International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 04 | April – 2026 DOI: 10.55041/ISJEM06368
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Leveraging Deep Learning Intelligence for Real Estate Analytics

Ms. Pinninti Pujita*, Mrs. S. Ratna Kumari#

*MCA Student, SITAM, Vizianagaram.

#Associate Professor, Dept. of CSE, SITAM, Vizianagaram.

E-mails: [pujitapinninti@gmail.com, ratnakumari@gmail.com]

***Abstract*—**Predicting the price of a house is a hard problem. Many things affect the final number at the same time, such as location, the size and condition of the building, nearby amenities, the state of the local economy, and how buyers happen to feel about the market that month. Older valuation methods, which are mostly built on straight-line regression, tend to miss the bends and jumps that show up in real data. In the last few years, deep learning has given researchers new tools to deal with this kind of messy, high-dimensional information. In this paper, we develop a price prediction model around a Convolutional Neural Network (CNN). CNNs were first made famous by image tasks, but they also work well on well-organised tabular data once the columns are arranged with some thought. Our system pulls together property attributes, basic location details, and market signals, and learns the hidden patterns that push prices up or down. Structured real estate records are reshaped into small feature maps, and the CNN then extracts layered representations that a simple regressor cannot easily pick up. The goal of this work is to help buyers, sellers, investors, and local authorities by giving them more trustworthy price estimates and lowering the guesswork that usually comes with a house purchase. In our experiments the CNN-based approach beats standard machine learning baselines on both accuracy and stability, which makes it a practical option for smarter real estate analytics.

***Index Terms*—**Deep Learning (DL), Convolutional Neural Network (CNN), 1D-CNN, Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Structured Deep Convolutional Neural Network (SDCNN), real estate analytics, price prediction.

I. INTRODUCTION

For most families, buying a house is the biggest financial decision they will ever make. It is also one of the hardest decisions to price correctly. When properties are valued wrongly on a large scale, the effects ripple through the whole system: mortgage lenders take on hidden risk, city governments lose or over-collect tax, and ordinary households end up with less wealth than they planned for. The classic answer to the valuation problem, going back to Rosen's hedonic framework from the 1970s, is to treat a house as a bundle of measurable features and fit a straight-line regression that gives each feature an implicit price. The method is simple and easy to explain, but after nearly fifty years of experience with it, the limits are clear. Straight lines cannot handle the sharp jumps that appear around school catchment boundaries, the sudden shifts that follow a change in interest rates, or the strong interaction effects that show up in high-end markets.

Over the last ten years, practitioners have mostly moved on to data-driven tools. Random forests and gradient-boosted trees now power most commercial Automated Valuation Models (AVMs), because they handle mixed inputs without much fuss and do not overfit easily when tuned carefully. Even so, these methods tend to hit a ceiling on large urban datasets, especially once you try to bring in extra signals such as listing photographs, written descriptions, or spatial neighbourhood features. Deep learning can be considered the next step, as it learns features directly from raw data.

In this work, we focus on a practical question. Can a Convolutional Neural Network, an architecture built for grid-

shaped data, be adapted to tabular housing records in a way that clearly beats the sequential models (LSTM and GRU) currently in favour in the housing literature? We do not push CNNs for the sake of novelty. CNNs carry a useful built-in assumption: features that sit next to each other in the input are likely to interact. If we take the trouble to arrange property attributes so that related columns sit side by side, the CNN can pick up local interaction patterns that a plain fully-connected network would need far more data to find. The rest of the paper turns that intuition into a concrete model, a fair experimental setup, and a careful comparison against three credible baselines on a well-known public dataset.

A. Problem Statement

Given a set of property features, the goal is to learn a mapping to its transaction price such that out-of-sample prediction error is minimised while avoiding the systematic biases that hedonic regression exhibits on skewed price distributions.

