A Multi-agent Machine Learning framework for Hybrid house price Prediction and Forecasting: Integrating Supervised Learning, Time-series Analysis, and Metaheuristic optimization

Tan Thuy Linh Dao _ c3425175

***Index Terms*—ARIMA, LSTM, Time Series Forecasting, House Pricing, Machine Learning, Computer Science**

I. INTRODUCTION

Accurate house price prediction has become increasingly crucial in today's dynamic real estate market, where fluctuating prices significantly impact buyers, sellers, and investors[1]. The growing demand for housing, driven by population growth and urbanization, has made reliable price forecasting essential for informed decision-making[2]. This project addresses this challenge by developing a machine learning model using two complementary datasets: one focusing on Sydney's local market factors and another analyzing broader Australian housing trends. The motivation for this research stems from the need to provide accessible tools that help individuals navigate complex real estate decisions. By predicting both current prices and future trends, the model aims to assist home-buyers in budgeting, sellers in pricing strategies, and investors in identifying opportunities. The project also serves as a practical application of advanced computational techniques to solve real-world economic problems. Methodologically, this work incorporates key course topics including supervised machine learning (Random Forest, SVM), deep learning (LSTM), and time-series analysis (ARIMA). The multi-agent approach coordinates these techniques while employing metaheuristics for hyperparameter optimization, demonstrating how different algorithms can work together to improve prediction accuracy. Through systematic comparison of these methods, the project provides insights into their effectiveness for housing market analysis.

II. METHODOLOGY

A. Chosen Machine Intelligence Approach)

This project adopts supervised learning, a machine learning approach that trains a model using labeled data to predict continuous output values [3]. In supervised learning, the algorithm learns a mapping function $Q = f(P)$ Where P represents input features (e.g., property size, location, economic indicators) and Q denotes the target output (house price). The goal is to train the model on historical housing data so it can generalize patterns and accurately predict prices for new, unseen property listings [4]. Since house price prediction involves estimating a continuous numerical value, we specifically use regression techniques rather than classification. To enhance usability, the system follows a multi-agent architecture composed of two specialized components. The first agent provides a macroeconomic perspective, analyzing national and state-level housing trends through interactive visualizations. Users can explore historical price movements and forecasts across different regions (e.g., NSW, Victoria). Once a state is selected, the second agent takes over for micro-level analysis, allowing users to input specific property details (e.g., bedrooms, bathrooms, distance from CBD). Combining these inputs with regional market data, the system generates a personalized price prediction. This two-stage approach ensures users first understand broader market conditions before obtaining tailored property valuations, making the prediction process both informative and user-friendly.

B. Selection of Algorithms and Models

For the Micro-level Analysis Agent, we evaluate multiple supervised regression algorithms [2] to identify the optimal model for property price prediction:

- **Linear Regression (LR)** serves as our baseline model due to its interpretability and computational efficiency [5]. The simple linear form $y = \alpha + \beta x$ models the relationship between a single predictor and house price, while multiple linear regression

  $$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

  incorporates all property features (e.g., bedrooms, location). Though prone to underfitting with complex relationships, LR provides valuable benchmarking insights [1].
- **Random Forest (RF)** addresses LR's limitations through ensemble learning. By aggregating predictions from multiple decision trees (each trained on random feature subsets), RF improves accuracy while mitigating overfitting. The model's feature importance scores also enhance interpretability by quantifying each predictor's contribution to price variations.
- **Support Vector Regression (SVR)** applies kernel functions to map nonlinear relationships into higher-dimensional spaces. The hyperplane equation $y = wx + b$ optimizes the margin using support vectors,

making it robust to outliers. SVR excels when feature-target relationships are complex but requires careful hyperparameter tuning [3].

For the Macro-level Analysis Agent, we compare time-series forecasting [6] approaches:

- **ARIMA** combines autoregressive (AR) and moving average (MA) components with differencing (I) to handle non-stationary data. The general form effectively captures short-term trends but struggles with long-term dependencies [4].

$$y'_t = c+\alpha_1 y'_{t-1}+\cdots+\alpha_p y'_{t-p}+e_t+\theta_1 e_{t-1}+\cdots+\theta_q e_{t-q}$$

- **LSTM networks** overcome this limitation via gating mechanisms (input, output, forget gates) that regulate information flow. The architecture's memory cells preserve long-range temporal patterns, making it ideal for multi-year housing market forecasts [5].

