

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/347350046>

# An Extreme Learning Machine Model Approach on Airbnb Base Price Prediction

Article in *International Journal of Advanced Computer Science and Applications* · January 2020

DOI: 10.14569/IJACSA.2020.01111123

---

CITATIONS

3

---

READS

511

2 authors, including:



**Agus Sihabuddin**

Universitas Gadjah Mada

14 PUBLICATIONS 37 CITATIONS

SEE PROFILE

# An Extreme Learning Machine Model Approach on Airbnb Base Price Prediction

Fikri Nurqahhari Priambodo<sup>1</sup>, Agus Sihabuddin<sup>2\*</sup>

Department of Computer Science and Electronics  
FMIPA, Universitas Gadjah, Yogyakarta  
Indonesia

**Abstract**—The base price of Airbnb properties prediction is still a new area of prediction research, especially with the Extreme Learning Machine (ELM). The previous studies had several suggestions for the advantages of ELM, such as good generalization performance, fast learning speed, and high prediction accuracy. This paper proposes how the ELM approach is used as a prediction model for Air BnB base price. Generally, the steps are setting hidden neuron numbers, randomly assigning input weight and hidden layer biases, calculating the output layer; and the entire learning measure finished through one numerical change without iteration. The performance of the model is estimated utilizing mean squared error, mean absolute percentage error, and root mean squared error. Experiment with Airbnb dataset in London with twenty-one features as input generates a faster learning speed and better accuracy than the existing model.

**Keywords**—Airbnb; base price prediction; extreme learning machine; fast learning

## I. INTRODUCTION

Airbnb is a property-sharing marketplace that allows property holders and tenants to put their properties on the web, with the goal that guests can pay to stay in them as part of the hospitality business. In the hospitality domain, evaluating and income, the executives, are two top now and again explored zone because of the theoretical and practical criticality of room estimating. Airbnb has must ace room valuing to build their gainfulness after fulfilling visitor desires [1].

Paid third-party pricing software is available, but generally, the property owners are needed to put their regular daily value (base cost); and the calculation will differ everyday price around that base cost. As a platform provider, Airbnb does not control how their hosts set prices for their postings, yet it gives an assortment of apparatuses to enable their hosts to put their prices all the more adequately. For instance, they permit hosts to set altered day by day rates, end of the week costs, and limits for long haul stays, so the base price determination becomes an essential process [2].

To be able to determine the property price, some researchers use various methods with three components to determine the price: i) a binary classification model predicts the booking probability of each listing night, ii) a regression model predicts the ideal cost for each listing night, iii) personalization reasoning on top of the yield from the resulting model to deliver the last cost suggestions [2].

In contrast to evaluating issues where valuing techniques are applied to an enormous number of indistinguishable items, there are no identical items on Airbnb. Each listing property on the Airbnb platform offers unique qualities and encounters to visitors. A geographically weighted regression (GWR) approach to distinguish a few variables corresponded with Airbnb listing costs has been implemented. The unique nature of Airbnb listing makes it exceptionally hard to appraise a precise interest bend needed to apply traditional income expansion evaluating techniques. The offline and online evaluation result of this pricing model show that the proposed model performs better than an immediate max-fire up evaluating methodology because of max-fire up pricing strategy are likely to suffer from demand curve which is hard to estimate [3].

Several machine learning algorithms such as a ridge, random forest regressor, linear regression, decision tree regressor, and lasso have been used in the forecast of housing selling costs for prediction. The result shows that feature selection is a significant component. Two special exhibitions of machine learning are the precision of forecast and averaging of errors or fitness, which might be influenced by the highlights chose with various gatherings of relationship levels [4].

The ELM has excellent potential for system prediction and modeling, i.e., an ELM based indicator for genuine frequency stability assessment (FSA) of power systems [5], electricity price forecasting [6], sales forecasting [7], security evaluation of wind power system [8], and drying system modeling [9]. However, there is no discussion about the Extreme Learning Machine (ELM) for Airbnb price forecast.

Over the past few years, a simple learning algorithm ELM for a single hidden layer feedforward network (SLFNs) was introduced [10]. ELM has superior faster-learning speed and better generalization performance than traditional feedforward network learning algorithms such as backpropagation (BP) algorithm. ELM achieves similar or preferred speculation execution over Support Vector Machine for regression and binary class, and much better speculation execution for multiclass classification cases [11].

