Deep Learning in Predicting Real Estate Property Prices: A Comparative Study

Donghui Shi
Department of Computer Engineering
School of Electronics and Information Engineering
Anhui Jianzhu University
Hefei, China 230601
sdonghui@gmail.com

Hui Zhang
Department of Computer Engineering
School of Electronics and Information Engineering
Anhui Jianzhu University
Hefei, China 230601
zh_289@163.com

Jian Guan
Department of Computer Information Systems
College of Business, University of Louisville
Louisville, KY 40292
jeff.guan@louisville.edu

Jozef Zurada
Department of Computer
Information Systems, College of Business
University of Louisville
Louisville, KY 40292
jozef.zurada@louisville.edu

Zejun Chen
Department of Computer Engineering
School of Electronics and Information Engineering
Anhui Jianzhu University
Hefei, China 230601
chenzejun1997@qq.com

Xiyang Li
Department of Computer Engineering
School of Electronics and Information Engineering
Anhui Jianzhu University
Hefei, China 230601
2572081284@qq.com

Abstract

The dominant methods for real estate property price prediction or valuation are multi-regression based. Regression-based methods are, however, imperfect because they suffer from issues such as multicollinearity and heteroscedasticity. Recent years have witnessed the use of machine learning methods but the results are mixed. This paper introduces the application of a new approach using deep learning models to real estate property price prediction. The paper uses a deep learning approach for modeling to improve the accuracy of real estate property price prediction with data representing sales transactions in a large metropolitan area. Three deep learning models, LSTM, GRU and Transformer, are created and compared with other machine learning and traditional models. The results obtained for the data set with all features clearly show that the RF and Transformer models outperformed the other models. LSTM and GRU models produced the worst results, suggesting that they are perhaps not suitable to predict the real estate price. Furthermore, the implementations of Transformer and RF on a data set with feature reduction produced even more accurate prediction results. In conclusion, our research shows that the performance of the Transformer model is close to the RF model. Both models produce significantly better prediction results than existing approaches in terms of accuracy.

1. Introduction

The dominant methods for real estate property (hereafter simply referred to as property) price prediction are regression-based [1]. Regression methods suffer from several deficiencies including multicollinearity and heteroscedasticity [2-4]. In the last few decades the introduction of machine learning and AI methods [5, 7-10] brought hope, but the results have been mixed so far. Other methods, such as Autoregressive Moving Average (ARMA) [11], have also been used for predicting property price changes. As sales transactions increase and sales data become more readily available, it is necessary for researchers to examine the effect of newer platforms using deep learning for implementing predictive models for property prices [12-13]. This research addresses this gap by applying deep learning models for property prices. Our results obtained from tests on sales transactions from a large metropolitan city clearly demonstrate that the deep learning Transformer models provide accurate predictive performance comparable to that of a reliable traditional machine learning model such as the Random Forest (RF).

2. Methods

2.1. Traditional machine learning models

URI: https://hdl.handle.net/10125/102749
978-0-9981331-6-4

The following traditional models are used in this research to compare with the new deep learning models. MLR model is a popular multi-variate predictive analysis method [1]. Decision Tree (DT) is a tree induction algorithm. The model is good at dealing with disordered and irregular data. DTs can be used for target variable with continuous values. In 1986, Quinlan released the ID3 algorithm that is still widely used today [14]. In 1993, Quinlan released the C4.5 algorithm based on the ID3 algorithm [15]. Random Forest (RF) [16] is one of the Bagging integration algorithms for classification and regression. The model is actually a combination of multiple DT models for training to maximize the advantages of a DT model. RFs are used in many applications, both in classification and prediction. The actual effect of the combined predictive models in RF is often far greater than that of a single predictive model.

Autoregressive moving average (ARMA) [17], which is used for modeling time series data, is often given as ARMA(p,q), where the parameters p and q respectively represent the number of autoregressive and the number of moving average terms. The values of the parameters are obtained by observing the autocorrelation function and partial autocorrelation function in the experiment.

2.2. LSTM model

The deep learning Long short-term memory (LSTM) model [18] is a type of Recurrent Neural Network (RNN) structure [19]. LSTM model is also a chain structure composed of multiple identical modules. The principle of RNN calculation is overwriting, but the derivation process follows the chain rule, which would cause the gradient to be connected and vanish. LSTM model uses a cumulative calculation method, which avoids the problem of vanishing gradient. Figure 2 shows the cell structure in the hidden layer of an LSTM model.