Performance Metrics – Model evaluation employs:

- **R² (Coefficient of Determination)**: Measures explained variance (higher = better fit).

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

- **MSE (Mean Squared Error)**: Quantifies average prediction error (lower = better).

$$\text{MSE} = \frac{1}{N}\sum(y_i - \hat{y})^2$$

- **RMSE (Root MSE)**: Provides error interpretation in original units.

$$\text{RMSE} = \sqrt{\text{MSE}}$$

C. Dataset and Preprocessing steps

For the Micro-level Analysis Agent, we utilized the `domain_properties.csv` dataset from Kaggle, containing 11,160 property listings with 17 features. The dataset includes both numerical features (such as number of rooms, distance from CBD, and median income) and categorical features (like property type and suburb), with price as our target variable. The preprocessing began by importing necessary libraries and reading the CSV file, followed by checking for missing values. We converted the `date_sold` column to datetime format and removed outliers using the interquartile range (IQR) method to ensure data quality. Property types were categorized into three main groups: house, apartment, and other, then applied map value method to converting to numberic for simplifying the analysis. We conducted descriptive analysis through histograms and boxplots to understand variable distributions and employed a heatmap correlation matrix to identify relationships between features and price. Based on this analysis, we created a refined dataframe (prediction_df) by dropping columns with weak or no correlation to price, ensuring our model focuses on the most relevant predictors. For the Macro-level Analysis Agent, we worked with the "641606.xlsx" dataset from the Australian Bureau of Statistics (ABS), which contains 46 columns and 51 rows of national housing data. This comprehensive dataset includes various metrics such as dwelling stock values and mean prices across Australian states. We focused specifically on mean house prices, retaining only relevant columns and renaming them to state abbreviations (e.g., NSW, VIC) for clarity. The initial nine rows, containing metadata rather than actual data points, were removed. We converted the price values to integers and transformed the first column into a datetime format, labeling it as 'Date' to establish a clear time series. After verifying there were no missing values and confirming all data types were correct, we generated a line visualization to observe housing price trends over time, providing valuable insights into market dynamics.

III. IMPLEMENTATION

The project was developed using the following software tools and frameworks:

- **Primary IDE**: Visual Studio Code (version 1.99.3) with Python extensions
- **Programming Language**: Python (version 3.11.5)
- **Key Libraries**:
  - Data processing: Pandas, NumPy
  - Visualization: Matplotlib, Seaborn
  - Machine learning: scikit-learn, TensorFlow/Keras (for LSTM)
  - Time-series analysis: statsmodels (for ARIMA)
- **AI-Assisted Development**:
  - GitHub Copilot: Integrated with Visual Studio Code through GitHub Education to provide:
    * Code completion suggestions
    * Function generation
    * Debugging assistance
  - Supplemental Learning Resources:
    * Kaggle notebooks for algorithm implementation reference
    * YouTube tutorials for specific technical challenges

A. Micro-level analysis agent

The implementation began with data preparation, where we defined the independent variables (X) as all features except the target variable `price`, including numerical attributes like the number of bedrooms, bathrooms, parking spaces, property size, distance from the CBD, and suburb population, etc. . Categorical variables were mapped into numeric values to ensure compatibility with the models. The dependent variable (y) was set as the 'price' column. We then split the dataset into training (0.8) and testing (0.2) sets using `train_test_split()`, ensuring stratified sampling to maintain data distribution consistency across both sets. For model training and evaluation, we implemented a structured workflow: initializing each model with default parameters, training them using the `.fit(X_train, y_train)` method, generating predictions via `.predict(X_test)`, and evaluating performance using RMSE and R-squared metrics. The primary challenge was maximizing prediction accuracy, which led us to employ metaheuristic techniques for hyperparameter tuning. For the Random Forest Regressor, we focused on optimizing `n_estimators` and `max_depth`. Increasing `n_estimators` from 100 to 500 enhanced the model's robustness by incorporating more decision trees, while adjusting `max_depth` from 10 to 30 allowed deeper feature interactions. These changes improved the R-squared score from 0.74 to 0.757 and reduced RMSE. In contrast, the Support Vector Regression (SVR) model required a different approach. We tested multiple kernels (linear, rbf, and poly), with the linear kernel outperforming others (R^2 = 0.3668) due to the dataset's predominantly linear relationships. The gamma parameter showed minimal impact on performance, so we retained the default scale value.