ELM does not have to build up an exact numerical model of the object and appreciate the characteristics of the item. The technique accomplishes deficiency areas only by restricted defect tests for preparing and learning. ELM picks input loads and hidden biases randomly and then analytically computes

\*Corresponding Author

yield loads with Moore-Penrose derive pseudo-inverse [12]. The entire cycle is assessed without iterations, so the learning cycle is a very time-productive strategy. ELM Method overcomes numerous issues in gradient-based learning algorithms, for example, learning rate, stopping criterion, reducing local minima, and the number of epochs [13]. The objective of this research are: 1) to provide a new approach for the base price prediction for Airbnb; 2) to give a more accurate and faster prediction model for Airbnb base price prediction.

## II. RELATED WORKS

In previous studies, price forecasting has been carried out. There are several ELM methods that have been applied to price forecasting, namely gold price forecasting [14]; in this paper, a learning algorithm for a single hidden layered feedforward neural network called Extreme Learning Machine (ELM) is utilized, which has good learning capacity. Also, this examination dissects the five models, explicitly feedforward backpropagation networks, feedforward networks without feedback, radial basis function, ELMAN networks, and ELM learning model. The outcomes demonstrate that ELM learning performs in a way that is better than different techniques.

Based on [15], this research focus on the issue of how to plan an approach that can improve the forecast exactness just as accelerate expectation measure for stock market prediction. Initially, so as to get the most critical highlights of the market news records, this article is proposed a new feature selection algorithm called NRDC, just as another component weighting calculation (N-TF-IDF) to help increase the expected precision. Exploratory outcomes demonstrate that the N-N-K-ELM model can accomplish better execution on the thought of both forecast exactness and expectation speed as a rule.

In [5], ELM is used as the predictor for real-time. This article surveys the ELM's applications in power planning and a short time later develops an ELM-based indicator for real-time frequency stability assessment (FSA) of power systems. The contributions of the indicator are power system operational parameters, and the yield is recurrence soundness edge that measures the security level of the power system subject to a chance. By disconnected training with a frequency stability database, the indicator can be online was applied for real-time FSA. Profiting by the rapid speed of ELM, the predictor can be online are refreshed for upgraded robustness and reliability.

ELM is also used for electricity market prices by [6]. In this article, a fast electricity market price forecast is proposed dependent on an as of late developed learning technique for single hidden layer feedforward neural networks, the extreme learning machine (ELM), to defeat these disadvantages. The new methodology additionally has improved value stretches gauge exactness by incorporating a bootstrapping method for vulnerability assessments. The outcomes show the extraordinary capability of this proposed approach for online price forecasting at the spot market costs assessment.

In this research, ELM will be used for predicting Airbnb property base price with one preferred position is predominant

quicker learning pace and better speculation execution with a theory that will improve the accuracy of the model.

## III. THEORETICAL BASIS

### A. Extreme Learning Machine

Extreme Learning Machine is a single hidden layer feedforward neural network (SLFNs) and is a sort of straightforward and powerful learning algorithm [12]. It merely needs to set the hidden layer nodes and infinitely differentiable actuation work before preparing by clarifying the minimum norm least-squares of a linear equation to the ideal arrangement. Arbitrarily picked the hidden biases and input weights, and the yield loads are determined systematically with a given number of hidden neurons. The entire cycle of computation finishes once at a time without iteration.

The output function of ELM for generalized SLFNs as shown in (1).

$$f_L(x) = \sum_{i=1}^L \beta_i h_i(x) = h(x) \beta \quad (1)$$

Where  $\beta = [\beta_1, \dots, \beta_L]^T$  is the output weight vector between the hidden layer of  $L$  nodes to the  $m \geq 1$  output nodes, and  $h(x) = [h_1(x) h_L(x)]$  is a nonlinear feature mapping. The output (row) vector of the hidden layer with respect to the input  $x$ .  $h_i(x)$  is the output of the  $i^{th}$  hidden node output. The output functions of hidden nodes may not be unique. Different output functions may be used in other hidden neurons. In particular, in a real application  $h_i(x)$  can be formulated in (2).

$$h_i(x) = G(w_i, b_i, x) \quad w_i \in R^d, b_i \in R \quad (2)$$

Where  $G(w, b, x)$  (with hidden node parameters  $(w, b)$ ) is a nonlinear piecewise continuous function satisfying ELM universal approximation capability theorems [11], [16].

