



RESEARCH ARTICLE

House price prediction: A data-centric aspect approach on performance of combined principal component analysis with deep neural network model

Fatemeh Mostofi , Vedat Toğan , Hasan Basri Başağa 

Karadeniz Technical University, Department of Civil Engineering, Trabzon, Turkey

Abstract

High dimensionality and skewness are two intrinsic characteristics of real estate dataset that affects the price prediction accuracy of deep neural network (DNN). The objective of this study is to investigate the effect of skewness in prediction accuracy of combined principal component analysis (PCA) with DNN (PCA-DNN) model. This research follows a threefold approach over a high dimensional and positively skewed real estate price dataset. Firstly, data distribution is to conform with normality using three conventional skewness reduction techniques, namely as square root transformation (SRT), cube root transformation (CRT), and logarithmic transformation (LT) methods. Secondly, the high dimensionality of original, SRT, CRT and LT skewed datasets are to be reduced using PCA. Thirdly, price prediction accuracy of PCA-DNN model over datasets with different skewness levels are to be compared by observing their error values. The results suggest that CRT method can considerably improve both prediction accuracy and computational time of PCA-DNN model, while displaying a good generalization ability. Despite CRT method, SRT and LT methods resulted in high error values and overfitting issues, respectively.

Keywords

Skewness; Kurtosis; House price prediction; Principal component analysis; Deep neural network

Received: 14 June 2021; Accepted: 27 June 2021

ISSN: 2630-5771 (online) © 2021 Golden Light Publishing All rights reserved.

1. Introduction

The introduction of artificial neural network (ANN) in 1940 has attracted several research attention [1], while being evolved into variety of algorithms. Amongst these evolutions, development of deep neural networks (DNNs) by Hinton et al. [2], renewed the research attraction ever since Awad and Khanna [1]. Compared to ANN, DNN offers a simpler sequential model with ability of fine-tuning the neural network result. DNN was used for wide range of applications, such as risk assessment [3,4], damage and fault detection [5,6] and forecasting of

traffic flow [7,8]. Regardless of wide application of DNN within different fields, the body of literature on house price prediction still needs further attention [9].

House price prediction using DNN has only been recently used by few literature, despite its vital role in real estate planning and appraisal applications. Wang et al. [10] adopted DNN model for house price prediction, where the results shown a good match between the predicted values and actual house prices. They highlighted the need for adaptation of non-traditional price prediction

* Corresponding author
Email: fatemmmehhh@gmail.com

approaches with capacity for full utilization of existing big data. Similarly, a good price prediction performance was observed in adoption of a multi-layer feedforward neural network model [11]. A more recent work was assessed DNN for utilization of various sample sizes for rental price prediction [12]. Their results validated the DNN ability in utilization of big data for price prediction, while demonstrating the higher accuracy of DNN for price data deviating from the median. Nevertheless, there have not been found any literature with focus on skewness reduction techniques on prediction accuracy of DNN model.

To improve the prediction accuracy of machine learning (ML) algorithms, Principal Component Analysis (PCA) has been used for reduction of high dimensional features. For example, PCA was used for dimensionality reduction of stock [13–15], air quality [16,17] and building natural period [18] dataset. In these applications, the feature space was first projected into lower subspace, while preserving variance of dataset using a simpler network architecture. In addition to high dimensionality, the skewed data distribution also effects the ML performance.

Skewed distribution is frequently observed in real estate dataset. Removing the skewness in dataset using transformation techniques, can further generalize the data for prediction purposes. The trained machine can learn beyond the skewed characteristics of data, and accordingly perform the prediction over the new dataset, after the transformation of new set into normal distribution. There have found no literature addressing the effect

of different skewness transformation techniques on prediction accuracy of either PCA or DNN. Therefore, since real estate dataset is commonly associated with high dimensionality and skewness, this study follows a threefold approach over a high dimensional and positively skewed real estate price dataset. Firstly, data distribution is to conform with normality using conventional skewness reduction techniques, namely as square root transformation (SRT), cube root transformation (CRT), and logarithmic transformation (LT) methods. Secondly, the high dimensionality of original dataset along with SRT, CRT and LT skewed reduced datasets is to be reduced using PCA, which is a popular unsupervised ML method for dimensionality reduction. Thirdly, price prediction accuracy of PCA-DNN model over dataset with different skewness levels are to be compared by observing their mean absolute percentage error (MAPE), mean squared error (MSE), and root-mean squared error (RMSE) values.