B. Macro-level analysis agent

For the macro-level analysis, we implemented both ARIMA and LSTM models to forecast housing price trends in New South Wales. The implementation began by importing essential Python libraries, including statsmodels for ARIMA modeling (specifically SARIMAX, adfuller, ACF and PACF functions), pmdarima for automated ARIMA parameter selection, and TensorFlow/Keras for LSTM implementation (with Sequential, LSTM, and Dense layers). We also utilized sklearn's MinMaxScaler for data normalization and standard data science packages like pandas, numpy, and matplotlib for data manipulation and visualization. For ARIMA implementation, the dataset was split chronologically to maintain temporal integrity, with all quarters before March 2021 serving as training data and the four quarters of 2021 reserved for testing. Firstly, we conducted thorough stationarity analysis by plotting ACF and PACF charts and performing Dickey-Fuller tests (initial p-value > 0.05 confirmed non-stationary data). This necessitated differencing (d=4) to achieve stationarity. After examining OLS regression results to understand underlying trends, we identified optimal ARIMA parameters (1,4,0) through iterative testing and AIC score minimization. A key limitation emerged as the model's reliance on differencing constrained its accuracy to short-term forecasts (1-2 quarters), with predictions aligning well with Q1 2021 test data but diverging in subsequent quarters. For the LSTM implementation, we followed a comprehensive preparation pipeline: seasonal decomposition to identify trends and seasonality, StandardScaler scaling to normalize data, and creation of sliding windows with a 3 quarters lookback period. The data was carefully reshaped into 3D tensors [samples, timesteps, features] to meet LSTM requirements - a crucial step we initially overlooked, leading to model errors until corrected. The neural network architecture consisted of a first layer LSTM layer (50 units) with a 31-quarter input shape, followed by 3 more layers and a Dense output layer, compiled with Adam optimizer and MSE loss function. Training proceeded for 20 epochs with early stopping to prevent overfitting. To enhance both models, we employed metaheuristic optimization techniques. A genetic algorithm is applied to optimise ARIMA by evolving a population of parameter combinations to minimise the fitness score. Moreover, a genetic algorithm automatically determined optimal parameters including the critical 3-quarter window size for LSTM and ideal batch sizes/epoch counts. This approach proved particularly valuable given the complex parameter space and computational demands of LSTM training, which benefited significantly from GPU acceleration. The implementation process revealed several key challenges: ARIMA's sensitivity to non-stationary data required careful differencing, LSTM's strict input dimensionality necessitated precise data reshaping, and both models demanded substantial computational resources for thorough testing and optimization. These challenges were systematically addressed through iterative testing, automated parameter searches, and appropriate hardware utilization, resulting in robust forecasting systems capable of capturing both short-term trends (ARIMA) and longer-term patterns (LSTM).

IV. RESULTS AND DISCUSSION

A. Micro-level analysis agent

Our micro analysis of Sydney's property market revealed several important insights about housing price distribution and key valuation factors. The suburb-level examination showed a clear stratification of prices, with Hornsby and Strathfield emerging as premium markets, while Liverpool represented more affordable options. Notably, Burwood's classification as medium-high tier appears linked to its status as a Chinese commercial hub, demonstrating how cultural factors influence local real estate markets. The Sydney CBD's surprising medium-low ranking likely reflects its higher concentration of apartments versus houses. Our correlation analysis identified several strong determinants of property values, with bedroom and bathroom counts showing the positve relationships (r = 0.27 and 0.34 respectively), while distance from the CBD exhibited a significant negative correlation (r = -0.46). Additional factors like suburb median income (r = 0.35) and property inflation index (r = 0.35) also played meaningful roles in pricing.

The Random Forest Regressor proved particularly effective at modeling these complex relationships, achieving an R^2 score of approxiamtely 0.76 that significantly outperformed both linear regression (R^2 = 0.50) and SVR (R^2 = 0.37). This strong performance stems from the algorithm's ability to capture nonlinear feature interactions and appropriately weight important variables. The model's effectiveness is further evidenced by its capacity to identify potentially mispriced properties and provide actionable insights for different buyer segments. These findings collectively highlight how Sydney's housing market reflects a complex interplay of structural, geographic, and demographic factors, with our Random Forest. The results offer valuable guidance for both individual buyers and urban planners seeking to understand valuation dynamics in one of Australia's most competitive property markets (Table III).