A standard SLFNs with  $L(N_0 \geq L)$  hidden layer nodes and the activation function  $g(x)$  are mathematically modeled as (3).

$$\sum_{i=1}^L \beta_i g_i(x_j) = \sum_{i=1}^L \beta_i g_i(w_i, x_j + b_i) = o_j \quad j = 1, 2, \dots, N \quad (3)$$

where  $w_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$  is the weight vectors connecting the  $i^{th}$  hidden node and input nodes,  $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$  is the weight vectors connecting the  $i^{th}$  hidden node and output nodes,  $b_i$  is the bias of the  $i^{th}$  hidden node,  $w_i \cdot x_j$  represents the inner product of  $w_i$  and  $x_j$ , the network structure, as shown in Fig. 1.

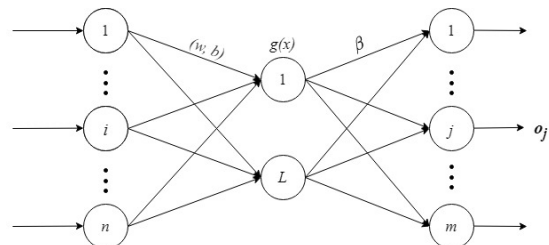


Fig. 1. A General ELM Network Structure.

To make SLFNs with  $L$  hidden layer nodes and the activation function  $g(x)$  approximate the  $N$  samples with zero error mean that  $\sum_{i=1}^L \|o_j - t_j\| = 0$ , namely, existing  $\beta_i, w_i$  and  $b_i$  make formula (4) established.

$$\sum_{i=1}^L \beta_i g(w_i \cdot x_j + b_i) = t_j, j = 1, 2, \dots, N \quad (4)$$

The equation (4) can be written compactly as (5)

$$H\beta = T \quad (5)$$

where,

$$H(w_1, \dots, w_L, b_1, \dots, b_L, x_1, \dots, x_N) \\ \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_L \cdot x_1 + b_L) \\ \vdots & \dots & \vdots \\ g(w_1 \cdot x_N + b_1) & \dots & g(w_L \cdot x_N + b_L) \end{bmatrix}_{N \times L}, \\ \beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, \text{ and, } T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \quad (6)$$

From (6)  $H$  is the hidden layer output matrix of the neural network, the  $i$ th column of  $H$  represents the output matrix about  $x_1, \dots, x_N$  of the  $i$ th hidden layer node.

When the activation function infinitely differentiable, the input connection weights  $w_i$  moreover, hidden layer bias  $b_i$  can randomly set at the beginning of the training, and they will be fixed in the training process; the output connection weights  $\beta$  are obtained by solving the least-squares solution of linear (7), the result as (8).

$$\min \|H\beta - T\| \quad (7)$$

$$\beta = H^+ T \quad (8)$$

where  $H^+$  is the Moore-Penrose generalized inverse of the hidden layer output matrix  $H$ . The mathematical transformation determines the output weights. This ensures that the long training phrase when network boundaries are iteratively changed with some reasonable learning boundaries (like iterations and learning rate) is not needed.

ELM learning algorithm steps can be summarized as follows:

Step 1: Given a training set  $(x_i, t_i) = (i = 1, 2, \dots, N)$ , the activation function is  $g(x)$ , number of hidden layer nodes is  $L$ , setting the input weights  $w_i$  and hidden layer bias  $b_i$  randomly.

Step 2: Calculate the output matrix  $H$  of the hidden layers.

Step 3: Calculate the output weights  $\beta$  by (8).

Because the ELM algorithm does not require iterative input weights and bias in training adjustment, it reduces the complexity of the training, and the training speed improved obviously.

#### B. Performance Evaluation

An evaluation needs to be performed to measure performance and to provide feedback that can serve to improve the model. In this research, root mean squared error

(RMSE), mean square error (MSE), and mean absolute percentage error (MAPE) measurements were performed [17].

MSE is a measure of prediction accuracy by squaring each error for each observation in a data set and then obtaining the average number of squares. MSE gives greater weight to the error compared to a small error because the error is raised before adding up. MSE can be calculated by (9).

$$MSE = \frac{1}{n} \times \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (9)$$

RMSE is the square root value of the average square error and formulated as in (10).

$$RMSE = \sqrt{\frac{1}{n} \times \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (10)$$

MAPE is the average percentage of the sum of the differences between prediction results with actual data. The formula for the mean absolute percentage error can be written as follows in (11).

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i} \times 100\% \quad (11)$$

Given  $y_i$  is actual value,  $\hat{y}_i$  is predicted value,  $i$  is the  $i$ th data,  $n$  is the number of data.

