

Article

Optimizing Machine Learning Models for Urban Sciences: A Comparative Analysis of Hyperparameter Tuning Methods

Tris Kee  and Winky K.O. Ho * 

Department of Building and Real Estate, The Hong Kong Polytechnic University, Hong Kong SAR, China; tris.kee@polyu.edu.hk

* Correspondence: winky.ho@polyu.edu.hk

Abstract

Advancing urban scholarship and addressing pressing challenges such as gentrification, housing affordability, and urban sprawl require robust predictive models. In urban sciences, the performance of these models depends heavily on hyperparameter tuning, yet systematic evaluations of tuning approaches remain limited. This study compares two traditional hyperparameter tuning methods, Random Search and Grid Search, with Optuna, a more recent and advanced optimization framework, using housing transaction data as an illustrative case. Our findings show that Optuna substantially outperforms the other methods, running 6.77 to 108.92 times faster while consistently achieving lower error values across multiple evaluation metrics. By demonstrating both efficiency and accuracy gains, this research underscores the potential of advanced tuning strategies to accelerate urban analytics and provide more reliable evidence for policy-making.

Keywords: hyperparameter tuning; random search; grid search; Optuna



Academic Editor: Dario Domingo

Received: 13 July 2025

Revised: 26 August 2025

Accepted: 29 August 2025

Published: 31 August 2025

Citation: Kee, T.; Ho, W.K.O.

Optimizing Machine Learning Models for Urban Sciences: A Comparative Analysis of Hyperparameter Tuning Methods. *Urban Sci.* **2025**, *9*, 348.

<https://doi.org/10.3390/urbansci9090348>

Copyright: © 2025 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/>).

1. Introduction

Machine learning (ML) has emerged as a transformative technology, revolutionizing a wide array of academic disciplines and industries through its ability to derive valuable insights from complex and large datasets. Integrating statistical techniques, computational algorithms, and artificial intelligence, ML has attracted widespread attention due to its robust capacity to identify patterns, make predictions, and automate decision-making processes. Machine learning is becoming more commonly used in urban sciences; in such settings, it is used to analyze complex data, uncover patterns, and generate predictions across diverse areas such as transportation, urban meteorology, and real estate. The versatility and scalability of ML have positioned it as an indispensable tool for researchers and professionals aiming to unlock the full potential of their datasets.

Several studies demonstrate the application of ML in transportation research, primarily for improving transportation systems. Deep neural networks, specifically convolutional and long short-term memory (LSTM) layers, are used to predict urban bus travel times [1]. This approach captures spatiotemporal correlations and uses an encoder/decoder architecture for accurate multi-time-step predictions, outperforming traditional methods and even Google's model in the Greater Copenhagen Area. Another study uses Context-dependent Random Forests to predict traffic flow based on Global Positioning System (GPS) data from a small percentage of vehicles, significantly improving prediction accuracy for intelligent GPS navigation systems [2].

ML models like Stochastic Gradient Boosting and boosted regression trees are employed to identify hazardous traffic conditions and analyze crash data [3,4]. Stochastic Gradient Boosting is particularly effective at handling different types of predictors and missing values, making it suitable for real-time risk assessment, and it can identify crash cases with high accuracy and a low false-positive rate [3]. Boosted regression trees models, on the other hand, are used to investigate the complex, nonlinear relationships in high-variance crash data, providing improved transferability over traditional logistic regression models [4]. Gradient boosting regression trees are also used to analyze and model freeway travel time, offering improved prediction accuracy and model interpretability by strategically correcting mistakes from previous models [5].

In engineering- and physics-based models, a data-driven subspace identification technique, which is a form of ML, can be used to estimate damage to structures by determining damage coefficients that define the reduction of stiffness or damping [6]. The minimization of a residual function helps provide accurate damage coefficient values. AI, a broader field encompassing ML, is being reviewed for its impact on concrete mix design, offering improvements in optimal proportioning, property prediction, and quality control [7]. It helps overcome the limitations of traditional approaches, though challenges remain with regard to data availability and model interpretability.

In climate and weather modeling, ML provides a powerful way to recognize complex patterns and model nonlinear dynamics in climate and weather processes [8]. Researchers have successfully incorporated physics and domain knowledge into ML models to improve emulating, downscaling, and forecasting. An algorithm based on generalized Hidden Markov Models has been used to track the best-performing climate model over time, with its average prediction loss matching that of the best model in hindsight and outperforming traditional methods [9].

There has been a growing consensus that complex scientific and engineering problems require methodologies that integrate traditional physics-based models with state-of-the-art ML techniques [10]. These physics-guided ML models can lead to greater physical consistency, reduced training time, and better generalization. For example, physics-based morphodynamic modeling is used to quantify the impact of climate change on coastal erosion [11].

ML has also become an essential analytical framework to interpret and manage the complexities of contemporary urbanization. As cities generate increasingly large and heterogeneous datasets, from traffic sensors and satellite imagery to housing market transactions and environmental monitors, ML algorithms provide the computational power and flexibility to make sense of this information. They enable researchers to explore the underlying structures and interrelationships among various urban phenomena, such as housing density, transportation networks, and public health [12,13].

The predictive capabilities of ML allow for dynamic modeling of urban systems. These include forecasting urban growth, simulating the impacts of policy interventions, and evaluating sustainability metrics with a data-driven foundation. Machine learning can support the study of spatial segregation and gentrification [14], allowing researchers and policymakers to identify patterns and predict future trends in these areas. For example, by analyzing demographic and socioeconomic data, ML models can pinpoint neighborhoods at risk of gentrification, providing a basis for targeted policy interventions.

ML is instrumental in addressing urban environmental concerns. Its approaches are used to forecast the hourly concentration of air pollutants like ozone, particle matter (PM_{2.5}), and sulfur dioxide [15]. These predictions are vital for public health warnings and for evaluating the effectiveness of air quality regulations. ML models can also monitor urban expansion and detect land use changes, as demonstrated by a study in Cameroon [16].

This capability helps urban planners optimize land allocation and infrastructure development to promote sustainable growth. In a similar vein, an artificial neural network model is used to predict urban growth in five major Greek cities by 2030, offering a tool for local policymakers to analyze future development scenarios across Europe [17].

While supervised ML methods are prevalent, some research also considers the potential of unsupervised methods and algorithm selection paradigms [18]. The integration of various ML approaches can lead to more accurate results and more thorough analyses, especially in areas like image processing and analytics. However, it is important to acknowledge that certain aspects, such as image acquisition and decision-making, still require human supervision [18]. This highlights the ongoing need for an expert-guided approach, ensuring that ML tools are used to augment, rather than replace, human expertise in urban planning and decision-making.

The application of ML to urban planning has recently advanced through the integration of reinforcement learning and graph neural networks. A notable example is an AI model that represents urban spaces as graphs and solves spatial planning as a sequential decision-making problem [19]. This model outperforms human experts in objective evaluations and supports collaborative workflows for planning, demonstrating the profound potential of computational methods to tackle complex urban challenges.

The emergence of mixed-use functional area detection using spatiotemporal data and entropy-based methods [20], as well as the growing use of unsupervised learning (UL) approaches such as clustering and topic modeling [21], further highlight ML's expanding role in urban sciences. UL methods, in particular, allow for the discovery of hidden structures in unlabeled urban datasets, revealing insights into urbanization patterns, built environments, sustainability, and urban dynamics. Despite certain challenges pertaining to data quality, interpretability, and validation, UL provides a promising pathway for understanding and managing urban complexity.

In addition, ML is reshaping the broader urban discourse by contributing to the emergence of new paradigms in city governance and development. Recent studies suggest a shift from "smart urbanism" to "autonomous cities," where AI systems increasingly mediate urban life, raise ethical concerns, and challenge traditional notions of control and transparency in governance [22].

A critical element underpinning the success of predictive ML models in urban sciences is hyperparameter tuning. Hyperparameters are user-defined settings that govern the learning process, such as tree depth in decision trees, learning rates in gradient boosting machines, or the number of clusters in unsupervised algorithms. Unlike model parameters learned from data, hyperparameters must be optimized externally to improve model accuracy, robustness, and generalizability. In high-stakes urban applications where predictions may influence transportation systems, housing policies, or environmental monitoring, well-tuned models can lead to reliable insights and guided decisions.

Systematic hyperparameter optimization (e.g., via Random Search, Grid Search, or advanced techniques like BayesOpt and Optuna) enhances the predictive performance of ML models by identifying the best combination of settings. This is especially important in applications such as sustainable transport modeling [23], where ML models classify freight transport sustainability levels with high accuracy, and real estate valuation [24–27], where ML-based approaches outperform traditional methods in forecasting property prices and identifying investment opportunities.

In summary, the synergy between machine learning and urban sciences is yielding novel frameworks for understanding, modeling, and managing cities. As ML methods become increasingly sophisticated, the role of hyperparameter tuning becomes more central,

not only for achieving high-performance predictions but also for ensuring the interpretability, reliability, and applicability of ML models to real-world urban challenges.

Despite the growing integration of machine learning techniques in urban studies, several methodological gaps persist. Many existing studies continue to rely on traditional hyperparameter tuning methods such as Random Search and Grid Search [28–31]. While these approaches have been widely adopted, they are becoming increasingly outdated and inefficient, particularly when applied to large datasets, due to their exhaustive and time-consuming search strategies. As such, they may hinder computational efficiency and limit the scalability of machine learning applications in urban research contexts.