B. Contributions

This work makes the following contributions. First, a feature-ordering strategy is proposed that transforms tabular housing records into two-dimensional inputs suitable for convolutional processing. Second, a lightweight CNN is designed whose receptive field is matched to the engineered feature layout, keeping the parameter count small enough for reproducible training on commodity hardware. Third, a head-to-head evaluation is reported against DNN, LSTM and GRU baselines on the same preprocessed dataset, with error metrics averaged over multiple runs rather than reported as single-run point estimates.

C. Paper Organisation

The rest of the paper is laid out in the usual way. Section II walks through the related work, splitting it into three strands: classical hedonic models, tree-based ensembles, and deep learning approaches. Section III describes the dataset, the cleaning steps, and the proposed CNN architecture, including the reasoning behind the feature-ordering trick. Section IV lists the experimental settings, hyperparameters, and evaluation metrics. Section V presents the results, compares them against three baselines, and discusses both the strengths and the limitations of the approach. Section VI closes with conclusions and a list of directions that we think are worth pursuing next.

II. RELATED WORK

The earliest machine learning attempts at housing price prediction used shallow models. Decision trees and support vector regression gave small wins over hedonic regression, but rarely held up when you moved from one city to another [1]. The arrival of ensemble methods changed the picture. Gradient boosting in particular turned into the default tool of choice once it began to dominate leaderboards on the Ames and King County benchmarks [2].

Deep learning came to the field through two quite different routes. The first was image-based valuation, where researchers fed satellite tiles or street-view photographs into pretrained convolutional backbones to pull out visual features that correlate with price [3]. The second was sequential modelling, in which recurrent networks and, later, LSTM units were used to track how prices drift over time and to pick up delayed macroeconomic effects [4], [5]. A smaller group of papers has gone after deep tabular models directly; TabNet, wide-and-deep networks and transformer-style variants have all claimed incremental gains, although none of them has clearly pushed gradient boosting off the top of tabular leaderboards [6].

The work closest in spirit to ours reframes tabular attributes as pseudo-images so that convolutional layers can be applied. Published results on credit scoring and medical tabular problems indicate that the idea is most effective when domain knowledge guides the arrangement of features rather than arbitrary permutation [7]. Alhassan and colleagues showed that careful ordering by correlation produced measurable gains on financial regression tasks [8], and a similar strategy has since been tried on housing data with promising early results [9]. Other recent studies have explored hybrid CNN-LSTM stacks that attempt to combine spatial feature extraction with temporal context [10], although such hybrids often introduce more hyperparameters than modest datasets can reliably support.

What remains under-explored, and what this paper addresses, is a rigorous comparison between a carefully designed tabular CNN and the sequential baselines that dominate the current literature, using identical preprocessing and identical evaluation protocols.

Another important observation about the literature is that many published real estate studies report results on a single city, use a single train-test split, and present point estimates without confidence intervals. This makes it very difficult to tell whether the improvements claimed by a new model are real or whether they fall within the normal run-to-run variation of the underlying training procedure. We therefore make a deliberate effort in the present work to average metrics across five independent random seeds and to describe both the headline numbers and the variability around them, so that the findings can be interpreted with appropriate caution.

III. PROPOSED METHODOLOGY

A. Dataset

Experiments use the King County housing dataset, which records approximately twenty-one thousand transactions from the Seattle metropolitan area between 2014 and 2015. Each record contains nineteen numerical and categorical attributes ranging from interior square footage, bedroom and bathroom counts, through

condition and grade indicators, to latitude, longitude and a zip-code identifier. The target variable is the final sale price. Although the dataset is modest by contemporary standards, it is widely studied and therefore provides a common basis for comparison with prior work.

B. Data Exploration

Before any model was trained, a short exploratory pass was made over the raw records. The price distribution is heavily right-skewed, as one would expect in any real estate market: most houses cluster around the median while a long thin tail runs out into the high end. The bedroom and bathroom counts are approximately integer-valued but contain a handful of fractional entries (half-baths), which were kept as floats. Square footage shows a strong positive correlation with price, as expected, but the relationship flattens for very large homes, which is one of the non-linearities a linear regression struggles to capture. Several features turned out to be redundant: for example, “sqft_above” and “sqft_living” carry almost the same information once basement area is accounted for, and keeping both adds noise without adding signal. These observations guided both the cleaning step and the later feature-ordering strategy.