2. Methodology

2.1. Preprocessing of data

This study uses 1381 price records with 19 features obtained through scrapping real estate price units from Trabzon city in north-eastern area of Turkey. Before conducting the analyses, data was cleaned by removing monthly charge feature with more than 60% empty data records. Besides, net area and gross area features were displaying very high correlation (Fig. (1)), and therefore net area feature is removed.



Fig. 1. High correlation between net area and gross area.

Additionally, categorical features are encoded using ordinal encoding method. The resulted cleaned data comprised of 18 variable columns with 1244 price record, described in Table 1.

The cleaned dataset includes $n = 1244$ observation and $m = 18$ variable columns as presented in Eq. (1).

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix} \quad (1)$$

$$a_{ij} \in R^{n \times m}, \forall i = 1, 2, \dots, n \quad \forall j = 1, 2, \dots, m$$

where, \mathbf{A} is cleaned dataset, with n records and m variables.

2.2. Data transformation using skewness reduction techniques

Skewness presents the level of asymmetry in data distribution. It is computed using the third center

moment of distribution, which is being normalized with standard deviation of power three.

$$S(a_j) = \frac{1}{n-1} \times \frac{\sum_{i=1}^n (a_{ij} - \mu_j)^3}{\sigma_j^3} \quad (2)$$

in which, n is number of sample records and $S(a_j)$ is skewness of each data column a_j , while a_{ij} represents i th data record, in j st column. Here, μ_j and σ_j are mean and standard deviation of each data column a_j , which are calculated using Eqs. (3) and (4).

$$\mu_j = \frac{1}{n} \sum_{i=1}^n a_{ij} \quad (3)$$

$$\sigma_j = \left[\frac{1}{n-1} \sum_{i=1}^n (a_{ij} - \mu_j)^2 \right]^{\frac{1}{2}} \quad (4)$$

Table 1. Statistical description of cleaned dataset.

| | mean | std | min | 0.25 | 0.50 | 0.75 | max |
|---------------------|-----------|-----------|--------|--------|--------|--------|---------|
| Price | 452768.49 | 229739.95 | 105000 | 295000 | 400000 | 550000 | 1850000 |
| Gross area | 164.8 | 56.84 | 50 | 135 | 160 | 180 | 570 |
| Number of bedrooms | 3.08 | 0.86 | 1 | 3 | 3 | 3 | 7 |
| Number of saloons | 1.04 | 0.2 | 0 | 1 | 1 | 1 | 2 |
| Building age | 11.14 | 11.95 | 0 | 0 | 10 | 18 | 41 |
| Floor number | 3.91 | 3.05 | 0 | 1 | 3 | 6 | 15 |
| Number of floors | 7.28 | 2.66 | 1 | 5 | 7 | 9 | 16 |
| Number of bathrooms | 1.61 | 0.59 | 1 | 1 | 2 | 2 | 4 |
| Number of balconies | 0.05 | 0.22 | 0 | 0 | 0 | 0 | 1 |
| Furniture | 0.97 | 0.18 | 0 | 1 | 1 | 1 | 1 |
| Amenities | 0.68 | 0.47 | 0 | 0 | 1 | 1 | 1 |
| Credit availability | 0.1 | 0.3 | 0 | 0 | 0 | 0 | 1 |
| Video call | 0.53 | 0.5 | 0 | 0 | 1 | 1 | 1 |
| Swap | 0.84 | 0.37 | 0 | 1 | 1 | 1 | 1 |
| Heating | 0.79 | 1.66 | 0 | 0 | 0 | 0 | 6 |
| Condition | 0.54 | 0.8 | 0 | 0 | 0 | 1 | 2 |
| Seller agency | 0.18 | 0.46 | 0 | 0 | 0 | 0 | 2 |
| Neighbourhood | 21.72 | 12.53 | 0 | 9 | 26 | 31 | 41 |

In this study, the skewness of data columns is calculated and a threshold of -1 and 1 are set for selection of highly skewed numerical features. Therefore, continuous data columns with skewness greater than 1 and less than -1 are to be transformed, using three skewness reduction levels with different skewness reduction powers. Additionally, Kurtosis is a quantitative measurement of thickness in distribution tail. Thence, for measuring the tail of the distributions, Kurtosis is computed using forth center moment of distribution and is normalized with standard deviation to the power of four, using Eq. (5).

$$K(a_j) = \frac{1}{n-1} \times \frac{\sum_{i=1}^n (a_{ij} - \mu_j)^4}{\sigma_j^4} \quad (5)$$

in which, $K(a_j)$ is kurtosis of each vector column. Based on the values obtained from Eqs. (3) and (4), kurtosis value of three, resulted from Gaussian distribution, and its skewness form is called mesokurtic.