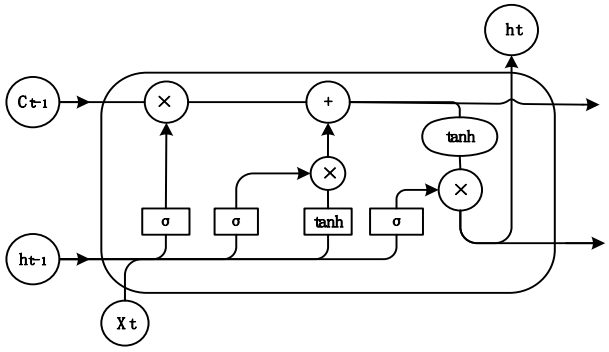


Figure 2. Hidden layer cell structure of LSTM model.

A common architecture of LSTM is composed of a cell, called the memory part of the LSTM unit, and three "regulators," called gates: an input gate, an output gate, and a forget gate. The input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell, and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit. The activation function of the LSTM gates is often the sigmoid function to control the output value from 0 to 1, indicating that the information is selectively received. The calculation process is as follows:

Forgotten gate: $f_t = \sigma(W_f.[h_{t-1}x_t] + b_f)$;

Input gate: $i_t = \sigma(W_i.[h_{i-1}, x_t] + b_i)$;

$$\check{C}_t = \tanh(W_c.[h_{t-1}x_t] + b_c);$$

Update cell status: $C_t = f_t * C_{t-1} + i_t * \check{C}_t$;

Output gate: $O_t = \sigma W_0[h_{t-1}, x_t] + b_0$;

$$h_t = O_t * \tanh(C_t);$$

where W is the weight of each layer of the network, b is the offset value of each layer, σ represents the sigmoid activation function, and tanh represents the hyperbolic tangent activation function. (https://en.wikipedia.org/wiki/Long_short-term_memory).

2.3. Gated Recurrent Unit (GRU) model

Another optimization of the RNN model is the Gated Recurrent Unit (GRU) [20]. Memory cells are actually composed of a couple of elements, gates, that are recurrent and control how information is being remembered and forgotten. GRU has the advantage of long-term learning, and it eliminates the input gate, forget gate, and output gate found in the structure of LSTM. Instead, the reset gate and update gate are combined, and the cell state and output vector are integrated into a vector. Figure 3 shows the hidden layer cell structure.

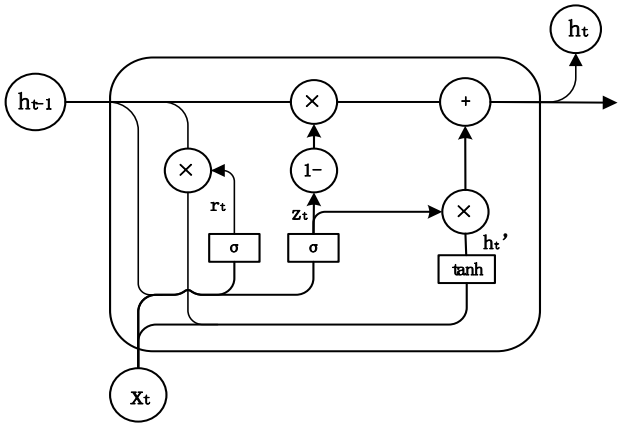


Figure 3. Hidden layer cell structure of GRU.

The following are the equations representing the reset gate and update gate to receive signals:

$Z_t = \sigma(W_{zg}[h_{t-1}, x_t])$;

$r_t = \sigma(W_{rg}[h_{t-1}, x_t])$;

The equation of the update gate is:

$h_t = z * h'_t + (1 - z) * h_{t-1}$;

where W is the weight of each layer of the network and σ represents the Sigmoid activation function.

After obtaining the gate control signal, the reset gate works to get the reset data $h'_{t-1} = h_{t-1} * r$, then combines h'_{t-1} with x_t, uses the tanh activation function to limit the obtained data to between -1 and 1, and calculates h'_t. Then the core step of GRU will update memory. This step performs two operations of forgetting and remembering. The innovation of GRU lies in using an update gate to combine forgetting and selecting memory, while LSTM structure requires two "gates" to complete this operation.

Both LSTM and GRU have the ability to learn and memorize long-term information, and both are good at processing time series data. The difference is that the GRU model automatically determines whether the generation of new information will be affected by the old information when processing the current cell state. In real applications, because the GRU model has fewer parameters, it can run faster, thus it is a good model for larger datasets.

2.4. Transformer

A transformer deep learning model was introduced in 2017 [21]. It applies the mechanism of self-attention, differentially weighting the significance of each part of the input data. It is used primarily in the field of natural language processing (NLP) replacing RNN models such as long short-term memory (LSTM). Like RNNs, transformers are suitable for processing sequential input data, such as natural language, with applications towards tasks such as translation and text summarization. However, unlike RNNs, transformers are different in that it can process the entire input all at once. The attention mechanism allows the decoder to use the most relevant parts of the input sequence in a flexible manner, by a weighted combination of all the encoded input vectors, with the most relevant vectors being attributed the highest weights.

