Optimizing the Shanghai Second-Hand Housing Price Prediction with CNN Model

Xiaoyang Li[1,a,*]

[1]School of Mathematics and Statistics, Beihua University, Jilin City, Jilin Province, China

[a]bhdx_lxy@163.com

[*]Corresponding author

Keywords: Second-Hand Housing Price Forecasting, Convolutional Neural Network (CNN), Model Optimization, Spatial Feature Mining

Abstract: This study focuses on optimizing the prediction of Shanghai's second-hand housing prices using a Convolutional Neural Network (CNN) model. Traditional second-hand housing price prediction methods have limitations in handling high-dimensional data and mining spatial correlations, leading to suboptimal accuracy. To address this, we collected 2024 Shanghai second-hand housing data from Kaggle, involving 171048 samples and nearly 30 features, which were preprocessed (including cleaning, encoding, and dimensionality reduction). A 1D-CNN model was constructed, and optimization strategies were implemented: deepening the network structure (adding convolutional and fully connected layers), strengthening regularization (incorporating L2 regularization and Dropout), and testing combinations of activation functions (ReLU, LeakyReLU, ELU) and optimizers (Adam, RMSprop). Experimental results show that the optimized model (ELU-Adam combination) outperforms the original CNN model, with MSE reduced by 59.0% (to 8362.02), MAE reduced by 42.7% (to 54.46), and R^2 improved to 0.9546, indicating stronger predictive accuracy and generalization ability. This research provides a reliable decision-making basis for stakeholders in the real estate market.

1. Introduction

Real estate industry serves as a crucial pillar of the national economy, and its stable development is closely linked to social wellbeing and economic structure. Among them, the Second-hand housing market, by directly reflecting the urban housing supply-demand relationship and residents' real housing needs, has become a core indicator for measuring the health of the Real estate market. As urbanization enters the middle and late stages, the increment of new housing has slowed down, while the proportion of second-hand housing transactions continues to rise. Its price fluctuations not only affect residents' asset allocation and home purchase decisions but also relate to the stability of the financial system and the effectiveness of macroeconomic regulation. Therefore, accurate prediction of Second-hand housing prices holds significant practical importance.

Second-hand housing prices are influenced by the interplay of multiple factors, exhibiting significant complexity and nonlinear characteristics. There are many prediction methods. Reference [1] comprehensively considers the attributes of second-hand houses and surrounding environmental factors, and uses the classic algorithm GBDT regression algorithm in the field of Ensemble learning to predict the transaction prices of second-hand houses in Beijing. Reference[2] establishes a Random Forest (RF) model, visually analyzes the results, re-evaluates model variables, studies the key factors affecting housing prices, and optimizes the model to predict housing prices. Reference [3] adopts the Linear regression statistical model to explore the factors affecting the price of second-hand houses. In different regions and under different circumstances, the most influential factor on housing prices is whether the house is renovated, followed by whether there are high-rise buildings. Reference [4] establishes feature data and label data based on different characteristics such as region, house age, nearby geographical conditions, cultural and transportation factors. House prices are predicted and analyzed using LightGBM regression model and LGBMRegressor algorithm. Reference [5] predicts

  DOI: 10.25236/icceme.2025.007

house prices through Artificial Neural Network (ANN), Support Vector Machine (SVM), and Classification and Regression Tree (CART). The performance of 16 different models was evaluated using metrics such as Linear correlation and Mean absolute error, and it was found that artificial neural networks outperformed other models in both larger and smaller clusters.

Traditional housing price prediction methods rely on manual feature selection, struggle to handle high-dimensional multimodal data, and fail to effectively uncover implicit spatial correlation patterns, resulting in limited prediction accuracy. Convolutional Neural Network (CNN) excels at capturing spatial correlations at different scales through multi-layer convolution[6], enabling precise quantification of location value's impact on housing prices. This study focuses on the application of CNN in Second-hand housing prices prediction, constructing relevant models to enhance modeling performance. Empirical comparisons with traditional methods highlight its advantages, providing accurate decision-making support for homebuyers, real estate developers, and others, thereby promoting the stable development of the Real estate market.