A study from [18] shown that using MAPE as a measure of quality for regression models is feasible both on a practical point of view and on a theoretical one.

## IV. METHODS

### A. Data Description

The dataset utilized for this examination originates from InsideAirbnb.com. The dataset was downloaded on 9 April 2019 and contained data on all London Airbnb listings that were live on the webpage on that date, which is 79,671 Airbnb listing. The data itself has 106 features and saved in .csv format.

In this study, the base price prediction is predicted from 21 input features that will be chosen after the data preprocessing of the Airbnb listing. In the sample dataset, the input features that contains a price value are the advertised price of its Airbnb listing, security deposit fee, cleaning fee, and extra people fee.

### B. Preprocessing

The dataset needs to be changed or prepared according to the needs. The original dataset has 106 features, including quite a few text columns of all the different description fields.

Some features or columns will be dropped because of: not indicated to be useful for predicting price, and there are many null or NaN entries. Some entries connected with the fee and having missing values were replaced with the median to avoid fractions by 0.

### C. Network Architecture

To solve the Airbnb base pricing issue, the ELM model was designed. The proposed methods can see in Fig. 2. Twenty-one input was picked to ELM from the dataset,

though the base price is assigned the yield of ELM. The performance of ELM relies upon be utilized the type of activation function and the number of hidden neurons. A sigmoid function was chosen as it is not too delicate to the user-determined parameters and determined by a testing parameter to find the best model of ELM.

Twenty-one features are chosen after the preprocessing as input due to their relationship to basic fee types of a property price are shown in Table I.

The test will be divided into two parts. The first test was conducted to learn the amount of training data and test data against the evaluation parameter value. The second test is to determine how the number of hidden neurons affects the evaluation parameter value and execution time; for this purpose, 100 steps regularly increase the number of hidden layer neurons from 100 to 1000. Each test will be conducted ten attempts and will be evaluated based on the average evaluation value and execution time.

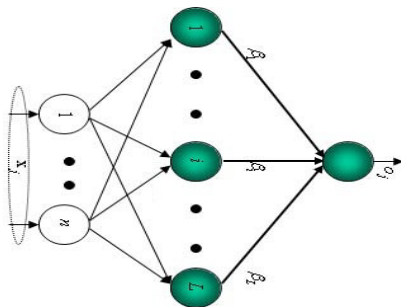


Fig. 2. The Architecture of Extreme Learning Machine.

TABLE I. FEATURES INPUT

No	Feature
1	number of people that a property can accommodate
2	the cleaning fees
3	number of days available for the next 90 days
4	the fee per extra person
5	total reviews that a property has
6	the number of bathrooms
7	the security deposit
8	the minimum night stays
9	the maximum night stays
10	property type
11	room type
12	location of the property
13	review ratings
14	the duration between the first and most recent review
15	the type of cancellation policy
16	whether the property is instantly bookable
17	the presence of a wide range of amenities
18	host response times in the past 30 days
19	host percentage of response rates
20	Super-host
21	total host listings

## V. RESULT AND DISCUSSION

The ELM parameter testing aims to determine the best parameters for benchmarking and results in Airbnb base price predictions. The first tests were conducted to learn the amount of training data and test data against the evaluation parameter value; the second is to resolve how the count of hidden neurons affects the evaluation parameter value and execution time. Each parameter value will be tested as many as ten times the experiment.

The first test conducted is to determine the size ratio of data training and data testing. The size ratio of training and testing data is 70:30 and 80:20. At this stage, the count of hidden neurons will be used as a control variable in testing. The number of hidden neurons used is 100. The results from testing the parameter ratio of training and testing data are indicated in Table II.

From the test results at the training data ratio and test of the ELM model, it was apparent that the training and test data with a ratio of 70%:30% had better average performance compared to the training and test data ratio of 80%:20% for all accuracy parameters.

### A. Number of Hidden Neuron Testing

The next test is determining the count of hidden neurons on a hidden layer. The count of hidden neurons to be tested are 100, 200, 300, 400, 500, 600, 700, 800, 900, and 1000 as indicated in [20]. The MSE accuracy from this step is shown in Table III.

The MSE result shows that accuracy tends to decrease as the number of hidden neurons increases, and it is more evident in the chart of average MSE versus hidden neurons size in Fig. 3.

The experiment results, as depicted in Fig. 3, implicates that 1000 hidden neurons perform better MSE accuracy than the other hidden neurons for the lowest, highest, and average MSE training and testing except for 500 hidden nodes on average MSE testing.

