

Article

Leveraging Machine Learning for Sophisticated Rental Value Predictions: A Case Study from Munich, Germany

Wenjun Chen ¹, Saber Farag ¹, Usman Butt ²  and Haider Al-Khateeb ^{3,*} 

¹ Faculty of Engineering and Environment, Northumbria University London, London E1 7HT, UK; saber.farag@northumbria.ac.uk (S.F.)

² Faculty of Engineering and IT, British University in Dubai, Dubai P.O. Box 345015, United Arab Emirates; usman.butt@buid.ac.ae

³ Cyber Security Innovation (CSI) Research Centre, Operations & Information Management, Aston University, Birmingham B4 7ET, UK

* Correspondence: h.al-khateeb@aston.ac.uk

Abstract: There has been very limited research conducted to predict rental prices in the German real estate market using an AI-based approach. From a general perspective, conventional approaches struggle to handle large amounts of data and fail to consider the numerous elements that affect rental prices. The absence of sophisticated, data-driven analytical tools further complicates this situation, impeding stakeholders, such as tenants, landlords, real estate agents, and the government, from obtaining the accurate insights necessary for making well-informed decisions in this area. This paper applies novel machine learning (ML) approaches, including ensemble techniques, neural networks, linear regression (LR), and tree-based algorithms, specifically designed for forecasting rental prices in Munich. To ensure accuracy and reliability, the performance of these models is evaluated using the R^2 score and root mean squared error (RMSE). The study provides two feature sets for model comparison, selected by particle swarm optimisation (PSO) and CatBoost. These two feature selection methods identify significant variables based on different mechanisms, such as seeking the optimal solution with an objective function and converting categorical features into target statistics (TSs) to address high-dimensional issues. These methods are ideal for this German dataset, which contains 49 features. Testing the performance of 10 ML algorithms on two sets helps validate the robustness and efficacy of the AI-based approach utilising the PyTorch framework. The findings illustrate that ML models combined with PyTorch-based neural networks (PNNs) demonstrate high accuracy compared to standalone ML models, regardless of feature changes. The improved performance indicates that utilising the PyTorch framework for predictive tasks is advantageous, as evidenced by a statistical significance test in terms of both R^2 and RMSE (p -values < 0.001). The integration results display outstanding accuracy, averaging 90% across both feature sets. Particularly, the XGB model, which exhibited the lowest performance among all models in both sets, significantly improved from 0.8903 to 0.9097 in set 1 and from 0.8717 to 0.9022 in set 2 after being combined with the PNN. These results showcase the efficacy of using the PyTorch framework, enhancing the precision and reliability of the ML models in predicting the dynamic real estate market. Given that this study applies two feature sets and demonstrates consistent performance across sets with varying characteristics, the methodology may be applied to other locations. By offering accurate projections, it aids investors, renters, property managers, and regulators in facilitating better decision-making in the real estate sector.



Citation: Chen, W.; Farag, S.; Butt, U.; Al-Khateeb, H. Leveraging Machine Learning for Sophisticated Rental Value Predictions: A Case Study from Munich, Germany. *Appl. Sci.* **2024**, *14*, 9528. <https://doi.org/10.3390/app14209528>

Academic Editors: Grigorios Beligiannis, Nadia Nedjah and Luiza de Macedo Mourelle

Received: 30 July 2024

Revised: 2 October 2024

Accepted: 11 October 2024

Published: 18 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: PyTorch; neural network; machine learning; feature selection; PSO; Germany; comparative analysis; property valuation; artificial intelligence

1. Introduction

The advent of digitalisation has ushered in an era with an enormous amount of available data [1]. In the context of the real estate market, multiple approaches, including

conventional statistical calculations, data mining strategies, and ML methods, have been applied in prior research to forecast property values [2]. However, predicting housing costs with precision is particularly challenging due to the strong correlation between real estate prices and factors such as geographical location, property size, and other attributes. Notably, the results of the same approach can vary significantly from one location to another [3]. These characteristics imply that simple methods may not effectively represent the ever-changing nature of data [4,5]. Previous techniques have exhibited limitations in scalability and flexibility, which has hindered the ability to meet the needs of stakeholders seeking access to pertinent information in the housing industry [1,6].

As rental value estimation undergoes rapid transformation, the application of advanced AI-based algorithms, such as ML models or deep learning (DL) methods, has introduced nuanced alternatives and offered promising solutions [7,8]. However, leveraging the PyTorch framework has not yet been thoroughly investigated in the existing literature. Compared to the application of standalone algorithms, utilising a framework like PyTorch could offer superior accuracy and efficiency. This advantage can be attributed to factors such as the highly optimised libraries, predefined neural network layers, and its ability to sustain improved accuracy across various features, uncover complex patterns, and provide more precise, adaptable, and timely estimations [1].

The primary objectives, aims and contributions of this study are outlined below:

- A variety of ML models were carefully selected for this regression task, including LR, random forest regression (RFR), gradient boosting regressors (GBRs), and XGB regressors.
- Two neural networks, specifically a fully connected neural network (FNN) and a PNN, were evaluated to assess the effectiveness of DL methods in predicting rental values.
- The PyTorch framework was applied to individual ML models, including those stacked with RFR, GBR, XGB, and LR, to enhance prediction accuracy.
- A thorough assessment of each model was conducted using metrics such as RMSE and the R^2 score.
- To verify the adequacy and reliability of the PyTorch framework, two different feature sets chosen by the CatBoost and PSO algorithms were used.
- This research aimed to enhance the effectiveness of AI techniques within the housing industry, bridging the gap between computer science and property investment, and expanding the scope of current research on the German property market.

This study employed AI-based algorithms to forecast rental prices in the German real estate market, with a focus on Munich. A variety of algorithms were employed as feasible solutions to provide more accurate and reliable predictions for market price evaluations, assisting agents, property owners, and renters in making objective assessments [9]. While this comprehensive investigation provided insights into the prediction capabilities of these approaches, the data source was gathered in 2020, underscoring the need for regular updates to maintain reliability over time. Some essential characteristics might be overlooked, or the attributes may not contribute meaningful predictive value during data pre-processing. Furthermore, each model was based on certain data assumptions, which could result in a misrepresentation of the effectiveness of AI-driven methods.

The paper is organised as follows: Section 2 delves into existing research and the prospective impacts of this study, providing a comprehensive review of recent and relevant literature on forecasting home values. Section 3 presents an in-depth examination of the selected dataset, which informs the development of this AI technique. Section 4 details the experimental methodology, findings, and an evaluation of the impact of implementing the AI-driven approach. Finally, Section 5 concludes the paper by discussing potential future efforts to enhance and build upon the insights and recommendations generated in this study.

2. Related Work

2.1. Existing Techniques for Predicting the Housing Market

Multiple AI-based methods have been implemented in the real estate industry to predict housing market trends in prior studies. The paper by [10] investigated the capabilities of neural networks in the domain of house price prediction, focusing on back propagation neural networks (BPNNs) and convolutional neural networks (CNNs) to identify complex relationships between property characteristics and sale prices. The authors examined these models over three different time periods (three, five, and six months) within the Taiwanese housing market, incorporating macroeconomic factors such as the housing investment demand-to-supply ratio, the housing price-to-income ratio, and two types of housing: land and buildings, with or without park access. The CNN model demonstrated superior performance compared to the BPNN algorithm over the five-month span, achieving a remarkable R^2 value above 0.945. Additionally, it yielded the lowest values for mean absolute error (MAE), RMSE, mean absolute percentage error (MAPE), and root mean squared logarithmic error (RMSLE). Nevertheless, macroeconomic factors had minimal impact on housing price evaluations. This study highlights the importance of using time-series data for real estate price estimation with CNNs. Although CNNs are typically associated with image processing tasks, the results showcase their versatility and effectiveness in property price prediction. This underscores the potential for novel applications of DL techniques in predictive work, and suggests the integration of advanced methods, such as PNNs, which could further improve prediction accuracy in the real estate market.

Subsequently, Pai et al. [11] employed various ML techniques, including BPNNs, classification and regression tree (CART), general regression neural networks (GRNNs), and least squares support vector regression (LSSVR), to predict housing prices in Taichung, Taiwan. This prediction was conducted without considering time series or macroeconomic data. Instead, the focus was on comparing real transaction prices with estimated ones to aid stakeholders, such as property owners and agents, in making informed judgments on the property valuations. The empirical data clearly indicated that LSSVR achieved higher predictive accuracy than the other methods, obtaining the lowest MAPE of 0.228. These results presented the efficacy of ML models, indicating that the regression model was more effective in forecasting actual and estimated prices, outperforming neural networks in this regard. The authors also recommend incorporating additional components, such as photographs of residences and economic data, as inputs for model training to improve performance. This research serves as an important reference for constructing a methodological framework by applying LR to estimate rental market trends, considering the relationship between housing features and rental prices in the real estate market.

Further insights were provided by [12], who utilised RF, extreme gradient boost (XGBoost), and LightGBM algorithms to forecast house values based on market preferences in the Chengdu market, China. The authors incorporated 12 significant property characteristics, including dwelling space, location, rent collection method, and transportation, to underscore the significance of comprehending local market preferences and behaviours. XGBoost, demonstrated remarkable performance, with an impressive accuracy score of 0.85 and a low mean square error (MSE) of 0.04, highlighting its potential for future investigation in similar scenarios. These chosen variables significantly influence rental cost in Chengdu, underscoring the advantages of utilising XGBoost for housing predictions and the need to select relevant features when evaluating ML models. This study provides a solid foundation for choosing suitable algorithms and testing models with appropriate feature sets. Another study by [13] employed a similar strategy, emphasising model efficiency through the selection of relevant variables for training. The authors investigated factors influencing commercial property prices by employing support vector machines (SVMs) to forecast pricing trends in the real estate market from 2021 to 2025. The dataset utilised was derived from a case study of Shenzhen, China, focusing on quantitative characteristics. The study employed a strategic method using the lasso variables filter (LVF) to identify key elements impacting prices. Results indicated that the SVM achieved superior accuracy

in the 5-year estimate and demonstrated an excellent fit, with minimal error reflected by an R^2 value of 1. The approaches used, including RF and XGBoost, offer valuable insights for predicting the rental market. This expanded the scope of model selection, providing a robust foundation for designing the research methodology in this study.

To achieve optimal outcomes, Wang & Xu [14,15] adopted various techniques to forecast housing prices, utilising the Boston housing dataset. To compare the effectiveness of ML and DL techniques, Xu [15] employed five methods—namely, LR, RF, XGBoost, SVM, and an artificial neural network (ANN)—to assess these models based on factors such as the proportion of residential land, the number of rooms, and the pupil–teacher ratio. The findings demonstrated that the ANN achieved outstanding outcomes, with an R^2 score of 0.8776 and the lowest MAE and RMSE values of 2.1085 and 3.1921, respectively. To further enhance ML model performance, Wang [14] implemented a range of algorithms, such as SVM with a radial basis function (RBF) kernel, logistic regression, ridge regression, lasso regression, polynomial regression, and RF. These were combined with score function, k-fold, and shuffle cross-validation techniques. The empirical result demonstrated that SVM with an RBF kernel outperformed the other approaches, achieving an R^2 score of 0.799. This result emphasises the importance of adjusting hyperparameters through grid search to enhance model accuracy. The outcomes from both studies highlight the versatility and adaptability of AI-driven techniques. These methodologies laid the groundwork for choosing and employing both ML and DL models to forecast rental prices in the German real estate sector.