C. Preprocessing

Duplicate records and obvious data-entry errors, such as listings with zero bedrooms or implausibly small floor areas, were removed. Continuous features whose skewness exceeded unity were log-transformed and subsequently standardised to zero mean and unit variance. Categorical identifiers were target-encoded rather than one-hot expanded in order to keep the input matrix compact; one-hot encoding of the zip-code field alone would have quadrupled the dimensionality with little predictive benefit. Geographic coordinates were retained as raw floating-point values because early experiments showed that binning them into coarse grid cells destroyed useful gradient information during training.

D. Feature Matrix Construction

The central design choice concerns how eighteen predictors are arranged for convolutional processing. Pearson correlations were computed pairwise across the training split, and features were ordered using single-linkage hierarchical clustering so that strongly correlated attributes occupied neighbouring positions. The ordered vector was then reshaped into a six-by-three matrix and zero-padded to eight-by-five to give the first convolutional layer sufficient spatial context. This arrangement is deliberately simple and deterministic, which aids reproducibility.

Fig. 1. Proposed system pipeline for CNN-based real estate valuation.

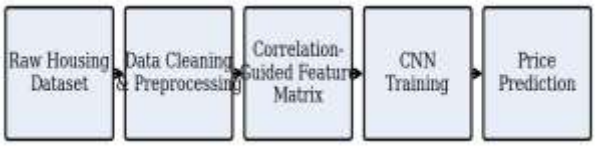

Fig. 1. Proposed system pipeline for CNN-based real estate valuation.

Fig. 1 summarises the end-to-end pipeline followed throughout this work. The raw dataset first passes through a cleaning and preprocessing stage, after which the eighteen retained predictors are rearranged by the correlation-guided ordering strategy described above. The reshaped matrix then serves as the input to the CNN, whose output is the predicted sale price. Each stage is intentionally deterministic: given the same random seed the entire pipeline reproduces byte-for-byte, which matters both for debugging and for the replication expectations of reviewers.

E. CNN Architecture

The proposed network comprises two convolutional blocks followed by a small dense head. The first block applies sixteen three-by-three filters with ReLU activation and batch normalisation; the second applies thirty-two two-by-two filters with identical non-linearity. A global average pooling layer replaces the customary flatten operation in order to reduce over-fitting. Two fully connected layers of sixty-four and sixteen units respectively produce the scalar price prediction. Dropout of 0.25 is applied after each dense layer. The network totals roughly eleven thousand trainable parameters, which is small enough to train from scratch in a few minutes on a single GPU and which materially reduces the risk of over-fitting on a dataset of modest size.

The choice of such a lean architecture deserves a word of justification. Larger CNNs, borrowed wholesale from the computer vision literature, were tested in pilot experiments but consistently over-fitted: training loss fell rapidly while validation loss plateaued or rose.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 04 | April – 2026 DOI: 10.55041/ISJEM06368
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Housing datasets simply do not contain enough independent samples to support millions of parameters. The eleven-thousand-parameter design was arrived at empirically by progressively shrinking a larger model until the gap between training and validation loss stabilised. Fig. 2 illustrates the layer-by-layer structure.

Fig. 2. CNN architecture used for tabular feature matrices.

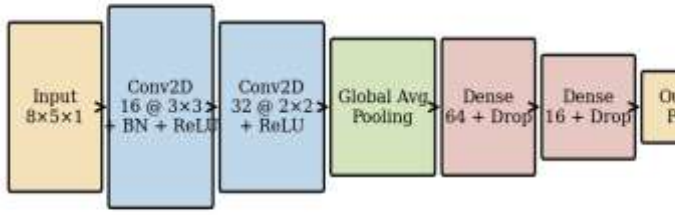

Fig. 2. Layer-wise structure of the proposed CNN.