2. Second-hand housing price prediction Based on CNN model

The data in this study is sourced from kaggle website, focusing on Shanghai's second-hand housing prices in 2024. After data collection, the paper first conducts Data cleaning. The final Data set consists of 171,048 samples, with nearly 30 features undergoing Feature processing to obtain Original feature. These include: Numerical Feature, Boolean feature , Low-cardinality categorical feature , - High-cardinality feature (Title, Floor distribution, Property type, Property right nature, District).To protect privacy and prevent Dimensionality explosion caused by High-cardinality feature, the High-cardinality feature was removed. To facilitate subsequent Second-hand housing price prediction, Partial Feature in the Original feature was encoded as shown in Table 1 below.

Table 1 Partial Feature encoding Reference Table.

Encoded Feature	Feature Type	Original feature Category/Value
F18	Category Feature	Unfinished
F19	Category Feature	Simple decoration
F20	Category Feature	Fine decoration
F21	Category Feature	Luxury decoration
F22	Category Feature	40-year property right
F23	Category Feature	50-year property right
F24	Category Feature	70-year property right
F25	Category Feature	More than two years
F26	Category Feature	More than five years

Among them, Numerical Feature (Number of living rooms, Number of halls, Number of bathrooms, Total area, Construction year, Total number of floors in residential buildings, Number of households in the community, Community greening rate, Property management fees, Average price of the community, these original features correspond to the encoded features F1-F10 respectively, These Original feature are Numerical feature, where the categories/values of Original feature correspond to Standardized value, which are not listed one by one in the table.) Boolean feature(South-facing, South and north, Near the subway, Adequate parking spaces, Regular layout, Many people's attention, With elevator, these original features correspond to the encoded features F11-F17 respectively. These Original feature are Boolean feature, where the Original feature categories/values correspond to 0/1, and thus are not listed individually in the table.) Decoration, Property right years, Property certificate years, these original features correspond to the encoded features F18-F26 respectively. We perform one-hot encoding on categorical features .

3. Convolutional Neural Network Model

In deep learning, Convolutional Neural Networks (CNNs), as a type of feedforward neural network based on convolutional structures capable of deep optimization, represent one of the most important deep learning frameworks.

In computer vision, two-dimensional convolutional neural networks are commonly used, such as references[7-12], while one-dimensional convolutional neural networks (1D-CNNs) extract local features from sequential data through convolution algorithms. Their core advantage lies in capturing relational patterns between features while preserving the temporal or structural characteristics of the sequence[13-15]. For the task of second-hand house price prediction, this model treats housing features as structured sequential data and employs 1D convolution operations to uncover latent relationships between features (such as the combined influence of "Total area-Number of living rooms-Average price of the community"), ultimately achieving accurate price predictions. The overall model architecture consists of an input layer, convolutional layer, pooling layer, fully connected layer, and output layer. Each layer progressively performs feature extraction and prediction through mathematical transformations, with the specific structure illustrated in Figure 1.

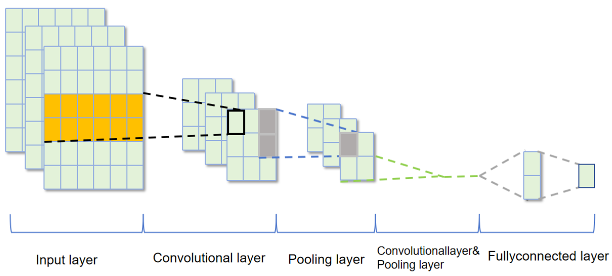

Figure 1 Convolutional Neural Networks.

The input layer receives preprocessed second-hand housing feature data, which is essentially structured sequential data. Let the input data be $X \in R^{s \times d}$, where: s is the length of the feature sequence (i.e., the total number of features involved in modeling); d is the dimensionality of each feature (here d=1, as individual features are scalars); X_t represents the value of the t-th feature in the sequence (e.g., standardized values of individual features such as "Total area" or "Construction year"). The input sequence takes the form:$X = [x_1, x_2, x_3, \dots, x_t, \dots, x_s]$, where X_t is the preprocessed value of the t-th feature (Numerical Feature is standardized, Boolean feature is 0/1, and categorical features undergo One-hot encoding). For sequential data, it will be processed through a one-dimensional convolution algorithm to reflect it in the convolutional layer:

$$a_j = f_1(X * W_j + b_j) \tag{1}$$

$$f_1(z) = max(z, 0) \tag{2}$$

The convolutional layer contains n_f convolutional kernels (kernels in this model, $n_f = 64$), each with a size of $W_j \in R^{m \times 1}$ (where is the kernel size, and =3 in this model). Where a_j represents the computation of the j-th feature map, where: "*" denotes the one-dimensional convolution operation, which involves element-wise multiplication of the kernel with a local window (of length m) of the input sequence followed by summation; b_j is the bias term for the j-th convolutional kernel; $f_1(z)$ is the activation function, employing ReLU to introduce nonlinear transformation. The convolutional layer outputs n_f feature maps, each capturing local features of the input sequence under different convolutional kernels, with a dimensionality of $(s - m + 1) \times n_f$ (the sequence length is reduced due to kernel sliding).