Transformer is based on the coding-decoder structure, which was initially used to address the problem of quantity mismatch between input and output, and it has gradually been applied in various fields. Transformer is composed of one to multiple encoders and a corresponding number of decoders and some connected network layers. Each encoder has the same structure and independent parameters. Each encoder is made up of two parts, multi-head Attention and Feed Forward Neural Network. The decoder adds a masked multi-head attention on the basis of the encoder to mask the predicted information that is not used in training.

2.5. The Assessment Measure Methods

Our research conducts experiments on each model separately and measures the performance of the model one at a time. The following three assessment metrics are commonly used in the property price prediction domain [2,5,7,13].

(1) Root Mean Squared Error (RMSE) refers to the root mean error between the predicted value and the actual value. It is an indicator that represents the degree of data change. This error tends to magnify the effects of outliers.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_t - f_t)^2}\ ;$$

(2) Mean Absolute Error (MAE) is the average value of absolute error, which can accurately represent the error between the predicted value and the actual value.

$$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n}|y_t - f_t|\ ;$$

(3) Mean Absolute Percentage Error (MAPE):

$\text{MAPE} = \sum_{i=1}^{n} \left|\frac{y_t - f_t}{y_t}\right| * \frac{100}{n}$;

where n is the number of samples, y_t is the actual value, and f_t is the predicted value.

For all three error metrics the smaller the error values are, the better the prediction models.

2.5. A Deep Learning Approach to Property Price Prediction

In the study, we tested the traditional machine learning models and deep learning models. The general framework is as shown in Figure 4. The sales data for condominiums for 2 years were collected and data for other relevant factors were then collected. Traditional machine learning models and deep learning models were tested and compared to determine the best model.

3. Data set

We collected the condominium sales transaction records from a large real estate transaction website (Lianjia, https://hf.lianjia.com/) from 2017 to 2019 in Hefei, a large metropolitan in China. There are 20,843 transaction records. The data contain the following attributes: the district of the property, the sale date, the sale price (the dependent variable), the average property price in the community, the size of the condo in square meters, number of rooms, the floor number, the total number of floors, building year of the community, number of elevators in the building, and number of households on each floor. The data set is sorted by the sale date in ascending order. The price of each sold property is changed with time and the data set has time series characteristics.

At present, there is no commonly accepted set of attributes for predicting property price in China. We interviewed and consulted market professionals before

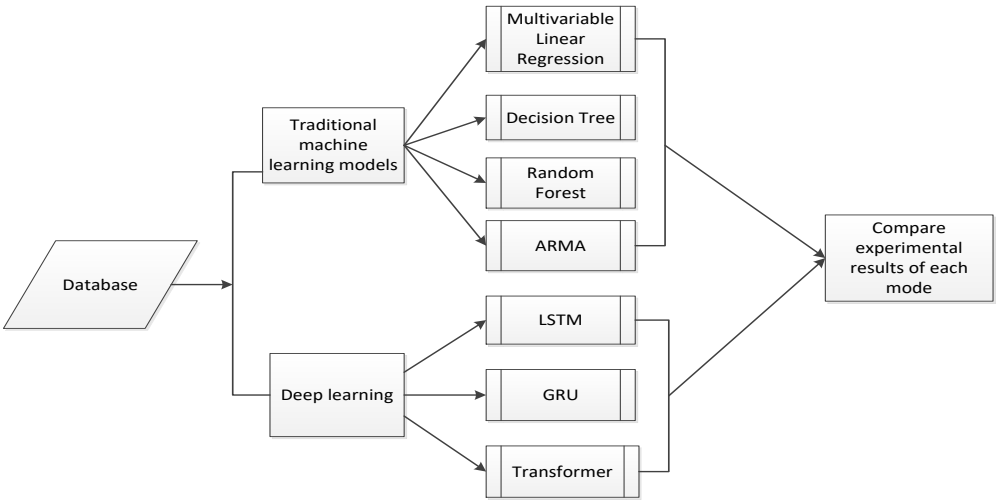


Figure 4. The research framework.

the study, and learned that there are certain factors that will affect the prices of condominiums in the market. The significant factors can be roughly divided into regulatory policies, economic factors, regional characteristics, community characteristics, and property characteristics. Relevant regulatory policies factors are currently difficult to collect. This paper uses Gross Domestic Product (GDP) and Per Capita Disposable Income (PCDI) values as macroeconomic factors. The main regional characteristics are the development of the administrative district where the property is located, the nearby traffic conditions, and the number of support facilities for the area. Property prices vary by administrative districts. In addition education resources, medical resources and commercial facilities are also included. For different cities, these factors have to be adjusted in accordance with the actual local conditions. The characteristics of the community variables are the age of the building and the average price of all the properties for sale in the community. The characteristics of the property itself, which are important to buyers, include the number of rooms, the floor number, and the number of households per elevator. Table 1 presents all the variables and categories of variables. Table 2 lists the descriptive statistics of the data set.