On the other hand, the complex nature of property attributes in housing datasets presents challenges related to high dimensionality. Sakri et al. [16] skilfully tackled these issues by utilising the PSO technique as a feature selection tool on the Ames house dataset to identify the most significant features from a total of 82 features. The predicting tasks were conducted using six algorithms: RF, multilayer perceptron (MLP), naive Bayes (NB), K-nearest neighbour (K-NN), fast decision tree learner (REPTree), and LR. The findings indicated that the RF model achieved substantial improvement, with an accuracy score increase from 0.814 to 0.844, and a two-fold decrease in the error rate from 13.3% to 6.9%. This improvement was achieved by removing non-informative and statistically insignificant variations, offering valuable insights into addressing high dimensionality effectively. Furthermore, the potential benefits of hybrid approaches were supported by [17], which combined PSO with XGBoost, resulting in significant gains in prediction accuracy. The authors conducted a comparative assessment of model efficacy across five algorithms: ridge regression, lasso regression, LightGBM, XGBoost, and PSO-XGBoost. The results highlighted the superiority of PSO-XGBoost over non-hybrid models, especially in capturing nonlinear interactions within the dataset. The PSO methodology significantly enhanced the R^2 score, which rose from 0.9517 to 0.9887 while the RMSE value decreased substantially from 0.1251 to 0.1015 (a reduction of 0.0236). These improvements have spurred increased interest in hybrid models for boosting prediction reliability and accuracy. PSO was effective in both studies for addressing high dimensionality, identifying key features that significantly impact predictive models. Applying PSO as a feature selection tool provides a valuable opportunity to explore, validate, and reduce the complexities of predictive modelling in this domain. In exploring hybrid models, Yang et al. [18] questioned the prevailing reliance on singular learning algorithms for housing price forecasting, which frequently fail to leverage the diverse capabilities of different ML models or the full potential of ensemble learning. To address this, the authors the D-Stacking technique, a diversity-driven ensemble approach. The initial layer of the D-stacking ensemble consisted of gradient-boosted decision tree (GBDT), XGBoost, and BPNet as base learners, while the second layer employed XGBoost. The model was evaluated on housing datasets originating from both China and the USA. Compared to individual prediction models, the D-Stacking technique demonstrated superior predictive, achieving RMSE values of 0.869 and 1.029, accordingly. This approach highlights that combining various algorithms can improve predictive outcomes and provide deeper insights into the strengths and limitations of

each model. The D-Stacking technique offers significant improvement in performance and demonstrates considerable potential in the evolving field of real estate forecasting. This reinforces the need to adopt various algorithms for enhanced accuracy, contributing valuable insights into the research methodology for this study.

The aforementioned literature primarily focuses on theoretical aspects. However, Disha et al. [19] makes a notable advancement by transitioning from theoretical concepts to practical implementation through the development of a smartphone application equipped with ML technology to estimate property prices for real estate professionals in Bangalore, India. This application evaluated seven distinct ML models, including logistic regression, lasso regression, ridge regression, decision tree (DT), RF, XGBoost, and Extreme Gradient Random Forest Boost (XGRFBoost), to identify the optimal solution for use within the software. Testing results indicated that XGBoost was the most effective at predicting the price of flats and independent houses, achieving accuracy rates of 0.9110 and 0.6465, respectively. Integrating ML techniques into a digital platform bridges complex theoretical algorithms with practical applications, demonstrating the feasibility of these approaches in real-time situations. Moreover, the demonstrated efficacy of XGBoost provides insights that inform the use of a similar approach for predicting rental values within the German dataset.

2.2. Methods of Feature Selection for Predicting Housing Prices

PSO is a meta-heuristic technique used to reduce the complexity of a data source, inspired by the collective behaviours observed in bird migration and foraging, where flocks of birds communicate iteratively to find the optimal solution based on an objective function [20–22]. An important benefit of this approach is its ability to optimise intricate structures and effectively handle datasets that contain both discontinuous and continuous data, as well as nonlinear functions [23]. Hybridising a neural predictive model with other optimisation strategies is an optimal choice [24,25]. In addition, PSO can identify the most informative and distinguishing characteristics through constant performance evaluation and comprehensive search space exploration, leading to improved overall performance, robustness, and generalisation capabilities of the model [26]. The research undertaken by [16,17] effectively employed PSO as a feature selection tool to find crucial traits from a pool of 82 variables in the Ames housing dataset. This approach significantly enhanced the accuracy of house price estimates. Furthermore, a distinctive method of combining PSO with XGBoost was implemented to develop a hybrid model. The result demonstrated improved performance compared to non-hybrid models, particularly in capturing the complex nonlinear relationships that exist in the data. Sakri et al. [16] employed this method in conjunction with other ML methods, showcasing its capability to improve prediction. The incorporation of this integration significantly impacts predictive models, leading to substantial increases in accuracy scores and a reduced error rate, demonstrating the increased reliability and precision of the forecasts. The research reviewed provides useful insights on applying ML and DL techniques for predicting housing prices. They emphasise the significance of employing a variety of methods, ensuring transparency in the model, and demonstrating the wide range of approaches in this field. The combination of PSO with XGBoost is a methodological breakthrough that might improve model performance by targeting the most influential predictors. Nevertheless, these studies also uncover certain constraints. Relying just on individual models without considering a range of ensemble or hybrid methods limits the capacity to apply findings broadly. Furthermore, a lack of appropriate data pre-processing or model selection consideration raises concerns about potential biases in the results. Although employing PSO for feature selection is novel, the criteria used and the influence on model performance require more investigation. The techniques applied show excellent prediction accuracy within those datasets, but the applicability of the results to various scenarios is still unidentified, which may not be universally applicable across different scenarios.

CatBoost is a gradient boosting technique that uses an advanced feature selection process since it was created especially to deal with category information [27]. This method can be used to handle class features in a flexible manner by transforming categorical features into a TS, an additional numerical characteristic that estimates the expected target for every category by adding super parameters. As a result, the model may capture intricate relationships within the dataset by using categorical data without adding bias, avoiding the need for data pre-processing [28]. In the context of high cardinality features, One-hot encoding produces an excessive number of extra binary attributes, which leads to a high-dimension problem [29]. Since the German housing dataset consists of categorical values that might cause the same issue, CatBoost offers a more effective and flexible way to address this problem by choosing features with high prediction potential while reducing noise from unnecessary factors [30].

2.3. Justification for Model Selection and Performance Metric

The use of 10 models appears to be a thoughtful choice to leverage the capabilities of various algorithms and to evaluate the effectiveness in forecasting rental prices. Following is the rationale for selecting these models:

- LR is a widely used approach for analysing relationships between variables and the target in many regression contexts [31]. This statement suggests that alterations in the independent components result in a proportional modification in the dependent variable in a linear correlation [32]. This technique may be used for prediction and estimate assistance. The use of this algorithm in many studies highlights its significance as a core methodology in the field of housing price prediction research. The research of [14,15] employed LR as a benchmark to evaluate the effectiveness of various ML models in predicting housing prices. In addition, Sakri et al. & Dishia et al. [16,19] implemented LR as a constituent of an algorithm to approximate house values in the regional real estate industry, based on the assumption that there is a direct relationship between the predictor variables and the housing prices. These studies collectively illustrate the enduring significance of LR in the field of forecasting property value in the real estate sector. It provides a basic and easily understood mechanism, making this algorithm highly valuable for comparing baselines in the regression task.
- RFR is an ML model that utilises many regression trees, each having many hierarchically organised criteria [33]. Applying ensemble learning with a single decision tree improves the predictive capability to anticipate the impact [34]. This approach demonstrates high efficacy in handling a substantial quantity of variables and offers several advantages, such as exceptional precision and efficiency, as well as efficiently addressing the issue of overfitting in high-dimensional data without requiring feature selection, regardless of whether the data are numerical or categorical [35]. Consequently, the utilisation of the RF algorithm in forecasting home values has emerged as a prevalent and significant subject in several research endeavours, all of which examine its effectiveness and adaptability. Ming et al. [12] applied RF to assess the Chengdu housing market and obtained an R^2 score of 0.83, indicating that it performed on par with the top models chosen for their research. In addition, Xu [15] successfully incorporated this algorithm in a comparison analysis employing the Boston housing dataset, resulting in an impressive R^2 score of 0.8329. This outcome demonstrated its exceptional capacity to appropriately evaluate pricing. In the research conducted by [14], RF was deployed for the purpose of prediction due to its ability to maintain the performance of decision trees while minimising the risk of overfitting. This strategic choice emphasises the equilibrium between complexity and abstraction in a model. By incorporating PSO feature selection approaches, Sakri et al. [16] successfully integrated RF and achieved notable improvements in accuracy. This may be attributed to its capability to efficiently manage many input variables. According to the study by [19], RF performed exceptionally well in predicting the values of residential properties in Bangalore when compared to other ML models and achieved a remarkable

R^2 score of 0.8512, which was the second highest among all the models evaluated, highlighting the robust forecasting ability of RF. Additionally, Yang et al. [18] used this algorithm as the main part of a stacking ensemble model, combining it with other methods to improve prediction results because it can handle data in many dimensions. All the results illustrated above show that RF has a diverse array of applications in predicting home values and can effectively contribute to both individual and integrated ensemble methods. This highlights its enduring importance in different housing markets and research approaches; therefore, RF was chosen as one of the models in this study.

