IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.19, No.4, October 2025, pp. 349~360
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: https://doi.org/10.22146/ijccs.103967

Predicting Resale Prices using Random Forests with Fine-Tuning Hyperparameters

Herman Widjaja[1], Nanda Perdana[2], Ito Wasito[3]
[1, 2, 3] Department of Information Technology, Pradita University, Scientia Business Park, Banten, Indonesia
e-mail: ***[1] herman.widjaja@student.pradita.ac.id** , [2]nanda.perdana@student.pradita.ac.id , [3]ito.wasito@pradita.ac.id

Abstrak

Prediksi harga properti yang akurat sangat penting bagi pembeli, penjual, dan pembuat kebijakan untuk membuat keputusan yang tepat di pasar real estat yang dinamis. Penelitian ini mengkaji penerapan model pembelajaran mesin—Random Forest, XGBoost, Decision Tree, dan LightGBM—untuk memprediksi harga jual kembali flat di Singapura. Penelitian ini memberikan wawasan berharga mengenai penggunaan pembelajaran mesin di pasar perumahan, terutama untuk dataset dengan ukuran, kompleksitas, dan jenis data yang serupa. Tujuan penelitian ini adalah mengembangkan model regresi prediktif untuk harga properti serta menganalisis dan membandingkan kinerja model-model tersebut. Kontribusi utamanya mencakup pengembangan alat untuk memperkirakan harga properti secara objektif dan peningkatan penelitian prediksi harga melalui perbandingan menyeluruh model pembelajaran mesin. Meskipun studi sebelumnya telah menunjukkan kemampuan prediktif model-model ini, penelitian ini berfokus pada dampak tuning hyperparameter terhadap kinerja model Random Forest. Dengan mengoptimalkan parameter seperti max_depth, n_estimators, dan n_jobs secara sistematis, waktu komputasi berhasil dikurangi lebih dari 93% (dari 865 detik menjadi 50 detik) dengan penurunan akurasi yang minimal. Melalui tuning hyperparameter yang tepat, Random Forest mencapai kinerja terbaik berdasarkan nilai MAE (26.555), mengungguli XGBoost (27.552), Decision Tree (28.832), dan LightGBM (29.752).

Kata kunci*— Random Forest, Hyperparameter Tuning, Prediksi Harga Perumahan, XGBoost, Decision Tree*

Abstract

The accurate prediction of housing prices is essential for informed decision-making by purchasers, sellers, and policymakers in dynamic real estate markets. This study investigates the application of machine learning models—Random Forest, XGBoost, Decision Tree, and LightGBM—to predict resale flat prices in Singapore. It provides valuable insights into the use of machine learning in housing markets, particularly for datasets with similar size, complexity, and data types. The objectives are to develop predictive regression models for property prices and to analyze and compare the performance of these models. Key contributions include the development of tools to objectively estimate suitable property prices and the advancement of price prediction research through an extensive comparison of machine learning models. While previous studies have demonstrated the predictive capabilities of these models, this research focuses on the impact of hyperparameter tuning on the performance of the Random Forest model. By systematically optimizing parameters such as max_depth, n_estimators, and n_jobs, computation time was reduced by over 93% (from 865 seconds to 50 seconds) with minimal loss in accuracy. With proper hyperparameter tuning, Random Forest achieved the best performance in terms of MAE score (26.555), outperforming XGBoost (27.552), Decision Tree (28.832), and LightGBM (29.752).

Keywords—*Random Forest, Hyperparameter Tuning, Housing Price Prediction, XGBoost, Decision Tree*

Received January 17th,2025; Revised July 27th, 2025; Accepted August 21th, 2025

1. INTRODUCTION

The real estate market plays a vital role in economy. Creating an accurate property price prediction is essential for homepurchasers, sellers, and policymakers. In Singapore, the resale flat market has gathered attention due to its unique characteristics within a highly developed and regulated housing system. Although the terminology used is 'resale', however most of ownership titles of the samples observed (HDB flats) in this paper are 99-years lease hold. Practically, it is transferable of the leasing right. [1] Typically, during transactions' processes, sellers tend to overvalue their asking prices while purchasers undervalue offers, creating challenges in closing transactions. This gap highlights the need for an objective tool to determine fair asking and offering prices.

Machine learning algorithms have developed as promising tools for property price prediction, addressing the limitations of traditional hedonic pricing models. Techniques such as decision trees, regression analysis, and neural networks have shown potential for achieving high accuracy. [2] However, the effectiveness of these machine learning models in the framework of Singapore's resale flat market remains a subject of further research, offering opportunities to refine and enhance predictive capabilities tailored to this market.

This paper has two contributions, firstly, creating a gauging tools for purchaser / seller / policymakers of a suitable price for a property with certain features. So that purchaser and seller has more objective of an anticipated transacted price. Secondly, this study aims to contribute to the growing body of research on real estate price prediction by conducting a wide-ranging comparison among various models of machine learning.

The primary goals of this study are namely to create model(s) of the price predictive regression. Secondly, to analyze and compare the performance of various machine learning models, namely: Decisision Tree, Random Forest (with tuning hyperparameters), XGBoost and LightGBM.