Table 1. Categories of variables and variables.

Categories	Variables	Type of Value
Time factors	Transaction time	Number of days from transaction date to Dec. 31, 2019
Economic factors	2019 GDP of the district	GDP value
	2018 GDP of the district	GDP value
	2018 PCD of the district	PCDI value
Location	Administrative District	District code
	Near government	Municipal and provincial governments; If the property is close to the location of government, the value is set to 1; If not close, it is set to 0
	Population	Population values
	Subway	Subway line number
	Large business district	Number of large business districts
	Shopping mall	Number of shopping malls
	Top hospitals	Number of top hospitals
	Medical institutions	Number of medical institutions
	Park	Number of parks
	Primary school	Number of primary schools
	Middle school	Number of secondary schools
Community characteristics	Building age	Building years
	Average price	Average price
Property characteristics	Number of rooms	A Room B living room C Kitchen D Bathroom Total number of rooms A+B+C+D
	Size of the condo	Area values (m^2)
	Floor number	High 1; Low 2; Medium 3
	Total floor number of the building	Total floors in the building
	Number of elevators	Number of elevators in the building
	Number of households	Number of households on each floor

Table 2. The descriptive statistics of partial fields of the data set.

Variables	Mean	Maximum	Minimum	StdDev
2019 GDP (10,000 RMB Yuan)	864.35	1200.00	520	210.78
2018 GDP (10,000 RMB Yuan)	765.44	1050.00	477.72	185.04
2018 PCDI (10,000 RMB Yuan)	45373.73	48972.00	25161.00	7062.07
Number of shopping malls	15.86	21	2	6.21
Number of top hospitals	3.98	7	0	2.58
Number of medical institutions	25.24	30	5	7.11
Number of parks	14.02	17	6	2.82
Number of primary schools	51.81	63	20	12.20
Number of middle schools	23.26	29	10	6.50
Building years (Year)	9.74	57	0	6.13
Average price (RMB Yuan)	16982.50	41031	2012	4507.06
Size of the condo (m^2)	88.41	789.00	15.05	32.75
Total number of floors of the building	23.43	54	1	11.06
Population (10,000)	103.08	128.70	66.07	22.88

4. Simulation and discussion of the results

Python 3.0 was used for the study. The Keras tool, a Python interface for convolutional and RNN neural networks, was used for constructing LSTM and GRU,

and PyTorch, an open source machine learning architecture, was used for constructing Transformer models in the study. 80% of the data in terms of sale date was selected as training data and the remaining 20% of the data was selected as testing data in the traditional machine learning and deep learning approaches. For machine learning models, the training data was used for constructing the models, and these models were used for the testing data. For LSTM, GRU and Transformer models, we compared the simulation results with different parameters to obtain optimal models. For example, we can select a parameter for epochs based on the Loss curve based on the different epochs in Figure 5, while the Loss function of the LSTM model was set to MAE. Later in the paper one can see that when the parameter for epochs is set to 50, the loss value is stabilized. Similarly, we compared the results of the Loss curve obtained with different parameter values for timesteps and batch sizes, and finally set timesteps to 50 and batch size to 1000. The main parameters of LSTM and GRU are listed in Table 3. Table 4 lists Transformer model parameters.

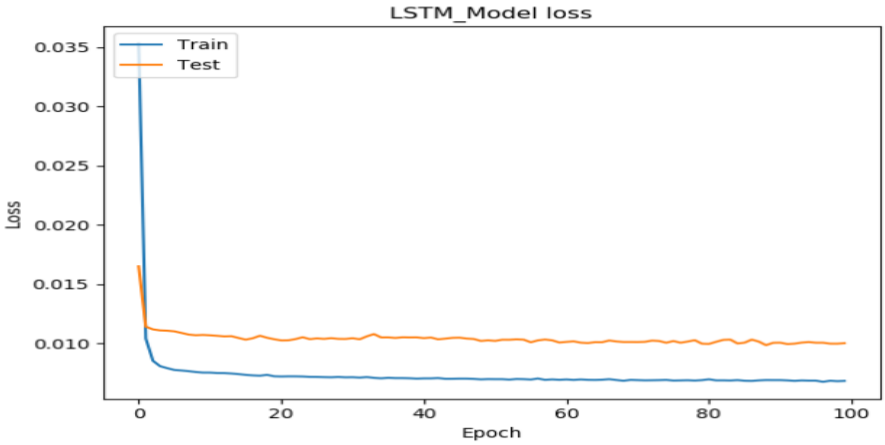


Figure 5. Loss curve for LSTM model.

