@vit.ac.in

B.G.Dharsana

Division of Mathematics

School of Advanced Sciences

Vellore Institute of Technology Chennai

Tamil Nadu -600127

dharshana.2023@vitstudent.ac.in

Abstract-Accurate forecasting of property sales prices is critical in the real estate sector, impacting investment strategies, market analysis, and economic forecasting. This project explores a deep learning approach using a Long Short-Term Memory (LSTM) model to predict sales prices of residential properties with varying bedroom counts (1-5) for houses and units over a 12-year period (2007-2019) in a specific region. Unlike traditional univariate models, this multivariate time series model captures not only the temporal dependencies within each property category but also cross-dependencies across property types and bedroom counts, enhancing prediction accuracy. The LSTM model architecture is particularly suited for handling long-term dependencies, making it an ideal choice for complex, multivariate time series data. By leveraging LSTM's ability to model both individual and interrelated patterns, this study aims to provide more reliable forecasts for property sales, supporting informed decision-making in real estate planning and financial analysis.

Keywords- Property Sales Forecasting, LSTM Model, Multivariate Time Series, Cross-Dependencies, Real Estate Market Analysis

I. INTRODUCTION

The real estate market is a dynamic sector where accurate price forecasting plays a vital role in guiding investment decisions, property valuations, and economic planning. Traditional methods for predicting property prices often focus on univariate time series models that analyse each property type in isolation, overlooking the interrelated dynamics that exist between various types of properties. In practice, the sales prices of houses and units with different bedroom counts are interconnected, influenced not only by their historical prices but also by market trends affecting related property types.

This study leverages the capabilities of Long Short-Term Memory (LSTM) networks to address the multivariate nature of property sales data, incorporating both temporal dependencies and cross-variable influences in a unified forecasting model. Spanning a 12-year period (2007-2019), our dataset includes sales prices for houses and units with 1-5 bedrooms, offering a robust foundation for multivariate time series analysis. By capturing complex, long-term dependencies, the LSTM model provides a more holistic view of market trends, allowing for improved accuracy in forecasting. This research aims to contribute to the development of advanced predictive tools that offer deeper insights into property price dynamics, enabling real estate professionals and investors to make more informed, data-driven decisions.

II. OBJECTIVE

The objective of this study is to develop a predictive model using Long Short-Term Memory (LSTM) networks to accurately forecast residential property sales prices in a specified region over the 2007-2019 period. By leveraging multivariate time series data encompassing sales prices for houses and units with varying bedroom counts (1-5), this research aims to capture both temporal dependencies and cross-variable relationships within the real estate market. The model seeks to enhance forecasting accuracy and provide valuable insights into property price trends, thereby supporting data-driven decision-making for real estate investors, analysts, and planners.

III. LITERATURE REVIEW

Papastefanopoulos, V., Linardatos, P., Panagiotakopoulos, T., & Kotsiantis, S. (2023) explains that Smart cities are increasingly relying on technologies like the Internet of Things (IoT) to manage urban growth and optimize resources. Time-series forecasting, a method of predicting future values based on historical data, is crucial for these applications. Deep learning techniques like LSTM networks, CNNs, and Transformer-based models are used to handle multivariate time-series data. These models are applied in various domains, including traffic management, energy management, waste management, healthcare, environmental monitoring, and public safety and security. The application of deep learning not only enhances operational efficiency but also contributes to creating more resilient, sustainable, and livable urban environments. Advances in machine learning and deep learning methodologies are actively addressing challenges in multivariate time-series forecasting. **Shi, S. (2023)** tells that the study investigates the performance of Long Short-Term Memory (LSTM) and LightGBM models on Beijing's

second-hand housing price dataset. LSTM outperforms LSTM in predicting housing prices based on historical time series data, but LightGBM outperforms it in incorporating multiple factors beyond time series. The results suggest that LSTM is more effective in time series predictions, but less effective in multi-factor models. Both models have strengths depending on the input data, with LSTM suitable for time series forecasting and LightGBM better for complex, multi-factor input.

Rostami, J., & Hansson, F. (2019) this thesis compares two machine learning methods, Long Short-Term Memory (LSTM) and Support Vector Machine (SVM), for forecasting monthly house prices in Stockholm and Uppsala from 2005 to 2019. The results show that LSTM outperforms SVM in accuracy for a 12-month horizon but lags behind ARIMA over the same period. This suggests that while LSTM shows promise, traditional methods like ARIMA may still offer more reliable forecasts for certain time series data, such as housing prices. **Nguyen, H. D., Tran, K. P., Thomassey, S., & Hamad, M. (2021)** this paper presents two data-driven approaches to improve supply chain management decision-making, focusing on forecasting and anomaly detection. The first uses Long Short-Term Memory (LSTM) networks for multivariate time series data, while the second uses an LSTM Autoencoder network and a SVM algorithm for anomaly detection. The methods are tested on benchmarking datasets and real-world data from the fashion retail industry.