First transformation method adopted in this study is square root transformation (SRT) method, which reduces data skewness using Eq. (6).

$$A_{Square_Root} = A^{\frac{1}{2}} \quad (6)$$

where, A is a specific vector column in original dataset and A_{Square_Root} is the modified vector column, having its skewness adjusted using square root skewness reduction method. Similarly, the cube root transformation (CRT) method, is a stronger reduction method than SRT, which is used over the skewed columns, as displayed in Eq. (7).

$$A_{Cube_Root} = A^{\frac{1}{3}} \quad (7)$$

where, A_{Cube_Root} is vector column in original dataset, whereby its skewness adjusted using cube root skewness reduction method. Finally, a stronger logarithmic transformation (LT) method is used for reducing the price skewness using Eq. (8).

$$A_{log} = \log(A) \quad (8)$$

where, A_{log} is vector column in original dataset, which is adjusted using logarithmic function. Despite being a strong transformation method, and

therefore causing major effect on distribution shape of data, the downside of LT is its inability in handling zero or negative values.

2.3. Dimensionality reduction using PCA

Real estate dataset is often presented with several features for explanation of a resultant price unit. So, high dimensional data records are to become soon sparse in large dimensional space of several explanatory features. In this regard, PCA is an unsupervised ML tool for feature extraction and dimensionality reduction of high dimensional dataset. Since PCA is unsupervised ML method, therefore only $d = 17$ number of feature columns are to be processed, using Eq. (9), without learning from resultant price column.

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix} \quad (9)$$

$$a_{ij} \in R^{n \times d}, \quad \forall i = 1, 2, \dots, n \quad \forall j = 1, 2, \dots, d$$

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$$y_i \in R^n, \quad \forall i = 1, 2, \dots, n$$

where, \mathbf{X} is a $(n \times d)$ matrix, presenting explanatory feature describing the separated price column \mathbf{Y} .

The first step of PCA is associated with normalization of dataset. This makes all the features unit free, while placing them on the origin of space. Afterwards, the covariance matrix is to be calculated through Eq. (10).

$$\mathbf{\Sigma} = \begin{bmatrix} \varsigma_{11} & \cdots & \varsigma_{1d} \\ \vdots & \ddots & \vdots \\ \varsigma_{d1} & \cdots & \varsigma_{dd} \end{bmatrix} \quad (10)$$

$$\varsigma_{kr} \in R^{d \times d}, \quad \forall k = 1, 2, \dots, d \quad \forall r = 1, 2, \dots, d$$

where, $\mathbf{\Sigma}$ is $(d \times d)$ matrix which shows the correlation among the several column features, which is calculated using Eq. (11).

$$\mathbf{\Sigma} = \frac{1}{n} \sum_{i=1}^n \mathbf{X}_i \mathbf{X}_i^T \quad (11)$$

Afterwards, the eigenvalues and eigenvectors are to be calculated by solving Eq. (12).

$$e_{ij} (\Sigma - \lambda) = 0 \quad (12)$$

in which, e_{ij} and λ are standing for eigenvector and eigenvalue. Then, the eigenvectors are to be sorted based on the descending order of eigenvalues. Afterwards, the sorted eigenvectors are to be used for computation of principal components, Eq. (13).

$$P_k = \sum_{r=1}^d e_{rk} X_{ij} \quad (13)$$

where, P_k is kth principal component, which is the projection of matrix \mathbf{X} into lower subspace.

Following the computation of principal components, selected k principal components are to be fed into DNN for prediction of price values, using PCA-DNN model.

In this study mean absolute error (MAE) is adopted for loss calculation within PCA-DNN structure, using Eq. (14).

$$MAE = \frac{\sum_{i=1}^n |Y_i - Y_i^P|}{n} \quad (14)$$

The activation function used for PCA-DNN model is softsign activation function, as stated in Eq. (15).

$$\varphi = \frac{Z}{1 + |Z|} \quad (15)$$

So, Eq. (15) is used for activation of all layers except last layer which is activated using linear activation. Besides, the Adam is used as optimization function for all the layers, except the last layer. The described loss, activation and optimization functions are to be used within the PCA-DNN model, comprised of 5 layers, having 7 neurons in each layer as shown in Fig. (2).

For evaluation of PCA-DNN performance with respect to adopted transformation methods, three error measurements, namely as MAPE (Eq. (16)), MSE (Eq. (17)), and RMSE (Eq. (18)) are used.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{\hat{y}_i} \times 100 \quad (16)$$

$$MSE = \frac{1}{n} \sum_i |y_i - \hat{y}_i|^2 \quad (17)$$

$$RMSE = \sqrt{MSE} \quad (18)$$

where, \hat{y}_i and y_i are the predicted and actual price values, respectively.