2. METHODS

2.1 Research Framework

In this research, the framework below (Figure 1) are inspired by other previous frameworks [3] and [4]

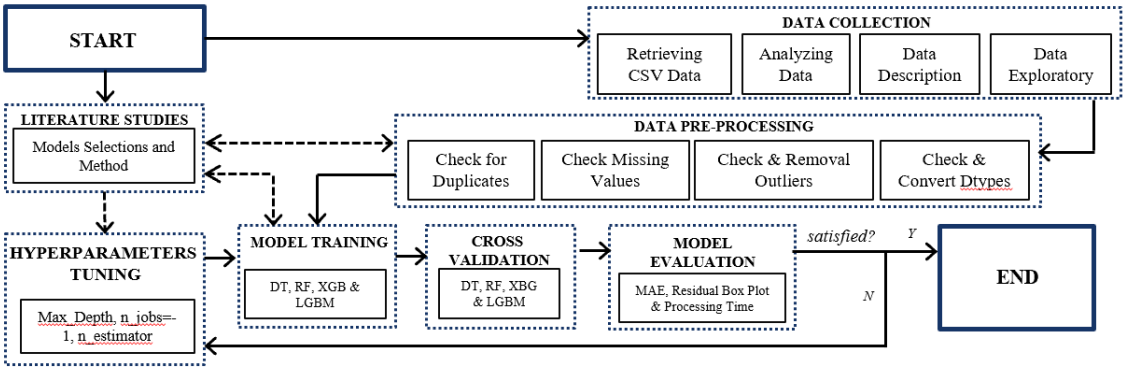

Figure 1 Research Framework

The research / experiment starts from Data Collection, includes retrieving dataset, analyzing dataset, description and exploratory. After the collection, the dataset preprocessed by checking duplicates, missing values, outliers removal and converting data types.

After the dataset ready, the preprocessed data run in Model Training of all models (Decision Tree, Random Forest, XGBoost and LightGBM. The result will be cross checked by 10 folds cross validation. Both results then evaluated, especially the MAE and processing time. Later, when the result not satisfied, the model then reiterated by adjusting / tuning the hyperparameters (max_depth, n_estimator and n_jobs). And the model rerun again. Otherwise, when the model satisfied, the experiment stopped.

2. 2 Data Description and Exploratory Data Analysis

The dataset was acquired from Singapore Government of all transactions of Resale Flat Prices based on recording date starting 2017 January until October 2024.[5] The dataset consists of 191,667 entries with total 11 columns (Table 1.). The scatter plot shows the distribution of units Flat Type and theirs Area (sqm), shows that in general that more population in the 3 Room Type and the Area Range is quite wide (startsfrom less than 30 m2 up to 350 m2).

Table 1 Dataset Columns and Description

#	Column	Description
0	month	Month of registered transaction
1	town	Designated area of with its own characteristics (features, infrastructure and community facilities)
2	flat_type	Units classification by number of rooms of unit. It ranges from two to five (bed) rooms, 3Gen units, and Executive units.
3	block	Classification of units by room size. They range from 2 to 5 rooms, 3Gen units, and Executive units.
4	street_name	Name of the HDB (Housing & Development Board) building comprising multiple flats / apartments
5	storey_ range	The location of unit sold within the building / tower (estimated range of floors)
6	floor_area_ sqm	Total unit's interior floor, measured in m^2
7	flat_model	Units classification by generation of which the flat was made. It ranges from New Generation, DBSS, Improved or Apartment
8	lease_ commence_date	The commencement date of the lease agreement, marking the starting of the term of lease during which the tenant (owner) has the right to use and occupy the property
9	remaining_ lease	Remaining time available on the lease.
10	resale_price	Price of the transacted flat unit

In Figure 2, we can see the histogram of frequency distribution of Area (sqm), it shows that the histogram not normally distributed. Unit size $\pm$ 60 m2 has broken the pattern of the histogram. Meanwhile peak of the frequency occurs at area approximately 90 m2. In Figure 3., the graph has a long tail to the right, indicating that while the majority of resale prices are concentrated on the lower end (around S$ 300,000-600,000), there are a few homes with much higher prices that extend up to S$ 1.6 million.

The distribution in Figure 4 is bimodal, with two noticeable peaks. The highest peak is around 90 years remaining, followed by 60-70 years. The most common lease remaining is around 90 years (over 35,000 properties) followed by lease range between 60-70 years. Figure 5. shows there are significant peaks around the late 1980s, late 1990s, and around early 2010. Each of these periods likely reflects times when a larger number of properties had their leases start. There was a sharp rise in lease commencements, peaking around 2010, followed by a slight decline closer to 2020. The histogram highlights three main waves of lease commencements: the late 1970s, late 1990s to early 2000s, and the early 2010s. These peaks likely reflect periods of increased property development or government housing projects.