TABLE II. MSE, RMSE, AND MAPE ON DIFFERENT DATA SPLIT

Data Split	MSE Training		MSE Testing	
	80:20	70:30	80:20	70:30
Lowest	0.590	0.573	0.622	0.589
Highest	0.901	0.910	1.027	0.940
Average	0.770	0.713	0.802	0.743
Data Split	RMSE Training		RMSE Testing	
	80:20	70:30	80:20	70:30
Lowest	0.768	0.757	0.789	0.767
Highest	0.949	0.954	1.013	0.970
Average	0.875	0.842	0.893	0.860
Data Split	MAPE Training		MAPE Testing	
	80:20	70:30	80:20	70:30
Lowest (%)	14.027	13.374	14.113	13.417
Highest (%)	17.054	17.497	17.624	17.393
Average (%)	15.821	15.241	15.879	15.148

TABLE III. MSE ACCURACY ON A DIFFERENT NUMBER OF HIDDEN NEURONS

Neuron	MSE TRAINING			MSE TESTING		
	Low	High	Avg.	Low	High	Avg
100	0.314	0.385	0.350	0.338	0.425	0.374
200	0.108	0.135	0.124	0.129	0.303	0.194
300	0.081	0.097	0.086	0.096	0.302	0.170
400	0.066	0.072	0.068	0.086	0.230	0.134
500	0.055	0.061	0.058	0.072	0.151	0.096
600	0.051	0.055	0.053	0.070	0.251	0.135
700	0.046	0.048	0.047	0.064	0.238	0.134
800	0.043	0.046	0.044	0.061	0.299	0.106
900	0.040	0.042	0.041	0.064	0.247	0.137
1000	0.038	0.039	0.039	0.057	0.198	0.098

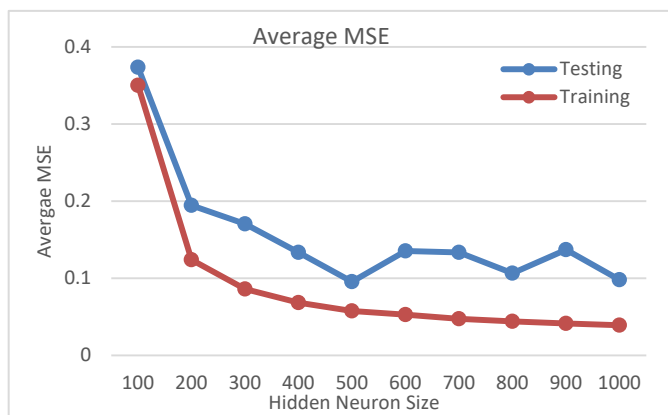


Fig. 3. Average MSE Testing Result.

The next accuracy testing is for RMSE, as shown in Table IV. It has the same behavior as MSE in which the more hidden nodes in the model, the smaller RMSE is.

As the average RMSE is transformed in to chart for the average RMSE, it is more obvious too that the behavior is detected as seen in Fig. 4.

TABLE IV. RMSE TESTING ON THE NUMBER OF HIDDEN NEURONS

Neuron	RMSE TRAINING			RMSE TESTING		
	Low	High	Avg.	Low	High	Avg
100	0.560	0.621	0.592	0.582	0.652	0.611
200	0.329	0.368	0.352	0.360	0.550	0.437
300	0.285	0.311	0.293	0.310	0.549	0.405
400	0.256	0.268	0.261	0.293	0.480	0.358
500	0.234	0.247	0.240	0.268	0.389	0.306
600	0.225	0.234	0.230	0.264	0.501	0.357
700	0.214	0.220	0.218	0.252	0.488	0.354
800	0.208	0.214	0.210	0.247	0.547	0.315
900	0.201	0.206	0.203	0.253	0.497	0.359
1000	0.196	0.198	0.198	0.239	0.445	0.304

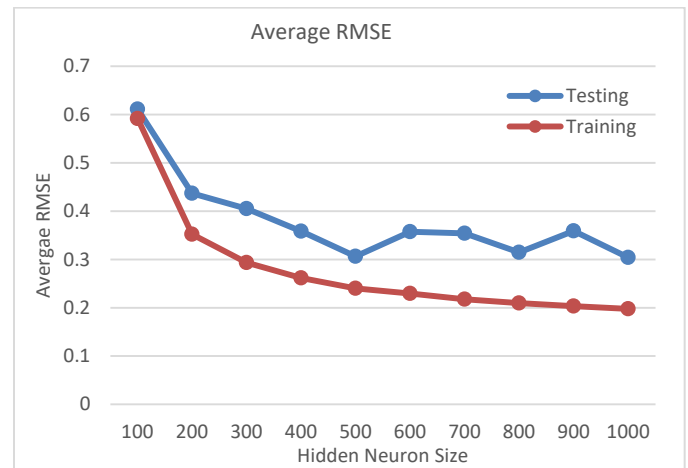


Fig. 4. Average RMSE Testing Result.