- GBR: this is an ensemble approach that constructs models sequentially to rectify the mistakes made by the previous models. It is frequently successful for sophisticated datasets with complex structures, especially this dataset.
- XGBoost, known as Extreme Gradient Boosting, improves its predictive accuracy by using several tree models [36]. This strategy mitigates overfitting problems by incorporating a regularisation term into the model framework, which penalises complicated models and minimises the sensitivity of predictions to individual data points [37]. The tree-based architecture of XGBoost demonstrates its advantage in terms of robustness against outliers and prevention of overfitting [38]. Thus, XGBoost is an exceptionally efficient method for resolving regression tasks due to its incorporation of regular terms and second-order derivatives, resulting in the enhanced learning and processing speed of the algorithm [39,40]. The exceptional efficacy and versatility of this model have garnered notoriety in the current body of work on house price prediction employing ML techniques. The research of [12] demonstrated the effectiveness of XGBoost in analysing the Chengdu housing market, suggesting that this approach has high accuracy and a lower MSE with broader applications. In addition, XGBoost was employed by [15] as one of the models in a comparative comparison of ML methods, applying the Boston housing dataset. Despite achieving the highest R^2 value of 0.8776, the XGBoost algorithm proved its competitiveness by achieving comparable performance to the ANN, with an R^2 score of 0.8495, resulting in a difference of 0.0281 compared to the performance of the ANN with its R^2 score of 0.8776. XGBoost was deployed by [19] in practical use among the seven selected models. The result showed it has the most effectiveness in achieving accuracy rates of 0.9110 and 0.6465 on the flat and independent housing costs estimation. Additionally, Sheng et al. [17] employed a novel technique by integrating XGBoost with PSO to significantly improve the accuracy of the model on the Ames housing dataset. The PSO-XGBoost hybrid method outperformed other individual models and demonstrated the benefits of integrating XGBoost with optimisation techniques to capture nonlinear data patterns, leading to improved predictive reliability and precision at 0.9887. In a similar vein, Yang et al. [18] disrupted the traditional method of using a single algorithm by inventing a novel ensemble learning methodology called D-Stacking, which is based on a diversity approach. The method involved adding XGBoost to a complicated prediction framework. When validated, the results of the model's performance in the Chinese and American housing datasets were better than standalone models. The collaborative efforts demonstrate the robust and durable performance of XGBoost in the domain of an AI-based approach for property price prediction, illustrating its potential as an independent solution when combined with other methods such as feature selection optimisations, hybrid models, and ensemble techniques. These integrations establish a solid foundation for the reference of methodology design in the experiment section, strengthening the significance of applying XGBoost in advancing the analysis of the real estate market.
- An FNN, also referred to as a multilayer perceptron, is the most fundamental type of DL network, typically comprised of a layer for input, multiple hidden units, and an output layer [41]. Every neuron in the input layer obtains information, while the weighted total of data without bias is activated in the hidden layer; the preceding

and following layers are connected to each other with the bias-weighted sum of the outcomes [42,43]. The activations are calculated at every layer until they reach the output ones, while the activated score creates the predicted result [8]. FNNs can execute regression tasks and learn intricate characteristics of the supplied data, as well as fit most functions and have strong nonlinear fitting capabilities with a simple principle and feasible implementation. Thus, the task of forecasting rental value in the German housing sector could be completed by its application [43]. A model that has the capability to represent intricate connections between input and output variables. An FNN model is well suited for collecting complex patterns in the housing market data due to the capacity to learn nonlinear representations.

- PNN: The neural network (NN) models based on PyTorch can be defined, trained, and inferred by applying Python language with a Tensor type of data [44]. The process is more rapid and effective when PyTorch is used. Moreover, it is achievable to optimise the efficiency of DL models while maintaining complete flexibility throughout the model development and execution procedures [45].
- Stacked models: this is a technique that involves merging the predictions of multiple models to enhance overall performance and accuracy. This study intends to combine PNN with RFR, GBR, XGB, and LR to obtain superior accuracy compared to using a standalone model.
- The R^2 score evaluates the effectiveness of regression models by displaying the variance proportion of the dependent variable, which together establishes the independent factor [46]. The range of values for R^2 is 0 to 1. A score of 0 indicates that the model explains no variability around the mean of the response variable, while a score of 1 indicates that all the variability around the mean is explained by the response variable [47]. A higher R^2 score denotes the level of quality of the trained model [48].
- The RMSE is applied to evaluate the efficiency and performance of various prediction models by comparing the difference between predicted and actual values [47]. The model performs better at estimates when the value is smaller [49].

This methodology of the study stresses a well-balanced combination of traditional, ensemble, and neural network models, supported by extensive literature, to guarantee a thorough examination of rental price prediction techniques.

In conclusion, the literature review emphasises the extensive application of AI-based techniques for forecasting housing market trends. It also underscores the crucial role of feature selection in improving model performance. Prominent predictive models, including CNNs, XGBoost, RF, and hybrid models like PSO-XGBoost, have demonstrated high accuracy in various real estate markets. These models exhibit remarkable proficiency in handling complex relationships and high-dimensional datasets, showcasing their ability to capture the nonlinear interactions essential for precise home price forecasting.

In the field of feature selection, methods such as lasso filtering and PSO have shown promise in reducing dimensionality and identifying relevant factors. Research shows that combining these feature selection techniques with ML models can significantly improve prediction accuracy, particularly when dealing with high-dimensional datasets, such as those found in real estate [50]. Advanced models and feature selection algorithms work together to address issues related to overfitting, model robustness, and generalisation across various market conditions.

Nonetheless, prior studies reveal significant gaps. More comprehensive research is required, as indicated by the limited attention given to ensemble and hybrid approaches as well as the inconsistent assessment of model performance across various feature sets. Additionally, concerns about the generalisability of current research results arise due to a focus on specific market scenarios or individual datasets. To address these gaps, this work evaluates a variety of individual models and feature sets selected using PSO and CatBoost in different settings. The goal of this research is to create a method that consistently achieves an accuracy rate averaging 0.90 across different models and features, providing

deeper insights into the expected precision and adaptability of ML models for real estate market analysis.

3. Dataset

The immo_data dataset was chosen for this research on the German real estate market due to its extensive and broader usage. This dataset was sourced from ImmoScout24, the most prominent and well-recognised real estate portal in Germany, and is publicly accessible at <https://www.kaggle.com/datasets/corrieaar/apartment-rental-offers-in-germany> (accessed on 11 October 2023). The initial dataset comprised 268,850 entries and 49 variables, providing detailed information on various characteristics of residential units within the German real estate market. Specifically for this study, data from the Munich housing market were extracted, resulting in 4383 records and 49 variables. After feature engineering, the dataset was refined to include 4369 records and 23 features, each representing distinct types of data.

The housing attributes are described in Table 1:

- The Munich property market includes 23 distinct parameters for each residential unit, including living space size, rent, and number of rooms.
- The dataset contains a total of 4369 records, of which 20% are specifically designated for testing purposes.
- The data comprises four types: objects, floats, integers, and Booleans.

Table 1. Dataset dictionary.

Feature	Description
ServiceCharge	Auxiliary costs (electricity, water, etc.)
NewlyConst	Is the building newly constructed?
Balcony	Does it have a balcony?
Picturecount	How many photos are on the listing?
Pricetrend	The price trend as calculated by Immospot
TelekomUploadSpeed	How fast is the internet upload speed?
TotalRent	Total rent (sum of the base rent and other costs)
NoParkSpaces	Number of parking spaces
HasKitchen	Has a kitchen
Cellar	Has a cellar
YearConstructedRange	Binned construction year, 1 to 9
BaseRent	Base rent without electricity and heating
LivingSpace	Living space in sqm
Lift	Is an elevator available
BaseRentRange	Binned base rent, 1 to 9
Geo plz	ZIP code
NoRooms	The number of rooms
ThermalChar	Energy efficiency class in kWh/(m ² a)
Floor	Which floor is the flat on
NumberOfFloors	Number of floors in the building
NoRoomsRange	Binned number of rooms, 1 to 5
Garden	Has a garden
LivingSpaceRange	Binned living space, 1 to 7

However, facilitating the rigorous evaluation and comparative analysis of various algorithms on this dataset is challenging due to its complexity. Improper feature engineering may lead to overfitting. Moreover, it is essential to note that there are several housing datasets available for study on housing prediction, including the Boston, Ames, and Chinese datasets.

Currently, there is only one dataset pertaining to the German housing industry that has been made publicly accessible. This dataset stands out due to its distinctive formats, structures, and properties, which may require specific procedures and techniques for processing and analysis. Furthermore, its availability enhances its usefulness and replicability. Considering these factors, the immo_data dataset is well-suited for conducting research in the German market.

The immo_data dataset was selected for this study because of its extensive and heterogeneous collection of data that encompass several facets of the German real estate market. Although it has well-defined objectives and assessment criteria for comparing various techniques and models, this dataset has some limitations, including:

- The data was collected on a specific fixed date, emphasising the importance of frequent updates to maintain its reliability over time.
- The presence of duplicated and irrelevant entries may compromise the quality and accuracy of the models.
- An uneven distribution of labels may lead to biased outcomes.
- The dataset typically includes outliers or extreme observations, which are common in the real estate market. If these observations are not handled properly, they may affect the results of the models.

The shape of the label exhibits an upward skew (see Figure 1), showing a longer tail on the right side of the distribution. The significant decline in frequency as rent increases, with a small number of observations in the upper rent categories (e.g., rentals above 5000), implies the presence of outliers compared to average rental prices. This non-symmetry may negatively affect model estimation. Hence, when forecasting rental prices, it is crucial to consider the skewness of the rent values to normalise the data.

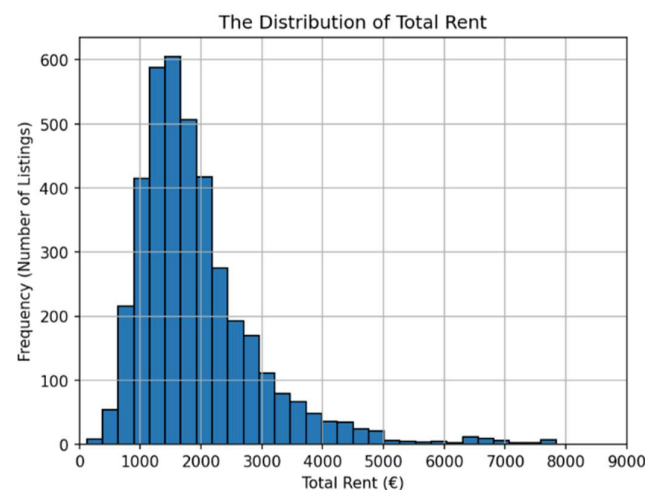


Figure 1. The shape of the target.

Two feature selection algorithms, CatBoost and PSO, were used to analyse feature significance within the dataset. The findings led to the creation of two unique feature sets, Sets 1 and 2, which demonstrate the significance of various characteristics in determining rental rates. Below is a breakdown of the pipeline used by both approaches to extract features for Sets 1 and 2.

After splitting the dataset into features (X) and the target variable (y) with an 80-20 training and testing split, feature set 1 was generated using the CatBoost algorithm. The CatBoost Regressor was configured with RMSE as the loss function, 100 iterations, a depth

of 6, and a learning rate of 0.1. Following training, the `get_feature_importance()` function was used to generate the feature importance scores, prioritising the top 10 features that contributed the most to prediction accuracy. Feature set 1 was then constructed using these top features.

PSO was employed to extract features from feature set 2 after pre-processing the features using normalisation and categorical handling. Several subsets of attributes were assessed using an objective function based on MSE, incorporating feature weights in each iteration. These weighted variables were used to train a basic linear regression model, with MSE serving as the optimisation guide. PSO parameters were defined to explore the solution space and identify the most optimal weights, incorporating cognitive and social factors. After averaging the relevance of each characteristic across several rounds, the top ten features were chosen for set 2 based on their highest ratings.