A critical gap lies in the limited adoption of more advanced, automated optimization frameworks such as Optuna [32]. Built on Bayesian optimization and equipped with pruning techniques, Optuna offers a more intelligent and adaptive search strategy, leading to faster convergence and improved predictive performance. However, its application remains underrepresented in urban analytics, particularly among researchers less familiar with recent developments in machine learning.

Another notable shortcoming in current urban studies is the predominant focus on the test set results, with little to no evaluation of the model's performance with regard to the training data. Several recent real estate studies [33–36] which have employed machine learning algorithms combined with Shapley values to predict property prices reported performance metrics solely based on the test set. While test set evaluation provides insight into the model's predictive accuracy on unseen data, it offers a limited view of the model's overall behavior. Without examining training set metrics, it becomes difficult to assess the model's generalization capability or to identify problems such as overfitting (where the model memorizes training data) or underfitting (where it fails to capture patterns). A more comprehensive evaluation should include a diverse set of metrics for both training and test sets. This dual reporting allows for a clearer understanding of how well the model learns from data, maintains accuracy across different subsets, and ensures its robustness, reliability, and practical applicability in real-world real estate prediction tasks.

These research gaps highlight the critical need for more efficient hyperparameter tuning frameworks and more rigorous model evaluation protocols within the urban research domain. As machine learning becomes increasingly integrated into urban sciences, ranging from real estate to transportation and environmental monitoring, ensuring that models are both accurate and generalizable is essential. Current practices often lack transparency and consistency, particularly in model validation and performance reporting. By adopting systematic tuning approaches and comprehensive evaluation metrics, researchers can better assess model reliability, identify overfitting or underfitting, and ultimately produce more trustworthy and actionable insights to support urban planning, policy development, and decision-making processes.

2. Literature Review

This section explores two machine learning algorithms frequently used in urban problems. We start with Gradient Boosting Machine (GBM), a powerful ensemble learning algorithm that combines multiple weak models to create a strong and accurate predictor. It is effective for complex datasets and can be used for regression, classification, and feature selection. Next, we examine Random Forest (RF), a popular ensemble learning method that combines multiple decision trees to create a robust and accurate model. It is known for handling large datasets and can be used for various tasks, including classification, regression, and clustering.

GBMs have been widely adopted in real-world applications due to their strong predictive capabilities. Using machine learning to detect and prevent Internet of Things (IoT)

attacks has been suggested by [37]. Their study focuses on detecting noise level anomalies in a city suburb using machine learning algorithms, including Gradient Boosting Machine and Deep Learning. Two types of attacks are tested: sudden and gradual changes in noise levels. Their results show that their approach can effectively detect such anomalies, even for small changes in noise levels. This can help cities protect their infrastructure and ensure a safer and more sustainable environment for citizens.

Three machine learning algorithms, including Support Vector Machine (SVM), Random Forest (RF), and Gradient Boosting Machine (GBM), are employed to predict property prices in the new town “Tseung Kwan O” in Hong Kong [38]. This study applies these methods to a dataset of 39,554 housing transactions in Hong Kong during the period between June 1996 and August 2014, and then compared their results. In terms of predictive power, GBM and RF outperform SVM, as demonstrated by their higher values of R^2 and lower values of standard evaluation metrics.

Based on GBM, a non-parametric method has been introduced to measure how connected financial institutions are, and how this affects their risk of failing [39]. This paper studies the connections between banks in China, and finds that banks with similar characteristics tend to be more connected and riskier, especially those heavily involved in interbank activities like industrial banking. The banking system becomes more connected during times of financial stress, and state-owned banks are more likely to be the source of risk during instability.

GBM can effectively impute missing values related to monitoring and reporting greenhouse gas emissions in healthcare facilities [40]. By using larger datasets and GBM, researchers can improve the accuracy of filling missing values, providing a more precise depiction of GHG emissions from healthcare facilities. This could influence policy-making and regulatory practices for monitoring and reporting GHG emissions. In this paper, hyperparameters are optimized by tuning with Grid Search.

RF has seen extensive application in multiple domains, with one notable example being urban planning. Although urban planning is becoming increasingly complex due to the vast and growing volume of available data, the integration of advanced information technologies into the field has advanced at a relatively slow pace. This is particularly noticeable in the recent emergence and implementation of ‘smart city’ technologies within urban environments [41,42]. Despite the rapid development of information and communication technologies (ICTs) in other sectors, their adoption in urban planning practices has lagged behind, falling short of leveraging the full potential these innovations offer. This gap presents both challenges and opportunities for urban planners, policymakers, and researchers to bridge the divide between cutting-edge technologies and their practical application in creating more efficient, sustainable, and livable cities. One promising solution lies in the use of machine learning techniques. For example, a study combines data from various sources and applies a Random Forest classifier alongside other machine learning models to evaluate their performance [43]. The findings indicate that the Random Forest model outperforms others in terms of accuracy, precision, and other evaluation metrics.

To demonstrate how machine learning can be used to deliver more accurate pricing predictions, the real estate market has been taken as an example [44]. Utilizing 24,936 housing transaction records, this paper employs Extra Trees, k -Nearest Neighbors, and RF, followed by hyperparameter tuning by Optuna, to predict property prices and then compares their results with those of a hedonic price model. Their results suggest that these three algorithms markedly outperform the traditional statistical techniques in terms of explanatory power and error minimization.

A study by [45] analyzes survey data from 2016 to 2021 and finds that Islamic banks’ accuracy in survey responses improves with a country’s level of development. The study

also finds a significant reduction in credit portfolio risk due to improved risk management practices, global economic growth, and stricter regulations. Additionally, concerns about terrorism financing and cybersecurity threats have decreased due to strengthened anti-money-laundering regulations and investments in cybersecurity infrastructure and education.

An approach to analyze large amounts of exposome data from long-term cohort studies has been proposed, using machine learning to identify predictors of health and applied to the 30-year Doetinchem Cohort Study [46]. Utilizing Random Forest (RF), their study finds that 9 exposures, from different areas, are most important for predicting health, while 87 exposures have little impact. The approach shows an acceptable ability to distinguish between good and poor health, with an accuracy of 70.7%.

3. Methodology

To ensure a fair comparison, we apply three hyperparameter tuning methods (Random Search, Grid Search, and Optuna) to two distinct machine learning models: GBM and RF. Each model uses its own specific set of hyperparameter spaces, as detailed in Table 1. Experiments are conducted on a dataset that reflects real-world scenarios to ensure the relevance of our findings. We assess the performance of each tuning method based on two primary criteria: computational efficiency and model performance metrics. Figure 1 illustrates the workflow of the methodology adopted in this study.

Table 1. Hyperparameter spaces for GBM and RF.

	GBM	RF
bootstrap		True
criterion	friedman_mse	friedman_mse
learning_rate	0.1	
loss	squared_error	
max_depth	5, 6, ..., 10	5, 6, ..., 10
max_features	sqrt	sqrt
min_impurity_decrease	0.1, 0.2	0.07, ..., 0.1
min_samples_leaf	10, 11, ..., 15	15, 16, ..., 20
min_samples_split	10, 11, ..., 15	10, 11, ..., 15
min_weight_fraction_leaf	0.1, 0.2	0.03, ..., 0.05
n_estimators	550, 560, ..., 600	50, 60, ..., 100
subsample	0.6	

3.1. Materials

Gradient Boosting Machine (GBM), introduced by Friedman [47], is a powerful ensemble learning technique that constructs a predictive model by sequentially combining multiple weak learners, typically decision trees. The fundamental idea behind GBM is to convert weak learners, which perform poorly on their own, into a strong predictive model by iteratively correcting the errors of previous models. During each iteration, the algorithm computes the gradient of the loss function with respect to the model predictions and adjusts the parameters of the new tree to minimize this loss. This process is repeated for a specified number of iterations, often referred to as boosting rounds, which is one of the key hyperparameters of the model.

GBMs are highly effective in handling heterogeneous data types, capturing complex feature interactions, and modeling nonlinear relationships. These properties make them particularly suitable for applications across a wide range of fields, including urban analytics, finance, healthcare, and environmental studies. The flexibility of GBM allows it to adapt to different data structures and learning tasks, making it a preferred choice for predictive modeling in both research and industry.