The last accuracy data is for the MAPE on various hidden neurons, as shown in Table V. The result is similar with two other accuracy parameters where the more neurons, the less accuracy parameter for the MAPE. It demonstrates the same behavior too for the nodes of 500 and 800.

Table V shown the average MAPE results for each testing parameter; like the previous evaluation method, testing the number of neurons on a hidden layer indicates that with many neurons, it will result in a small error value. If the average MAPE is charted, the same pattern is appearing for the MAPE, as shown in Fig. 5.

From Fig. 3, 4, and 5, it can be seen when the number of neurons increased, and the accuracy tends to increase too. It could be caused by the more neurons, the more link between the input and output layer, and leads to a better quality of learning process. The more link between input and output layer needs more computation time, as indicated in Table VI.

From Table VI, we can see that 1000 hidden neurons in terms of duration cost almost six times longer than 100 hidden neurons. The time increasing for the computation tend to be a linear graph, as shown in Fig. 6.

TABLE V. MAPE TESTING ON SOME HIDDEN NEURONS

Neuron	MAPE TRAINING			MAPE TESTING		
	Low	High	Avg.	Low	High	Avg
100	9.87	10.99	10.39	9.93	11.15	10.38
200	5.84	6.52	6.24	6.25	10.87	7.87
300	4.97	5.47	5.14	5.05	11.16	7.43
400	4.38	4.65	4.50	4.55	9.52	6.22
500	3.94	4.17	4.05	4.01	6.84	4.90
600	3.75	3.94	3.83	3.79	10.00	6.31
700	3.51	3.62	3.57	3.51	9.98	6.23
800	3.35	3.47	3.40	3.40	11.03	5.14
900	3.19	3.34	3.25	3.56	9.84	6.33
1000	3.06	3.15	3.11	3.07	8.57	4.88

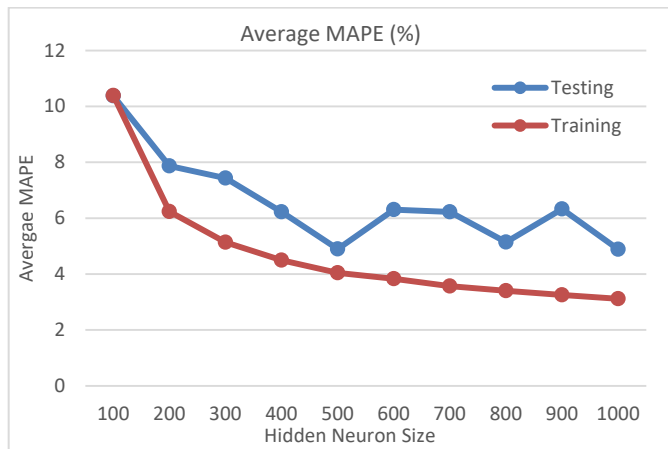


Fig. 5. Average MAPE Testing Result.

TABLE VI. ELM COMPUTATIONAL TIME

Neuron	TRAINING TIME (s)		
	Low	High	Avg.
100	0.931	1.062	1.032
200	1.833	2.154	1.967
300	2.248	2.692	2.366
400	2.623	2.833	2.723
500	3.015	3.206	3.097
600	3.447	3.991	3.566
700	3.929	4.111	4.029
800	4.300	4.565	4.472
900	4.884	5.260	5.033
1000	5.420	5.653	5.522

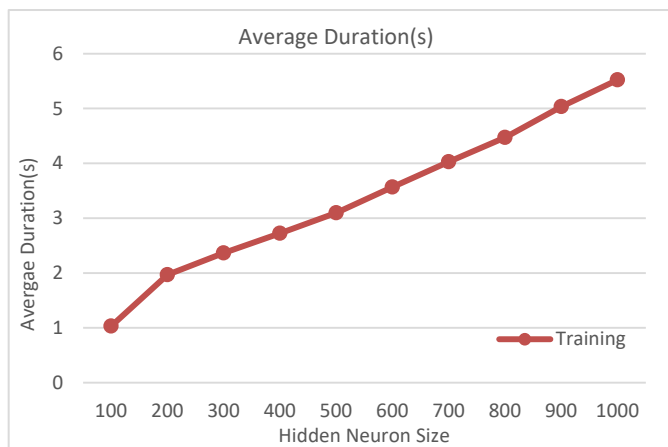


Fig. 6. The Average Computational Time for Test Data.