The pooling layer downsamples the feature maps output by the convolutional layer, preserving key information while reducing computational load. This model employs max pooling, which consists of two steps:

1) Local max pooling: The feature maps are divided into windows of size 2, and the maximum value within each window is selected. The formula is:

$$a_{b,j}(k) = \max\left(a_j(2k-1), a_j(2k)\right) \tag{3}$$

Where a_j represents the j-th feature map output by the convolutional layer, and k is the index of the pooled sequence. After this operation, the sequence length is reduced to half of the original.

2) Flattening: The multi-dimensional feature maps after pooling are transformed into a one-dimensional vector to serve as input for the fully connected layer. If the length of the pooled feature map is the length of s' ,the flattened vector becomes $n_f \times s'$.

The fully connected layer integrates the global features output by the pooling layer through a multi-layer neural network, achieving a nonlinear mapping from features to predicted values. This model consists of two fully connected layers:

1) The first fully connected layer: Given the input a_p as a flattened feature vector, the weight matrix $W_f \in R^{n_1 \times (n_f \times s')}$ ($n_1 = 64$ is the number of neurons), the output is:

$$a_{f1} = f_2(a_p W_f + b_f) \tag{4}$$

Where b_f is the bias term, and $f_2(z)$ still uses the ReLU activation function(same as the above equation).

2) Dropout layer: To mitigate overfitting, a Dropout operation is introduced after the first fully connected layer, randomly dropping 20% of neuronal connections to preserve feature diversity.

3) The second fully connected layer: The weight matrix $W_{f2} \in R^{n_2 \times n_1}$ (n=32), the output is:

$$a_{f2} = f_2(a_{f1} W_{f2} + b_{f2}) \tag{5}$$

The output layer maps the high-dimensional features from the fully connected layer to the final house price prediction value, using a single neuron without an activation function (since house prices are continuous values):

$$\tilde{y} = f_3(a_{f2} W_o + b_o) \tag{6}$$

Where: $W_o \in R^{1 \times n_2}$ is the output layer weight matrix; b_o is the output layer bias term; $f_3(z) = z$ (identity function), directly outputting the predicted house price $\tilde{y}$.

4. Original CNN Model

The original model adopts a simplified architecture for performance comparison. Input layer: Receives preprocessed Feature vector (dimension 64), Reshape to (64,1) for 1D convolution compatibility. Convolutional layer: Single Conv1D layer (64 Filters, Kernel size=3, Stride 1, Activation function ReLU), extracting Local features. Pooling layer: MaxPooling1D (Pool size=2), compressing Feature dimension to 32. Fully connected layer: Two Dense layers (64→32 Neurons, ReLU activation) with Dropout (0.2) to mitigate Overfitting. Output layer: One Neuron (Linear activation) for Predicted price output. Model compilation: Optimizer set to Adam (Learning rate 0.001), Loss function as MSE.

4.1 Optimized model Design

To address the shortcomings of the original model, the optimization strategies are as follows.

Table 2 The structure of the Optimized model

Layer	Optimized model (using ELU-Adam as an example)	Original Model
Convolutional layer1	Conv1D(64,3)+ELU+MaxPooling1D(2)	Conv1D(64,3)+ReLU+MaxPooling1D(2)
Convolutional layer2	Conv1D(128,3)+ELU+MaxPooling1D(2)	-
Fully connected layer1	Dense(128,L2 Regularization)+ELU+Dropout(0.3)	Dense(64,ReLU)+Dropout(0.2)
Fully connected layer2	Dense(64,L2 Regularization)+ELU	Dense(32,ReLU)
Output layer	Dense(1,Linear activation)	Dense(1,Linear activation)

1) We deepen the network structure: add 1 convolutional layer and 1 fully connected layer to enhance deep feature extraction capability. 2) We strengthen regularization: Incorporate L2

Regularization (λ=0.001) and Dropout (0.3) to mitigate Overfitting. 3) We diversify Activation function: Compare ReLU, LeakyReLU (Alpha=0.1), and ELU (Alpha=1.0) to adapt to the housing price data distribution. 4) We optimizer selection: Test Adam and RMSprop (Rho=0.9) to validate the effect of Adaptive learning rate. The structure of the Optimized model is shown in Table 2.