The differences between these two sets stem from the various techniques applied for feature selection. Set 1 is more model-specific, as it leverages the gradient boosting process of CatBoost to discover features that minimise loss. In contrast, feature set 2 employs a general-purpose PSO strategy that optimises feature weights to reduce prediction errors. Some variables, such as “baseRent” and “serviceCharge”, are scored highly in both sets, while other attributes differ, indicating that various approaches capture distinct portions of the data for predictions. These differences imply that combining techniques might result in a more thorough and reliable feature selection process, with PSO providing flexibility across various model types and CatBoost being more suitable for tree-based models. This comparative research shows that different methods have distinct effects on feature selection, which can enhance model design and improve prediction accuracy.

Figure 2 shows the importance of features selected by these tools. The vertical axis (y-axis) lists the names of the features in the dataset, while the horizontal axis (x-axis) represents the importance scores of the characteristics. A higher or longer bar indicates that a feature is highly relevant and has a greater impact on model predictions. According to the figure, “baseRent”, “baseRentRange”, and “serviceCharge” were the three most crucial characteristics in set 1, whereas “newlyConst_true”, “telekomUploadSpeed”, and “priceTrend” were the crucial features in set 2.

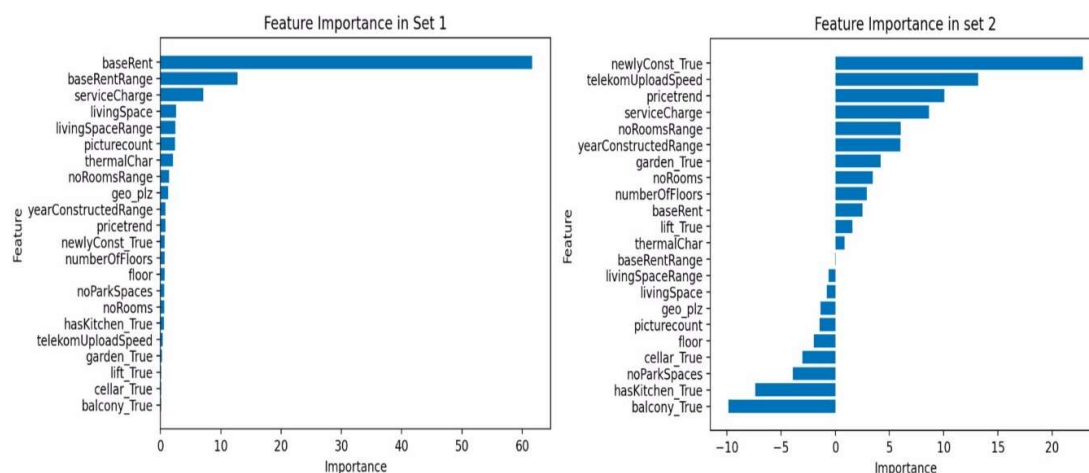


Figure 2. Feature importance in two sets.

The differences indicate the weight-specific attributes of various algorithms when projecting rental prices, demonstrating that the selection and ranking of elements vary based on the approach utilised. In both sets, for example, “baseRent” and “serviceCharge” consistently exhibited high relevance, highlighting their crucial role in establishing rental prices in the Munich market. In contrast, characteristics such as “lift true”, “cellar true”, and “balcony true” in set 1 and “thermalChar” in set 2, had low relevance, implying that their influence on the prediction was minimal.

Furthermore, the presence of negative relationship scores in set 2 highlights features that exhibit an inverse correlation with rental prices. For instance, as “thermalChar” increased, the rental price tended to decline. These revelations emphasise the significance of considering not just highly rated features but also comprehending the negative correlations that certain characteristics have with the target variable.

This thorough examination underscores the importance of choosing the appropriate characteristics based on the specific modelling method being implemented. It reveals that different variables produce varying predicted results depending on the algorithm, while certain features remain consistently significant.

4. Experiment

In the real world, stakeholders often face poorly informed choices that are both technically challenging and financially burdensome when forecasting rental values. Although ML models have been used in the existing literature, the instability of these results has made it difficult to provide accurate forecasts for stakeholders. This study aims to employ PyTorch as a sophisticated AI-based framework to analyse the complex dataset, as current methods have struggled to adapt to rapid market changes while maintaining consistent accuracy. The immo_data dataset, which is exclusive and openly accessible, will be used in this study to predict rental costs in the German market. Implementing PyTorch for rental value forecasting in this industry enhances accuracy, simplifies the process, and increases reliability through scalability and flexibility.

4.1. Experiment Design

The purpose of leveraging ML to forecast rental values in Munich, Germany, stems from the lack of current rental forecasting research in this city. As a prestigious city in Germany, Munich is experiencing significant growth due to its role as a vital hub for corporations and organisations. This expansion is driven by a steady influx of both local and international individuals. In this study, ten algorithms were used to predict rental expenses in the city, such as LR, RFR, GBR, XGB, PNN, FNN, and hybrid models. The models were trained and assessed using two distinct feature sets identified by CatBoost and PSO.

Figure 3 illustrates the approach and design of the experiment, showing the arrangement for predicting rental prices using a dataset called Immo_data. The figure explains the use of the ten algorithms, including LR, RF, GBR, XGB, and DL algorithms such as PNN and FNN. It also discusses stacked models that combine PNNs with other models. This study applied two distinct feature selection techniques, CatBoost and PSO, to generate two independent feature sets for evaluating the stability and performance of models.

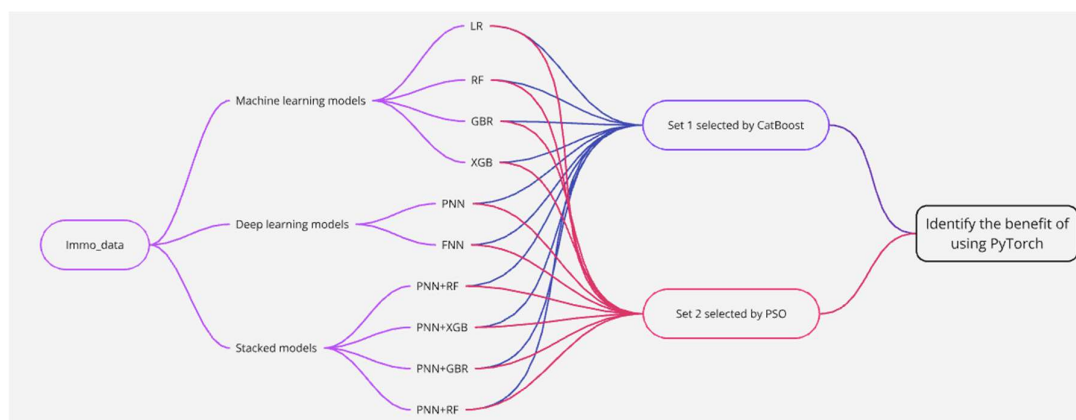


Figure 3. Experiment design map.

The study aims to provide insights into the application of PyTorch for rental price prediction tasks through evaluation and comparison. Employing a wide range of algorithms for this predictive work offers flexibility in model design and the ability to identify the optimal technique for improved accuracy.

4.2. Model Training and Evaluation

Given the comprehensive description of the training phase, several essential steps were taken throughout the learning process to guarantee accurate and reliable rental pricing projections for the ML models. Below is a detailed explanation of the procedure:

Data preparation

- Data cleaning: To improve the quality of the data, unnecessary observations, outliers, and inconsistencies were eliminated. K-nearest neighbours (KNNs) imputations was used to address missing numerical values, while missing categorical values were filled in using the fill-in approach.
- Feature engineering: Building predictive models involves a critical stage called feature engineering, especially in the real estate industry, where the quality of features can significantly impact model performance. Two feature engineering approaches were used in this case. The One-hot encoding technique transforms categorical variables like “newlyConst”, “balcony”, “Kitchen”, “cellar”, “lift”, and “garden” into numerical values for use in a ML model. Normalisation was applied to adjust features such as rental price, space, and range, ensuring these factors are on a comparable scale.
- Feature selection: Recent research has highlighted the application of hybrid models and meta-heuristic algorithms, such as PSO and CatBoost. These two methods were applied to choose the most pertinent characteristics. While CatBoost reduced dimensionality by converting categorical data into TSs, PSO optimised feature selection using an objective function.

Model architecture

- A variety of models were investigated based on their demonstrated performance in comparable predicting tasks and their capacity to manage intricate connections within the data. The algorithms used encompassed LR, tree-based algorithms, neural networks, and stacked models. In terms of neural network structure, regularisation techniques such as dropout were applied to both the PNNs and FNNs to prevent overfitting and provide a fair comparison between these two algorithms. Dropout regularisation was used in the hidden layers with ReLU activation for both models. Moreover, both FNN and PNN models shared a similar architecture, consisting of two hidden layers with 256 neurons, respectively, and ReLU activation functions in both layers to ensure a fair comparison. The only difference was the framework used (TensorFlow for FNN and PyTorch for PNN); however, the overall model structure and hyperparameters were consistent across both frameworks. This provides clarity on the use of frameworks and ensures a direct comparison between models built with TensorFlow and those built with PyTorch.

Training process

- Training–validation split: An 80:20 ratio was used to divide the dataset into training and validation sets. This ratio ensures that there is enough data for both learning and evaluation the predicted accuracy of the models on new instances, with the goal of reducing overfitting and enhancing generalisation. The training set contained 3495 records, while the validation set included 874 items after the split.
- Hyperparameter tuning: Key hyperparameters were methodically optimised across multiple rounds to improve model performance. These hyperparameters included the learning rate, batch size, number of layers, and number of neuron

per layer. The study assessed learning rates between 0.00001 and 0.1, examined batch sizes ranging from 8 to 152, modified the number of layers from one to four, and varied the number of neurons per layer from 8 to 520. The optimal configuration was found to be a learning rate of 0.001, a batch size of 64, two layers, and 256 neurons per layer.

- Epochs and iterations: A range of epochs, from 10 to 4000, was used to train the models. The models were continuously updated based on the error resulting from the difference between the actual and predicted values for each epoch. The best-performing epoch was found at 410 epochs.
- Loss function and optimisation: The Adam optimiser was utilised to minimise the loss, with MSE serving as the loss function. The ReLU activation function was employed in the neural networks, aiding the ability of the model to learn intricate patterns, maintain gradients, and reduce the vanishing gradient issue. This approach is computationally inexpensive, facilitating quicker training and more efficient learning.
- Regularisation: A regularisation approach, such as dropout, was applied in the PNN to address overfitting issues by randomly disabling a subset of neurons during training. This implies that a portion of neurons is temporarily eliminated from the network, including their connections, to compel the network to develop more robust features from the data that are not dependent on any specific group of neurons. This method enhances the ability of the model to generalise to new data by simplifying its complexity and avoiding over-reliance on certain features or patterns from the training set.

Post-training analysis

- Model comparison: To determine which models performed the best, all models were compared after training. The models stacked with PyTorch increased the accuracy of the lowest-performing models.