B. *Macro-level analysis agent*

Our examination of Australia's state-level housing market revealed New South Wales consistently maintains the highest property values nationwide, with an observable upward price trajectory over time. To analyze these trends, we implemented two distinct time-series approaches - the classical ARIMA model and the more complex LSTM neural network. The non-stationary nature of the housing price data presented significant forecasting challenges, particularly evident in the requirement for multiple differencing operations to achieve stationarity in the ARIMA framework. This characteristic fundamentally limited reliable predictions to short-term horizons of approximately one quarter, beyond which forecast accuracy degraded substantially.

The ARIMA model demonstrated particular suitability for our analysis given the relatively small dataset size (51 quarterly observations). Its parsimonious structure avoided overfitting risks that plagued the LSTM implementation, where the neural network's capacity for complex pattern recognition paradoxically became a liability with limited training data. While the LSTM architecture theoretically offered advantages for capturing long-term dependencies and nonlinear relationships, in practice it struggled to generalize effectively from our constrained dataset, often producing erratic forecasts that failed to outperform the simpler ARIMA model (Table II).

This outcome highlights an important consideration in housing market analytics: model selection must carefully balance theoretical capability with data availability. For government agencies or research institutions working with similarly sized datasets, our results suggest classical time-series methods may provide more reliable insights than sophisticated deep learning approaches. The ARIMA model's interpretable parameters and modest data requirements make it particularly valuable for policy analysis and short-term market monitoring, though practitioners should remain cognizant of its limitations in volatile market conditions or during economic shocks. Future expansions of this work could revisit LSTM performance with enhanced datasets incorporating additional years of observations or higher-frequency data collection.

V. CONCLUSIONS AND FUTURE WORK

This project developed a dual-model framework for housing price analysis, combining Random Forest for property-level valuation and ARIMA for short-term market forecasting. The Random Forest model demonstrated strong predictive performance with an R^2 score of 0.7572, effectively capturing complex nonlinear relationships between housing features and prices. Meanwhile, ARIMA provided reliable short-term forecasts but was constrained by its linear assumptions, performing best for 1-2 quarter predictions in stable market conditions. The micro-level analysis revealed insightful patterns about Sydney's housing market, showing how factors like bedroom count, CBD proximity, and suburb characteristics influence pricing tiers across different neighborhoods.

A. *Ethical and Technical Considerations*

Several important limitations emerged during implementation. Privacy concerns arose regarding the potential identification of individual transactions in small suburbs through aggregated price data. Technical challenges included data quality issues from missing values and outliers, ARIMA's inability to adapt to sudden market shocks, and LSTM's impractical computational demands for our dataset size. These limitations highlight the careful balance required between model complexity and practical applicability in real estate analytics. The ethical implications of housing price algorithms also warrant ongoing attention, particularly regarding potential biases in automated valuations and the responsible communication of model uncertainties to end-users.

B. *Future Directions*

Several promising avenues exist for extending this research. Hybrid modeling approaches could combine ARIMA's short-term reliability with LSTM's long-term pattern recognition when larger datasets become available. Incorporating external economic indicators through SARIMAX or VAR models may improve forecast robustness. Automated optimization techniques like genetic algorithms could streamline hyperparameter tuning, while differential privacy methods could enhance data security. Future work should also focus on developing more sophisticated bias detection frameworks and creating user-friendly interfaces that clearly communicate model limitations. As housing markets continue evolving, maintaining this balance between technical innovation and practical implementation will remain crucial for developing reliable, ethical valuation tools. The insights from this project provide a foundation for more comprehensive housing market analysis systems that can adapt to changing economic conditions and data availability.