Kalinga, E., Nyamle, E., & Abdalla, A. (2023) describes that a real estate price prediction model using Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) to address the challenge of predicting residential housing prices. The model identifies key factors influencing prices, including size, location, time, property quality, and land prices. The model outperforms Convolutional Neural Networks by 50%, achieving accuracies of 97.45%, 79.23%, and 53.8% for different market classes. **Hong, J. K. (2021)** this study presents a model using Long Short-Term Memory (LSTM) networks to predict product sales based on temperature changes. It predicts sales for clothing items like short pants, flip-flop sandals, and winter outerwear. The model's results align with real-world patterns, demonstrating its effectiveness in forecasting seasonal product sales.

Wan, H., Guo, S., Yin, K., Liang, X., & Lin, Y. (2020) explains that a new approach to forecasting correlated time series using Long Short-Term Memory (LSTM) called CTS-LSTM. This method addresses the challenges of modeling both intra-sequence temporal and inter-sequence spatial dependencies, which are crucial when dealing with multiple correlated time series. The proposed solution, CTS-LSTM, explicitly models and maintains both spatial and temporal correlations in separate components, capturing complex, non-linear patterns observed in correlated time series. The model outperforms state-of-the-art baselines on real-world datasets, achieving lower errors in RMSE, MAE, and MAPE. This approach is applicable for various domains, such as demand forecasting and environmental monitoring, and could be useful for air quality control projects or other applications dealing with correlated time series prediction.**Wang, X., Liu, H., Du, J., Dong, X., & Yang, Z. (2023)** mentioned that SDCNet, a novel model for multivariate time series forecasting, aimed at improving long-term forecasting accuracy and efficiency by addressing temporal patterns and complex inter-variable dependencies. The model decomposes the time series into seasonal and trend-cyclical components, using Convolutional Neural Networks (CNNs) for temporal and feature convolution. SDCNet outperforms competing methods, showing a 16.73% improvement in relative accuracy across four real-world datasets and a 23.87% performance gain on datasets without significant periodicity.

Zupan, M. (2024) this paper discusses the challenges faced by small entrepreneurs in the commodities trading business, particularly in building materials, due to price volatility, logistic issues, raw material shortages, and inflation. It suggests that historical data from accounting books can provide valuable information about previous warehouse dynamics, improving COGS forecasting. The study demonstrates that transforming historical accounting data into time series data suitable for machine learning algorithms can improve forecasting of COGS in the commodities trading industry, particularly for warehouse management**. Ryu, H. S., Kim, H. J., & Kim, M. K. (2023)** this paper presents a two-level optimal bidding strategy framework for load aggregators to address demand response (DR) challenges, particularly in distributed energy resources like renewable energy. The framework combines a data-driven forecasting model and a data-driven agent-based model to provide a more realistic estimate of DR's impact. Key components include a Long Short-Term Memory (LSTM) autoencoder model for predicting aggregated load and market prices, a Data-Driven Agent-Based Model (D-ABM) for estimating DR's impact, and bidding strategy optimization. The framework effectively handles residential electricity consumption uncertainties, leading to improved profits and better resource management.

IV. METHODOLOGY

1. **Data Structure:** The dataset used in this study was sourced from Kaggle (https://www.kaggle.com/), one of the most well-known and regarded platforms for hosting datasets pertaining to various fields. The dataset contains four columns

    - **Date:** Represents the time of the observation.
    - **MA (Moving Average):** The target variable you want to predict, which is a numerical representation of the median price over a specified period.
    - **Type:** A categorical variable indicating whether the property is a "house" or a "unit."
    - **Bedroom:** A numerical variable indicating the number of bedrooms in the property.

2. **Data Preprocessing:**

    - **Date Handling**: Convert the date column into a datetime format to facilitate time series analysis. Setting the date as the index

allows for easier manipulation and visualization of time series data.

- **Handling Missing Values:** Check for any missing values in the dataset. Depending on the context, you may choose to fill these gaps (e.g., using forward fill or interpolation) or remove rows with missing values.
- **Encoding Categorical Variables:** Since LSTM models require numerical input, categorical variables like "type" need to be converted into a numerical format. One-hot encoding is a common technique where each category is represented as a binary vector.

3. Feature Selection:

- **Identifying Features and Target:** The target variable is the moving average (MA). The features can include the one-hot encoded values of the "type" column, the "bedroom" count, and potentially lagged values of the MA itself (previous days' MA) to provide context for the prediction.
- **Feature Engineering:** Consider creating additional features that may help the model, such as: Lagged values of MA (e.g., MA from the previous day, week, etc.) and Rolling statistics (e.g., rolling mean, rolling standard deviation) to capture trends and seasonality.