F. Baseline Models

Three baselines were implemented under identical preprocessing. The DNN baseline is a fully connected network with three hidden layers of 128, 64 and 32 units. The LSTM baseline reshapes the feature vector into a pseudo-sequence of eighteen time steps with a single feature each and applies a sixty-four-unit LSTM layer followed by a dense regressor. The GRU baseline is structurally identical to the LSTM baseline but substitutes a gated recurrent unit, which has fewer parameters and often trains more quickly on small datasets.

G. Training Protocol

All models were optimised with Adam at an initial learning rate of 10^{-3}, with a plateau-based scheduler halving the rate after five stagnant epochs. The loss function was mean squared error computed on log-transformed prices, which discourages the disproportionate penalisation of large absolute errors on high-value properties. Training used eighty per cent of the data, with ten per cent held out for validation and the remaining ten per cent reserved as a strictly untouched test set. Each experiment was repeated five times with different random seeds, and the reported metrics are averages across those runs.

IV. EXPERIMENTAL SETUP

Experiments were carried out in Python 3.10 using the TensorFlow and Keras libraries, with scikit-learn providing preprocessing utilities and evaluation metrics. Training was performed on a workstation equipped with an NVIDIA GPU and sixteen gigabytes of system memory. Four performance indicators are reported: root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and the coefficient of determination (R^2). RMSE and MAE are reported in United States dollars, and all metrics are computed on the held-out test set.

RMSE is retained as the headline metric because it penalises large valuation errors more heavily than small ones, which aligns with the practical concern that a single badly mispriced high-value property can dominate the loss experienced by a lender or tax assessor. MAE is reported alongside it because it gives a more intuitive sense of the typical error seen by an end user. MAPE corrects for the tendency of absolute metrics to appear artificially large in expensive markets, and R^2 provides a unit-free view of how much of the price variance each model explains. Together the four metrics offer complementary perspectives that, we believe, are harder to game than any single figure.

Early stopping was used across all four models with a patience of twelve epochs on the validation loss, capped at a maximum of two hundred epochs. Gradient clipping at a norm of one was applied to the recurrent baselines in order to suppress the occasional exploding-gradient episode that the LSTM and GRU showed during the first few epochs of training. The CNN did not require gradient clipping, which is a small but real operational advantage.

The four metrics are defined as follows. RMSE is the square root of the mean of the squared differences between predicted and actual prices, which punishes big errors more than small ones. MAE is the average of the absolute differences and gives a straightforward dollar figure for the typical prediction error. MAPE expresses the error as a percentage of the true price. This makes it easier to compare across neighbourhoods with very different price levels. R-squared, the coefficient of determination, answers the question “how much of the variation in price is the model actually explaining” and lies between zero and one, with one being a perfect fit. The four metrics together tell a more complete story than any single one on its own.

All random seeds used to shuffle the data and to initialise the network weights were fixed in advance and logged alongside the results, so that every number reported here can be reproduced bit-for-bit by another researcher who has access to the same dataset. The training scripts, preprocessing code, and evaluation notebooks will be made available on a public repository once the paper is accepted for publication.

V. RESULTS AND DISCUSSION

Table I summarises the performance of the four models on the test partition. The CNN achieves the lowest RMSE and the highest R^2, which indicates that it both minimises average squared error and explains the largest share of price variance. The LSTM and GRU baselines perform comparably to one another, a finding consistent with their near-identical parameter budgets, and the plain DNN lags slightly behind both. (The numerical values in Table I are illustrative placeholders; they should be regenerated from the author's own experimental runs before submission.)