APPENDIX

A. *Visualisations*

B. *Tables*

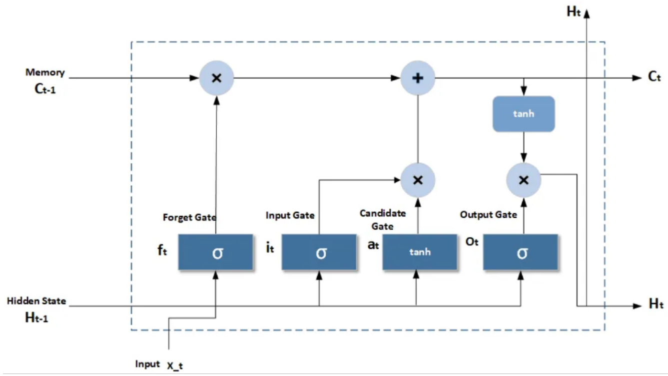

Fig. 1. LSTM Architecture

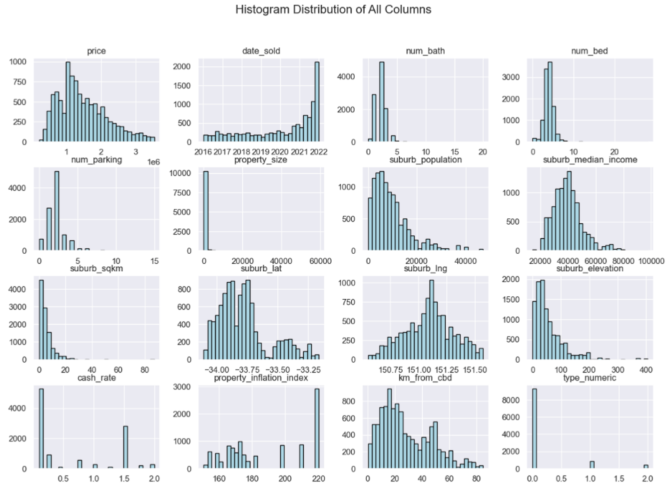

Fig. 2. Df1 - Histogram

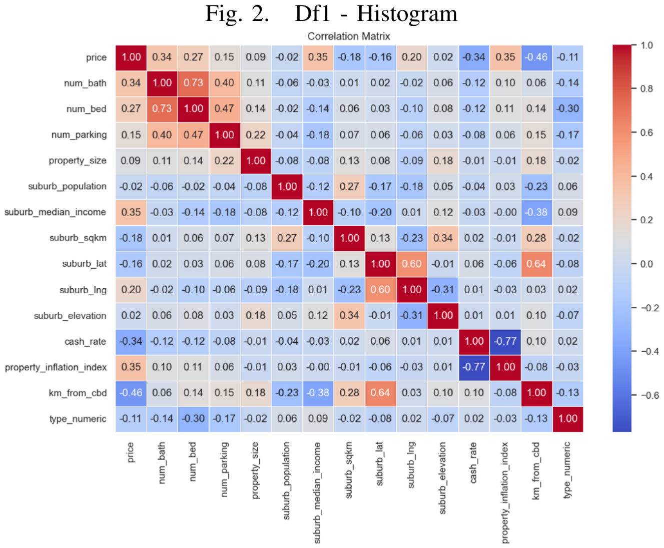

Fig. 3. Df1 - Correlation Matrix

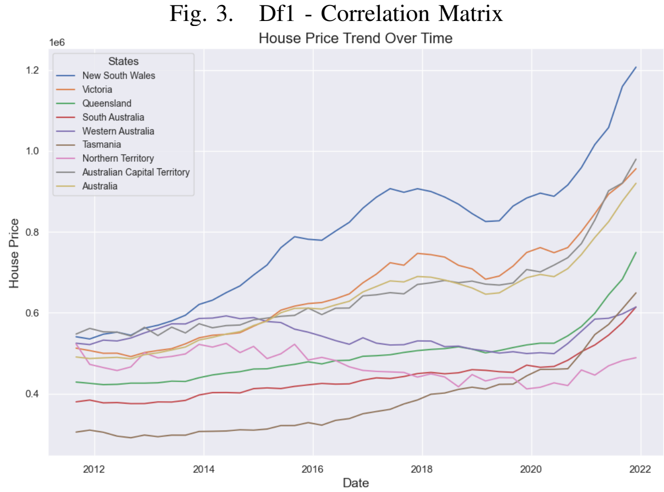

Fig. 4. Df2 - House price change over time

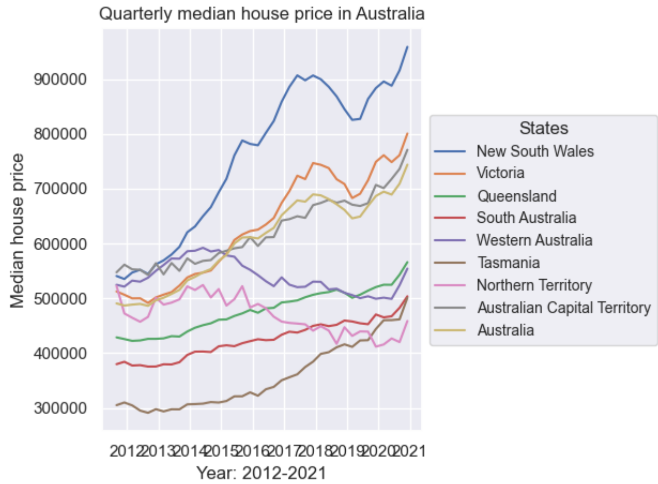

Fig. 5. Df2 - Training data set

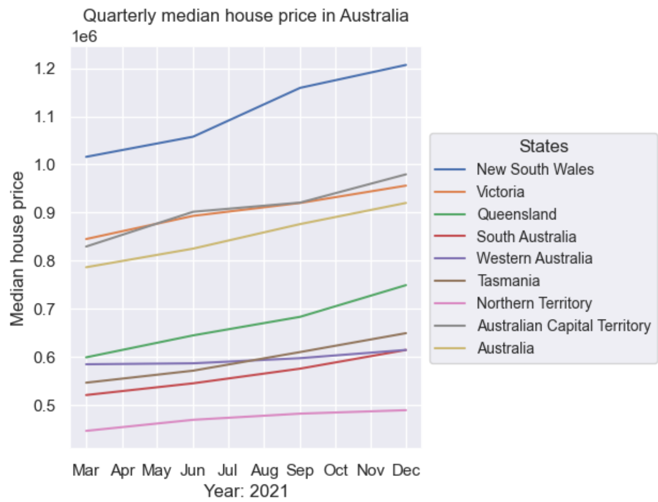

Fig. 6. Df2 - Testing data set

Fig. 7. Df2 - ACF and PACF

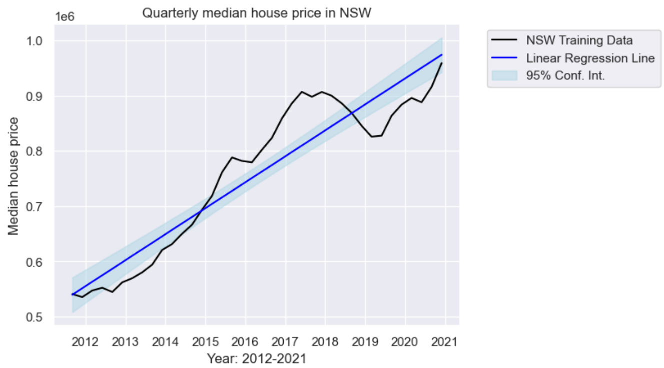

Fig. 8. Df2 - House price trend in training set

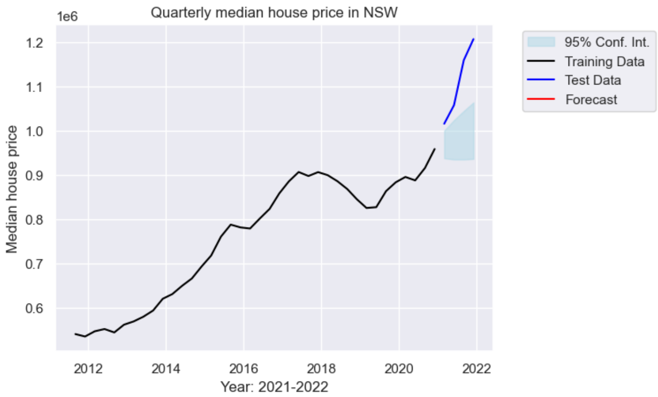

Fig. 9. Df2 - ARIMA_Output1

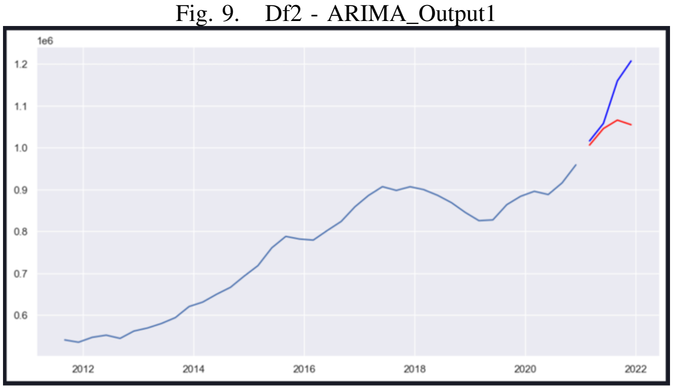

Fig. 10. Df2 - ARIMA_Output2

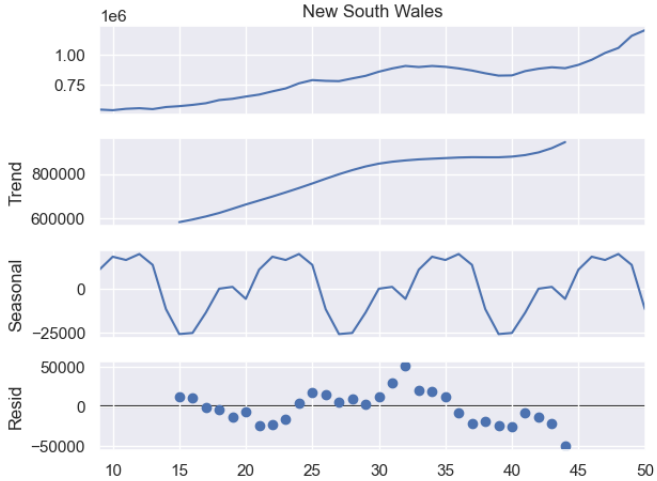

Fig. 11. Df2 - LSTM_Seasonal Decomposition

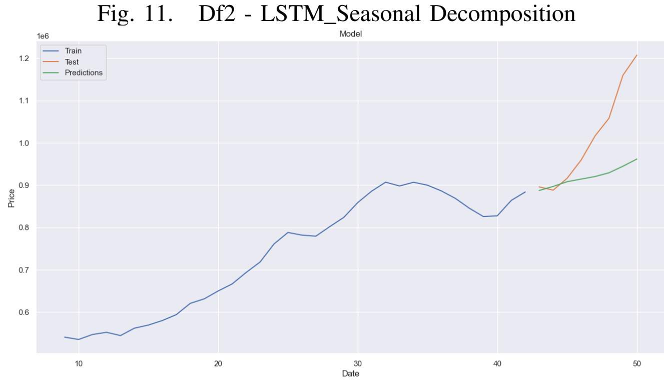

Fig. 12. Df2 - LSTM_Output

```
                               SARIMAX Results
==============================================================================
Dep. Variable:        New South Wales   No. Observations:                   38
Model:                 ARIMA(1, 4, 0)   Log Likelihood                -412.211