4.2 Experimental Parameter Settings

Among them, Epochs: Optimized model 100 epochs, original model 30 epochs (terminated by Early stopping mechanism), Batch size: 32 (Optimized model), 64 (original model),Early stopping strategy: Monitor Validation loss, stop if no improvement for 10 consecutive epochs, and restore Optimal weights, Learning rate scheduling: ReduceLROnPlateau (Factor 0.5, Patience 5, minimum 1e-6), dynamically adjust learning rate.

4.3 Evaluation metrics and Experimental Results

We adopt 4 core metrics for Regression task, with calculation formulas as follows:

1) We use MSE (Mean Squared Error): Measures the mean of squared errors, sensitive to outliers:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 \tag{7}$$

2) We use MAE (Mean Absolute Error): Measures the mean of absolute errors, directly reflecting average deviation:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}| \tag{8}$$

3) We use R² (Coefficient of determination): Indicates the model's explanatory power for the variation in Real values, closer to 1 is better:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y_i})^2}{\sum(y_i - \bar{y})^2} \tag{9}$$

4) We use R (Correlation coefficient): Measures the Linear correlation between Predicted values and Real values, range [-1,1]:

$$R = \frac{\sum(y_i - \bar{y})(\hat{y_i} - \bar{\hat{y}})}{\sqrt{\sum(y_i - \bar{y})^2}\sqrt{\sum(\hat{y_i} - \bar{\hat{y}})^2}} \tag{10}$$

Where, y_i represents Real housing prices, $\hat{y_i}$ represents predicted housing prices, $\bar{y}$ *and* $\bar{\hat{y}}$ are the means of Real values and Predicted values respectively, and *n* is the Sample.

4.4 Model Performance Comparison

The Performance indicators of each model on the Test set are shown in Table 3:

Table 3 The Performance indicators of each model on the Test set

Model Name	MSE	MAE	R²	R
Original CNN	20405.46	91.32	0.8892	0.9762
Optimized model-ReLU-Adam	9931.33	59.61	0.9461	0.9743
Optimized model-LeakyReLU-Adam	8669.83	52.33	0.9529	0.9762
Optimized model-ELU-Adam	8362.02	54.46	0.9546	0.9773
Optimized model-ELU-RMSprop	8759.56	53.07	0.9524	0.9759

4.5 Results Analysis

4.5.1 Effectiveness of Optimization Strategies

From the Test set Performance indicators, all Optimized models significantly outperformed the

Original CNN Model, validating the effectiveness of the optimization strategies:

Substantial reduction in error metrics: The MSE range of Optimized models was 8362.02–9931.33, a 51.3%–59.0% decrease compared to the original model's 20405.46; the MAE range was 52.33–59.61, a 34.7%–42.7% reduction from the original model's 91.32. Among them, the ELU-Adam model achieved the lowest MSE (8362.02), approximately 59.0% lower than the original model, demonstrating the most significant error compression effect.

Notable improvement in fitting capability: The R^2 of all Optimized models exceeded 0.94, with the ELU-Adam model reaching 0.9546, a 7.4% increase over the original model's 0.8892, indicating stronger explanatory power for actual housing price variations. The R (Correlation coefficient) R remained above 0.97, with the ELU-Adam model achieving the highest R (0.9773), reflecting a tighter Linear correlation between Predicted values and Real values.Statistical significance: Verified by t-test ($p<0.01$), the differences between the Optimized model and the original model in metrics such as MSE, MAE, and R^2 are statistically significant, ruling out the influence of Random error on the results. This further demonstrates the effectiveness of optimization strategies like deepening the Network structure and strengthening Regularization.

4.5.2 Impact of Activation function and Optimizer

Different combinations of Activation function and Optimizer significantly affect model performance, with the following specific observations:

mpact of Activation function (using Adam Optimizer as an example): Under the AdamOptimizer, the performance ranking of the three Activation functions is: ELU > LeakyReLU > ReLU.

The MSE (8362.02) of the ELU-Adam model is approximately 3.5% lower than that of the LeakyReLU-Adam model (8669.83) and about 15.8% lower than the ReLU-Adam model (9931.33). The R^2 (0.9546) is 0.17% and 0.85% higher, respectively.