4. Data Scaling:

Normalization: Scale the target variable (MA) to a range between 0 and 1 using techniques like Min-Max scaling. This is crucial for LSTM models, as they are sensitive to the scale of input data. Normalization helps in faster convergence during training and improves model performance. Normalization is a critical step in LSTM modeling, especially for time series data, due to several key reasons related to the model's efficiency and performance:

Stabilizes Training and Improves Convergence:

- Time series data often has varying scales (e.g., temperature in Celsius vs. revenue in thousands of dollars). LSTMs and other neural networks perform better when inputs are normalized to a consistent scale, often between 0 and 1 or -1 and 1.
- This scaling reduces the range of the input values, making the model's learning process more stable and enabling faster convergence during training, as it prevents gradients from becoming too large or too small (i.e., reducing the risk of gradient vanishing or exploding).

Enables Better Handling of Long Sequences:

- LSTMs are designed to handle sequences of data, but if the values vary widely across time steps, it becomes difficult for the model to maintain the necessary temporal relationships.
- Normalization ensures that variations in data are relative rather than absolute, helping the LSTM maintain consistent weight updates over long sequences. This is especially beneficial for remembering patterns in time series over many time steps.

5. Creating Sequences for LSTM

- **Time Series Data Preparation:** LSTMs require input in the form of sequences. This means you need to create overlapping sequences of data where each sequence consists of a fixed number of time steps (e.g., the last 10 days of MA) to predict the next value.
- **Sequence Length:** The choice of sequence length (time step) is important. A longer sequence may capture more temporal dependencies but can also introduce noise, while a shorter sequence may miss important trends.

6. Splitting the Data

Train-Test Split: Divide the dataset into training and testing sets. A common practice is to use 80% of the data for training and 20% for testing. This ensures that the model is trained on a substantial amount of data while still being evaluated on unseen data to assess its generalization ability.

7. Building the LSTM Model

Model Architecture: An LSTM model typically consists of:

Input Layer: Accepts the input sequences.

LSTM Layers: These layers are designed to capture temporal dependencies in the data. You can stack multiple LSTM layers to allow the model to learn more complex patterns.

Dropout Layers: These are used to prevent overfitting by randomly setting a fraction of the input units to 0 during training.

Output Layer: A Dense layer that outputs the predicted value of MA.

8. Compiling the Model

Loss Function: The model is compiled with a loss function suitable for regression tasks, such as Mean Squared Error (MSE), which measures the average

squared difference between predicted and actual values.

Optimizer: An optimizer like Adam is commonly used for training LSTM models due to its adaptive learning rate properties.

9. Training the Model

Fitting the Model: The model is trained on the training dataset for a specified number of epochs. During training, the model learns to minimize the loss function by adjusting its weights based on the input data.

Batch Size: The batch size determines how many samples are processed before the model's internal parameters are updated. Smaller batch sizes can lead to more stable convergence.

10. Making Predictions

Using the Trained Model: After training, the model can be used to make predictions on the test dataset. The model will output

V. LONG SHORT-TERM MEMORY (LSTM) ARCHITECTURE

Long Short-Term Memory (LSTM) networks are specialized types of Recurrent Neural Networks (RNNs) that excel at learning from sequences and handling time-dependent data. Unlike basic RNNs, which struggle with long-term dependencies, LSTMs maintain information over extended time frames. They achieve this by using a unique cell structure comprising a memory cell and three gates—forget, input, and output gates—that carefully control the information flow at each time step.

Core Components of LSTM Architecture

1. **Memory Cell**: This cell acts as the LSTM's "memory," retaining information across time steps. It helps overcome the vanishing gradient problem, which often affects simpler RNNs, by storing relevant data across longer sequences.
2. **Gates in LSTM**:
    - **Forget Gate**: Controls what portion of the prior memory to discard. It uses a sigmoid activation to determine which data points are less relevant as new information arrives.
    - **Input Gate**: Manages which new information will be added to the cell's memory. This gate has two parts: a sigmoid layer that decides what parts to update, and a tanh layer that provides candidate values for updating.
    - **Output Gate**: Decides what information from the cell's memory to output at each time step, producing the hidden state that will move to the next cell or feed into the final output layer.
3. **Updating the Cell State**: LSTM cells continually update their memory by selectively discarding and adding information based on the forget and input gates' output. This regulated update allows the model to learn patterns over long sequences.
4. **Hidden State**: This is the LSTM's immediate output at each step, encapsulating the relevant information for predicting the next time step or passing data to subsequent layers.

How LSTMs Function Over Time

At each time step:

- The **forget gate** removes unimportant parts of the previous cell state.
- The **input gate** selects new information to incorporate into the memory.
- The **output gate** generates the hidden state that serves as output, helping capture essential sequence information.