4.3. Methodology of Stacking Models

Stacking, also known as stacked generalisation, is an ensemble learning method that combines multiple algorithms to increase prediction accuracy. Using the same dataset, several base models are trained, and the predictions from these models are then used as inputs for a meta-model. Known by another name, the “stacker”, this meta-model learns to combine the outputs of the base models to generate a final forecast.

In the stacking process, several base models are initially chosen and trained on the training dataset. These algorithms can vary widely, including LR, tree-based models like RF and GBR, as well as neural networks, including FNNs and PyTorch-based neural networks. Every base model learns to forecast results using the provided feature set. After training, these models provide level-1 predictions, which are estimations made on a validation set. The next stage involves creating a new dataset, with each occurrence represented by the expected values from the base models. This dataset is then used to train the meta-model, which learns to optimally combine the predictions of the base models to produce an output. The meta-model delivers the final aggregated forecast for novel information by merging the predictions of the underlying models, which can be any predictive model.

In predictive modelling, the stacking strategy provides several crucial benefits. Stacking technique employs the distinct strengths of various model types-including linear models, tree-based models, and neural networks-while mitigating their weaknesses. This approach frequently yields better performance than using a single method alone. Predictive accuracy is typically improved by the ensemble technique due to the meta-model, or “stacker”, which learns how to effectively incorporate projections from base models using complementary strengths. Furthermore, stacking method minimises individual model deficiencies, reducing variance and overfitting, resulting in more robust and trustworthy predictions.

This study used a PNN in conjunction with various other methods, such as RF, GBR, XGB, and LR, to build stacking models. The goal of this hybrid strategy was to achieve better accuracy and stability than individual models by leveraging the distinctive capabilities of each algorithm. Performance significantly improved with PNN stacking, as evidenced by higher R^2 scores and lower RMSE values across various feature sets. The research effectively solved the difficulties involved in rental price prediction by layering several models, greatly improving overall predictive performance.

The developed methodology provided stakeholders with important insights by accurately forecasting rental prices in the German real estate market. Table 2 presents the R^2 score and RMSE values for the four ML methods without the PyTorch framework across the two feature sets. These four algorithms demonstrated a satisfactory level of accuracy. Nevertheless, for both feature sets, the singular models that did not use PyTorch exhibited lower R^2 scores, particularly evident for XGB and LR in set 1, as well as all tree-based models in set 2. These results suggest a reduced level of predictive accuracy, with elevated RMSE values suggesting more substantial prediction errors.

Table 2. Performance Comparison of ML Models Without PyTorch.

Model	Feature Set 1		Feature Set 2	
	R^2 Score	RMSE	R^2 Score	RMSE
RF	0.9187	289.87	0.8866	342.33
GBR	0.9158	294.89	0.8946	330.00
XGB	0.8903	336.68	0.8717	364.23
LR	0.8928	375.57	0.9045	314.21

The models in set 1 exhibited R^2 values ranging from 0.8903 to 0.9187, with the XGB model having the lowest score and the RF model achieving the highest score, with RMSE values ranging from 375.57 to 289.87. In set 2, the models displayed R^2 values that varied from 0.8717 to 0.9045, with the XGB model obtaining the lowest score and the LR model receiving the highest score, corresponding to RMSE values ranging from 364.23 to 314.21.

While RF achieved the highest accuracy of 0.9187 in set 1, its performance was somewhat lower in set 2, with an R^2 score of 0.8866. This suggests that the performance of a single model can be compromised when the characteristics change, thus impeding the dependability and reliability of the model.

Table 3 highlights the R^2 scores and RMSE values for FNN and PNN models using two different feature sets. The PyTorch NN model exhibited superior performance compared to the FNN model in terms of both R^2 score and RMSE for every feature set, suggesting a more optimal fit and more precise predictions. This consistent performance indicates that the PNN was somewhat more effective in this scenario.

Table 3. Performance Metrics of Neural Network Models.

Model	Feature Set 1		Feature Set 2	
	R^2 Score	RMSE	R^2 Score	RMSE
FNN	0.9000	321.47	0.9015	319.08
PyTorch NN	0.9041	314.83	0.9054	312.61

In Table 4, the performance of the FNN and PNN, as measured by R^2 scores across six separate runs, is compared to the implementation of TensorFlow in the FNN. Tests were conducted using the identical architecture and hyperparameters for both methods to determine if the PNN provided a statistically significant improvement over the FNN. The table illustrates that for each run, the PNN consistently outperformed the FNN in terms of R^2 scores. Even if the changes in R^2 scores appeared to be small, there may have been an overall increase in performance when employing the PyTorch-based model.

The RMSE values for the two models are compared, in addition to the assessment of R^2 scores. The PNN performed better than the FNN in all six runs, as shown in Table 5, with a consistently lower RMSE value. When predicting rental prices, the lower RMSE of the PNN implies improved prediction accuracy and decreased error. The variations in these results indicate that the PyTorch implementation enhances the performance of models in real-world prediction tasks.

Table 4. The Comparison of R^2 Scores.

Run	FNN	PNN
1	0.905	0.914
2	0.904	0.913
3	0.906	0.912
4	0.907	0.916
5	0.903	0.915
6	0.905	0.914

Table 5. The Comparison of RMSE.

Run	FNN	PNN
1	315.12	300.45
2	314.21	299.32
3	313.45	300.67
4	312.87	299.89
5	314.90	301.23
6	313.67	300.12

A paired t -test and p -value analysis for both R^2 scores and RMSE values were conducted to evaluate the differences in performance between the FNN and PNN, as shown in Table 6. With a p -value of 8.29×10^{-5} and a t -statistic of -11.619 for the R^2 scores, the results are well below the conventional threshold of 0.05, indicating that the PNN shows slight improvements over the FNN in terms of R^2 scores.

Table 6. The Comparison of Statistical Test Results.

	t -Statistic	p -Value
R^2 Score	-11.61895003862225	8.29×10^{-5}
RMSE	39.03519035424145	2.08×10^{-7}

Comparatively, the t -statistic for the RMSE was 39.035 with a p -value of 2.08×10^{-7} , suggesting that the PNN outperformed the FNN according to the RMSE results. The small p -values provide solid evidence for the improved efficiency of the PyTorch model, indicating that these results were not due to random chance. The statistical tests support the conclusion that the PNN performed considerably better than the FNN with regard to both R^2 and RMSE, thereby verifying the advantages of using PyTorch in rental price prediction.

Table 7 exhibits the R^2 score and RMSE values for four distinct ensemble models that integrate PNNs with other techniques, namely RF, GBR, XGB, and LR. These evaluations were conducted on both feature set 1 and feature set 2.

The R^2 value for feature set 1 exhibited a high level of correlation between the anticipated and actual values across all methods, and the respective RMSE values indicated a high level of precision in the forecasts, with minimal error. In contrast, feature set 2, while showing somewhat lower overall R^2 scores compared to feature set 1, still demonstrated a satisfactory level of predictive capability. The results of the RMSE exhibited a modest increase, suggesting a greater degree of prediction error in comparison to feature set 1.

These findings clearly demonstrate an overall improvement in the performance metrics of this integration compared to the previous research. This implies that the performance of

each model was effectively improved across various feature sets by combining a single ML algorithm with PyTorch. More specifically, the R^2 score showed improvement, while the RMSE decreased in comparison to the individual models. Although the magnitude of the increase may differ among the models, all models consistently exhibited an improvement exceeding 90% accuracy. Specifically, XGB exhibited the lowest performance among the 10 models in both feature sets prior to stacking with the PNN. However, the accuracy in both sets improved to 0.9097 and 0.9022, respectively, after being combined with the PNN.

Table 7. Performance Comparison of ML Models With PyTorch.

Model	Feature Set 1		Feature Set 2	
	R^2 Score	RMSE	R^2 Score	RMSE
PyTorch+RF	0.9181	290.87	0.9040	314.95
PyTorch+GBR	0.9150	296.35	0.9055	312.49
PyTorch+XGB	0.9097	305.51	0.9022	317.86
PyTorch+LR	0.9043	314.52	0.9071	309.88

These results indicate that integrating PNNs with ML algorithms yields more reliable predictive performance, regardless of the different circumstances tested. In existing studies, applying the same models to various datasets generates different outcomes, highlighting the unreliability of a model. In the context of applying PyTorch, it consistently provides superior R^2 scores and lower RMSE values in both feature sets, suggesting that the integration enhances the performance of the standalone ML models.

4.4. Predicted vs. Actual Plot

Figure 4 depicts four scatter plots comparing actual values with expected values obtained from various ML models and ensemble combinations in Set 1. Each plot includes a diagonal line representing parity between projected and real values, where proximity of points to this line indicates higher prediction accuracy.

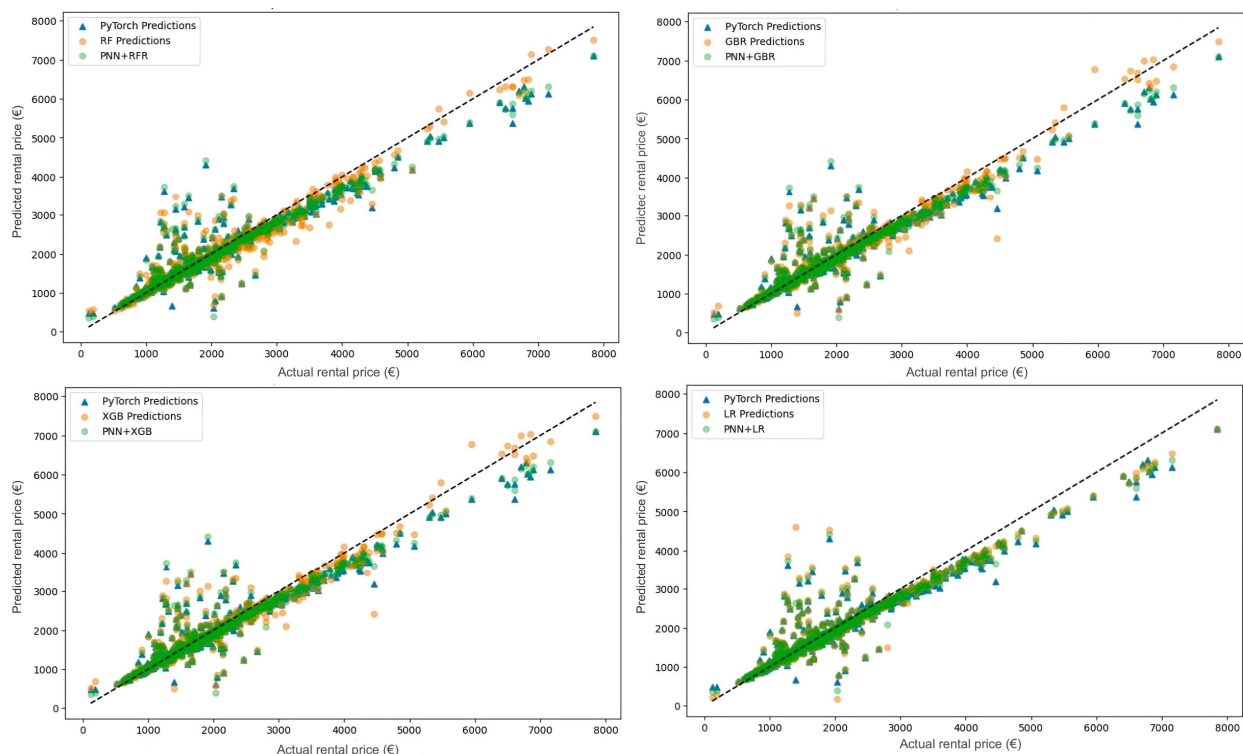


Figure 4. Comparison of model performance across feature set 1.