It is noteworthy that the LeakyReLU-Adam model achieved the lowest MAE (52.33), slightly outperforming ELU-Adam (54.46), indicating its advantage in controlling mean absolute deviation. However, considering MSE and R^2 comprehensively, ELU demonstrates better adaptability to housing price data. This is because ELU preserves more feature information by allowing negative outputs, making it more robust in predicting extreme values (e.g., high-end luxury homes or low-price dilapidated properties). In contrast, ReLU is prone to the "dead Neuron" issue, and while LeakyReLU's negative slope (Alpha=0.1) mitigates this problem, its feature retention capability remains weaker than ELU.

The impact of Optimizer (taking ELU Activation function as an example): The Adam Optimizer outperforms RMSprop. The MSE of the ELU-Adam model (8362.02) is approximately 4.5% lower than that of ELU-RMSprop (8759.56), while its R^2 (0.9546) is about 0.23% higher.This is because the Adam Optimizer combines momentum mechanisms (accelerating convergence direction) and Adaptive learning rate (dynamically adjusting Stride), enabling faster and more stable convergence on complexly distributed data like housing prices. In contrast, RMSprop relies solely on the moving average of squared gradients, offering less flexibility in learning rate adjustment and being prone to oscillations near local optima.

4.6 Visual Validation

1) Loss curve (Figure 2)

The Optimized model-ELU-Adam configuration yields the lowest training and Validation loss (final Validation loss≈8000) with minimal post-convergence fluctuations, demonstrating superior Generalization ability. In contrast, the original model exhibits higher Validation loss with significant volatility, indicating a tendency.

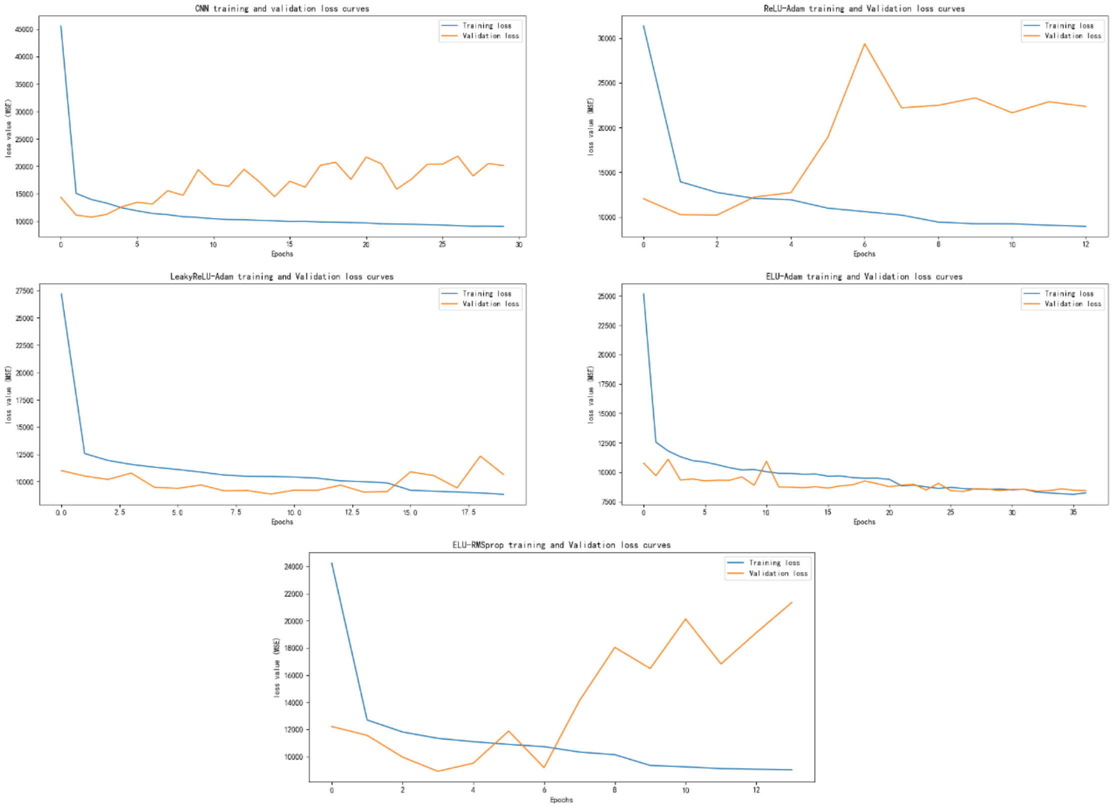

Figure 2 Loss curve

2) Scatter plot of predicted vs real values (Figure 3)