Table 3. LSTM and GRU model parameters.

Parameters	Description	Value for LSTM	Value for GRU
timesteps	Time window length	50	150
data_dim	Input data dimension	22	22
output_dim	Output data dimension	1	1
train_size	Proportion of training set	0.8	0.8
rnn_units	Number of neural nodes	10	10
dropout	Overfitting parameters	0.5	0.5
batch_size	Number of training samples in a batch	1000	1000
Epochs	Data training rounds	50	50

Table 4. Transformer model parameters.

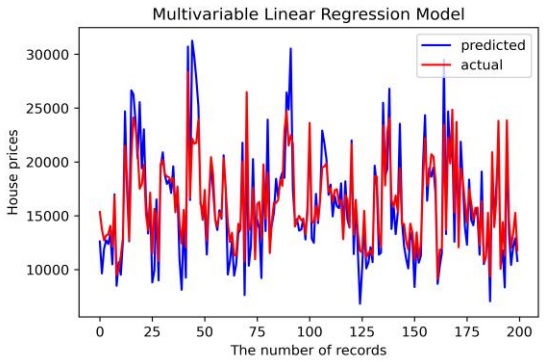
Parameters	Description	Values
Hidden_size	The number of the hidden layers	2
Hidden_unit	The number of the nodes in the hidden layer	64
Hidden_activation	Hidden_activation fuction	GELU
Loss	Loss function	rmse
Optimizer	Optimizer	adam
Learning_rate	Learning rate	0.005
Weight_decay	Weight decay	2
Epoch	Data training rounds	200
Batch_size	Number of training samples in a batch	64

Table 5 presents the results of the models based on the whole data set. It can be seen that the MAE, RMSE and MAPE values of the deep learning models LSTM and GRU are worse than those of the traditional machine learning models. The transformer model is close to the RF model, and they both have the best performance. The prediction results of the two deep learning models GRU and LSTM are very close, mainly because the internal design logic of the two models is similar. Figure 6 provides the experiment results. (Due to the limited space, only the latest two hundred prediction results are shown). The vertical axis represents property prices, the red solid line is the actual sale price, and the blue dotted line is the predicted price. The results shown in Figure 6 are consistent with those presented in Table 5.

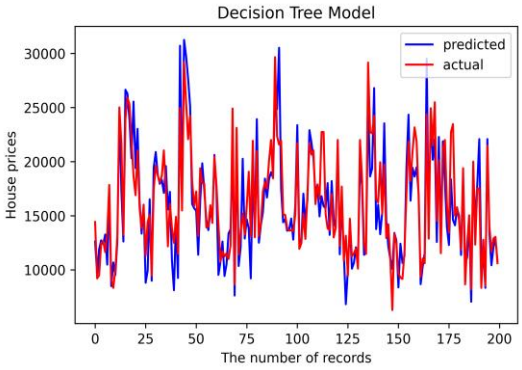
Table 5. The results of models based on whole the data set.

Method	MLR	DT	RF	ARMA	LSTM	GRU	Transformer
MAE	1375.33	1561.70	**1016.02**	2820.70	3835.20	3777.11	**1175.43**
RMSE	2007.56	2345.91	**1563.26**	3651.41	4947.48	4969.12	**1797.71**
MAPE	9.36	10.16	**6.82**	19.40	26.95	25.86	**7.87**

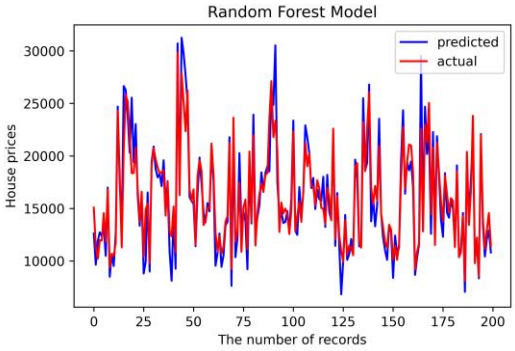
The XGBRegressor model in XGBoost (eXtreme Gradient Boosting), an open-source software library which provides a regularizing gradient boosting framework, was used to calculate the importance of the features from the data set. The top 10 features were then selected according to their importance. They include: 1. Transaction time (1024); 2. Size of the condo (1014); 3. Average price (948); 4. Total floor number of the building (385); 5. Building age (354); 6. The ratio of the number of households to the number of elevators (268); 7. Number of rooms (243); 8. Floor number (166); 9. Decoration (120); and 10. Administrative District (86). The numbers in the parenthesis are the importance values computed by XGBoost.