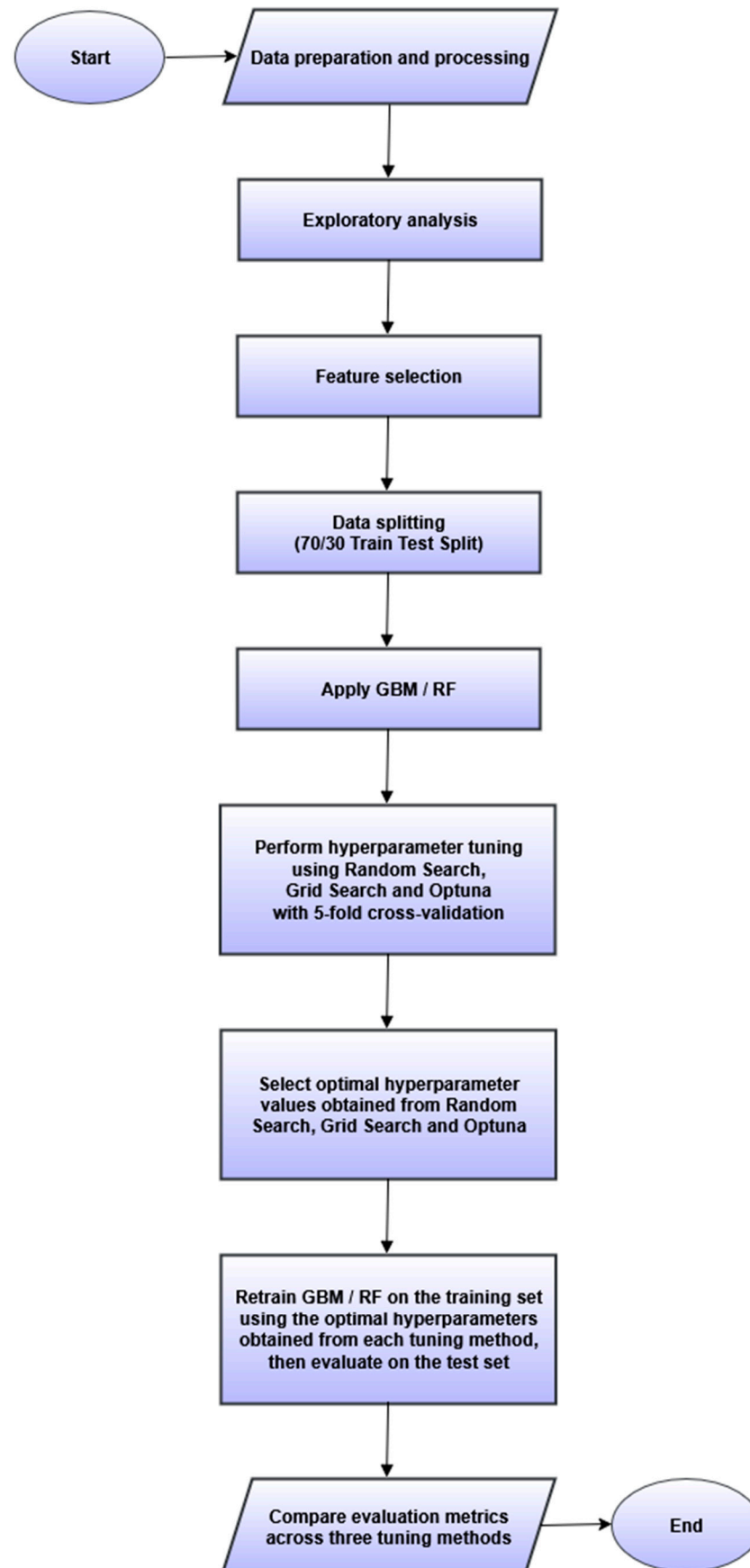


Figure 1. Methodological workflow of the research.

However, the performance of GBMs is heavily dependent on careful hyperparameter tuning. Important hyperparameters include the maximum depth of trees, the learning rate, the number of boosting iterations, and regularization parameters that control model complexity and prevent overfitting. Given the large and high-dimensional hyperparameter space, manual tuning can be labor-intensive and inefficient. Techniques such as Grid Search, Random Search, or automated optimization methods like Optuna are often employed to identify optimal parameter settings. Despite this complexity, when properly tuned, GBMs consistently deliver high predictive accuracy and robustness, which has contributed to their widespread adoption in applied machine learning.

Random Forest (RF), introduced by Breiman in 2001 [48], is a widely used ensemble learning method that combines bootstrap aggregation, or bagging, with decision trees to produce robust predictive models. As a supervised learning algorithm, RF can be applied to both classification and regression tasks. The algorithm works by constructing a large number of decision trees during the training phase, with each tree trained on a randomly sampled subset of the data. The final prediction is obtained by aggregating the outputs of all individual trees, typically through majority voting for classification or averaging for regression. This ensemble approach improves predictive accuracy and stability compared to a single decision tree, reducing variance and mitigating overfitting.

RF offers several practical advantages that contribute to its popularity in applied research. It can effectively handle large datasets with high dimensionality, including cases with numerous irrelevant features, without a significant loss in predictive power. The algorithm also manages missing values naturally, making it suitable for real-world datasets where incomplete information is common. Furthermore, the averaging of predictions across multiple trees enhances generalization and decreases sensitivity to noise in the training data.

Despite its robustness, RF includes a variety of hyperparameters that must be carefully tuned to achieve optimal performance. Key hyperparameters include the maximum depth of trees, the number of trees, and regularization parameters that control model complexity. While tuning RF hyperparameters is generally less time-consuming and complex compared to GBM, careful selection is still important to ensure that the model balances bias and variance effectively. Overall, the combination of accuracy, stability, and flexibility makes RF a highly versatile tool in diverse domains, from urban planning and finance to healthcare and environmental studies.

3.2. Gradient Boosting Machine

For a continuous target variable Y , the GBM model constructs an approximation as a weighted sum of weak learner functions:

$$F_m(x) = \sum_{k=1}^m \rho_k h_k(x) \quad (1)$$

To minimize the average loss function over the training set, we begin with a constant function model, $F_0(x) = \arg \min_{\rho} \sum_{i=1}^n L(y_i, \rho)$, and iteratively update it according to $F_m(x) = F_{m-1}(x) + \rho_m h_m(x)$. Here, $h_m(x)$ represents the base learner function, with h being the optimal function at each step. Since finding the exact function h that minimizes the loss function L is computationally infeasible [46], we use a steepest descent approach. The model is updated as follows:

$$F_m(x) = F_{m-1}(x) + \rho_m g_m(x) \quad (2)$$

where

$$g_m(x) = \left[\frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)} \right]_{i=1}^n \quad (3)$$

and

$$\rho_m = \arg \min_{\rho} \sum_{i=1}^n L(y_i, F_{m-1}(x_i) - \rho g_m(x_i)) \quad (4)$$

While this method provides an approximate solution, GBM estimations remain approximations.

Table 1 documents the hyperparameter spaces for GBM, which are critical for optimizing model performance and generalization in the experiments described. These hyperparameters control the model's complexity, learning behavior, and computational efficiency, directly influencing the accuracy and robustness of predictions for the continuous target variable Y . Hyperparameters for GBM are optimized using various techniques such as Random Search, Grid Search and Optuna to further enhance model accuracy and generalization. Well-tuned hyperparameters significantly improve a model's performance, efficiency, and robustness, enabling better capture of data patterns and more reliable predictions.

The performance of the model is evaluated using the coefficient of determination R^2 , mean absolute error (MAE), median absolute error (MedAE), mean squared error (MSE), mean absolute percentage error (MAPE), symmetric mean absolute percentage error (sMAPE), root mean squared error (RMSE), and root mean squared log error (RMSLE) on the test set, defined as

$$MAE = \frac{\left| \left(h(x^{(i)}) - y^{(i)} \right) \right|}{m} \quad (5)$$

$$MedAE = \text{median} \left(\left| \left(h(x^{(i)}) - y^{(i)} \right) \right| \right) \quad (6)$$

$$MSE = \frac{1}{m} \sum_{i=1}^m \left(h(x^{(i)}) - y^{(i)} \right)^2 \quad (7)$$

$$MAPE = \frac{100\%}{m} \sum_{i=1}^m \left| \frac{\left(h(x^{(i)}) - y^{(i)} \right)}{y^{(i)}} \right| \quad (8)$$

$$sMAPE = \frac{100\%}{m} \sum_{i=1}^m \frac{2 \cdot \left| \left(h(x^{(i)}) - y^{(i)} \right) \right|}{\left| \left(h(x^{(i)}) + y^{(i)} \right) \right|} \quad (9)$$

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m \left(h(x^{(i)}) - y^{(i)} \right)^2} \quad (10)$$

$$RMSLE = \sqrt{\frac{1}{m} \sum_{i=1}^m \left(\log_e \left(h(x^{(i)}) + 1 \right) - \log_e \left(y^{(i)} + 1 \right) \right)^2} \quad (11)$$

where $h(x^{(i)})$ and $y^{(i)}$ represent the predicted value and actual value of the target variable Y , respectively; and m represents the number of observations in the test data.

In this research, we employ MAE, MedAE, MSE, MAPE, sMAPE, RMSE, and RMSLE as evaluation metrics because they are widely accepted and conventional measures in machine learning analysis. Each metric offers a different perspective on model performance. MAE provides a straightforward interpretation of average prediction error, while MedAE is the median of the absolute differences between predicted and actual values. Unlike MAE, MedAE is more robust to outliers because it uses the median instead of the mean. MSE penalizes larger errors more heavily, making it sensitive to outliers. MAPE measures relative error, allowing for intuitive comparisons across different scales. sMAPE is a percentage-

based error metric that calculates the absolute difference between predicted and actual values, divided by the average of the two. It is “symmetric” because it treats over- and under-predictions equally, avoiding bias from large actual values (unlike MAPE). RMSE expresses error in the same units as the target variable, which aids interpretability. RMSLE measures the square root of the average squared differences between the log-transformed predicted and actual values. It penalizes under-predictions more than over-predictions, and is less sensitive to large differences in absolute values.

Using this set of complementary metrics ensures a more comprehensive evaluation of model accuracy and robustness. These metrics are standard in both academic literature and practical machine learning applications, enabling comparisons with other studies and supporting the credibility and reproducibility of the findings.