The ML models combined with PyTorch performed exceptionally well, as shown by the clustering of data points (represented by green dots) along the diagonal line. This suggests that the stacked model effectively captures underlying data patterns, yielding both higher R^2 scores and lower RMSE values. These metrics reflect more precise and reliable predictions compared to models that do not use PyTorch.

Set 2, depicted in Figure 5, shows scatter plots comparing actual values (x-axis) to anticipated values (y-axis) for each model. These diagrams indicate ensemble models combining PNN with RF, GBR, XGB, and LR. The plots illustrate that the stacked models exhibit tighter clustering around the diagonal line, implying enhanced prediction accuracy over standalone models (shown by orange dots). However, some data points deviate from the diagonal line, particularly at higher actual values, suggesting that prediction errors increase with the scale of the target variable.

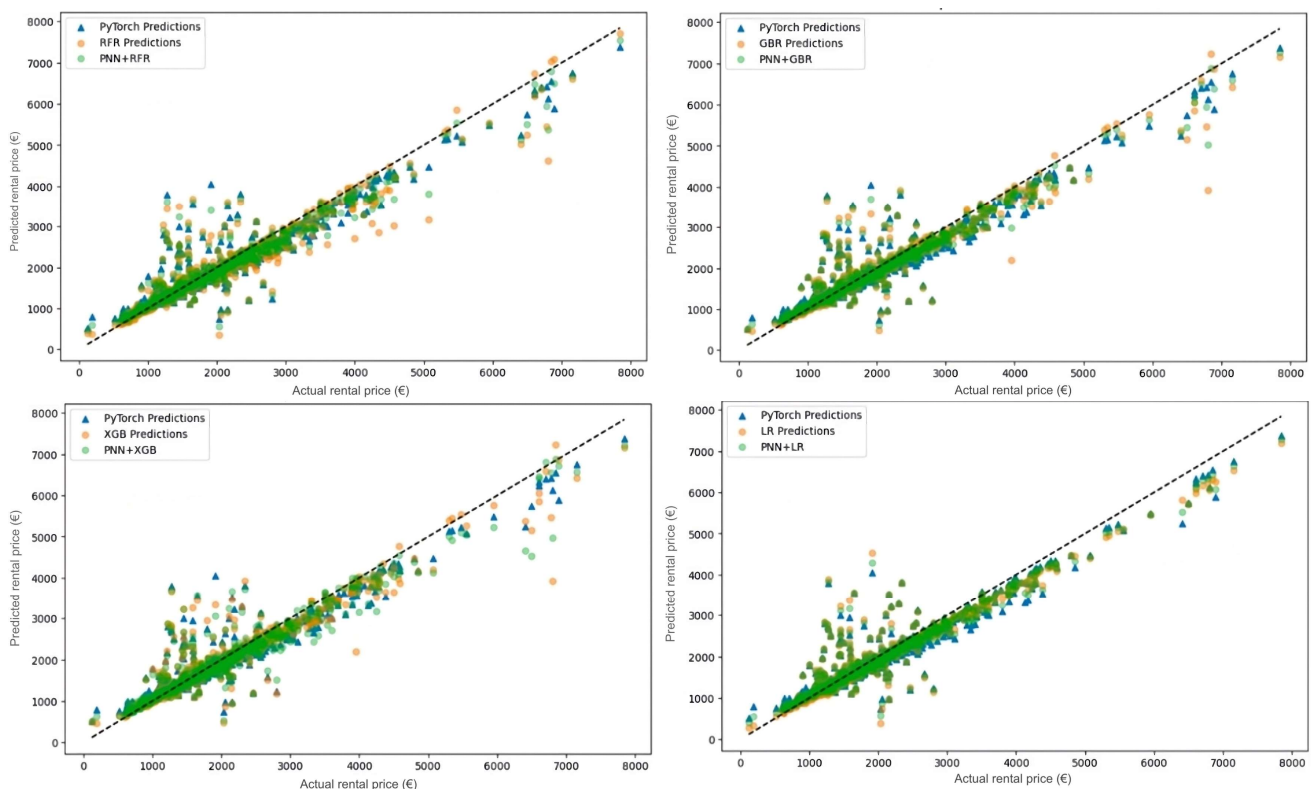


Figure 5. Comparison of model performance across feature set 2.

5. Investigation Assessment

PyTorch is an ML framework built upon the Torch library and is used for multiple applications, including computer vision and NLP. According to the findings, PyTorch framework, when used with standalone ML models, can also be applied to different scenarios for predicting rental values in the real estate industry.

In feature set 1, it is worth noting that while RF and GBR exhibited greater performance compared to the stacked models, their performance decreased when the features were altered. This highlights the limitations of relying solely on individual models. In set 2, the tree-based models exhibited poor performance, whereas these algorithms performed well in set 1.

The variation in performance across distinct feature sets may be attributed to the use of various feature selection techniques. CatBoost demonstrated robust predictive capabilities and greater accuracy across various datasets. This discrepancy can be attributed to the ways in which each approach processes the data. Changes in predicted accuracy may result from PSO performance variability, especially if the original feature set is insufficient.

While PSO can fine-tune feature weights, it may struggle with generalisation if the features chosen fail to capture the whole spectrum of significant variables. On the other hand, the ensemble technique of CatBoost improves its ability to generalize to untested data by combining several weak learners. Moreover, the feature relevance rankings of CatBoost provide insights into which factors influence predictions, crucial for those in the real estate industry to comprehend the primary attributes affecting rental rates. As opposed to this, the feature significance of PSO is less apparent, as it depends on the weighted combination of specific characteristics rather than an evaluation of individual features contributions to the model.

As a result, while CatBoost and PSO both provide useful techniques for predictive modelling, the distinctions between these methods in feature selection and performance underscore the importance of choosing the right algorithm for certain tasks. PSO may need thorough evaluation of initial feature sets and optimisation procedures; however, the strong handling of categorical data of CatBoost and ensemble learning capabilities often lead to better outcomes, especially for complex datasets. Comprehending these variations not only aids in selecting suitable models but also amplifies the overall efficacy of predictive analyses in real estate.

This study utilised a stacking strategy, integrating a PyTorch-based neural network with various other ML models. A variety of ML methods, such as LR, XGB, GBR, and RFR, were employed as base models, with a PNN implemented as the meta-model. The base models generated predictions, which were then used as input features for the meta-model. This stacking approach consistently enhanced the performance of single models. For instance, the R^2 score of XGB rose from 0.8903 to 0.9097 in feature set 1 and from 0.8717 to 0.9022 in feature set 2. Stacked models performed consistently across various feature sets, indicating increased resilience and generalisation. By optimally integrating predictions from multiple models, stacking helps overcome the shortcomings of a single model. In predicting rental prices in the Munich real estate market, this technique shows the potential for combining conventional ML models with neural networks via PyTorch to improve forecast accuracy and reliability.

Previous research primarily utilised standalone models such as CNNs, BPNNs, and SVMs for predicting tasks. This study explored the benefits of integrated techniques to improve model performance and reliability, which can be applied to various scenarios, such as investigating the housing market in other regions. Further research could explore further applications of PyTorch for more sophisticated methodologies within this field.

The immo_data dataset provides useful insights into the German housing market, although its discrete data and complex structure may limit the efficacy of ML models. Consequently, two features sets were employed to assess the durability and reliability of PyTorch, thereby enhancing the precision of the ML models.

It is notable that certain characteristics in the dataset, such as base rent, recent construction, and picture trend, strongly correlate with rental prices. These factors are more likely to impact outcomes and are essential for effective model training. The concentration of labels in the lower rental price range can lead to biased outcomes, but normalising the data and removing inconsistent observations help mitigate this skewness.

On the other hand, this work possesses several limitations that could be addressed in further investigations. One limitation is the use of only two evaluation metrics, which may not fully capture the most effective assessment strategy for this regression task. Various feature engineering techniques, such as KNN imputation, mean imputation, outlier reduction, or data removal, could lead to diverse outputs and may require specific adjustments. Furthermore, this study used a single dataset sourced from a real estate website, which may not fully represent the complexities and variations within the German real estate market. Future research could apply more authoritative data from the German government to provide a more accurate depiction of this property market.

In summary, the key findings of this research include:

- The PyTorch framework is highly robust and adaptable for all ML algorithms, even with different feature sets. With an average accuracy level of 90%, the statistical significance test results for the R^2 score and RMSE indicate that PyTorch demonstrates minimal sensitivity to feature changes. However, standalone ML models, such as RF, XGB, GBR, and LR, are more affected by changes in features, while neural networks remain unaffected.
- Among the models, XGB model shows the most significant variability across feature sets, resulting in greater performance differences.
- Of the neural network models, PNNs exhibit superior accuracy on both feature sets, achieving higher accuracy than FNNs, as verified by statistically substantial improvements in both R^2 and RMSE scores. These DL models employ numerous layers of nonlinear transformations to capture intricate characteristics in the data, leading to an excellent R^2 score.

This study contributes to the existing literature on housing prediction by examining the impact of PyTorch on ML models in this domain. Moreover, the development and implementation of more robust, integrated methodologies can effectively address the challenges posed by complex datasets when predicting rental values.

6. Summary and Discussion

Based on the constraints and research gaps identified in the current literature, the improvements introduced by this study are presented in Table 8.

Previous research has primarily utilised basic ensembles or single-model techniques, achieving accuracy levels ranging from 0.6465 to 0.9887. However, this investigation incorporated ten distinct models, including advanced neural networks and stacked techniques, providing a more comprehensive analysis and improved forecast reliability. Unlike other studies concentrated on simple feature sets and one-dimensional reduction methods, this work leveraged various feature sets through a more sophisticated feature engineering process. The robustness of the approach is evident in the consistent performance observed across various sets.

The R^2 scores of the 10 ML models used in this study to forecast Munich rental values ranged from 0.8903 to 0.9187 for feature set 1 and from 0.8717 to 0.9071 for feature set 2. These scores underscore the consistent accuracy achieved by integrating the PyTorch framework with ML methods. In addition, the superiority of PyTorch-based models was further validated through a comparison between FNNs and PNNs. With p -values of 8.29×10^{-5} for R^2 and 2.08×10^{-7} for RMSE, the statistical significance test confirms that the PNN performed more accurately than other methods on both R^2 and RMSE, proving these enhancements are not due to random variation. The stacked strategy consistently exhibited improved performance, aligning with previous research that supported the integrated D-stacking model for its higher accuracy compared to single-model methods.