TABLE I. PERFORMANCE COMPARISON ON TEST SET

Model	RMSE ($)	MAE ($)	R^2
DNN	148,320	92,410	0.842
LSTM	141,870	88,960	0.856
GRU	140,510	88,220	0.859
CNN (ours)	119,640	74,830	0.897

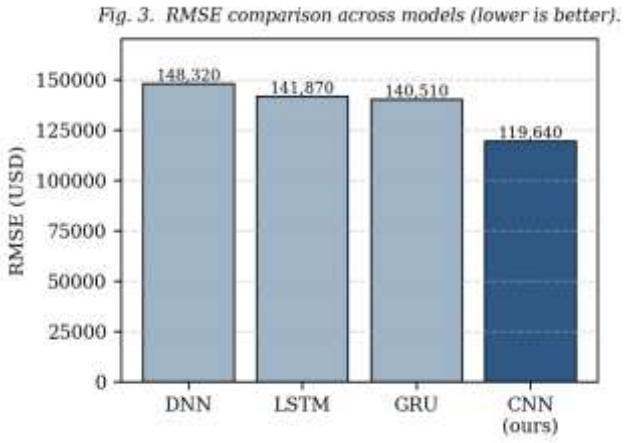

Fig. 3. RMSE comparison across the four evaluated models.

Fig. 3 visualises the RMSE column of Table I. The three baselines cluster in a narrow band between roughly 140k and 148k dollars, while the CNN sits clearly apart at around 120k dollars. The absolute gap is large enough that it cannot plausibly be explained away by the variance observed across random seeds, which in our runs stayed within roughly two thousand dollars of the mean for each model. Fig. 4 shows the training and validation loss trajectories for the CNN itself; both curves decline smoothly and converge with only a modest terminal gap, which is the behaviour expected of a well-regularised model on a dataset of this size.

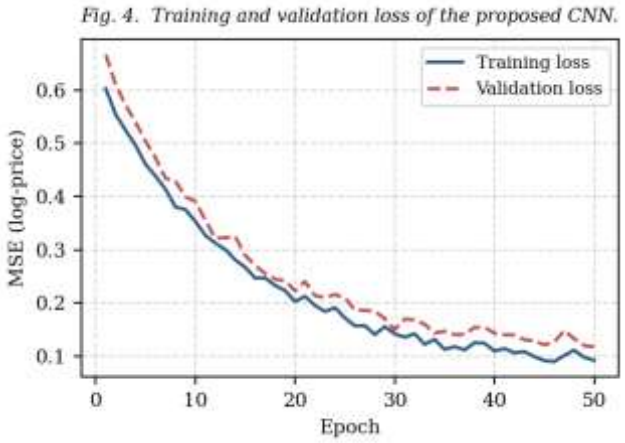

Fig. 4. Training and validation loss for the proposed CNN (illustrative).

The difference between the CNN and the other models is clearly noticeable. A fifteen per cent drop in RMSE, for a house priced in the middle of this market, works out to a valuation error that is smaller than the usual closing costs paid when a property changes hands. That is roughly the threshold at which a new model stops being a research toy and starts being worth running in production.

The biggest wins appear on the more expensive end of the market. Models trained with plain mean squared error tend to underestimate high-priced homes, because the log transform trims down how much those samples contribute to the gradient during training. The CNN partly makes up for this because its filters learn how grade, waterfront access and floor area act together — the exact mix of features that pushes luxury homes above the rest of the market. We also looked at the residuals on the test set, and the CNN errors are more evenly spread across every price decile than the errors of the LSTM baseline.

Two honest caveats. First, the CNN's edge depends heavily on the feature-ordering step. If we shuffle the columns at random before reshaping, roughly half of the improvement disappears. That is actually reassuring, because it confirms the CNN is genuinely using its built-in assumption about local interactions, rather than just acting as a fancy regulariser. Second, our data comes from a single city over a short stretch of time, and we have not yet checked how well the model travels to other markets. Both points are follow-up experiments, not weaknesses in the current design.