3. Results and discussions

Based on Table 1., the resulted cleaned dataset possesses two continuous numerical variables, namely as price, and gross area, whereby both are positively skewed with high degree of skewness and fat tails. All three skewness reduction methods are considerably influenced the exiting distributions observed in price and gross area features. Therefore, using three discussed skewness reduction methods, the existing skewness in price and gross area variables are reduced, and resulted distributions are presented in Fig. (3).

As it is shown in Fig. (3), all three transformation methods are reducing the existing skewness of original dataset into range -1 to 1. Amongst the transformation methods, LT is resulted in strongest transformation, changing the data distribution very close to normal distribution. Accordingly, Skewness and Kurtosis of resulted transformed data using SRT, CRT and LT methods (Eqs.(6)-(8)) are presented in Table 2.

Based on Table 2, the transformation of original dataset, using SRT method is changed price and gross area skewness from highly skewed to moderately positive skewed distribution. Besides, using SRT method is highly adjusted the fat tailed Kurtosis closer to three.

Furthermore, Table 3 shows structured DNN and PCA-DNN architectures used for all four datasets, with respect to selected hyperparameters.

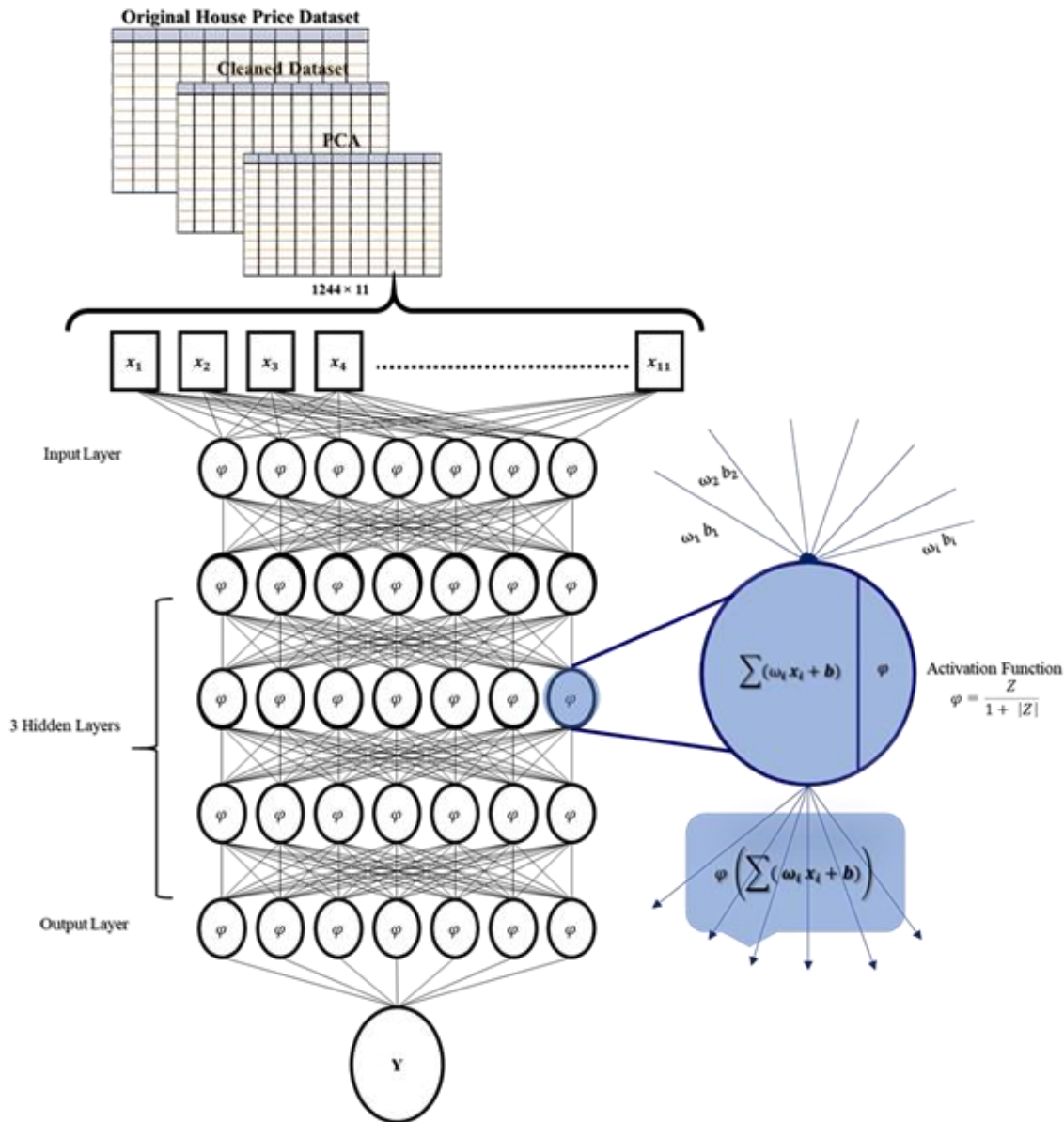


Fig. 2. PCA-DNN model

Table 2. Skewness and Kurtosis of transformed dataset

| Dataset | Skewness | | | | Kurtosis | | | |
|------------|----------|-------|-------|-------|----------|------|------|------|
| | Original | SRT | CRT | LT | Original | SRT | CRT | LT |
| Price | 1.714 | 0.868 | 0.607 | 0.101 | 4.48 | 1.22 | 0.67 | 0.17 |
| Gross area | 1.928 | 0.929 | 0.624 | 0.037 | 6.09 | 2.54 | 1.97 | 1.56 |

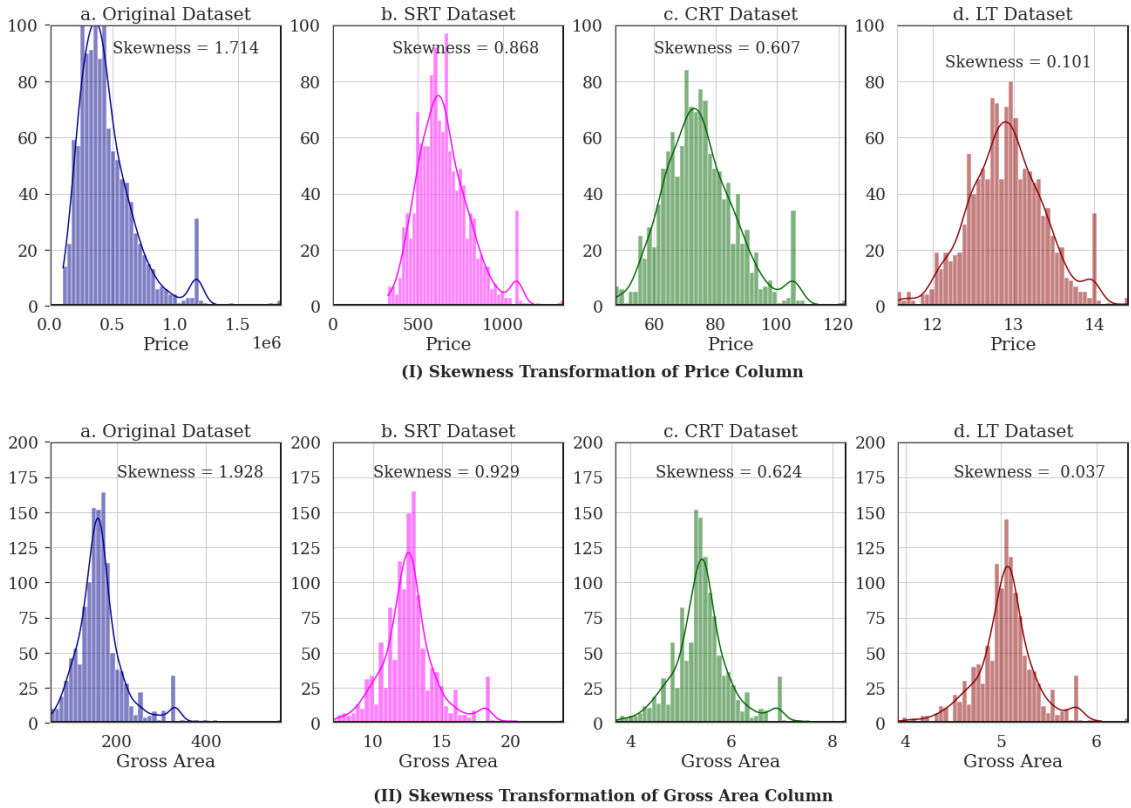


Fig. 3. Distribution of price (I) and gross area (II) columns (Original dataset (a), SRT adjusted dataset (b), CRT adjusted dataset (c) and LT adjusted dataset (d)).