The objective of this study was to improve ML model precision in predictive tasks by leveraging the PyTorch framework, which has not been thoroughly explored in the existing literature. While previous research focused on individual model performance, this study proposed that exploring the use of PyTorch alongside ML models yields substantial benefits, offering a novel contribution to the field. The increased precision across both feature sets further highlights the advantages of this technique, suggesting a transition in home value estimation towards a more sophisticated, data-centric approach.

Table 8. Comparative Analysis of Previous Research and Improvement.

Reference	Dataset	Methods Used	Preferred Model	Result	Limitations	Improvement from Proposed Study
[10]	Taiwan	CNN, BPNN	CNN	$R^2 > 0.945$	In existing research, single models are frequently employed to estimate real estate prices; nevertheless, comparative analyses consistently show that neural networks perform better.	Unlike other research, this study divided the dataset into two sets with distinct relevant features, using two feature selection techniques: PSO and CatBoost. Ten different models, incorporating both machine learning (ML) and deep learning (DL) techniques, were applied for comparative analysis. The study also introduces PyTorch, a versatile framework that optimizes the training process to enhance model performance. Additionally, steps were taken to further increase accuracy, particularly for models with lower performance.
[11]	Taichung, Taiwan	LSSVR, CART, GRNN, BPNN	LSSVR	MAPE = 0.228		
[12]	Chengdu, China	RF, XGBoost, LightGBM	XGBoost	Accuracy = 0.85 MSE = 0.04		
[13]	Shenzhen, China	SVM	SVM	$R^2 = 1$		
[15]	Boston, the US	LR, RF, SVM, ANN, XGBoost,	ANN	$R^2 = 0.8776$		
[14]	Boston, the US	LR, Ridge, Lasso, RF, SVM with RBF, Polynomial Regression	SVM with RBF	Score = 0.799		
[19]	Bangalore, India	LR, Lasso, Ridge, DT, RF, XGBoost, XGRFBoost	XGBoost	Accuracy = 0.6465	The PSO method was used to reduce dimensionality. Various models performed differently, with some demonstrating higher prediction accuracy than others. However, it remains a reliable reference approach for the proposed study.	
[16]	Ames, the US	Compare models with and without PSO: RF, NB, LR, MLP, KNN, REPTree	RF with PSO	Accuracy = 0.844 RMSE = 0.0699 MAE = 0.019		
[17]	Ames, the US	Ridge, Lasso, XGBoost, LightGBM, PSO-XGBoost	PSO-XGBoost	$R^2 = 0.9887$ RMSE = 0.1015		
[18]	CS House US House	KNN, SVM, RF, GBDT, XGBoost, BPNet	Stacking Models	RMSE = 0.869 RMSE = 1.029	One stacking strategy, integrated with multiple models was used to predict the price. It's challenging to assess the effectiveness of this method without making comparisons with alternative approaches.	

This advancement has practical implications for stakeholders in the real estate sector. The integration of PyTorch with diverse ML models has resulted in notable enhancements in forecast precision, providing stakeholders with more reliable rental prices estimates. This development facilitates better financial planning and investment decisions, enabling investors and landlords to set competitive rental rates, maximise profits, and reduce vacancies. Real estate brokers can provide more accurate advice to clients, aiding strategic pricing, negotiations, as well as informed decisions regarding the purchase, sale, or rental of properties. In dynamic markets where rental prices fluctuate, PyTorch-based models offer flexibility, ensuring adaptability to changing feature sets and market conditions. In addition, the analysis of complex datasets enhances comprehension of stakeholders for the factors influencing rental prices, thereby guiding investment decisions and focused marketing tactics. Precise projections also support policymakers in monitoring market trends and formulating measures to stabilise rental markets, promoting accessibility and fairness. By adopting cutting-edge prediction models, real estate companies can gain a competitive edge, strengthening their market position and attracting new clients. Overall, this approach benefits industry players by incorporating these positive implications into their strategies.

Despite these improvements, limitations exist. Certain important attributes were either overlooked or lacked predictive value. Furthermore, the choice of ML algorithms for prediction was based on specific data assumptions that may vary from real-world circumstances. Limited exploration of the hyperparameter space often due to time or computational constraints, may also hinder model performance. Although R^2 and the RMSE are widely employed measures to evaluate ML methods, these may not fully capture the precision of models. Additionally, the complexity of neural networks and stacked approaches may pose a challenge for individuals without specialised knowledge.

Subsequent investigations can build upon these discoveries by exploring other methodologies based on PyTorch in different regions, assessing the efficacy of the proposed strategy across various real estate markets. This can be achieved by employing multiple datasets and incorporating other advanced AI techniques to enhance rental values predictions.

7. Conclusions

To conclude, this research presents a new strategy for rental price prediction in the German real estate market by employing AI-based approaches. The study shows how merging conventional models with advanced neural network frameworks can improve prediction accuracy through the integration of PyTorch with different ML approaches. By leveraging both neural networks and ensemble learning, this approach outperforms earlier methods that struggled with the complexity of large datasets.

The principal benefit of the suggested approach lies in its ability to address nonlinearity and high-dimensionality in rental pricing data using sophisticated techniques like XGBoost and PyTorch-based neural networks. Additionally, the stacking model technique effectively combines predictions from multiple models when working with heterogeneous feature sets, resulting in more reliable and accurate results.

Despite its advantages, the proposed approach has some limitations. Key features that could have enhanced prediction accuracy were excluded during feature selection. Time and computational resource constraints also limited the extent of hyperparameter tuning, potentially impacting model performance. While R^2 and RMSE are vital evaluation metrics, these may not fully capture the complex behaviour of advanced models. Furthermore, the complexity of stacked models and neural networks can present challenges for individuals without advanced ML expertise. Future research should concentrate on extending statistical validation across more datasets and real-world scenarios, as well as enhancing the transparency and interpretability of complex models.

This study offers several opportunities for further exploration by readers and future scholars. Utilising NLP techniques could allow prediction models to incorporate additional data sources, such as textual information from real estate listings, potentially boosting

prediction accuracy. Expanding this study to other regions, such as different German cities, China, and the UK, could help evaluate how well the recommended strategy generalizes to other real estate markets. Additionally, more advanced hyperparameter optimisation methods, like automatic machine learning (AutoML) or Bayesian optimisation, could further refine model performance.

Future work can examine other prevalent frameworks like TensorFlow, Keras, and MXNet to assess whether PyTorch offers unique benefits in rental price prediction. A comparative evaluation of these frameworks could highlight their respective advantages and help determine the most effective approach for various real-world applications. Comparative studies on training times and processing speeds across frameworks would also provide valuable insights, particularly for applications requiring rapid predictions.

Developing hybrid models that integrate traditional ensemble methods with DL approaches, such as transformer-based or recurrent neural networks (RNNs) is another promising direction. In addition, enhancing model interpretability using techniques like Local Interpretable Model-Agnostic Explanations (LIMEs) would be beneficial. LIME can help users understand the decision-making process of complex models and can be used to a range of inputs, including text, images, diagrams, and tabular data. Improving interpretability is vital for practical applications, particularly in fields where transparency is critical.

Extending statistical validation to more datasets and diverse market situations is another crucial subject for future research. While this study shown statistically substantial improvements in PNN models, further validation could reinforce the robustness and reliability of the suggested models, ensuring that they are applicable to a wider range of datasets and market scenarios.

Overall, the proposed framework offers a tool for accurate rental price prediction, enabling data-driven decision-making for real estate stakeholders. Although this study represents a major step forward in the field of rental price prediction, there remains ample opportunity for further discovery and refinement. As the field evolves, incorporating more complex AI techniques and addressing model limitations will contribute to advancements in both real estate practice and academic research.

Author Contributions: Methodology, W.C. and S.F.; validation, U.B. and H.A.-K.; investigation, all authors; formal analysis W.C. and S.F.; writing—original draft, W.C., S.F. and U.B.; writing—review and editing, U.B. and H.A.-K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhantileuov, E.; Smayil, A.; Aibatbek, A.; Kassymkhanov, S. A case study of machine learning comparisons for predicting apartment prices in astana. In Proceedings of the 2023 IEEE International Conference on Smart Information Systems and Technologies (SIST), Astana, Kazakhstan, 4–6 May 2023; pp. 305–309.
2. Khandaskar, S.; Panjwani, C.; Patil, V.; Fernandes, D.; Bajaj, P. House and rent price prediction system using regression. In Proceedings of the 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 14–16 June 2023; pp. 1733–1739.
3. Kindermann, F.; Le Blanc, J.; Piazzesi, M.; Schneider, M. *Learning about Housing Cost: Survey Evidence from the German House Price Boom*; Technical report; National Bureau of Economic Research: Cambridge, MA, USA, 2021.
4. Truong, Q.; Nguyen, M.; Dang, H.; Mei, B. Housing price prediction via improved machine learning techniques. *Procedia Comput. Sci.* **2020**, *174*, 433–442. [[CrossRef](#)]