As observed in Fig. 6, the execution time for the model continue expanding as the number of hidden neurons ascends because the learning time of ELM is nearly spent on computing the Moore-Penrose generalized inverse  $H^+$  of the hidden layer output matrix  $H$ .

Based on the tests already conducted, the parameters to be used in determining Airbnb price prediction results with the

ELM method are the ratio of training and test data of 70%:30% and the number of hidden neurons = 1000.

### B. Comparison with XGBoost

In the testing phase, the performance of the resulting model was evaluated using three selected methods, namely MAPE, MSE, and RMSE. As a comparison, other predictions are made using the XGBoost method [19]. The test results for the three methods can be seen in Table VII.

TABLE VII. ELM MODEL COMPARISON AGAINST XGBOOST

Method	ELM		XGBoost	
	Train Data	Test Data	Train Data	Test Data
MAPE (%)	3.06	3.07	6.70	6.75
MSE	0.038	0.057	0.181	0.198
RMSE	0.196	0.239	0.426	0.445

In the MAPE method, the model produces a value of 3.06% for the training data and 3.07% for the test data, while for the MSE method, the model has a value of 0.038 for the training data and 0.057 for the test data, and for the RMSE method, the model produces a value of 0.196 for the training data and 0.239 for test data. While for the XGBoost model, In the MAPE method, the model has a value of 6.70% for the training data and 6.75% for the test data, while for the MSE method, the model produces a value of 0.181 for the training data and 0.198 for the test data, and for the RMSE method, the model produces a value of 0.4260 for the training data and 0.4454 for test data. This is very clear that the ELM model outperforms the XGBoost model for the three accuracy parameters.

As observed in Table VIII, In terms of execution time, the ELM model takes 5.4 seconds for training data and 6.341 seconds for test data. On the other hand, the XGBoost model takes 12.8950 seconds for training data and 12.8950 seconds for test data.

TABLE VIII. THE AVERAGE DURATION TIME OF ELM AND XGBOOST

Algorithm	Training time (s)	Testing time (s)
ELM	5.419	6.341
XGBOOST	12.895	12.895

## VI. CONCLUSION

In conclusion, we have developed an ELM prediction model approach on Airbnb base price and tested it by London Airbnb Listing in April 2019. The model is trained using 21 features, 70%:30% data split, and a maximum of 1000 neurons. The experiment results show that the model is having a good accuracy with the best average MSE value of 0.096, RMSE value of 0.304, and MAPE value of 4.88% for the test data. These accuracies are mostly achieved by 1000 neurons. From the experiments, the model show as the count of neurons raises, the link between the input and output layers would consequently increase. This leads to a better quality of learning. These accuracy parameters are outperforming the XGBoost algorithm and having a much faster learning time with better accuracy.

For further research, the number of neurons for training can be expanded to more than 1000 neurons with more powerful hardware, so the convergence point with the number of neurons where the accuracy reached the optimum value could be found. The base price prediction can be expanded to a daily basis prediction with more features like scheduled events, holidays, and many other features.

#### ACKNOWLEDGMENT

The author would like to express our gratitude to the Direktorat Penelitian Universitas Gadjah Mada, Indonesia, for providing the research grant.