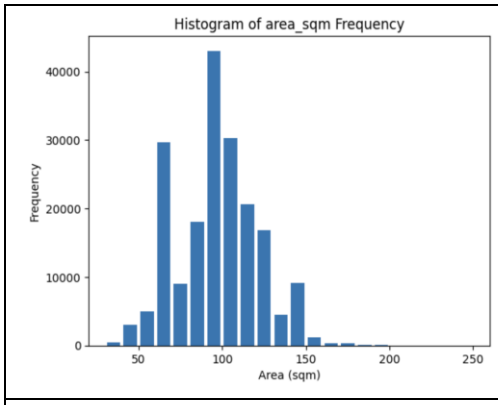

Figure 2 Histogram of area_sqm Frequency

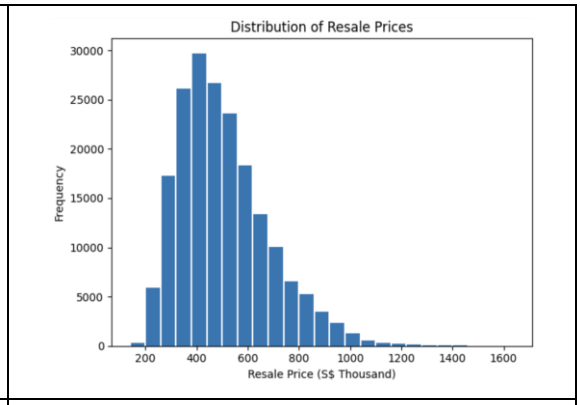

Figure 3 Distribution of Resale Prices Frequency

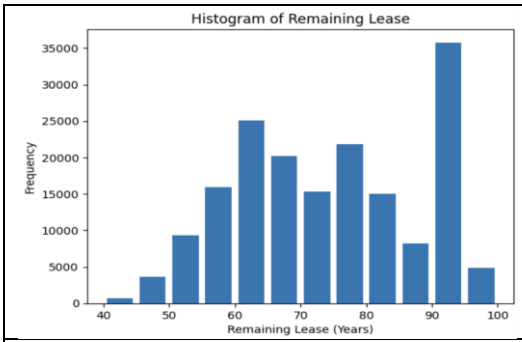

Figure 4 Histogram of area_sqm Frequency

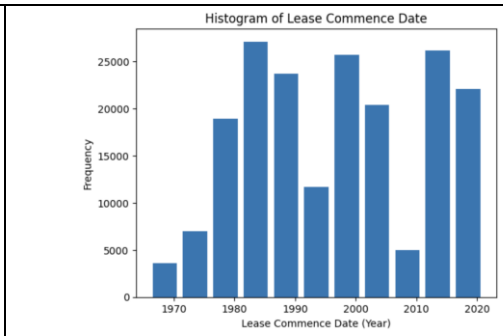

Figure 5 Distribution of Resale Prices Frequency

2. 3 Data Pre-Processing

In the pre-processing, the experiment drop the column block since it is not relevant to the study. Then, the experiment also removes outlier (one unit has size of 350 m2, and is seperated from the remaining, see Figure 1). Later, the experiment convert storey_range to numeric, binning the column month (semester_period), convert to categorical for several columns (‘town', 'street_name', 'flat_model’). One-hot decoding is used to convert these colums. At the end of pre-processing, the dataset shape is 191,666 with 10 columns.

2. 4. Models

Machine learning (ML) models in general will perform superior than statistical and econometric models. Furthermore, ensemble ML models have been proven in many literatures to perform even better performance than single ML models.[6]

Real estate predicting model / regression has long become an interesting topic of researches. Several models / algorithms of machine learning and deep learning have been developed. Some of the algorithms frequently used in these typical cases are: artificial neural network [2] , [7] , [8] , random forest [9] , [3] , [10], k nearest neighbor [2], [11], XGBoost [7] , [9] , [12]. In recent researches, XGBoost and Random Forest Regression are several times compared directly / head-to-head [9] [13] [14] [15] [16]

Paper [13] evaluates two machine learning algorithms, XGBoost and Random Forest. It is focusing on key features such as overfitting, hyperparameter tuning, the impact of leaf nodes, handling missing values, and their outcome in classification and regression tasks. The study concludes that XGBoost outperforms Random Forest (accuracy, recall, and F1-score) and matches it in precision.

A study evaluates housing price prediction models, including Random Forest (RF), XGBoost (XGB), LightGBM (LGBM), Stacked Generalization Regression and Hybrid Regression. Random Forest shows the lowest training error but is inclined to to overfitting & has

high complexity in processing time. XGBoost and LightGBM perform well in accuracy, with LightGBM excelling in efficiency. The study uses the "Housing Price in Beijing" dataset (300,000 records from 2009–2018 with 26 variables). [9]

Research [14] evaluates machine learning algorithms using large datasets with diverse property attributes. While Linear Regression provides simplicity and interpretability, XGBoost consistently outperforms Random Forest in prediction accuracy based on Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). Paper [15] compares models using metrics like R^2, RMSE, and MAPE, along with training and inference times. XGBoost outperforms Random Forest on both small and large datasets, achieving similar accuracy with significantly shorter training (2–50 times faster) and inference times (up to 40 times faster). This study utilizes a dataset of over 30,000 housing records from India with 12 features. Paper [16] analyzes 62,723 housing records from 2015–2019 and compares machine learning models for regression performance. The study concludes that XGBoost outperforms models like Decision Trees, Random Forest, SVR, and CatBoost in accuracy and computational efficiency.

Based on the above relevant literature studies, this study considers the selected as the most widely used models to process the dataset: DT, RF, XGB and LGBM.

2.5 Decision Tree, Random Forest, XGBoost and LightGBM

Decision Tree (DT) is easy to identify with its tree shape structure. Every node represents assessment on a characteristic and the branches out of the node represents the assessment outcomes. DT formation does not need any basic domain knowledge. DT can process dataset with high dimensions too. To avoid meaningless and unwanted rules in DT, the depth of tree should be tunned to avoid [17]. DT can describe intricate, non-linear relationship and interaction between features for the purpose of predicting housing prices. However, compared to linear techniques, their models may be harder to understand due to their propensity for overfitting, particularly with limited datasets [14].