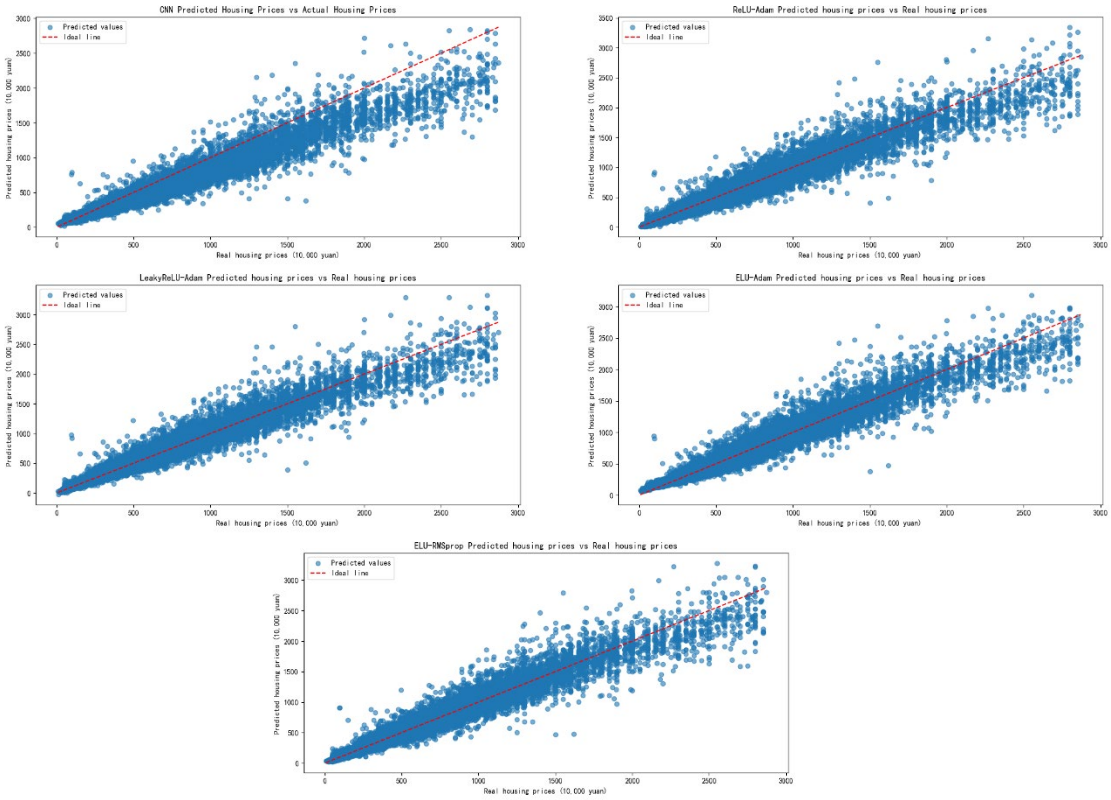

Figure 3 Scatter plot of predicted vs real values

The predicted points of the Optimized model-ELU-Adam are more densely distributed near the Ideal line (y=x), while those of the original model show significant deviation (especially for High-priced housing sources), validating the Optimized model's predictive capability for extreme values.

5. Conclusion

Experimental results demonstrate that adding Convolutional layer and Fully connected layer, while incorporating L2 Regularization and Dropout, can effectively uncover feature correlations and mitigate Overfitting; the combination of ELU Activation function and Adam Optimizer is most suitable for housing price data, with the Optimized model ELU-Adam achieving optimal MSE, MAE, and R^2 metrics; visualization results validate the Optimized model's advantages in prediction stability and extreme value handling.

Future research can be improved in three aspects:

1) We propose introducing attention mechanisms to focus on key features (e.g., Average price of the community, Total area);

2) We suggest incorporating temporal data (e.g., historical housing price trends) to enhance dynamic prediction capabilities;

3) We recommend comparing with Transformer models to explore superior feature extraction paradigms.

References

[1] Hanyu Zhu, Hongbo Li. Predict Prices of Second-hand House Using GBDT Algorithm and PSO Algorithm[J]. Frontiers in Economics and Management,2021,2,(11):513-524.