#### REFERENCES

- [1] M. Chattopadhyay and S. K. Mitra, "Do Airbnb host listing attributes influence room pricing homogenously?," *Int. J. Hosp. Manag.*, vol. 81, no. September 2018, pp. 54–64, 2019, doi: 10.1016/j.ijhm.2019.03.008.
- [2] P. Ye et al., "Customized regression model for Airbnb dynamic pricing," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 932–940, 2018, doi: 10.1145/3219819.3219830.
- [3] Z. Zhang, R. J. C. Chen, L. D. Han, and L. Yang, "Key factors affecting the price of Airbnb listings: A geographically weighted approach," *Sustain.*, vol. 9, no. 9, pp. 1–13, 2017, doi: 10.3390/su9091635.
- [4] T. Mohd, S. Masrom, and N. Johari, "Machine learning housing price prediction in Petaling Jaya, Selangor, Malaysia," *Int. J. Recent Technol. Eng.*, vol. 8, no. 2 Special Issue 11, pp. 542–546, 2019, doi: 10.35940/ijrte.B1084.0982S1119.
- [5] Y. Xu, Y. Dai, Z. Y. Dong, R. Zhang, and K. Meng, "Extreme learning machine-based predictor for real-time frequency stability assessment of electric power systems," *Neural Comput. Appl.*, vol. 22, no. 3–4, pp. 501–508, 2013, doi: 10.1007/s00521-011-0803-3.
- [6] X. Chen, Z. Y. Dong, K. Meng, Y. Xu, K. P. Wong, and H. W. Ngan, "Electricity price forecasting with extreme learning machine and bootstrapping," *IEEE Trans. Power Syst.*, vol. 27, no. 4, pp. 2055–2062, 2012, doi: 10.1109/TPWRS.2012.2190627.
- [7] F. L. Chen and T. Y. Ou, "Sales forecasting system based on Gray extreme learning machine with Taguchi method in retail industry," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 1336–1345, 2011, doi: 10.1016/j.eswa.2010.07.014.
- [8] Y. Xu, Z. Y. Dong, Z. Xu, K. Meng, and K. P. Wong, "An intelligent dynamic security assessment framework for power systems with wind power," *IEEE Trans. Ind. Informatics*, vol. 8, no. 4, pp. 995–1003, 2012, doi: 10.1109/TII.2012.2206396.
- [9] A. Balbay, Y. Kaya, and O. Sahin, "Drying of black cumin (*Nigella sativa*) in a microwave assisted drying system and modeling using extreme learning machine," *Energy*, vol. 44, no. 1, pp. 352–357, 2012, doi: 10.1016/j.energy.2012.06.022.
- [10] G. Bin Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006, doi: 10.1016/j.neucom.2005.12.126.
- [11] G. Bin Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 42, no. 2, pp. 513–529, 2012, doi: 10.1109/TSMCB.2011.2168604.
- [12] G. Bin Huang, X. Ding, and H. Zhou, "Optimization method based extreme learning machine for classification," *Neurocomputing*, vol. 74, no. 1–3, pp. 155–163, 2010, doi: 10.1016/j.neucom.2010.02.019.
- [13] G. Bin Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," *IEEE Int. Conf. Neural Networks - Conf. Proc.*, vol. 2, pp. 985–990, 2004, doi: 10.1109/IJCNN.2004.1380068.
- [14] S. K. Chandar, M. Sumathi, and S. N. Sivanadam, "Forecasting gold prices based on extreme learning machine," *Int. J. Comput. Commun. Control*, vol. 11, no. 3, pp. 372–380, 2016, doi: 10.15837/ijccc.2016.3.2009.
- [15] F. Wang, Y. Zhang, H. Xiao, L. Kuang, and Y. Lai, "Enhancing Stock Price Prediction with a Hybrid Approach Based Extreme Learning Machine," *Proc. - 15th IEEE Int. Conf. Data Min. Work. ICDMW 2015*, pp. 1568–1575, 2016, doi: 10.1109/ICDMW.2015.74.
- [16] G. Bin Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, no. 16–18, pp. 3460–3468, 2008, doi: 10.1016/j.neucom.2007.10.008.
- [17] J. G. De Gooijer and R. J. Hyndman, "25 Years of Time Series Forecasting," *Int. J. Forecast.*, vol. 22, no. 3, pp. 443–473, 2006, doi: 10.1016/j.ijforecast.2006.01.001.
- [18] A. de Myttenaere, B. Golden, B. Le Grand, and F. Rossi, "Mean Absolute Percentage Error for regression models," *Neurocomputing*, vol. 192, pp. 38–48, 2016, doi: 10.1016/j.neucom.2015.12.114.
- [19] L. Lewis, "Predicting Airbnb prices with machine learning and deep learning," *Medium - Toward Data Science*, 2019, [Online]. Available: <https://towardsdatascience.com/predicting-Airbnb-prices-with-machine-learning-and-deep-learning-f46d44afb8a6>.
- [20] F.H. Priambodo, A. Sihabuddin, *Extreme Learning Machine Prediction Model on Airbnb Base Price*, Thesis, Universitas Gadjah Mada, Yogyakarta, 2020.