Random Forest (RF) is a robust algorithm for both classifier and regressor. It uses bagging, combining multiple decision trees trained independently on separate bootstrap samples to produce more reliable predictions than any single tree. [2] [18]

RF constructs decision trees ensemble on bootstrapped examples of the training data and uses this to make predictions about new occurrences. [19][20] As the number of trees increases, the generalization error of the forest approaches a stable limit. Some strengths of RF are namely: better accuracy, robust to outliers and noise, faster than bagging or boosting, provides correlation, useful internal estimates of error and features importance and relatively simple. [13] [18]

Gradient Boosting Machines (GBMs) are advanced ensemble methods that iteratively build models to correct errors made by previous models. GBM algorithms include XGBoost (XGB), LightGBM (LGBM), and CatBoost. These methods are characterized by their high predictive accuracy, ability to capture non-linear relationships, and effective handling of feature interactions. Unlike RF, which train trees in parallel on bootstrap exsamples of the initial dataset, GBMs train trees sequentially, with each iteration aimed at improving the performance of the preceding model. [19] Compared to RF, XGB relies on a smaller set of initial parameters. It employs a flexible approach that aligns closely with the inherent characteristics of the data, iteratively adjusting leaf nodes to enhance model predictions [13]. Some researchers suggest that XGB is one of the most powerful algorithms in tackling regression & classification cases. [21]

The Light Gradient Boosting Machine (LGBM) was initially introduced in late 2017, followed by a steady release in 2020 (November). It has been widely adopted by researchers in fields such as medicine and biochemistry. However, its application in the real estate market remains relatively uncommon. [21] LGBM is among the gradient-boosted tree algorithms that have become more accessible and computationally efficient in recent years. [19]

Some studies indicate that LGBM can surpass traditional algorithms like Semi-Global Matching and XGB in memory efficiency and computational speed. Additionally, LGBM offers several advantages, including high accuracy, the ability to handle large-scale datasets, GPU

acceleration, parallel processing, and support for distributed computing. Other research suggests that LGBM can also outperform algorithms such as Random Forest Regressor and Logistic Regression in terms of accuracy. [20] [22] [23]

2.6 K-fold Cross Validation

According to Mohan [14], K-fold Cross Validation is a practical method for assessing how well a model performs. In this method, the dataset is separated into random K parts evenly, or folds. The model is trained for K times, with K minus 1 folds used for training and the remaining fold used for validation each time. This process helps to determine how well the model can take a broad view to new data. K-fold Cross Validation is also helpful for finding a good equilibrium for both bias and variance [9] [15]. Common choices for K include five, ten, or fifteen folds, but other values can also be used [24]. K-fold cross validation helps determine whether such model is accurate and reduces the risk of overfitting or underfitting [25].

2.7 Mean Absolute Error (MAE)

This study use MAE (Mean Absolute Error) as the measurement of deviation. A lower MAE indicates a more accurate prediction of the actual results by the model. Variable y_i is the predicted result for variable i, $\hat{y}_i$ is the actual result and n is the total number of results for that variable. [23]

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1}$$

3. RESULTS AND DISCUSSION

3.1. Feature Importance

From the training, the study provides highest three (3) features importance (Table 2) of each models, and it shows interesting result. While DT and RF have similar top 3, including their rank (area_sqm, semester_period, story_range), XGB has one different feature (flat_type), while the other two are similar to DT and RF (area_sqm and semester_period).

Table 2 Top 3 Features Importance of All Models

FEATURES	area_sqm	semester_period	storey_range	flat_type	town	street name	LCD
DT	#1	#2	#3				
RF	#1	#2	#3				
XGB	#1	#3		#2			
LGBM					#1	#2	#3

The suprising result comes from LGBM whose features importance are totally different from the other models. LGBM creates town, street_name and LCD as Top 3 features importance. LGBM puts more importance to the locational features. Feature town and street_name refer to the location of the properties / flat buildings.

3.2. Residual Box Plot

From the residual box plot analysis, the experiment gets DT as the benchmark for Median (0.00) followed by RF (-0.76), XGB (-1.75) and the biggest is LGBM (-2.12). Meanwhile for the Inter Quartile Range (IQR), the smallest is RF (37.55), followed by DT (40.00), XGB (41.82) and LGBM (45.05). And for the min-max range, the smallest again achieved by RF (150.21), DT

(160.00), XGB (167.28) and LGBM (180.20). Lastly, for outliers range, XGB is the most insensitive to outliers, has the narrowest range of outliers, followed by LGBM, RF and DT.

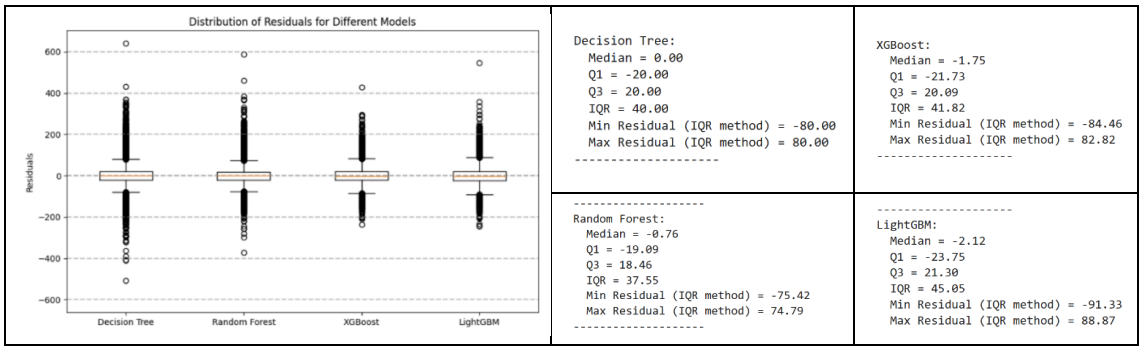

Figure 6 Distribution of Residuals and Result of Median, IQR, Min / Max of Models