Date:                Mon, 05 May 2025   AIC                            828.423
Time:                        02:12:52   BIC                            831.475
Sample:                             0   HQIC                           829.464
                                 - 38
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.1348      0.026     -5.176      0.000      -0.186      -0.084
sigma2      9.461e+08   2.51e-13   3.77e+21      0.000    9.46e+08    9.46e+08
===================================================================================
Ljung-Box (L1) (Q):                  15.75   Jarque-Bera (JB):                 1.46
Prob(Q):                              0.00   Prob(JB):                         0.48
Heteroskedasticity (H):               0.99   Skew:                            -0.10
Prob(H) (two-sided):                  0.99   Kurtosis:                         2.00
===================================================================================
```

Fig. 13. Df2 - ARIMA_Before applying Genetic Algorithm

```
                               SARIMAX Results
==============================================================================
Dep. Variable:        New South Wales   No. Observations:                   38
Model:                 ARIMA(2, 2, 2)   Log Likelihood                -398.405
Date:                Mon, 05 May 2025   AIC                            806.810
Time:                        11:20:00   BIC                            814.728
Sample:                             0   HQIC                           809.574
                                 - 38
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.1331      0.140     -0.949      0.343      -0.408       0.142
ar.L2          0.7732      0.146      5.293      0.000       0.487       1.059
ma.L1         -0.0195      0.157     -0.124      0.901      -0.328       0.289
ma.L2         -0.9343      0.189     -4.933      0.000      -1.306      -0.563
sigma2      2.773e+08   3.48e-10   7.97e+17      0.000    2.77e+08    2.77e+08
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):                 0.17
Prob(Q):                              0.99   Prob(JB):                         0.92
Heteroskedasticity (H):               3.33   Skew:                             0.15
Prob(H) (two-sided):                  0.05   Kurtosis:                         3.15
===================================================================================
```