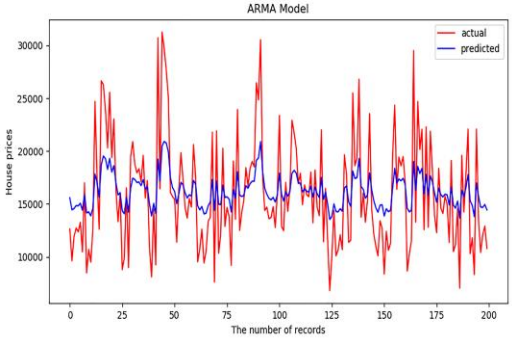
(a) MLR - actual and predicted prices



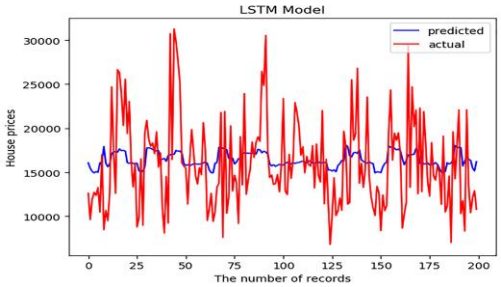
(b) DT - actual and predicted prices



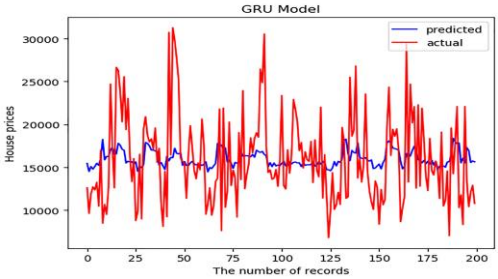
(c) RF - actual and predicted prices



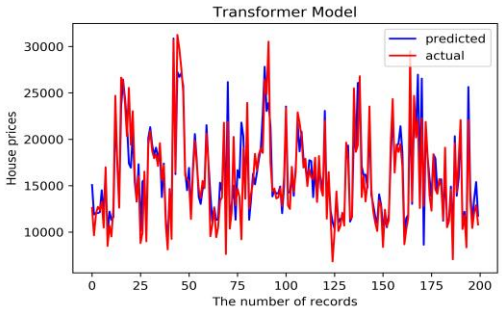
(d) ARMA - actual and predicted prices



(e) LSTM - actual and predicted prices



(f) GRU - actual and predicted prices



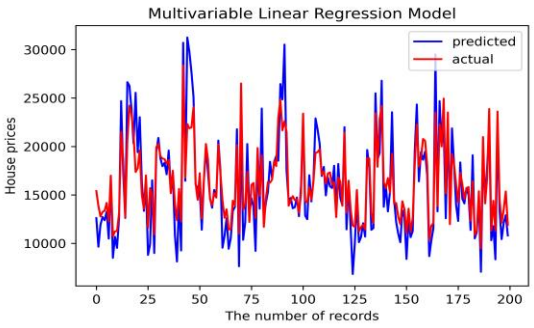
(g) Transformer - actual and predicted prices

Figure 6. Prediction results based on whole the data set.

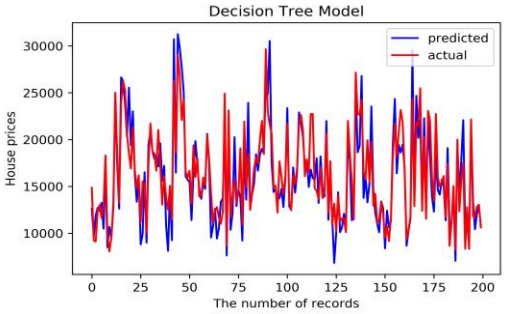
In addition to testing the models with the whole data set, we also tested the same models on the data set with the 10 input variables/features as described above. Table 6 shows the results of the models based on the data set with 10 features. From the results, we can see that the prediction results of the models are consistent with the results based on the whole data set. The results of ARMA are not listed in the table. The Univariate Time Series Forecasting, ARMA, predicts future prices based on its own past values (old prices). Thus, only one variable representing property prices is used in ARMA model, so the results for all variables in the data set and 10 variables in the data set are the same for ARMA. The deep learning models Transformer and RF continue to outperform the other models. The best prediction results are by RF and the second best prediction results are by Transformer. The Transformer deep learning model demonstrates further improvement. The MAE for the RF model, is 1017.86, the RMSE is 1569.66, and the MAPE is 6.72%, which are better than those for the other models. The MAE for the Transformer model, is 1157.81, the RMSE is 1747.51, and the MAPE is 7.63%, which are the better than those for the other models except for RF. Figure 7 presents the experiment results based on data set with 10 features. It is consistent with the results of Table 6.