3.3 Distribution of Percentage Error

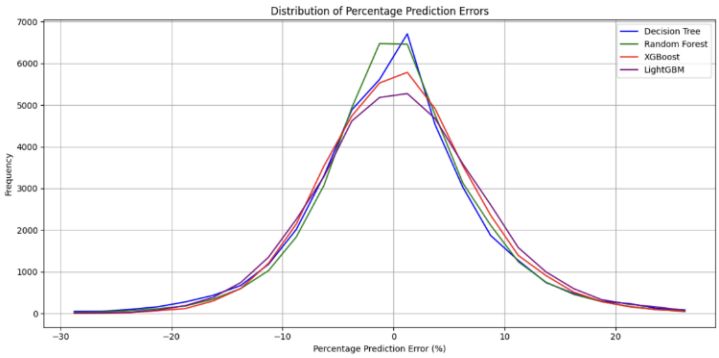

Figure 7 Distribution of Percentage Error of Models

The Figure 7 shows that RF has relatively more accurate prediction around 0. DT shows slightly higher score for error range 1%-2%. XGBoost comes in the third place, and yet again LGBM comes the last in this metric / comparison.

3.4 MAE, Processing Time and K-fold Cross Validation

In Table 3, based on MAE score, Random Forest performs the best with 26.55 followed by XGB (27.55), DT (28.83) and lastly LGBM (29.75). And follows the intuition, the processing time is exactly comes in opposite order. The lower the MAE, the longer the processing time. The fastest processing time is LGBM (8.31 s), followed by DT (15.7 s), XGB (31.14 s) and lastly is RF (58.45 s).

When the model of RF runs in the default hyperparameters (no max_depth, n_jobs=none, n_estimators=100), the processing time of RF is 865.55 seconds, more than 30 times slower than the second slowest (XGB, 27.949 seconds).

Table 3 MAE, Processing Time and K-fold Cross Validation Average

Model	MAE	Time (s)	MAE (K-fold)	Time (s) (K-fold average)
Decision Tree	28.832	16.169	31.517	8.727
Random Forest	26.555	50.830	26.594	41.666
XGBoost	27.552	27.949	34.293	11.033
LightGBM	29.752	7.751	36.420	2.967
Random Forest *	22.662	865.588	24.625	554.475

** = RF before hyperparameter tuning*

The study reviews some combinations of Hyperparameters (Table 4). When the n_estimator reduced from default (100) to 50, the processing time reduced to less than half (while maintaining the n_max to 25 / 30 / 35). Later, the n_estimator reduced to 25 and the processing time decreased ± 50% from n_estimator =50. Meanwhile the RMSE, MAE and R^2 only slightly worse. Sequentiallty when the n_estimator reduced to 10, the time decrease more than 90% than the default 865 seconds. Later, the experiment proposes n_estimator = 8 and max_depth = 30 to be compared with other models' metrics.

Altough the metrics are getting better when n_estimaotor reduced (and max_depth increased), the study try to avoid possible issues with accuracy and overfitting. And proposed the estimator of 8 as the comparable result against other models. In scikit-learn's RandomForestRegressor, the n_jobs parameter specifies the number of CPU cores utilized during model training and prediction. By default, n_jobs=None, which equates to single-core execution. Setting n_jobs=-1 enables the use of all available processors, facilitating parallel computation and potentially reducing execution time.[26]

If we compare the average result of K-fold, and MAE without K-fold cross validation, the rank between XGBoost and Decision Tree are switched. Meanwhile, the rank of processing time (average) in the cross validation step is remain the same. In the K-fold average MAE, DT has second best after RF, however, the processing time is also maintained at the second place.

Table 4 RF Hyperparameter Tuning and Processing Time

	#	Max Depth	RMSE	MAE	R^2	Processing Time (s)
# n_jobs=-1, estimator = 50	# 1	25	41.792468	29.146014	0.942619	309.791671
	# 2	30	37.316724	25.741113	0.954251	328.535040
	# 3	35	35.367517	24.197670	0.958906	348.437610
# n_jobs=-1, estimator = 25	# 1	25	41.874745	29.202638	0.942393	151.495807
	# 2	30	37.481803	25.860794	0.953846	160.962797
	# 3	35	35.576861	24.331527	0.958418	181.486496
# n_jobs=-1, estimator = 10	# 1	30	38.226176	26.376433	0.951994	66.021771
	# 2	35	36.402882	24.920965	0.956465	70.090555
	# 3	40	35.457884	24.144854	0.958696	71.658146
# n_jobs=-1, estimator = 8	# 1	25	42.864264	29.895943	0.939638	50.991992
	# 2	**30**	**38.465465**	**26.554824**	**0.951391**	**54.299659**
	# 3	35	36.653803	25.098185	0.955862	54.837746
# n_jobs=-1, estimator = 7	# 1	30	38.626795	26.655391	0.950983	47.655659
	# 2	35	36.846169	25.244985	0.955398	48.035209
	# 3	40	36.002295	24.545135	0.957417	49.389459
# n_jobs=-1, estimator = 5	# 1	50	36.419008	24.613192	0.956426	39.982511
	# 2	60	36.454829	24.651426	0.956340	38.632446
	# 3	70	36.459102	24.632764	0.956330	40.984387

Although showing different time counting, the order of computational speed / processing time of the four models are relatively consistent between the initial testing and K-fold cross validation average. The fastest is LGBM, DT, XGB and Random Forest as the latest.