Table 3. Hyperparameters used in DNN and PCA-DNN models

| Model Parameter | DNN Model | PCA-DNN Model |
|---|-----------------|-----------------|
| Number of Layers | 5 | 5 |
| Number of Neurons (except output layer) | 7 | 7 |
| Number of Neurons (output layer) | 1 | 1 |
| Total trainable parameters | 302 | 260 |
| Activation function (except output layer) | Softsign | Softsign |
| Activation function (output layer) | Linear | Linear |
| Patience Number | 10 | 10 |
| Monitor value | Validation Loss | Validation Loss |
| Loss Function | MEA | MEA |
| Optimization Function | Adam | Adam |
| Total data size | 1244 × 17 | 1244 × 11 |
| Training size | 995 × 17 | 995 × 11 |
| Validation size | 124 × 17 | 124 × 11 |
| Test size | 125 × 17 | 125 × 11 |

Using the hyperparameters described in Table 3, the observed performance for training of original, SRT, CRT and LT dataset, using DNN model is presented in Table 4.

It can be argued from Table 4 that CRT followed by LT is resulted in shortest training time, considering observed CPU and Wall times. Here, SRT is shown the poorest training time, with highest MAPE, MSE and RMSE values. Despite the poor performance observed by SRT method, LT followed by CRT are outperformed the performance of original dataset, considering all observed error values. Adding the PCA feature extraction to the training model, persevering 80% of variance in dataset, allows the model to be trained using 11 extracted features. Accordingly, Table 5 shows training results of adopted PCA-DNN model over the four sets of dataset.

As seen in Table 5, CRT shows significantly better performance compared to other three models, in terms of both accuracy and computational time. In addition to CRT, LT also improves the performance of original dataset, showing higher accuracy and smaller computation time compared to training the original dataset. Like DNN model, the observed performance of PCA-DNN model also highlights the poor performance of SRT transformation method. The unsupervised and supervised training performance of four sets of dataset are presented in Fig. (4).

Unsupervised and supervised learning patterns of all four dataset show a good training performance, even though SRT model has shown poor accuracy. Furthermore, the error values from validation and training losses for all the four dataset is observed, which the results are presented in Fig. (5).

Table 4. Training results of DNN model

| | Original Dataset | SRT Dataset | CRT Dataset | LT Dataset |
|-------------------|------------------|-------------|-------------|------------|
| CPU times | 3.19 s | 4.29 s | 2.21 s | 3.96 s |
| Wall time | 2.7 s | 3.91 s | 1.92 s | 4.43 s |
| MAPE Unsupervised | 50.17 % | 351.1 % | 43.15 % | 23.45 % |
| MAPE Supervised | 25.95 % | 207.65 % | 16.92 % | 18.03 % |
| MSE Unsupervised | 31.46 % | 1540.89 % | 23.27 % | 6.87 % |
| MSE Supervised | 8.42 % | 539.0 % | 3.58 % | 4.07 % |
| RMSE Unsupervised | 5.61 % | 39.25 % | 4.82 % | 2.62 % |
| RMSE Supervised | 2.9 % | 23.22 % | 1.89 % | 2.02 % |

Table 5. Training results of PCA-DNN model

| | Original Dataset | SRT Dataset | CRT Dataset | LT Dataset |
|-------------------|------------------|-------------|-------------|------------|
| CPU times | 3.46 s | 2.33 s | 3.97 s | 5.43 s |
| Wall time | 3.01 s | 1.99 s | 4.35 s | 6.83 s |
| MAPE Unsupervised | 46.31 % | 310.43 % | 36.42 % | 23.0 % |
| MAPE Supervised | 22.41 % | 174.45 % | 15.55 % | 23.26 % |
| MSE Unsupervised | 26.81 % | 1204.57 % | 16.58 % | 6.61 % |
| MSE Supervised | 6.28 % | 380.41 % | 3.02 % | 6.76 % |
| RMSE Unsupervised | 5.18 % | 34.71 % | 4.07 % | 2.57 % |
| RMSE Supervised | 2.51 % | 19.5 % | 1.74 % | 2.6 % |