Table 6. The results of models based on the data set with 10 features.

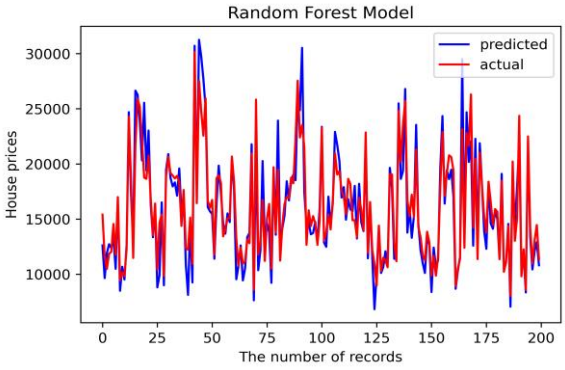
Model	MLR	DT	**RF**	LSTM	GRU	**Transformer**
MAE	1398.73	1568.09	**1017.86**	3835.30	3570.50	**1157.81**
RMSE	2018.44	2316.46	**1569.66**	4984.13	4947.43	**1747.51**
MAPE	9.62	10.15	**6.72**	26.38	25.20	**7.63**



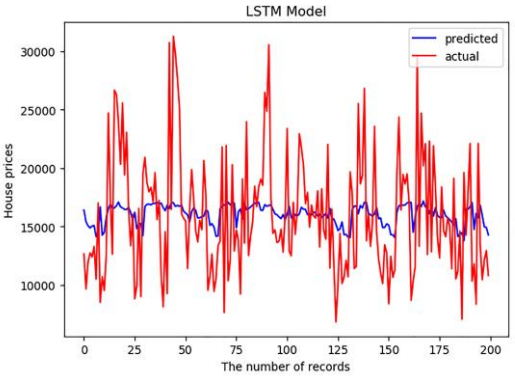
(a) MLR - actual and predicted prices



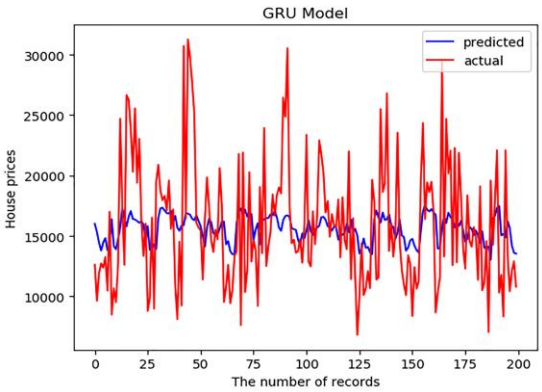
(b) DT - actual and predicted prices



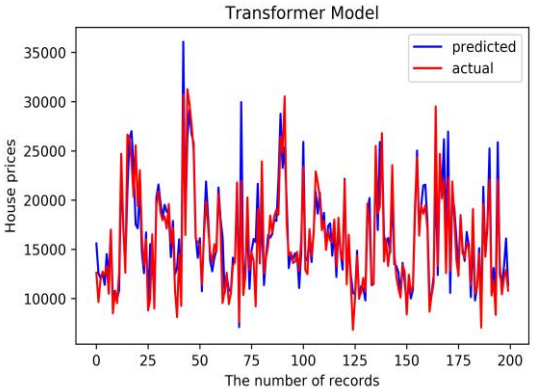
(c) RF - actual and predicted prices



(d) LSTM - actual and predicted prices



(e) GRU - actual and predicted prices



(f) Transformer - actual and predicted prices

Figure 7. The prediction results based on the data set with 10 features.

We calculated regression coefficients using the MLR model. They can be used for modeling the relationship between predicted prices and input variables. The coefficients for Transaction time, Size of the condo, Average price, Total floor number of the building, Building age, the Ratio of the number of households to the number of elevators, Number of rooms, Floor number, Decoration, and Administrative District are-331.18, 3680.26, 407.53, 97.78, -30.40, -444.08, 6.30, 151.50, 24.73, and -142.11, respectively. Among them, Size of the condo, Average price, and Floor number are positive on the price. Transaction time, the Ratio of the number of households to the number of elevators, and Administrative district are negative on the price. Similar results can be obtained using the RF model. The deep learning models cannot compute the importance of the features.