3.3. Random Forest

Random Forest is a supervised learning algorithm that employs an ensemble approach for both classification and regression. It builds a specified number of decision trees and aggregates their predictions to create a more accurate and robust model than a single tree. The algorithm uses random sampling with replacement, a technique known as “bagging” [49,50], to reduce variance. Given a training set with features X and outputs Y , bagging iteratively selects random samples from the training set K times and fits trees to these samples. For each tree, a random sample of instances is drawn with replacement, corresponding to a unique random vector \varnothing_k which shapes a specific tree. This variability introduces slight differences among the trees. The prediction of the k -th tree for input X is given by $f_x(x, \varnothing_k)$. At each node during tree splitting, features are randomly selected to minimize correlations. A node S is split into subsets S_1 and S_2 using a threshold c that minimizes the sum of squared errors:

$$SSE = \arg \min_{\theta} \left(\sum_{x_i \in S_1} (y_i - c_1)^2 + \sum_{x_i \in S_2} (y_i - c_2)^2 \right) \quad (12)$$

The final prediction is the average of all tree outputs:

$$\hat{f}(x) = \frac{1}{K} \sum_{k=1}^K f_k(x, \varnothing_k) \quad (13)$$

Table 1 also documents the hyperparameter spaces for RF, which are critical for optimizing model performance and generalization in the experiments described. Hyperparameters are optimized using techniques such as Random Search, Grid Search, and Optuna to enhance model performance and generalization. Finally, model performance is evaluated using the coefficient of determination R^2 , MAE, MedAE, MSE, MAPE, sMAPE, RMSE, and RMSLE on the test set.

3.4. Hyperparameter Tuning

Hyperparameter tuning is the process of finding the best combination of hyperparameters for a machine learning model. It plays a crucial role in maximizing performance and generalization while avoiding overfitting. However, the operation of hyperparameter tuning can be computationally expensive, especially for complex models with plenty of hyperparameters.

Several techniques have been proposed for hyperparameter tuning, including Random Search, Grid Search, Optuna, and more. Traditional methods like Random Search and Grid Search often struggle to efficiently explore the high-dimensional hyperparameter space, leading to suboptimal results or excessive computational costs. Random Search involves

randomly sampling hyperparameters from specified distributions. Despite its simplicity, Random Search often surpasses Grid Search in terms of efficiency as it explores the search space more diversely [51]. However, there is no guarantee of finding the global optimum, and the quality of the solution heavily relies on the number of random samples taken.

In contrast, Grid Search is a method that involves exhaustively evaluating preset hyperparameter values to determine the optimal combination. Although Grid Search ensures the discovery of the global optimum within the search space, it can incur high computational costs, particularly when dealing with numerous hyperparameters or continuous parameters requiring precise resolutions.

Optuna represents a recent advancement in hyperparameter tuning techniques, introducing a more streamlined and automated approach centered around Bayesian optimization and tree-structured Parzen estimators (TPEs). Diverging from the conventional Random Search and Grid Search methods, Optuna builds probabilistic models to capture the link between hyperparameters and the objective function, enabling informed decisions regarding which hyperparameters to investigate next.

Through the application of Bayesian optimization principles, Optuna intelligently navigates the hyperparameter space by emphasizing promising areas while disregarding less fruitful ones. This adaptive sampling methodology leads to notable reductions in computational expenses and faster convergence towards the optimal solution compared to traditional approaches. A key strength of Optuna lies in its lightweight and versatile design, facilitating seamless integration with a variety of machine learning libraries and frameworks. Its Pythonic approach to defining search spaces and simple parallelization capabilities further support its user-friendliness and scalability [52].

4. Data Definitions and Sources

In this study, we focus on the analysis of a specific private housing estate, Ocean Shores, located in the Tseung Kwan O district of Hong Kong. This estate is officially designated as one of the “selected popular residential developments” by the Rating and Valuation Department of the Hong Kong SAR Government. Ocean Shores comprises 15 residential blocks and a total of 5,726 domestic units. Our dataset spans from January 2010 to December 2020, encompassing 7,652 intertemporal and cross-sectional observations.

The dataset includes detailed information on individual buildings, locations, transaction dates, occupation permits, sale prices, square footage, and other property characteristics (e.g., the inclusion of a parking space). These records are maintained by the government and compiled by a commercial data provider known as EPRC. All property prices are inflation-adjusted using the private housing estate price index published by the Rating and Valuation Department.

To ensure data quality, we exclude transaction records with incorrect transaction dates or missing values for key variables such as sale price or gross floor area. Furthermore, to enhance the robustness of the analysis and reduce the impact of outliers and potential data errors, we apply a trimming approach, one of the standard methods to handle outliers, by removing the bottom and top 5% of observations based on transaction prices.

where

P_i^t represents the total transaction price of residential property i during time period t , measured in HK dollars and inflation-adjusted.

NFA_i^t represents the net floor area of residential property i .

AGE_i^t represents the age of residential property i in years during time period t , which can be obtained by determining the difference between the date of issue of the occupation permit and the date of housing sales.

FL_i^t represents the floor level of residential property i .

$E_i^t, S_i^t, W_i^t, N_i^t, NE_i^t, SE_i^t, SW_i^t$ & NW_i^t represent eight orientations that residential property i is facing. They are assigned to be 1 if a property is facing a particular orientation, 0 otherwise. The omitted category is Northwest so that coefficients may be interpreted relative to this category.

In this study, our primary objective is not to predict property prices per se but to systematically evaluate and compare the performance of different hyperparameter tuning methods, namely Random Search, Grid Search, and Optuna, using machine learning algorithms GBM and RF. We employ a relatively small dataset comprising 7,652 housing transaction records from the property market solely as a practical case study to demonstrate and illustrate the differences in computational efficiency and predictive performance among these tuning strategies. While the dataset provides a concrete and relatable context, it is important to note that the methodology and insights derived here are broadly applicable to any dataset or domain where hyperparameter tuning is necessary.

Despite its modest size, the dataset still imposes significant computational demands, particularly for exhaustive methods such as Grid Search. For example, tuning a GBM model with about 7,652 rows and 10 features using Grid Search can take nearly two hours, and the required time increases substantially with larger datasets or more complex models. This highlights the importance of efficient hyperparameter optimization techniques in real-world scenarios, where computational resources and time are often limited.

We have selected GBM and RF as representative algorithms due to their widespread adoption across various fields and their distinct computational characteristics. This choice enables a meaningful evaluation of the trade-offs between tuning efficiency and model accuracy. Our approach underscores the practical challenges researchers and practitioners face in selecting hyperparameter optimization methods, even when working with datasets of moderate size.

By focusing on GBM and RF and using the property market as a familiar example, this study provides valuable insights into optimizing model selection processes that are transferable across domains. The findings can guide practitioners in balancing the competing demands of computational speed and predictive performance, making them relevant for a wide range of applications beyond real estate analytics.

Exploratory Data Analysis

Exploratory statistics is a field that focuses on exploring and summarizing datasets to gain insights into patterns, trends, and characteristics before modeling or hypothesis testing. By using graphical and numerical methods, researchers can conduct exploratory data analysis to examine key characteristics. Figures 2–4 show histograms, correlations, and a correlation matrix for property prices and the selected features. The results reveal a high correlation between footage area and property prices at 0.8, as well as moderate correlations between southeast at 0.4 and property age at -0.4 , and floor level and south at 0.2. Table 2 provides a summary of descriptive statistics for the features analyzed in this study, offering a concise overview of the dataset.