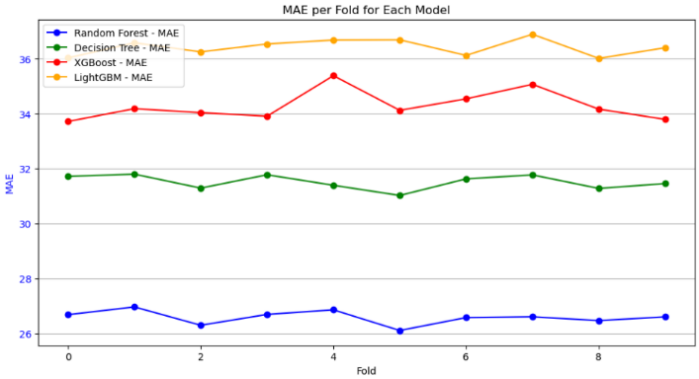

Figure 8 Graph of MAE per Fold of Each Model

The MAE scores among models show different order of result between before and after cross validation. Before crosss validation, RF performs the best MAE, followed by XGB, DT and LGBM. However, during cross validation, the order slightly changed. The smallest and highest

MAE still held by both RF and LGBM. Meanwhile the 2nd and 3rd place changed position between XGB and DT. XGB performs better MAE before cross validation (27.552 and 28.832), meanwhile DT performs better MAE in cross validation (31.517 and 34.293). The above graph (Figure 8.) shows relatively consistency of MAE per fod for each model. Slightly increased MAE occurs in 4th fold of XGB.

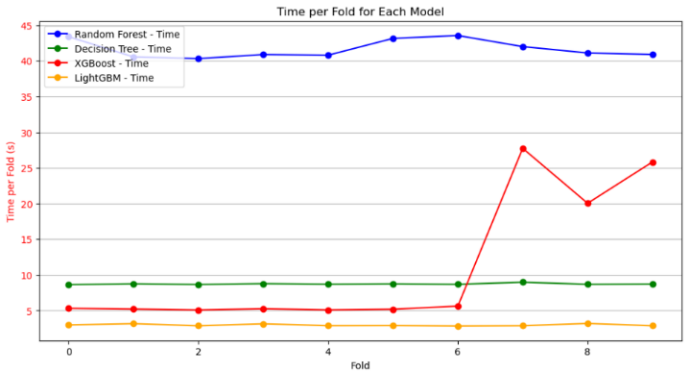

Figure 9 Graph of Processing Time per Fold of Each Model

Average time in 10-fold Cross Validation are ranging between 25% slower (Random Forest) up to 270% slower (XGBoost) than the initial metrics(3rd column times and 5th column-time K-fold average). The order of processing time also consistent before and after K-fold validation. The fastest to the slowest are: LGBM, DT, XGB and RF. Unlike the MAE which shows relatively consistent result, the processing time per fold of XGB shows anomalies at fold 7th – 8th and 9th. There are significant spike. Possibilities of this result may be: data skew / imbalance, data outliers / anomalies, features importance variabilities or else. This finding needs more exploration. However, the study limits this exploration to keep the focus to the initial goals.

3.5 Hyperparameters Tuning

Table 5 Pseudocode of Random Forest Model

```
# 1. Load the dataset
# Load dataset from CSV file into a DataFrame (df)
# 2. Preprocessing: one-hot encoding for categorical features
   # Define categorical columns: town, street_name, flat_type, flat_model, semester_period
   # Create OneHotEncoder with sparse=False and handle_unknown='ignore'
   # Fit and transform the encoder on the categorical columns
   # Create a DataFrame from the encoded data
   # Drop original categorical columns from the main DataFrame
   # Concatenate encoded DataFrame with the main DataFrame
# 3. Prepare the data
   # Define X (features) by dropping 'price_S$thousand' column from DataFrame
   # Define y (target) as the 'price_S$thousand' column
# 4. Split data into training & testing sets
   # Use train_test_split with test_size=0.2 and random_state=42
# 5. Create and train Random Forest model
   # Create RandomForestRegressor with adjusted hyperparameters:
      # n_estimators = 8
      # n_jobs = -1 (use all available cores)
      # random_state = 42
      # max_depth = 30
   # Fit the model to the training data (X_train, y_train)
# 6. Make predictions
   # Predict on the test data (X_test) using the trained model
# 7. (Optional) Evaluate the model
   # Calculate performance metrics (RMSE, Mean Absolute Error, R-squared)
   # Print or visualize the results
```

For particular Random Forest (RF), initial run of the model shows significant processing time 865 seconds. The research tunes the hyperparameters of max_dept from default unlimited (none) to only 30, the n_jobs = -1 and n_estimator from 100 become 8. Although there is an increase of MAE score from 22.66 to 26.55, the processing time reduced significantly from 865 seconds to 58 seconds (more than 93% time saving). This solution not fully similar as suggested by [9] that setting the max_depth as solution to limit the overall processing (training) time. The pseudocode of RF model with fine tuning will be as described in Table 5.

Eventhough hyperparameters tunning also applicable for DT (max_depth, min_samples_split, min_samples_leaf, max_features, min_impurity_decrease), XGB (n_estimators, learning_rate, max_depth, min_child_weight, gamma, subsample, colsample_bytree, reg_alpha and reg_lambda) and LGBM (n_estimators, learning_rate, num_leaves, max_depth, min_child_samples, subsample, colsample_bytree, reg_alpha and reg_lambda), this study only processes the hyperparamters tunning for RF. The reason is, in

the initial round of training, RF perform the best MAE among others. Thus, adjusting the hyperparameters, is the second round step trying to overcome the less desirable performance of its time processing.