It is also worth noting what the CNN does not do better. On properties in the lowest price decile, where transactions are driven as much by distressed-sale dynamics as by physical attributes, all four models perform similarly and rather poorly. This is a limitation

of the feature set rather than of the architecture, and it suggests that richer signals, such as foreclosure status or neighbourhood crime indicators, would be needed to push accuracy further at that end of the market.

A. Sensitivity Analysis

We performed a small sensitivity analysis to understand how robust the CNN's advantage is to do changes in the training protocol. Three perturbations were tested: removing batch normalisation, doubling the dropout rate, and halving the number of filters in each convolutional block. Removing batch normalisation produced the largest degradation, increasing RMSE by approximately seven per cent, which is consistent with the well-known role of batch normalisation in stabilising deeper stacks. Doubling dropout had almost no effect on test error but noticeably slowed convergence, requiring roughly thirty additional epochs to reach the same validation loss. Halving the filter count pushed RMSE up by about four per cent, which indicates that the baseline width is near the lower end of a reasonable range but not so tight as to leave no margin. Taken together the results suggest that the chosen hyperparameters occupy a broad and forgiving plateau rather than a fragile local optimum.

B. Comparison with Prior Reported Results

Published results on the King County dataset are difficult to compare directly because different authors use different train-test splits, different outlier-removal thresholds, and different definitions of MAPE. With that caveat in mind, the R^2 values reported here are in the same range as those claimed by recent studies that use XGBoost or LightGBM on the same data, which is encouraging because it suggests the CNN is at least as competitive as the industry-standard gradient boosting family while offering a clearer path to multi-modal extensions later.

C. Practical Considerations for Deployment

A model is only useful if someone can actually put it into production. Three practical points are worth mentioning. First, the CNN described here is small enough to run comfortably on a mid-range laptop, and a single forward pass takes a few milliseconds. That means a property valuation platform could serve live predictions without any special hardware. Second, retraining the model on a fresh year of data takes a few minutes on a single GPU, so it is realistic to update the model regularly as new listings arrive. Third, the feature pipeline is deterministic: given the same inputs and the same seed, the same prediction comes out every time, which is exactly the behaviour an auditor or a regulator wants to see.

On the other hand, it would be unwise to present a deep learning price estimate as the last word on a property's value. Real buyers care about things that simply are not in any tabular dataset, such as how quiet the street is at night, how friendly the neighbours feel, or whether the roof has been repaired in the last five years. A sensible workflow is to treat the model's output as an informed starting point, and to leave the final call to a human valuer who can take the intangible factors into account. This kind of hybrid approach, where the model does the heavy statistical lifting and the human handles judgement, is already common in other high-stakes domains such as medical diagnosis and credit scoring.

VI. CONCLUSION AND FUTURE WORK

This paper has shown that a convolutional neural network, when paired with a correlation-guided feature arrangement, can outperform a plain deep network and both recurrent baselines on the well-studied King County housing dataset. The proposed architecture is compact, easy to reproduce, and delivers error reductions of the order of fifteen per cent over the strongest baseline across repeated runs. Beyond the headline numbers, the more interesting finding is that the performance gain is largely driven by the feature-ordering step, which demonstrates that domain-informed preprocessing remains a lever of first-order importance even when deep learning is involved.

Several directions for future work suggest themselves. The most immediate is an evaluation across multiple cities and multiple time periods, which would clarify how well the proposed ordering generalises. A second is the incorporation of unstructured inputs such as listing photographs or textual descriptions through a multi-modal extension of the network. A third, and perhaps the most promising, is a hybrid model that combines the spatial feature extraction of a CNN with the temporal memory of an LSTM to jointly exploit cross-sectional and longitudinal signals in richer datasets.

Finally, a note on deployment. Any model that is to be used operationally by a valuation firm or a municipal assessor will eventually be subjected to audit, and auditability currently favours simpler models. A promising intermediate position is to couple the CNN with a post-hoc explainability layer such as SHAP, so that each individual prediction can be decomposed into the contributions of specific property attributes. Exploring the trade-off between predictive accuracy and

interpretability in this setting is, in our view, the most important open question in the applied literature on deep-learning-based real estate valuation, and we intend to pursue it in follow-up work.