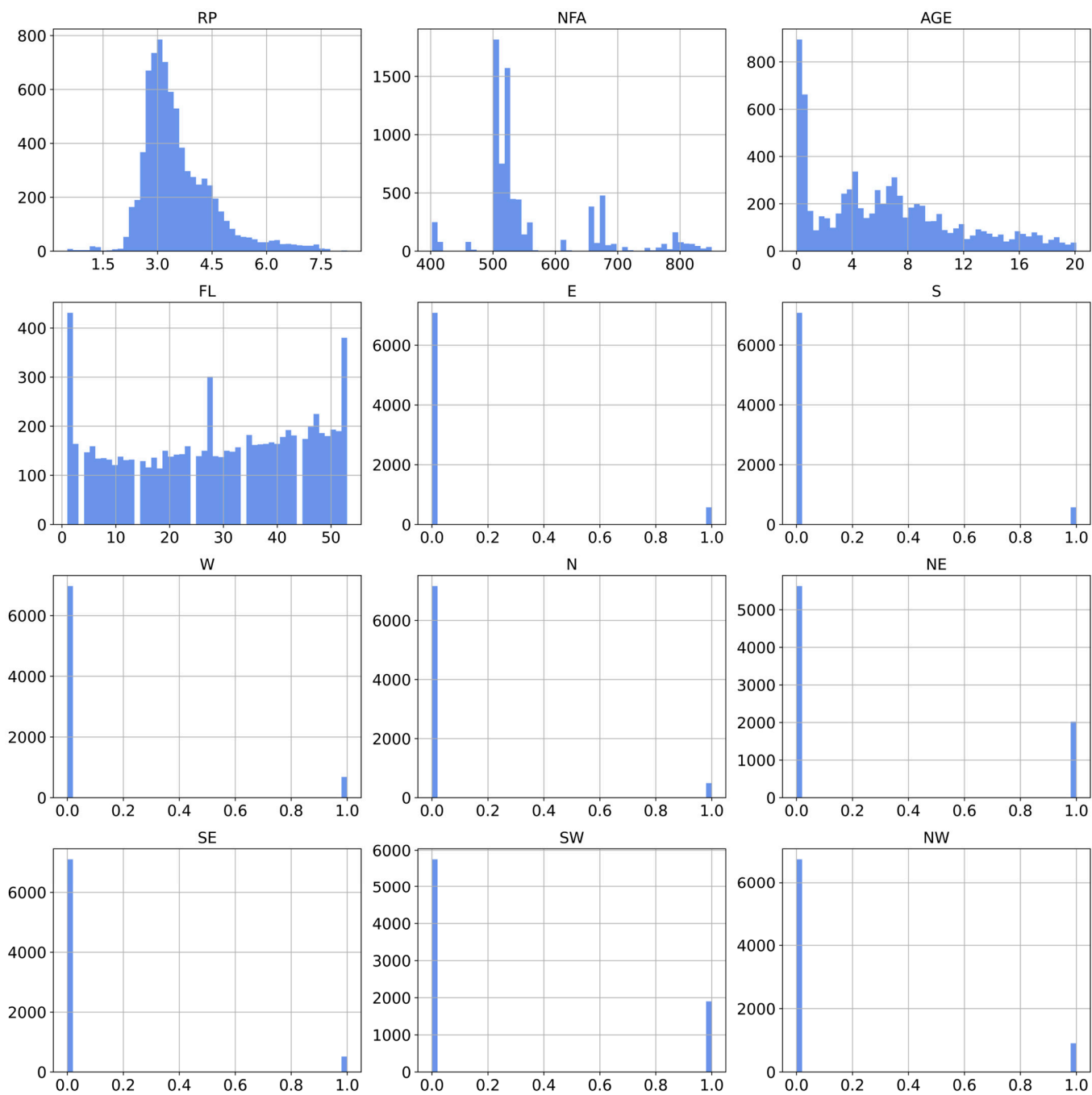


Figure 2. Distribution of property prices and key features.

Table 2. Descriptive statistics.

	RP	GFA	AGE	FL	E	S	W	N	NE	SE	SW	NW
Count	7,652	7,652	7,652	7,652	7,652	7,652	7,652	7,652	7,652	7,652	7,652	7,652
Mean	3.6547	560.2145	6.4579	28.6431	0.0746	0.0748	0.0890	0.0640	0.2641	0.0673	0.2480	0.1181
Std	0.9935	94.3491	5.1822	15.9485	0.2628	0.2630	0.2848	0.2448	0.4409	0.2506	0.4319	0.3228
Min	0.5187	402	0.0027	1	0	0	0	0	0	0	0	0
25%	2.9184	506	1.8658	15	0	0	0	0	0	0	0	0
50%	3.3113	552	5.8904	30	0	0	0	0	0	0	0	0
75%	4.0134	559	9.5014	42	0	0	0	0	0	0	0	0
Max	8.2193	851	20.0959	53	1	1	1	1	1	1	1	1
Skew	1.3376	1.3530	0.6693	-0.1753	3.2382	3.2346	2.8875	3.5623	1.0703	3.4547	1.1671	2.3666

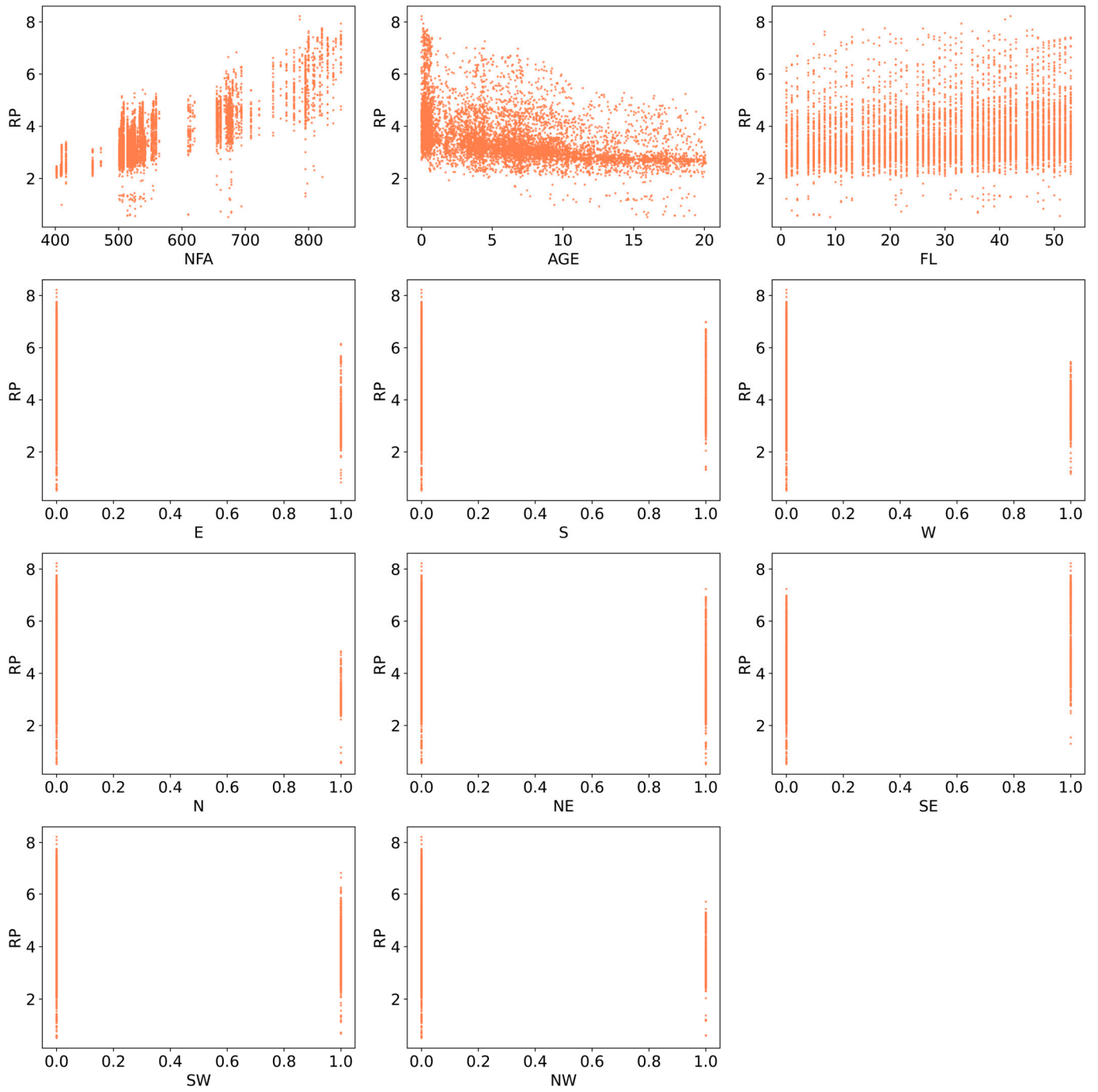


Figure 3. Data visualization of property prices and key features.

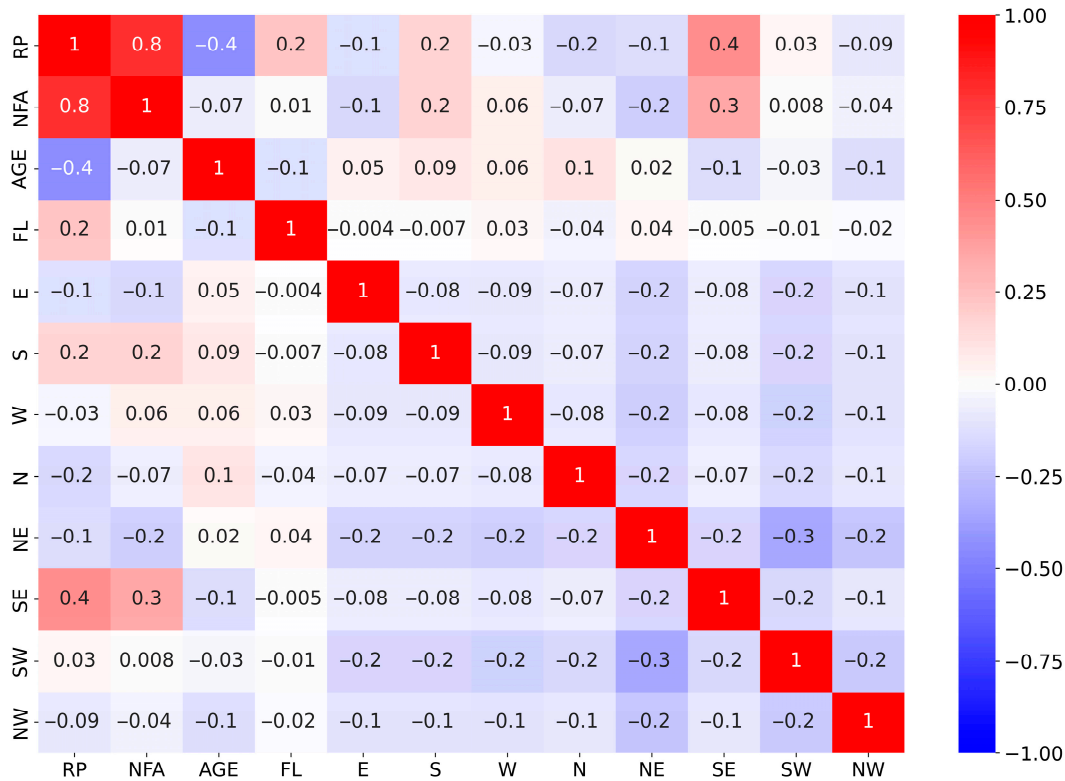


Figure 4. Correlation matrix of property prices and key features.