3.6 Discusion

Unlike previous research, [4], [13], [15], [16], [19], [21], [22], [23], [27], these research shows that Random Forest is more superior model compare to RF, LGBM, DT, for their respective dataset. Although the processing time is slightly slower, with fine tuning hyperparameters, this study result can provide the best MAE with slightly slower processing time. The time consuming process of RF is also supported by previous researches. In this research [23] LGBM needs 13 seconds, while XGBoost 296 seconds and RF needs 2,046 seconds (more than 100 times of LGBM). In another research [15] , eventough the MAPE score is smaller for RF than XGB, the training time is more than 50 times longer. Paper [9] also adressed the issue of RF processing time. Showing different results than previous research [9] and [13], among hyperparameters, *n_estimators* provide more significant time saving compared to *max_depth*.

In modern computational processes (research or applications), training of large-scale dataset of ML has become a decisive factor. The computational processes, require demands resource requirement, that mostly leads to costs and / or training time consumed. Therefore, strategies and techniques to increase the efficiency of large-scale training models (including reduced processing time), have been developed.[28]

The performance of RF in this research is better than XGB and other models. This result supported by less literatures. Only [9] shows that RF superior compare to XGB and LGBM especially in RMSLE (Root Mean Squared Logarithmic Error) metric. Meanwhile, results in [15] says RF is better in MAPE (Mean Absolute Percentage Error) score in India dataset, but inferior in R2, RMSE and training time for both India and Japan datasets. MAPE score is similar to MAE, the difference is MAPE uses Percentage Error (predicted divided per actual value in percentage), while MAE calculate the actual error value (difference between predicted value and actual value).

First limitation of this study are for other models (except for RF) the experiment does not conduct hyperparameter tuning, moreover not all RF's hyperparameters exercised (*max_features, max_samples, max_leaf_nodes, min_samples_split , min_samples_leaf)*. Second limitation, although the area information (not the exact geospatial data) of the properties are available in the dataset, the experiment does not process the geospatial information. Thirdly, although typically other metrics are measured for similar cases (R^2, RMSE and accuracy), for simpler consideration and analysis, those metrics are not reported.

Meanwhile, first idea of future studies / researches: other models / ensembles comparisons (Ridge Regression, Lasso , Logistic Regression, Multiple Linear Regression, KNN, CatBoost, Voting Regressor etc). Second idea is comparisons of the same models (DT, RF, XGB, LGBM) for other datasets. Thirdly, tuning other hyperparameters and comparing the results' metrics and processing time.

4. CONCLUSIONS

This study underline the result of importance of hyperparameter tuning in optimizing machine learning models for resale flat price prediction. Among the models tested, Random Forest consistently delivered superior accuracy, particularly when evaluated using Mean Absolute Error (MAE), distribution of percentage of error graph (Figure 8.). Fine-tuning parameters such as max_depth, n_jobs and n_estimators significantly improved computational efficiency, reducing processing time by over 93% with minimal influence on accuracy. For the same MAE result, it is much more effective to adjusting the n_estimator than limiting the max_depth. These findings establish Random Forest as a reliable tool for real estate price prediction when properly configured.

The comparison with other models, including XGBoost, LightGBM, and Decision Trees, revealed trade-offs between processing time and accuracy. While XGBoost and LightGBM were

faster, Random Forest consistently achieved the best MAE scores, making it ideal for scenarios that prioritizing accuracy. However, the lack of consistent hyperparameter optimization across all models is a limitation, suggesting the need for more balanced comparisons in future research.

Future studies could expand hyperparameter tuning across all models and incorporate additional metrics like R^2 or RMSE for a more comprehensive evaluation. Leveraging geospatial data, omitted in this study, could further improve model predictions. Testing these methods on datasets from other regions or markets could validate the findings and improve their generalizability. Addressing these areas would advance the utilization of machine learning in the real estate price prediction.

ACKNOWLEDGEMENTS

The authors wish to extend their gratitude to Universitas Pradita, Serpong, Indonesia for their support and the government of Singapore (Housing and Development Board) who collects the dataset with consistency and comprehensiveness.

REFERENCES

[1] Chua Beng Huat, "Public Subsidy/Private Accumulation: The Political Economy of Singapore's Housing ," in *Contemporary Southeast Asia* , vol. 46, Linda Y.C. Lim, Ed., ISEAS - Yusof Ishak Institute, 2024, pp. 499–401.

[2] M. Yazdani, "Machine Learning, Deep Learning, and Hedonic Methods for Real Estate Price Prediction," Oct. 2021, [Online]. Available: http://arxiv.org/abs/2110.07151

[3] A. G. Rawool, D. V Rogye, S. G. Rane, D. R. Vinayk, and A. Bharadi, "House Price Prediction Using Machine Learning," 2021.

[4] S. Bhushan Jha, R. F. Babiceanu, V. Pandey, and R. K. Jha, "Housing Market Prediction Problem using Different Machine Learning Algorithms: A Case Study."

[5] G. of S. Housing and Development Board, "Resale flat prices based on registration date from Jan-2017 onwards," data.gov.sg. Accessed: Oct. 23, 2024. [Online]. Available: https://data.gov.sg/datasets/d_8b84c4ee58e3cfc0ece0d773c8ca6abc/view

[6] E. K. Ampomah, Z. Qin, and G. Nyame, "Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement," *Information (Switzerland)*, vol. 11, no. 6, Jun. 2020, doi: 10.3390/info11060332.

[7] N. H. Zulkifley, S. A. Rahman, N. H. Ubaidullah, and I. Ibrahim, "House price prediction using a machine learning model: A survey of literature," *International Journal of Modern Education and Computer Science*, vol. 12, no. 6, pp. 46–54, 2020, doi: 10.5815/ijmecs.2020.06.04.