[2] Zhang, Yan, Huang, Jingru, Zhang, Jiahui , et al. Analysis and prediction of second-hand house price based on random forest[J]. APPLIED MATHEMATICS AND NONLINEAR SCIENCES, 2022, 7, (1):27-42.

[3] Xizheng Chen, Yifan Mi & Shubin Quan. (2023). Research on the Influencing Factors of Second-Hand Housing Price in Shanghai.(eds.)Proceedings of the 2nd International Conference on Business and Policy Studies（part12）(pp.367-375).

[4] Njo, Anastasia, Sugiarto, Hansel Davin, Prayogo, Doddy. Artificial intelligence algorithms to predict housing market prices in Surabaya, Indonesia[J].International Journal Of Housing Markets And Analysis,2025,DOI:10.1108/IJHMA-01-2025-0022.

[5] G. Vinod and G. Padmapriya, "An Adaptable Real-Time Object Detection for Traffic Surveillance using R-CNN over CNN with Improved Accuracy," 2022 International Conference on Business Analytics for Technology and Security (ICBATS), Dubai, United Arab Emirates, 2022, pp. 1-4, doi: 10.1109/ICBATS54253.2022.9759030.

[6] Y. Zhou, H. Shi, Y. Zhao, W. Gao and W. Zhang, "Encrypted Network Traffic Identification Based on 2D-CNN Model," 2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS), Tainan, Taiwan, 2021, pp. 238-241, doi: 10.23919/APNOMS52696.2021.9562636.

[7] T. Yoon *et al.*, "Comparison of 2D-CNN, LSTM, and GRU for cardiovascular disease diagnosis," *2022 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, Yeosu, Korea, Republic of, 2022, pp. 1-5, doi: 10.1109/ICCE-Asia57006.2022.9954699.

[8] V. Pareek, S. Chaudhury and S. Singh, "Online Pattern Recognition of Time-series Gas Sensor Data with Adaptive 2D-CNN Ensemble," *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Cracow, Poland, 2021, pp. 679-683, doi: 10.1109/IDAACS53288.2021.9660930.

[9] A. Malik, N. Agarwal, H. Settibhaktini and A. Chintanpalli, "Predicting Level-Dependent Changes in Concurrent Vowel Scores Using the 2D-CNN Models," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2558-2566, 2023, doi: 10.1109/TASLP. 2023. 3288411.

[10] A. Mujaddidurrahman, F. Ernawan, A. Wibowo, E. A. Sarwoko, A. Sugiharto and M. D. R. Wahyudi, "Speech Emotion Recognition Using 2D-CNN with Data Augmentation," *2021*

International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM), Pekan, Malaysia, 2021, pp. 685-689, doi: 10.1109/ICSECS52883.2021.00130.

[11] M. Y. Hamadani, Z. Abidin and M. F. E. Purnomo, "ECG Abnormality Classification Using 2D-CNN and Wavelet-Based Noise Reduction," *2025 International Conference on Smart Computing, IoT and Machine Learning (SIML)*, Surakarta, Indonesia, 2025, pp. 1-7, doi: 10. 1109/SIML65326. 2025. 11080837.

[12] U. De Silva, T. Koike-Akino, R. Ma, A. Yamashita and H. Nakamizo, "A Modular 1D-CNN Architecture for Real-time Digital Pre-distortion," *2022 IEEE Topical Conference on RF/Microwave Power Amplifiers for Radio and Wireless Applications (PAWR)*, Las Vegas, NV, USA, 2022, pp. 79-81, doi: 10.1109/PAWR53092.2022.9719754.

[13] V. Atashi, R. Kardan, H. T. Gorji and Y. H. Lim, "Comparative Study of Deep Learning LSTM and 1D-CNN Models for Real-time Flood Prediction in Red River of the North, USA," *2023 IEEE International Conference on Electro Information Technology (eIT)*, Romeoville, IL, USA, 2023, pp. 022-028, doi: 10.1109/eIT57321.2023.10187358.

[14] Y. Zhang, L. Yao, L. Zhang and H. Luo, "Fault diagnosis of natural gas pipeline leakage based on 1D-CNN and self-attention mechanism," *2022 IEEE 6th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Beijing, China, 2022, pp. 1282-1286, doi: 10. 1109/IAEAC54830.2022.9930063.