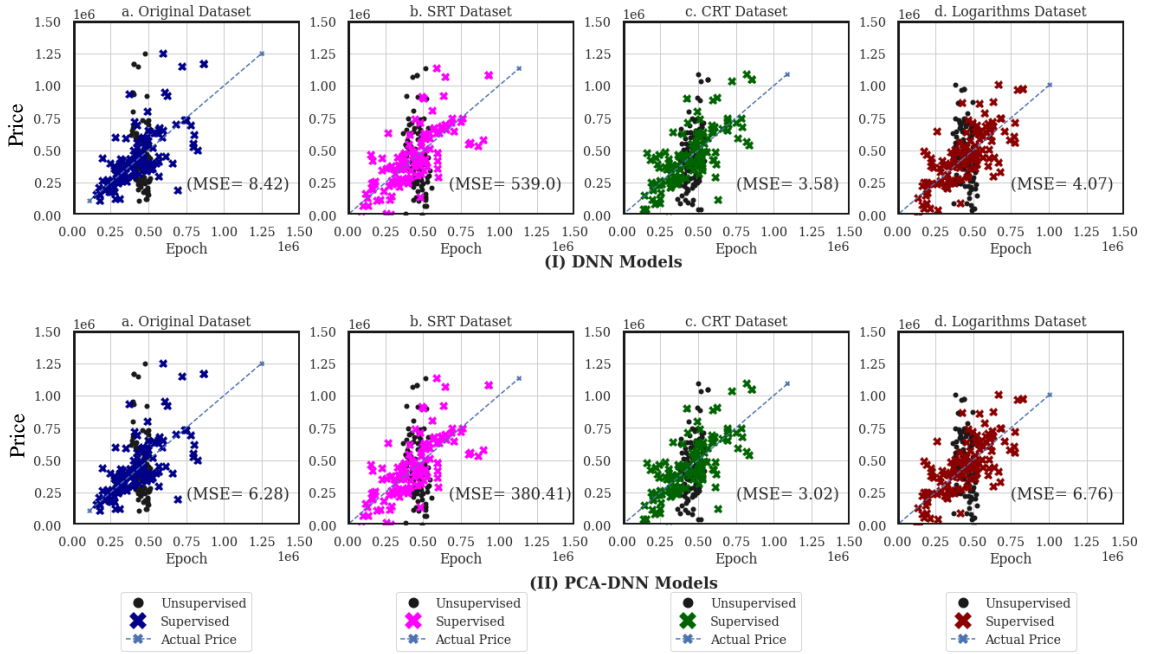


Fig. 4. Supervised and unsupervised training performance of original (a), SRT (b), CRT (c), and LT (d) dataset, adopting DNN (I) and PCA-DNN (II) models.

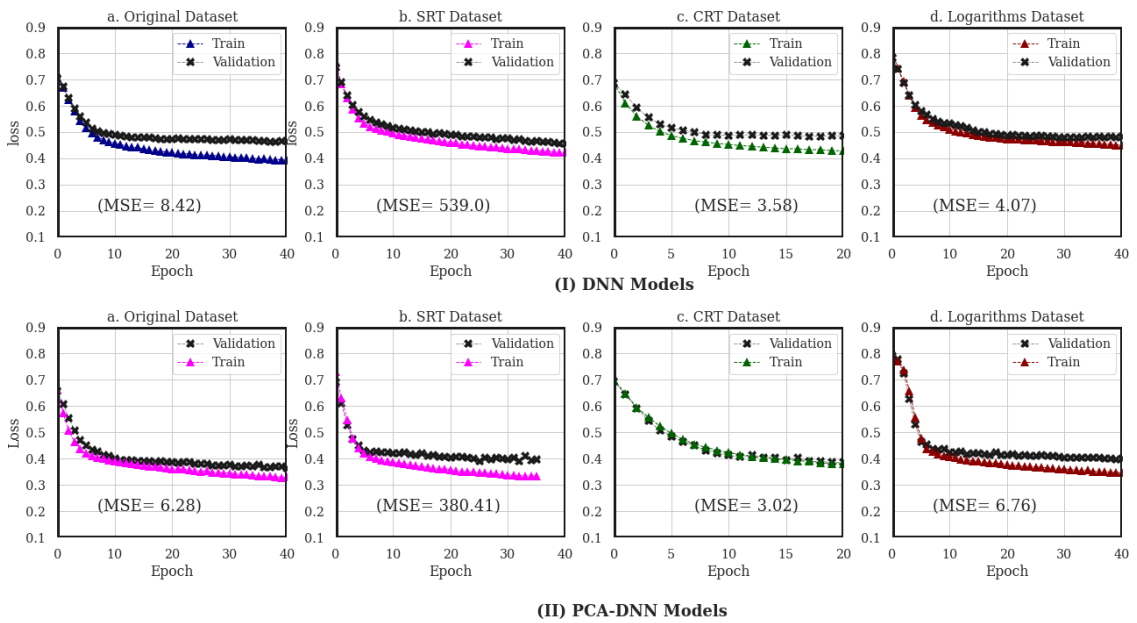


Fig. 5. Observed L for training and validation losses of original (a), SRT (b), CRT (c), and LT (d) datasets, using DNN (I) and PCA-DNN (II) models.

Fig. (5) demonstrates the good match between the training and validation loss values using CRT method over PCA-DNN model. This shows that PCA-DNN model can well generalized with CRT method, without encountering overfitting issue. Despite PCA-DNN model, DNN model shows possibility for overfitting for all of four dataset, except LT method.