[8] A. J. Aminuddin, "A Review on The Performance of House Price Index Models: Hedonic Pricing Model Vs Artificial Neural Network Model," *International Journal of Accounting, Finance and Business (IJAFB)*, no. 39, pp. 53–63, Mar. 2022, doi: 10.55573/IJAFB.073906.

[9] T. Quang, N. Minh, D. Hy, and M. Bo, "Housing Price Prediction via Improved Machine Learning Techniques," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 433–442. doi: 10.1016/j.procs.2020.06.111.

[10] W. K. O. Ho, B. S. Tang, and S. W. Wong, "Predicting property prices with machine learning algorithms," *Journal of Property Research*, vol. 38, no. 1, pp. 48–70, 2021, doi: 10.1080/09599916.2020.1832558.

[11] P. Kanadiya, P. M. Chawan, and P. J. Kanadiya, "A KNN-Linear Regression Fusion Approach for Improved Real Estate Price Estimation," *International Research Journal of Engineering and Technology*, 2024, [Online]. Available: https://www.researchgate.net/publication/384322683

[12] P. Y. Wang, C. T. Chen, J. W. Su, T. Y. Wang, and S. H. Huang, "Deep learning model for house price prediction using heterogeneous data analysis along with joint self-attention mechanism," *IEEE Access*, vol. 9, pp. 55244–55259, 2021, doi: 10.1109/ACCESS.2021.3071306.

[13] S. Fatima, A. Hussain, S. Bin Amir, S. Haseeb Ahmed, and S. Muhammad Huzaifa Aslam, "XGBoost and Random Forest Algorithms: An In-Depth Analysis," *Pakistan Journal of Scientific Research, PJOSR*, vol. 3, no. 1, pp. 26–31, Jun. 2023.

[14] M. Mohan Tito Ayyalasomayajula, S. Bussa, and S. Ayyalasomayajula, "Forecasting Home Prices Employing Machine Learning Algorithms: XGBoost, Random Forest, and Linear Regression," *ESP Journal of Engineering & Technology Advancements*, vol. 1, no. 1, pp. 125–133, Sep. 2021, doi: 10.56472/25832646/JETA-V1I1P114.

[15] E. Henriksson and K. Werlinder, "Housing Price Prediction over Countrywide Data A comparison of XGBoost and Random Forest regressor models," KTH Royal Institute of Technology, Stockholm, Sweden, 2021.

[16] S. Bhushan Jha, R. F. Babiceanu, V. Pandey, and R. K. Jha, "Housing Market Prediction Problem using Different Machine Learning Algorithms: A Case Study," 2020.

[17] M. Thamarai and S. P. Malarvizhi, "House Price Prediction Modeling Using Machine Learning," *International Journal of Information Engineering and Electronic Business*, vol. 12, no. 2, pp. 15–20, Apr. 2020, doi: 10.5815/ijieeb.2020.02.03.

[18] L. Breiman, *Random Forests*, vol. 45. Netherland: Kluwer Academic Publishers, 2001.

[19] A. Hjort, J. Pensar, I. Scheel, and D. E. Sommervoll, "House price prediction with gradient boosted trees under different loss functions," *Journal of Property Research*, vol. 39, no. 4, pp. 338–364, 2022, doi: 10.1080/09599916.2022.2070525.

[20] A. L. John and S. Shaikh, "Predicting House Prices using Machine Learning and LightGBM," 2022. [Online]. Available: https://ssrn.com/abstract=4108744

[21] S. Abdul-Rahman, S. Mutalib, S. Alam, M. Nor, H. Zulkifley, and M. I. Ibrahim, "Advanced Machine Learning Algorithms for House Price Prediction: Case Study in Kuala Lumpur," 2021. [Online]. Available: www.ijacsa.thesai.org

[22] H. Li, "House price prediction based on machine learning," *Applied and Computational Engineering*, vol. 4, no. 1, pp. 623–628, May 2023, doi: 10.54254/2755-2721/4/2023362.

[23] W. Mulia, M. Lista, and A. SSi, "Comparison of Random Forest, XGBoost, and LightGBM Methods in Estimating Airbnb Accommodation Rental Prices Based on Customers in New York City," 2023.

[24] R. F. N. Iskandar, D. H. Gutama, D. P. Wijaya, and D. Danianti, "Klasifikasi Menggunakan Metode Random Forest untuk Awal Deteksi Diabetes Melitus Tipe 2," *Jurnal Teknik Industri Terintegrasi*, vol. 7, no. 3, pp. 1620–1626, Jul. 2024, doi: 10.31004/jutin.v7i3.26916.

[25] H. Hafid, "Penerapan K-Fold Cross Validation untuk Menganalisis Kinerja Algoritma K-Nearest Neighbor pada Data Kasus Covid-19 di Indonesia," *Journal of Mathematics, Computations and Statistics*, vol. 6, no. 2, pp. 161–168, Oct. 2023, [Online]. Available: http://www.ojs.unm.ac.id/jmathcos

[26] scikit-learn.org, "RandomForestRegressor." Accessed: Dec. 03, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html?utm_source=chatgpt.com

[27] L. Zhang, "Housing Price Prediction Using Machine Learning Algorithm," *Journal of World Economy*, vol. 2, no. 3, pp. 18–26, Sep. 2023, doi: 10.56397/jwe.2023.09.03.

[28] S. Dahiya, "Techniques for Efficient Training of Large-Scale Deep Learning Models," *MZ Computing Journal*, vol. 4, no. 1, 2023, [Online]. Available: https://mzjournal.com/index.php/MZCJ