5. Results and Discussions

In this study, the dataset is first split into 70% for training and 30% for testing. Hyperparameter tuning is performed on the training set using five-fold cross-validation to identify the optimal parameter values. In each iteration, 80% of the training data (equivalent to 56% of the entire dataset) is used for model training, while the remaining 20% (14% of the full dataset) serves as the internal validation fold. Each fold is used once as the validation fold to ensure robust and unbiased evaluation. Each hyperparameter tuning method will be employed to optimize the hyperparameters by maximizing the average performance across the five folds, thereby enhancing the overall predictive accuracy of the model. After selecting the optimal hyperparameters, the model is retrained on the training set (70%), and its predictive performance is assessed on the completely unseen test set (30%), providing a robust estimate of real-world performance.

Before presenting the optimized model results, it is useful to first compare the performance of the untuned GBM and RF models to highlight key differences in their predictive behavior. As shown in Table 3, the untuned GBM achieves a very high training R^2 of 0.98732, but its test R^2 decreases to 0.91444, indicating overfitting. This is further evidenced by the substantial percentage increases between the training and the test sets for MAE (170.36%), MedAE (165.43%), MSE (586.48%), and RMSE (162.01%). By contrast, the RF model exhibits a more balanced performance, with a training R^2 of 0.91645 and test R^2 of 0.89652, reflecting only a modest decline of 2.18%. Nevertheless, its error metrics (MAE, MSE, and RMSE) still point to signs of overfitting. With respect to relative percentage errors (MAPE and sMAPE), both untuned models perform similarly, though GBM attains slightly lower test values. Overall, both models show overfitting in their untuned forms.

Table 3. Performance metrics of untuned GBM and RF models.

	GBM			RF		
	Training Set ^a	Test Set ^b	$(b - a)/a \times 100$	Training Set ^c	Test Set ^d	$(d - c)/c \times 100$
R ²	0.98732	0.91444	−7.38164%	0.91645	0.89652	−2.17557%
MAE	0.06290	0.17007	170.36426%	0.18404	0.20285	10.22216%
MedAE	0.03932	0.10437	165.42968%	0.12595	0.13542	7.51954%
MSE	0.01245	0.08547	586.47581%	0.08200	0.10336	26.05310%
MAPE	2.10411%	5.93201%	3.82791%	5.98480%	6.91522%	0.93041%
sMAPE	1.96253%	5.13475%	3.17222%	5.36070%	5.94096%	0.58026%
RMSE	0.11158	0.29235	162.00683%	0.28636	0.32150	12.27337%
RMSLE	0.03207	0.07767	142.24336%	0.00000	0.08354	Nan
Model accuracy (Average R ²)	0.89939	0.89039	−1.00137%	0.90928	0.89197	−1.90376%

5.1. Optimized Gradient Boosting Machine

Table 4 compares the hyperparameter selections made by three tuning methods after estimating the GBM. The hyperparameters criterion, learning_rate, loss, max_features, and subsample are fixed, while the tuning methods independently select optimal values for max_depth, min_impurity_decrease, min_samples_leaf, min_samples_split, min_weight_fraction_leaf and n_estimators. The results indicate that the three methods do not converge on the same optimal hyperparameters. In particular, Optuna selects a max_depth of eight while the other two tuning methods select a value of six. Optuna has a consensus with Random Search and Grid Search that they all select a min_impurity_decrease of 0.1, min_weight_fraction_leaf of 0.1 and n_estimators of 600. Optuna selects min_samples_leaf of 13, while Random Search and Grid Search select values of 12 and 10, respectively. Optuna selects a min_samples_split of 12 while Random Search and Grid Search select values of 15 and 10, respectively.

Table 4. Optimized hyperparameters for GBM and RF.

	GBM			RF		
	Random Search	Grid Search	Optuna	Random Search	Grid Search	Optuna
bootstrap				True	True	True
criterion	friedman_mse	friedman_mse	friedman_mse	friedman_mse	friedman_mse	friedman_mse
learning_rate	0.1	0.1	0.1			
loss	squared_error	squared_error	squared_error			
max_depth	6	6	8	8	8	9
max_features	sqrt	sqrt	sqrt	sqrt	sqrt	sqrt
min_impurity_decrease	0.1	0.1	0.1	0.1	0.1	0.08
min_samples_leaf	12	10	13	18	15	17
min_samples_split	15	10	12	15	10	12
min_weight_fraction_leaf	0.1	0.1	0.1	0.03	0.03	0.03
n_estimators	600	600	600	90	90	50
subsample	0.6	0.6	0.6			

Table 5 presents the results of using GBM optimized by each method. Our findings indicate that Optuna outperforms Random Search and Grid Search in terms of computational efficiency and model performance. Optuna accomplishes the hyperparameter optimization process in a duration of approximately 3.10 min, thereby demonstrating a remarkable acceleration of 6.77 times compared to Random Search (21.02 min) and an impressive acceleration of 34.50 times compared to Grid Search (107.08 min). This substantial reduction in computation time is particularly valuable in situations where resources are limited or time-sensitive applications are involved.

Table 5. Performance metrics of hyperparameter-optimized GBM and RF models.

	GBM			RF		
	Random Search	Grid Search	Optuna	Random Search	Grid Search	Optuna
R ²	0.86335 (0.85433) {−1.04438%}	0.86335 (0.85433) {−1.04438%}	0.86387 (0.85490) {−1.03858%}	0.79767 (0.78402) {−1.71011%}	0.79767 (0.78402) {−1.71011%}	0.79956 (0.78762) {−1.49321%}
MAE	0.24917 (0.25988) {4.29897%}	0.24917 (0.25988) {4.29897%}	0.24826 (0.25896) {4.30763%}	0.28991 (0.30138) {3.95743%}	0.28991 (0.30138) {3.95743%}	0.28770 (0.29875) {3.84135%}
MedAE	0.17997 (0.18300) {1.68537%}	0.17997 (0.18300) {1.68537%}	0.17962 (0.17898) {−0.35361%}	0.19038 (0.19344) {1.60476%}	0.19038 (0.19344) {1.60476%}	0.18975 (0.19540) {2.97871%}
MSE	0.13412 (0.14550) {8.48149%}	0.13412 (0.14550) {8.48149%}	0.13361 (0.14493) {8.47390%}	0.19859 (0.21573) {8.62736%}	0.19859 (0.21573) {8.62736%}	0.19673 (0.21213) {7.82822%}
MAPE	7.80118% (8.43231%) {0.63113%}	7.80118% (8.43231%) {0.63113%}	7.77224% (8.39544%) {0.62320%}	8.92081% (9.65347%) {0.73266%}	8.92081% (9.65347%) {0.73266%}	8.85157% (9.57079%) {0.71922%}
sMAPE	7.10543% (7.45360%) {0.34817%}	7.10543% (7.45360%) {0.34817%}	7.07753% (7.42096%) {0.34343%}	8.03486% (8.40656%) {0.37170%}	8.03486% (8.40656%) {0.37170%}	7.96494% (8.33739%) {0.37244%}
RMSE	0.36623 (0.38144) {4.15445%}	0.36623 (0.38144) {4.15445%}	0.36553 (0.38070) {4.15080%}	0.44564 (0.46446) {4.22445%}	0.44564 (0.46446) {4.22445%}	0.44355 (0.46058) {3.84037%}
RMSLE	0.08537 (0.09213) {7.91348%}	0.08537 (0.09213) {7.91348%}	0.08521 (0.09191) {7.86360%}	0.09790 (0.10573) {7.99380%}	0.09790 (0.10573) {7.99380%}	0.09749 (0.10499) {7.68494%}
5-fold CV						
1st fold R ²	0.84992 (0.84629) {−0.42640%}	0.84992 (0.84629) {−0.42640%}	0.85163 (0.84549) {−0.72133%}	0.80012 (0.76412) {−4.49867%}	0.80012 (0.76412) {−4.49867%}	0.78783 (0.76686) {−2.66177%}
2nd fold R ²	0.86898 (0.84802) {−2.41292%}	0.86898 (0.84802) {−2.41292%}	0.86906 (0.84937) {−2.26511%}	0.79991 (0.80333) {0.42709%}	0.79991 (0.80333) {0.42709%}	0.80027 (0.79964) {−0.07818%}
3rd fold R ²	0.85138 (0.84424) {−0.83828%}	0.85138 (0.84424) {−0.83828%}	0.85149 (0.84477) {−0.78891%}	0.77154 (0.78513) {1.76137%}	0.77154 (0.78513) {1.76137%}	0.76622 (0.79289) {3.48070%}
4th fold R ²	0.85408 (0.85411) {0.00294%}	0.85408 (0.85411) {0.00294%}	0.85013 (0.85472) {0.18638%}	0.80018 (0.78538) {−1.85010%}	0.80018 (0.78538) {−1.85010%}	0.78037 (0.78455) {0.53530%}
5th fold R ²	0.84097 (0.83122) {−1.15983%}	0.84097 (0.83122) {−1.15983%}	0.83911 (0.82962) {−1.13108%}	0.78252 (0.74013) {−5.41733%}	0.78252 (0.74013) {−5.41733%}	0.78374 (0.73947) {−5.64860%}
Model accuracy (Average R ²)	0.85307 (0.84478) {−0.97196%}	0.85307 (0.84478) {−0.97196%}	0.85288 (0.84479) {−0.94836%}	0.79086 (0.77562) {−1.92664%}	0.79086 (0.77562) {−1.92664%}	0.78369 (0.77668) {−0.89371%}
Average Std of R ²	+/− 0.01819 (+/− 0.01507)	+/− 0.01819 (+/− 0.01507)	+/− 0.01908 (+/− 0.01674)	+/− 0.02362 (+/− 0.04331)	+/− 0.02362 (+/− 0.04331)	+/− 0.02206 (+/− 0.04320)
Computational Speed (seconds)	1,261.07	6,425.08	186.22	344.39	5,176.05	47.52