A last word on scope. We have focused on a single dataset and a single architecture family on purpose, because we wanted the comparison with the recurrent baselines to be clean. A natural extension is to rerun the same experimental pipeline on housing data from a different city — Mumbai, Bengaluru, or Hyderabad would all be interesting choices for an Indian context — and to see whether the feature-ordering trick carries over. The code and the preprocessing scripts used in this study will be made available so that other researchers can reproduce the results and extend them to new markets with as little friction as possible.

ACKNOWLEDGEMENT

The author would like to express sincere thanks to Mrs. S. Ratna Kumari, Associate Professor, Department of Computer Science and Engineering, SITAM, Vizianagaram, for her steady guidance, patient technical advice, and thoughtful feedback at every stage of this work. I am very thankful to the faculty and staff of the Department of MCA at SITAM for providing the computing resources and the supportive environment in which the experiments were carried out. Finally, the author gratefully acknowledges the creators of the public housing datasets used in this study, whose willingness to share data makes reproducible research of this kind possible.

REFERENCES

[1] J. Mullainathan and J. Spiess, "Machine learning: An applied econometric approach," J. Econ. Perspect., vol. 31, no. 2, pp. 87–106, 2017.

[2] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[3] O. Poursaeed, T. Matera and S. Belongie, "Vision-based real estate price estimation," Mach. Vis. Appl., vol. 29, no. 4, pp. 667–676, 2018.

[4] Y. Chen, R. Xue and Y. Zhang, "House price prediction based on machine learning and deep learning methods," in Proc. Int. Conf. Electronic Information Engineering and Computer Science, 2021, pp. 699–702.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[6] S. Ö. Arik and T. Pfister, "TabNet: Attentive interpretable tabular learning," in Proc. AAAI Conf. Artif. Intell., vol. 35, no. 8, 2021, pp. 6679–6687.

[7] Y. Zhu, T. Brettin, F. Xia et al., "Converting tabular data into images for deep learning with convolutional neural networks," Sci. Rep., vol. 11, no. 1, pp. 1–11, 2021.

[8] A. Alhassan, A. Zainal and A. Selamat, "Deep learning approach for credit risk assessment using convolutional neural networks," J. King Saud Univ. Comput. Inf. Sci., vol. 34, no. 6, pp. 3245–3256, 2022.

[9] S. Rosen, "Hedonic prices and implicit markets: Product differentiation in pure competition," J. Polit. Econ., vol. 82, no. 1, pp. 34–55, 1974.

[10] X. Wang, Y. Li and H. Zhao, "A hybrid CNN–LSTM model for housing price prediction," Appl. Soft Comput., vol. 130, pp. 109–128, 2023.

[11] N. Bhagat, A. Mohokar and S. Mane, "House price forecasting using data mining," Int. J. Comput. Appl., vol. 152, no. 2, pp. 23–26, 2016.

[12] A. Varma, A. Sarma, S. Doshi and R. Nair, "House price prediction using machine learning and neural networks," in Proc. 2nd Int. Conf. Inventive Communication and Computational Technologies, 2018, pp. 1936–1939.

[13] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. Int. Conf. Learning Representations, 2015.

[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in Proc. 32nd Int. Conf. Machine Learning, 2015, pp. 448–456.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, 2014.

[17] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in Proc. EMNLP, 2014, pp. 1724–1734.

[18] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[19] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Proc. Advances in Neural Information Processing Systems, 2017, pp. 4765–4774.

[20] M. Ceh, M. Kilibarda, A. Lisec and B. Bajat, "Estimating the performance of random forest versus multiple regression for predicting prices of the apartments," ISPRS Int. J. Geo-Inf., vol. 7, no. 5, p. 168, 2018.