Benefits of LSTMs

- **Retain Long-Term Patterns**: LSTMs preserve dependencies over extended time steps, which is useful in tasks involving sequential data, like language and time series.
- **Reduces Gradient Vanishing**: By using memory cells, LSTMs keep gradients stable, making training easier for deeper models.

Common Uses of LSTMs

- **Forecasting Time Series**: Sales, air quality, or other sequential data forecasting.
- **Sequence Modeling**: Useful in machine translation, speech processing, and text generation.
- **Complex Event Analysis**: Applied in video and anomaly detection for quality control or security.

VI. MODEL EVALUATION

Post-training, each model's performance was rigorously evaluated using a range of metrics:

- **Mean Squared Error (MSE**): MSE measures the average of the squares of the errors, which indicates the average squared difference between predicted and actual values. A lower MSE indicates better model performance.

- **Mean Absolute Error (MAE):** MAE represents the average absolute difference between predicted and actual values. It provides a straightforward measure of prediction accuracy, with lower values indicating better performance.

Root Mean Squared Error (RMSE): RMSE is the square root of MSE and provides an error metric in the same units as the predicted values. It is sensitive to outliers and gives a

higher weight to larger errors, making it useful for understanding the model's performance in practical terms.

R-squared (R^2): R^2 indicates the proportion of variance in the dependent variable that can be explained by the independent variables in the model. An R^2 value close to 1 suggests that the model explains a significant portion of the variance, indicating strong predictive capability.

VII. ANALYSIS OF RESULT

The predicted values show a general upward trend, which aligns with the expected behaviour of median house sale prices over time. The model appears to capture the overall trend effectively, as evidenced by the high R^2 value of 0.921. However, there are noticeable discrepancies between predicted and actual values, particularly in the later predictions, where the predicted values are significantly higher than the actual values. This could indicate that the model may be overfitting or that there are external factors influencing house prices that are not captured in the training data.

MSE	393224622.73
MAE	10969.006
RMSE	19829.89
R^2	0.921

VIII. CONCLUSION

The LSTM model demonstrates strong predictive performance for the moving average of median house sale prices, as indicated by the evaluation metrics. While the model captures the general trend effectively, further refinement may be necessary to improve accuracy, particularly in later predictions. Future work could involve exploring additional features, tuning hyperparameters, or employing ensemble methods to enhance model robustness.

IX. REFERENCES

Papastefanopoulos, V., Linardatos, P., Panagiotakopoulos, T., & Kotsiantis, S. (2023). Multivariate Time-Series Forecasting: A Review of Deep Learning Methods in Internet of Things Applications to Smart Cities. *Smart Cities*, *6*(5), 2519-2552**.**

Shi, S. (2023). Comparison of Real Estate Price Prediction Based on LSTM and LGBM. *Highlights in Science, Engineering and Technology*, *49*, 294-301.

Rostami, J., & Hansson, F. (2019). Time series forecasting of house prices: An evaluation of a support vector machine and a recurrent neural network with LSTM cells.

Nguyen, H. D., Tran, K. P., Thomassey, S., & Hamad, M. (2021). Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management. *International Journal of Information Management*, *57*, 102282.

Kalinga, E., Nyamle, E., & Abdalla, A. (2023). Modelling Prediction of Cities Real Estate Price Trend Using Recurrent Neural Network: A Case of Dar es Salaam City. *Tanzania Journal of Engineering and Technology*, *42*(3), 64-77.

Hong, J. K. (2021). LSTM-based sales forecasting model. *KSII Transactions on Internet & Information Systems*, *15*(4).

Aseeri, A. O. (2023). Effective short-term forecasts of Saudi stock price trends using technical indicators and large-scale multivariate time series. *PeerJ Computer Science*, *9*, e1205.

Wan, H., Guo, S., Yin, K., Liang, X., & Lin, Y. (2020). CTS-LSTM: LSTM-based neural networks for correlatedtime series prediction. *Knowledge-Based Systems*, *191*, 105239.

Wang, X., Liu, H., Du, J., Dong, X., & Yang, Z. (2023). A long-term multivariate time series forecasting network combining series decomposition and convolutional neural networks. *Applied Soft Computing*, *139*, 110214.

Zupan, Mario. "Accounting journal entries as a long-term multivariate time series: Forecasting wholesale warehouse output." *Intelligent systems in accounting, finance and management* 31, no. 1 (2024): e1551.

Ryu, H. S., Kim, H. J., & Kim, M. K. (2023). Two-Level Optimal Bidding Strategy for Load Aggregator Based on a Data-Driven Approach Combined with LSTM-Based Forecasting and Agent-Based Models. *IEEE Access*.