Notes: Figures indicate the values for the training set; figures in brackets indicate the values for the test set; figures in curly parentheses indicate the difference in values between the training and the test sets. Our estimations are conducted on the following computing environment: Operating System—Microsoft Windows 11 Home (Version 10.0.26100, Build 26100); Processor—Intel Core i7-10510U CPU @ 1.80GHz with 4 cores and 8 threads; RAM—16 GB; System Type—64-bit, x64-based architecture; Machine Model—Dell Inspiron 7790 All-in-One (AIO).

Furthermore, GBM models optimized using Optuna exhibit superior performance metrics compared to those optimized using Random Search and Grid Search. Evaluation metrics such as MAE, MedAE, MSE, MAPE, sMAPE, RMSE, and RMSLE consistently

display lower values for models optimized by Optuna on the test set, suggesting that Optuna effectively traverses the hyperparameter space to identify optimal configurations that lead to models with enhanced predictive accuracy and generalization capabilities (see Table 5).

The acceptable threshold for overfitting varies depending on the application and dataset. As a general guideline, a difference of less than 5% between the training and test performance metrics is considered negligible, while differences greater than 10% may suggest overfitting. In this study, the observed differences across the three hyperparameter tuning methods for GBM range from 0.34% to 8.48%, indicating only minimal overfitting and supporting the conclusion that the models are well optimized for generalization. The consistently low error metrics on both the training and the test sets, together with high R^2 values, further confirm the absence of underfitting, where a model fails to capture essential data patterns. Taken together, the integration of hyperparameter tuning, cross-validation, and comprehensive performance evaluation has yielded an optimized GBM model that demonstrates strong predictive accuracy and robust generalization across all three tuning methods.

5.2. Optimized Random Forest

Table 4 also presents a comparison of the hyperparameter selections made by three different tuning methods after estimating a Random Forest (RF) model. In our experimental setup, the hyperparameters bootstrap, criterion, and max_features are fixed, while each tuning method independently determines optimal values for max_depth, min_impurity_decrease, min_samples_leaf, min_samples_split, min_weight_fraction_leaf, and n_estimators. Interestingly, Optuna does not select the same value for all hyperparameters when compared to the other two methods. Random Search and Grid Search select the same value for max_depth of 8, min_impurity_decrease of 0.1, min_weight_fraction_leaf of 0.03, and n_estimators of 90. For the optimal value of min_samples_leaf and min_samples_split, Random Search selects 18 and 15 while Grid Search selects 15 and 10, respectively.

In Table 5, our results based on RF indicate that Optuna surpasses both Random Search and Grid Search in terms of computational efficiency and model performance metrics. Notably, Optuna exhibits significantly faster computation speed, completing the hyperparameter optimization process in approximately 0.79 min, a reduction of 7.25 times compared to Random Search (5.74 min) and 108.92 times compared to Grid Search (86.27 min).

RF optimized using Optuna also displays superior performance metrics compared to those optimized using Random Search and Grid Search. Evaluation metrics such as MAE, MedAE, MSE, MAPE, sMAPE, RMSE, and RMSLE consistently show lower values for Optuna-optimized models based on the test set, suggesting that Optuna effectively traverses the hyperparameter space to identify optimal configurations that lead to models with enhanced predictive accuracy and generalization capabilities.

In this study, the observed differences across the three hyperparameter tuning methods for RF range from 0.37% to 8.63%, indicating only minimal overfitting and supporting the conclusion that the models are well optimized for generalization. The consistently low error metrics on both the training and the test sets, together with high R^2 values, further confirm the absence of underfitting. Taken together, the integration of hyperparameter tuning, cross-validation, and comprehensive performance evaluation has yielded optimized RF model that demonstrates strong predictive accuracy and robust generalization across all three tuning methods.

5.3. Five-Fold Cross-Validation

Table 5 also presents the fold-wise cross-validation results for GBM and RF under different hyperparameter optimization strategies. For GBM, test R^2 values ranged from 0.831 to 0.855, exhibiting modest variability across folds. In contrast, RF showed slightly higher variability, with larger fold-to-fold differences. Despite these fluctuations, the mean performance of each tuning strategy is consistent across models (GBM: $R^2 \approx 0.845$; RF: $R^2 \approx 0.776$), indicating that the relative ranking of Random Search, Grid Search, and Optuna is robust. Comparison of training and test scores shows minimal differences, with the average gap between train and test R^2 ranging from approximately -0.95% to -0.97% for GBM and from -0.89% to -1.93% for RF. These small differences indicate no signs of overfitting and suggest good model generalization across all three tuning methods. The reported standard deviations underscore the sensitivity of cross-validation to fold splits. Across the three tuning methods, the average standard deviation of test R^2 over the five folds is 0.015 to 0.016 for GBM and 0.043 for RF, further highlighting that RF predictions are slightly more sensitive to training-validation partitioning than GBM. Overall, these results emphasize the stability and generalizability of the models across folds and tuning strategies.

5.4. Discussions

The findings of this study extend beyond technical improvements in machine learning workflows and have direct implications for urban planning and policy-making. Predictive models are increasingly used to support urban scholarship and decision-making in areas such as housing affordability, gentrification dynamics, property valuation, land-use change, and transportation demand. In these contexts, efficiency and accuracy are critical, as models are often updated frequently with new data and must produce timely, reliable insights.

By empirically evaluating the efficiency and predictive accuracy of different hyperparameter tuning methods using a real-world housing dataset, our study demonstrates how more advanced approaches such as Optuna can significantly reduce computation time while enhancing model performance. Although hyperparameter tuning is a standard procedure in machine learning, its computational demands can become a critical bottleneck, particularly in urban research where datasets are increasingly large, complex, and high-dimensional. This issue is rarely addressed explicitly in the urban science literature. Our study fills this gap by illustrating how the choice of tuning method affects both the speed and accuracy of predictive modeling tasks.

The implications for urban researchers and policymakers are substantial. Our findings show that Optuna dramatically reduces computation time compared to Random Search and Grid Search (e.g., 186.22 s versus 1,261.07 and 6,245.08 s, respectively, for GBM and 47.52 s versus 344.39 and 5,176.05 s for RF), while consistently achieving lower error values and higher Average R^2 . Efficient tuning methods such as Optuna therefore allow for more timely model deployment, enabling decision-makers to act on insights without undue computational delays. They also make it feasible to analyze more extensive or more granular datasets without requiring high-end computing resources.

For example, housing market models tuned with Optuna could provide more reliable short-term forecasts of property prices, which in turn could inform affordability policies or investment strategies. Similarly, models of urban growth or land-use change optimized with Optuna could improve the design of zoning policies or infrastructure planning. By demonstrating that Optuna outperforms traditional hyperparameter tuning methods, our study highlights how methodological advances in machine learning can translate into more robust, efficient, and actionable tools for evidence-based urban planning and policy making.

Table 6 summarizes the main advantages and limitations of the machine learning algorithms (GBM, RF) and hyperparameter optimization methods (Random Search, Grid

Search and Optuna) considered in this study. While GBM and RF differ in flexibility, interpretability, and computational cost, the choice of hyperparameter tuning strategy strongly influences model performance and efficiency. The comparison highlights the trade-offs between simplicity, accuracy, and scalability that should guide method selection in practical applications.

Table 6. Pros and cons of GBM, RF, Random Search, Grid Search, and Optuna.