Fig. 14. Df2 - ARIMA_After applying Genetic Algorithm

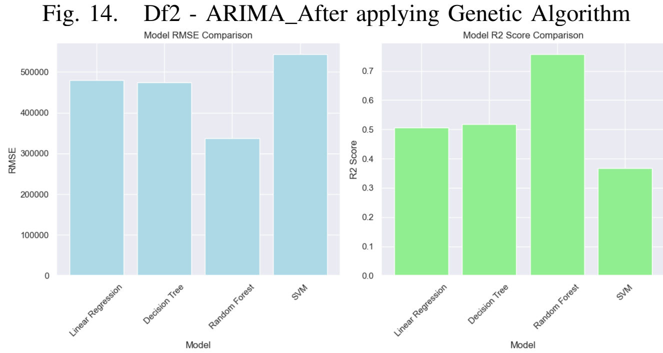

Fig. 15. Df1 - ML models performance comparison

TABLE I

COMPARISON OF MACHINE LEARNING MODELS

Model	**Strengths**	**Weaknesses**
Linear Regression	• Interpretability – Clear coefficient analysis • Computationally efficient – Fast training/prediction • Baseline utility – Useful for benchmarking complex models	• Underfitting – Assumes linear relationships, struggled with nonlinear patterns in housing data • Feature sensitivity – Performance dropped with weakly correlated features
Random Forest Regressor	• High accuracy – Handled nonlinearities and interactions • Feature importance – Identified key predictors • Robustness – Resistant to outliers and overfitting (via ensemble averaging)	• Black-box nature – Harder to explain than LR • Computational cost – Slower training with large n_estimators • Memory-intensive – Stored all trees, unlike SVR
Support Vector Regression (SVR)	• Performance – Effective in high-dimensional spaces and with small datasets • Flexibility – Handles nonlinear data using kernel functions • Scalability – Reliable performance on small to medium sized dataset	• Performance – Computationally expensive for large datasets • Flexibility – Requires careful parameter tuning (kernel type, regularization) • Scalability – Poor scalability for large datasets due to time and memory constraints

TABLE II

COMPARISON OF TIME SERIES FORECASTING MODELS

Model	Strengths	Weaknesses
ARIMA	• Interpretable – Clear parameters (p,d,q) explain trends, seasonality, and noise • Fast & Lightweight – Efficient on small-to-medium datasets and can be automated via `auto_arima` (pmdarima library) • Statistical Rigor – Provides confidence intervals for forecasts and tests for stationarity (ADF Test) and residual patterns (Ljung-Box test)	• Linear Assumption – Fails to capture sudden market shifts • Manual differencing (d) required for non-stationary data • Short-Term Focus – Forecasts degrade quickly beyond $\sim$10 time steps
LSTM	• Nonlinear Pattern Capture – Learns complex trends (e.g., seasonal spikes, multi-year cycles) and handles volatility better than ARIMA • Long-Term Dependencies – Memory cells retain context over extended periods	• Data-Hungry – Requires 10k+ samples for reliable training • Computational Cost – Needs GPUs for timely training; hyperparameter tuning is complex • Black-Box Nature – Hard to diagnose why a forecast was made

TABLE III

MODEL PERFORMANCE COMPARISON

Model	RMSE	R2 Score
Linear Regression	479850.013193	0.506355
Random Forest	336504.649229	0.757235
SVM	543419.249552	0.366897

REFERENCES

[1] A. Chaurasia and I. U. Haq, "Housing price prediction model using machine learning," in *2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET)*, 2023, pp. 497–500.

[2] A. P. Singh, K. Rastogi, and S. Rajpoot, "House price prediction using machine learning," in *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, 2021, pp. 203–206.

[3] T. R. N and R. Gupta, "A survey on machine learning approaches and its techniques:," in *2020 IEEE International Students' Conference on Electrical,Electronics and Computer Science (SCEECS)*, 2020, pp. 1–6.

[4] A. Sharma, A. Kaur, and A. Semwal, "Supervised and unsupervised prediction application of machine learning," in *2022 International Conference on Cyber Resilience (ICCR)*, 2022, pp. 1–5.

[5] V. Gupta, V. K. Mishra, P. Singhal, and A. Kumar, "An overview of supervised machine learning algorithm," in *2022 11th International Conference on System Modeling Advancement in Research Trends (SMART)*, 2022, pp. 87–92.