5. Yoshida, T.; Murakami, D.; Seya, H. Spatial prediction of apartment rent using regression-based and machine learning-based approaches with a large dataset. *J. Real Estate Financ. Econ.* **2022**, *69*, 1–28. [\[CrossRef\]](#)
6. Sharma, S.; Arora, D.; Shankar, G.; Sharma, P.; Motwani, V. House price prediction using machine learning algorithm. In Proceedings of the 2023 7th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 23–25 February 2023; pp. 982–986.
7. Cekic, M.; Korkmaz, K.N.; Müküüs, H.; Hameed, A.A.; Jamil, A.; Soleimani, F. Artificial intelligence approach for modeling house price prediction. In Proceedings of the 2022 2nd International Conference on Computing and Machine Intelligence (ICMI), Istanbul, Turkey, 15–16 July 2022; pp. 1–5.
8. Samek, W.; Montavon, G.; Lapuschkin, S.; Anders, C.J.; Müller, K.-R. Explaining deep neural networks and beyond: A review of methods and applications. *Proc. IEEE* **2021**, *109*, 247–278. [\[CrossRef\]](#)
9. d’Errico, A.; Michalski, N.; Brainard, J.; Manz, K.M.; Manz, K.; Schwettmann, L.; Mansmann, U.; Maier, W. *World Health Day 2022: Impact of COVID-19 on Health and Socioeconomic Inequities*; Frontiers Media SA: Munich, Germany, 2023; p. 42.
10. Zhan, C.; Wu, Z.; Liu, Y.; Xie, Z.; Chen, W. Housing prices prediction with deep learning: An application for the real estate market in taiwan. In Proceedings of the 2020 IEEE 18th International Conference on Industrial Informatics (INDIN), Warwick, UK, 20–23 July 2020; Volume 1, pp. 719–724.
11. Pai, P.-F.; Wang, W.-C. Using machine learning models and actual transaction data for predicting real estate prices. *Appl. Sci.* **2020**, *10*, 5832. [\[CrossRef\]](#)
12. Ming, Y.; Zhang, J.; Qi, J.; Liao, T.; Wang, M.; Zhang, L. Prediction and analysis of chengdu housing rent based on xgboost algorithm. In Proceedings of the 3rd International Conference on Big Data Technologies, New York, NY, USA, 18–20 September 2020; pp. 1–5.
13. Lv, C.; Liu, Y.; Wang, L. Analysis and forecast of influencing factors on house prices based on machine learning. In Proceedings of the 2022 Global Conference on Robotics, Artificial Intelligence and Information Technology (GCRAIT), Chicago, IL, USA, 30–31 July 2022; pp. 97–101.
14. Wang, Y. The comparison of six prediction models in machine learning: Based on the house prices prediction. In Proceedings of the 2022 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE), Guangzhou, China, 5–7 August 2022; pp. 446–451.
15. Xu, J. A novel deep neural network-based method for house price prediction. In Proceedings of the 2021 International Conference of Social Computing and Digital Economy (ICSCDE), Chongqing, China, 28–29 August 2021; pp. 12–16.
16. Sakri, S.B.; Ali, Z. Analysis of the dimensionality issues in house price forecasting modeling. In Proceedings of the 2022 Fifth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU), Riyadh, Saudi Arabia, 28–29 March 2022; pp. 13–19.
17. Sheng, C.; Yu, H. An optimized prediction algorithm based on xgboost. In Proceedings of the 2022 International Conference on Networking and Network Applications (NaNA), Urumqi, China, 3–5 December 2022; pp. 1–6.
18. Yang, Z.; Zhu, X.; Zhang, Y.; Nie, P.; Liu, X. A housing price prediction method based on stacking ensemble learning optimization method. In Proceedings of the 2023 IEEE 10th International Conference on Cyber Security and Cloud Computing (CSCloud)/2023 IEEE 9th International Conference on Edge Computing and Scalable Cloud (EdgeCom), Xiangtan, China, 1–3 July 2023; pp. 96–101.
19. Disha, U.B.; Saxena, S. Real estate property price estimator using machine learning. In Proceedings of the 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), Greater Noida, India, 28–30 April 2023; pp. 895–900.
20. Almohimeed, A.; Saad, R.M.A.; Mostafa, S.; El-Rashidy, N.; Farag, S.; Gaballah, A.; Elaziz, M.A.; El-Sappagh, S.; Saleh, H. Explainable artificial intelligence of multi-level stacking ensemble for detection of alzheimer’s disease based on particle swarm optimization and the sub-scores of cognitive biomarkers. *IEEE Access* **2023**, *11*, 123173–123193. [\[CrossRef\]](#)
21. Gad, A.G. Particle swarm optimization algorithm and its applications: A systematic review. *Arch. Comput. Methods Eng.* **2022**, *29*, 2531–2561. [\[CrossRef\]](#)
22. Joodaki, N.Z.; Bagher Dowlatshahi, M.; Joodaki, M. A novel ensemble feature selection method through Type I fuzzy. In Proceedings of the 2022 9th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), Bam, Iran, 2–4 March 2022; pp. 1–6.
23. Hassan, R.; Hamid, O.; Brahim, E. Induction motor current control with torque ripples optimization combining a neural predictive current and particle swarm optimization. In Proceedings of the 2023 9th International Conference on Control, Decision and Information Technologies (CoDIT), Rome, Italy, 3–6 July 2023; pp. 2067–2072.
24. Shami, T.M.; El-Saleh, A.A.; Alswaitti, M.; Al-Tashi, Q.; Summakieh, M.A.; Mirjalili, S. Particle swarm optimization: A comprehensive survey. *IEEE Access* **2022**, *10*, 10031–10061. [\[CrossRef\]](#)
25. Wu, X.; Li, C.; Jiang, J.; Sun, A.; Zhang, Q. Distribution network reconfiguration based on improved particle swarm optimization algorithm. In Proceedings of the 2023 IEEE 7th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 15–17 September 2023; Volume 7, pp. 971–975.
26. El Hammedi, H.; Chrouta, J.; Khaterchi, H.; Zaafour, A. Comparative study of mppt algorithms: PO, INC, and PSO for PV system optimization. In Proceedings of the 2023 9th International Conference on Control, Decision and Information Technologies (CoDIT), Rome, Italy, 3–6 July 2023; pp. 2683–2688.

27. Wang, Z.; Ren, H.; Lu, R.; Huang, L. Stacking based lightgbm-catboost-randomforest algorithm and its application in big data modeling. In Proceedings of the 2022 4th International Conference on Data-driven Optimization of Complex Systems (DOCS), Chengdu, China, 28–30 October 2022; pp. 1–6.
28. Zhong, C.; Geng, F.; Zhang, X.; Zhang, Z.; Wu, Z.; Jiang, Y. Shear wave velocity prediction of carbonate reservoirs based on catboost. In Proceedings of the 2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 28–31 May 2021; pp. 622–626.
29. Ye, X.; Li, Y.; Feng, X.; Heng, C. A crypto market forecasting method based on catboost model and bigdata. In Proceedings of the 2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP), Xi'an, China, 15–17 April 2022; pp. 686–689.
30. Zhang, C.; Chen, Z.; Zhou, J. Research on short-term load forecasting using k-means clustering and catboost integrating time series features. In Proceedings of the 2020 39th Chinese Control Conference (CCC), Shenyang, China, 27–29 July 2020; pp. 6099–6104.
31. Chen, Y.; Xue, R.; Zhang, Y. House price prediction based on machine learning and deep learning methods. In Proceedings of the 2021 International Conference on Electronic Information Engineering and Computer Science (EIECS), Changchun, China, 23–26 September 2021; pp. 699–702.
32. Kalaivani, K.; Kanimozhiselvi, C.; Bilal, Z.M.; Suresh, G.; Yokeswaran, S. A comparative study of regression algorithms on house sales price prediction. In Proceedings of the 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), Trichy, India, 23–25 August 2023; pp. 826–831.
33. Alshammari, T. Evaluating machine learning algorithms for predicting house prices in Saudi Arabia. In Proceedings of the 2023 International Conference on Smart Computing and Application (ICSCA), Hail, Saudi Arabia, 5–6 February 2023; pp. 1–5.
34. Zhou, Q.; Zhu, P.; Huang, Z.; Zhao, Q. Pest bird density forecast of transmission lines by random forest regression model and line transect method. In Proceedings of the 2020 7th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS), Guangzhou, China, 13–15 November 2020; pp. 527–530.
35. Kurniawati, N.; Novita Nurmala Putri, D.; Kurnia Ningsih, Y. Random forest regression for predicting metamaterial antenna parameters. In Proceedings of the 2020 2nd International Conference on Industrial Electrical and Electronics (ICIEE), Lombok, Indonesia, 20–21 October 2020; pp. 174–178.
36. Zhu, R.; Yang, Y.; Chen, J. Xgboost and cnn-lstm hybrid model with attention-based stock prediction. In Proceedings of the 2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI), Changchun, China, 26–28 May 2023; pp. 359–365.
37. El Houda, B.N.; Lakhdar, L.; Abdallah, M. Time series analysis of household electric consumption with xgboost model. In Proceedings of the 2022 4th International Conference on Pattern Analysis and Intelligent Systems (PAIS), Oum El Bouaghi, Algeria, 12–13 October 2022; pp. 1–6.
38. Zhang, X.; Yan, C.; Gao, C.; Malin, B.A.; Chen, Y. Predicting missing values in medical data via xgboost regression. *J. Healthc. Inform. Res.* **2020**, *4*, 383–394. [[CrossRef](#)] [[PubMed](#)]
39. Ge, J.; Zhao, L.; Yu, Z.; Liu, H.; Zhang, L.; Gong, X.; Sun, H. Prediction of greenhouse tomato crop evapotranspiration using xgboost machine learning model. *Plants* **2022**, *11*, 1923. [[CrossRef](#)] [[PubMed](#)]
40. Qiu, Y.; Zhou, J.; Khandelwal, M.; Yang, H.; Yang, P.; Li, C. Performance evaluation of hybrid woa-xgboost, gwo-xgboost and bo-xgboost models to predict blast-induced ground vibration. *Eng. Comput.* **2021**, *38*, 4145–4162. [[CrossRef](#)]
41. Wang, W.; Dong, W.; Yu, T.; Du, Y. Research on prs/irs time registration based on fully connected neural network. In Proceedings of the 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 11–13 December 2020; Volume 9, pp. 942–947.
42. Jia, B.; Zhang, Y. Spectrum analysis for fully connected neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 10091–10104. [[CrossRef](#)] [[PubMed](#)]
43. Li, Q.; Zhai, Z.; Li, Q.; Wu, L.; Bao, L.; Sun, H. Improved bathymetry in the south China sea from multisource gravity field elements using fully connected neural network. *J. Mar. Sci. Eng.* **2023**, *11*, 1345. [[CrossRef](#)]
44. Lee, K.H.; Park, J.; Kim, S.-T.; Kwak, J.Y.; Cho, C.S. Design of nnf-pytorch neural network model converter. In Proceedings of the 2021 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 20–22 October 2021; pp. 1710–1712.
45. Sawarkar, K. *Deep Learning with PyTorch Lightning: Swiftly Build High-Performance Artificial Intelligence (AI) Models Using Python*; Packt Publishing Ltd.: Birmingham, UK, 2022.
46. Rustam, F.; Reshi, A.A.; Mehmood, A.; Ullah, S.; On, B.-W.; Aslam, W.; Choi, G.S. Covid-19 future forecasting using supervised machine learning models. *IEEE Access* **2020**, *8*, 101489–101499. [[CrossRef](#)]
47. Sahoo, A.; Ghose, D.K. Imputation of missing precipitation data using KNN, SOM, RF, and FNN. *Soft Comput.* **2022**, *26*, 5919–5936. [[CrossRef](#)]
48. Almaslukh, B. A gradient boosting method for effective prediction of housing prices in complex real estate systems. In Proceedings of the 2020 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), Taipei, Taiwan, 3–5 December 2020; pp. 217–222.

49. Guang, W.; Zubao, S. Research on the application of integrated rg-lstm model in house price prediction. In Proceedings of the 2023 IEEE 5th International Conference on Power, Intelligent Computing and Systems (ICPICS), Shenyang, China, 14–16 July 2023; pp. 348–353.
50. Karamti, H.; Alharthi, R.; Anizi, A.A.; Alhebshi, R.M.; Eshmawi, A.; Alsubai, S.; Umer, M. Improving prediction of cervical cancer using knn imputed smote features and multi-model ensemble learning approach. *Cancers* **2023**, *15*, 4412. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.