5. Conclusion

This paper compares the performances of three deep learning models with those of regression, ARMA, and two traditional machine learning methods. The objective was to determine the applicability of deep

learning models to property sale price prediction. The selected models were tested under different data scenarios. The results indicate that in all scenarios the RF and Transformer models clearly outperformed the other models. The MAE, RMSE, and MAPE values of the LSTM and GRU models based on deep learning are worse than those of the other four machine learning techniques. It suggests that deep learning models LSTM and GRU may not be an effective modeling solution for property sale price prediction. The accuracy rates of the deep learning Transformer model on the data set with 10 features (obtained through feature reduction) yields a significant improvement compared to those on whole the data set. Our research shows that Transformer models are comparable to RF models, and have the better performance in property price prediction. A possible future research can focus on improving the internal structure of Transformer cells to achieve better prediction results.

References

[1]. Mark, J. and M. Goldberg. Multiple Regression Analysis and Mass Assessment: A Review of the Issues. *Appraisal Journal*, 1988, 56:1, 89–109.

[2]. Antipov, E.A. and E.B. Pokryshevskaya, Mass appraisal of residential apartments: An application of Random forest for valuation and a CART-based approach for model diagnostics. *Expert Systems with Applications*, 2012. 39(2): p. 1772-1778.

[3]. Kilpatrick, J., Expert systems and mass appraisal. *Journal of Property Investment & Finance*, 2011. 29(4/5): p. 529-550.

[4]. Moore, J.W., Performance comparison of automated valuation models. *Journal of Property Tax Assessment and Administration*, 2006. 3(1): p. 43-59.

[5]. Guan, J., J. Zurada, and A.S. Levitan. An Adaptive Neuro-Fuzzy Inference System Based Approach to Real Estate Property Assessment. *Journal of Real Estate Research*, 2008, 30:4, p.395–420.

[6]. McCluskey, W., P. Davis, M. Haran, M. McCord, and D. Ilhatton. The potential of artificial neural networks in mass appraisal: the case revisited. J*ournal of Financial Management of Property and Construction*, 2012. 17(3): p. 274-292.

[7]. Zurada, J., A.S. Levitan, and J. Guan., A Comparison of Regression and Artificial Intelligence Methods in a Mass Appraisal Context. *Journal of Real Estate Research*, 2011. 33(3): p. 349-387.

[8]. Krol, D., T. Lasota, W. Nalepa, and B. Trawinski. Fuzzy System Model to Assist with Real Estate Appraisals. *Lecture Notes in Computer Science*, 2007, 4570, 260–69.

[9]. Taffese, W.Z. Case-Based Reasoning and Neural Networks for Real Estate Valuation. *Proceedings of the 25th Conference on Proceedings of the 25th IASTED International Multi-Conference: Artificial Intelligence and Applications.* Innsbruck, Austria, 2007, 98–104.

[10]. Peterson, S. and A.B. Flanagan. Neural Network Hedonic Pricing Models in Mass Real Estate Appraisal. *Journal of Real Estate Research*, 2009, 31:2, p. 147–164.

[11]. Jadevicius, A., & Huston, S. ARIMA modelling of Lithuanian house price index. *International Journal of Housing Markets and Analysis*. 2015. 8:1: p.135-147.

[12]. Yu, L., C. Jiao, H. Xin, Y. Wang, and K. Wang. Prediction on housing price based on deep learning. *International Journal of Computer and Information Engineering*, 2018. 12(2), 90-99.

[13]. Shi, D., Guan, J., Zurada, J., and Levitan, A, S., 2021, "Predicting Home Sale Prices: A Review of Existing Methods and Illustration of Data Stream Methods for Improved Performance", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Disco*very, e1435, 1-23. https://doi.org/10.1002/widm.1435.

[14]. Quinlan, J. R. *Induction of Decision Trees*. Mach. Learn. 1, 1 (Mar. 1986), 81–106.

[15]. Quinlan, J. R. *C4.5*: *Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[16]. Ho, T.K. Random Decision Forests . *Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal*, QC, 14–16 August 1995. pp. 278–282.

[17]. Whittle, P. *Prediction and Regulation by Linear Least-Square Methods*. University of Minnesota 1983.Press. ISBN 0-8166-1148-3.

[18]. Hochreiter, S, and J. Schmidhuber. Long short-term memory. *Neural Computation*.1997. 9 (8): 1735–1780.

[19]. Dupond, S. A thorough review on the current advance of neural network structures *Annual Reviews in Contro*l. 2019.14: 200–230.

[20]. Cho, K., B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv*:1406.1078.2014.

[21]. Vaswani, A. Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L, Gomez, A.N. Kaiser, L., and Polosukhin, I. (2017-06-12). "Attention Is All You need". arXiv:1706.03762 (HotCloud). 2010