In brief, transforming data using CRT method closer to Gaussian normal distribution shows good improvement in terms of prediction accuracy, computational time, and generalization ability of model. In other words, using CRT method, with medium power for transformation the distribution shape of dataset, can improves the prediction accuracy of positively skewed dataset, using PCA-DNN model.

4. Conclusion

This study evaluates the effect of skewness reduction techniques on high dimensional and positively skewed real estate dataset for price prediction through PCA-DNN model. The high dimensionality of dataset is addressed with PCA, while highly skewed data columns are transformed using three commonly used distribution transformation methods, namely as SRT, CRT and LT. The results indicate that CRT method can considerably improve the prediction accuracy of PCA-DNN model, while using lesser computational time. Moreover, CRT has not displayed the slightest overfitting issue and can well generalized for the other dataset, using PCA-DNN model. Besides, SRT method shown a very poor performance, in terms of both prediction accuracy and computational time, as it impairs the ML performance compared to results obtained from training the original dataset. Lastly, this study is conducted over positively skewed dataset, and thus it is suggested to repeat the analyses for negatively skewed dataset, with different shape of Kurtosis.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

- [1] Awad M, Khanna R. *Efficient Learning Machines*. Apress, Berkeley, CA, USA, 2015.
- [2] Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. *Neural Computation* 18: 1527–1554.
- [3] Paltrinieri N, Comfort L, Reniers G (2019) Learning about risk: Machine learning for risk assessment. *Safety Science* 118: 475–486.
- [4] Zhang X, Huang C, Chen Y. Method of highway risk assessment and accident quantity prediction based on multi-source heterogeneous data and deep neural network. 13th Asia Pacific Transportation Development Conference, May 27–30 2020, Shanghai, China.
- [5] Jia F, Lei Y, Lin J, Zhou X, Lu N (2016) Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mechanical Systems and Signal Processing* 72–73: 303–315.
- [6] Lim S, Chi S. Bridge damage prediction using deep neural network. *International Conference on Computing in Civil Engineering*, June 17-19 2019, Atlanta, Georgia, USA.
- [7] Qu L, Li W, Li W, Ma D, Wang Y (2019) Daily long-term traffic flow forecasting based on a deep neural network. *Expert Systems with Applications* 121: 304–312.
- [8] Zhou L, Chen XM. Short-term forecasting of traffic flow and speed: a deep learning approach. 18th COTA International Conference of Transportation Professionals, July 5-8 2018, Beijing, China.
- [9] Rafiei MH, Adeli H (2016) A novel machine learning model for estimation of sale prices of real estate units. *Journal of Construction Engineering and Management* 142: 04015066.
- [10] Wang F, Zou Y, Zhang H, Shi H. House price prediction approach based on deep learning and ARIMA model. *IEEE 7th International Conference on Computer Science and Network Technology*, Oct. 19-20 2019, Dalian, China.
- [11] Jiang Z, Shen G. Prediction of house price based on the back propagation neural network in the keras deep learning framework. 6th International

- Conference on Systems and Informatics, 2-4 Nov. 2019, Shanghai, China.
- [12] Seya H, Shiroy D (2021) A comparison of residential apartment rent price predictions using a large data set: kriging versus deep neural network. *Geographical Analysis* 0: 1–22.
- [13] Liu H, Wang J (2011) Integrating independent component analysis and principal component analysis with neural network to predict chinese stock market. *Mathematical Problems in Engineering* 382659.
- [14] Zahedi J, Rounaghi MM (2015) Application of artificial neural network models and principal component analysis method in predicting stock prices on Tehran Stock Exchange. *Physica A: Statistical Mechanics and Its Applications* 438: 178–187.
<https://doi.org/10.1016/j.physa.2015.06.033>.
- [15] Berradi Z, Lazaar M (2019) Integration of principal component analysis and recurrent neural network to forecast the stock price of casablanca stock exchange. *Procedia Computer Science* 148: 55–61.
- [16] Kumar A, Goyal P (2013) Forecasting of air quality index in delhi using neural network based on principal component analysis. *Pure and Applied Geophysics* 170: 711–722.
- [17] Azid A, Juahir H, Toriman ME, Kamarudin MKA, Saudi ASM, Hasnam CNC, et al (2014) Prediction of the level of air pollution using principal component analysis and artificial neural network techniques: A case study in Malaysia. *Water, Air, and Soil Pollution* 225: 1–14.
- [18] Kuźniar K, Waszczyszyn Z (2006) Neural networks and principal component analysis for identification of building natural periods. *Journal of Computing in Civil Engineering* 20: 431–436.