	Pros	Cons
GBM	<ol style="list-style-type: none"> 1. Performs both classification and regression tasks. 2. Capable of handling different data types and loss functions. 3. Works well with missing data and categorical features. 4. Provides several hyperparameter tuning options which make the function fit flexible. 5. Manages high-dimensional data and multicollinearity well. 6. Provides highly accurate predictions due to sequential error correction. 7. Provides estimates of feature importance. 	<ol style="list-style-type: none"> 1. Training can be slow, especially with large datasets and many trees. 2. Sensitive to hyperparameter settings and overfitting. 3. Sensitive to outliers. 4. Requires careful tuning of multiple hyperparameters. 5. The ensemble of many trees makes the model difficult to interpret.
RF	<ol style="list-style-type: none"> 1. Performs both classification and regression tasks. 2. Works well with missing data and categorical features. 3. Manages high-dimensional data and multicollinearity well. 4. Requires less hyperparameter tuning. 5. Naturally resistant to overfitting due to ensemble averaging. 6. Provides accurate predictions. 7. Provides estimates of feature importance. 	<ol style="list-style-type: none"> 1. Overfits with noisy classification and regression. 2. The ensemble of many trees makes the model difficult to interpret.
Random Search	<ol style="list-style-type: none"> 1. More computationally efficient than Grid Search, especially when the number of hyperparameters is large. 2. Can discover good combinations faster. 3. Easy to implement and parallelize. 4. Reduces computation by evaluating fewer points. 	<ol style="list-style-type: none"> 1. Does not guarantee finding the optimal solution. 2. May miss optimal regions if not enough samples are drawn. 3. May spend time in unpromising regions of the search space.
Grid Search	<ol style="list-style-type: none"> 1. Guarantees finding the optimal combination of hyperparameters within the defined grid. 2. Simple and easy to implement. 3. Works well when the number of hyperparameters and their ranges are small. 	<ol style="list-style-type: none"> 1. Very computationally expensive, especially with many hyperparameters. 2. Does not scale well with large search spaces. 3. Wastes time evaluating every combination, even in regions of the search space that are clearly suboptimal.
Optuna	<ol style="list-style-type: none"> 1. Learns from past trials to intelligently sample new hyperparameter combinations, focusing on promising regions. 2. More efficient than Random Search and Grid Search. 3. Automatically prunes unpromising trials early. 4. Supports conditional parameters and complex search spaces. 5. Easily integrates with many ML frameworks. 	<ol style="list-style-type: none"> 1. More complex to set up than Random Search and Grid Search, especially for those new to Bayesian optimization. 2. May require multiple runs to stabilize results due to randomness. 3. Pruning and sampling may not always be beneficial for small datasets. 4. Does not guarantee finding the global optimum.

6. Conclusions

This study compares three hyperparameter optimization strategies—Random Search, Grid Search, and Optuna—using a housing estate dataset as a case study. The results highlight the advantages of Optuna in achieving higher predictive accuracy while reducing computational time, underscoring its potential for advancing methodological rigor in applied urban research. By facilitating more efficient model development, Optuna enables researchers to uncover complex patterns in urban data and generate insights that can inform policy and planning decisions in areas such as housing markets, traffic management, and land use. The adoption of such advanced optimization frameworks can therefore enhance the practical relevance of machine learning in urban sciences, where timely and data-driven decision-making is critical.

The potential applications of Optuna extend beyond housing data to a wide range of urban challenges. Its efficiency and scalability can support the development of cost-effective, data-driven approaches to urban planning, governance, and resource allocation. At the same time, its limitations should not be overlooked. Optuna's performance is influenced by model complexity, dataset characteristics, and search space configuration. In certain cases, alternative frameworks such as HyperOpt, BayesOpt, or Ray Tune may prove equally effective or even superior. Users must also be mindful that Optuna's default pruning strategies and parameter settings are not universally optimal, requiring careful adjustment to specific tasks.

This study has two limitations that should also be acknowledged. First, the analysis relies on a single housing estate dataset. While this dataset provides a concrete and well-defined context, its use inevitably restricts the external validity of the findings. Here, the housing market data serves primarily as a case study to demonstrate and compare the performance of different hyperparameter optimization strategies: Random Search, Grid Search, and Optuna. Its purpose is not to develop a universally generalizable housing price prediction model. Consequently, the empirical results should be interpreted as illustrative of methodological differences rather than definitive for all housing markets. Nevertheless, the comparative insights are transferable to a wide range of urban data science problems where predictive modeling and hyperparameter tuning are central. Second, the computational efficiency results are dependent on the specific experimental environment, including hardware and software configurations. Different platforms may yield different runtimes, although relative comparisons across optimization methods are expected to remain informative.

Future research should build upon this study by conducting comparative evaluations of Optuna and other state-of-the-art frameworks, including HyperOpt, BayesOpt, and Ray Tune, across multiple algorithms, datasets, and tasks. Investigations into hybrid approaches that integrate the strengths of different optimizers also hold promise for improving robustness and adaptability. In addition, further work should examine the interpretability, reproducibility, and usability of these tools in applied urban sciences. By addressing these avenues, future studies can provide deeper guidance on selecting and deploying hyperparameter optimization strategies to strengthen both methodological foundations and real-world impact in urban analytics.

Author Contributions: Conceptualization, W.K.O.H.; methodology, T.K. and W.K.O.H.; software, W.K.O.H.; validation, W.K.O.H.; formal analysis, W.K.O.H.; investigation, T.K. and W.K.O.H.; resources, T.K.; data curation, W.K.O.H.; writing—original draft preparation, T.K. and W.K.O.H.; writing—review and editing, T.K. and W.K.O.H.; visualization, W.K.O.H.; funding acquisition, T.K. and W.K.O.H. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by funding from the Department of Building and Real Estate, The Hong Kong Polytechnic University (Project ID: P0049970, Department Incentive Fund).

Data Availability Statement: The data supporting the study's findings are available in OSF at DOI 10.17605/OSF.IO/5SDCN.

Acknowledgments: The authors would like to thank the editor and the anonymous reviewers for their feedback and insightful comments on the original submission. All errors and omissions remain the responsibility of the authors.

Conflicts of Interest: The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

1. Petersen, N.C.; Rodrigues, F.; Pereira, F.C. Multi-output bus travel time prediction with convolutional LSTM neural network. *Expert Syst. Appl.* **2019**, *120*, 426–435. [[CrossRef](#)]
2. Hamner, B. Predicting travel times with context-dependent random forests by modeling local and aggregate traffic flow. Paper presented at 2010 IEEE International Conference on Data Mining Workshops, Sydney, NSW, Australia, 13 December 2010; pp. 1357–1359. [[CrossRef](#)]
3. Ahmed, M.M.; Abdel-Aty, M. Application of stochastic gradient boosting technique to enhance reliability of real-time risk assessment. *Transp. Res. Rec. J. Transp. Res. Board* **2013**, *2386*, 26–34. [[CrossRef](#)]
4. Chung, Y.S. Factor complexity of crash occurrence: An empirical demonstration using boosted regression trees. *Accid. Anal. Prev.* **2013**, *61*, 107–118. [[CrossRef](#)]
5. Zhang, Y.R.; Haghani, A. A gradient boosting method to improve travel time prediction. *Transp. Res. Part C* **2015**, *58*, 308–324. [[CrossRef](#)]
6. Grande, E.; Imbimbo, M. A data-driven approach for damage detection: An application to the ASCE steel benchmark structure. *J. Civ. Struct. Health Monit.* **2012**, *2*, 73–85. [[CrossRef](#)]
7. Barbhuiya, S.; Sharif, S. Artificial Intelligence in concrete mix design: Advances, applications and challenges. In Proceedings of the 2023 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies, Sakhir, Bahrain, 20–21 November 2023; University of Bahrain: Sakhir, Bahrain, 2023. [[CrossRef](#)]
8. Kashinath, K.; Mustafa, M.A.; Albert, A.; Wu, J.L.; Jiang, C.; Esmailzadeh, S.; Azizzadenesheli, K.; Wang, R.A.; Chattopadhyay, A.; Singh, A.; et al. Prabhat Physics-informed machine learning: Case studies for weather and climate modelling. *Philos. Trans. R. Soc. A* **2021**, *379*, 20200093. [[CrossRef](#)]
9. Monteleoni, C.; Schmidt, G.A.; Saroha, S.; Asplund, E. Tracking climate model. *Stat. Anal. Data Min.* **2011**, *4*, 372–392. [[CrossRef](#)]
10. Willard, J.; Jia, X.W.; Xu, S.M.; Steinbach, M.; Kumar, V. Integrating scientific knowledge with machine learning for engineering and environmental systems. *ACM Comput. Surv.* **2022**, *55*, 1–37. [[CrossRef](#)]
11. Jamous, M.; Marsooli, R.; Miller, J.K. Physics-based modeling of climate change impact on hurricane-induced coastal erosion hazards. *Npj Clim. Atmos. Sci.* **2023**, *6*, 86. [[CrossRef](#)]
12. Wang, R.Y.; Liub, Y.; Lud, Y.; Zhang, J.B.; Liub, P.H.; Yao, Y.; Grekousis, G. Perceptions of built environment and health outcomes for older Chinese in Beijing: A big data approach with street view images and deep learning technique. *Comput. Environ. Urban Syst.* **2019**, *78*, 101386. [[CrossRef](#)]
13. Feng, C.; Jiao, J.F. Predicting and mapping neighborhood-scale health outcomes: A machine learning approach. *Comput. Environ. Urban Syst.* **2021**, *101562*, 101562. [[CrossRef](#)]
14. Reades, J.; De Souza, J.; Hubbard, P. Understanding urban gentrification through machine learning: Predicting neighbourhood change in London. *Urban Stud.* **2019**, *56*, 922–942. [[CrossRef](#)]
15. Zhu, D.X.; Cai, C.J.; Yang, T.B.; Zhou, X. A machine learning approach for air quality prediction: Model regularization and optimization. *Big Data Cogn. Comput.* **2018**, *2*, 5. [[CrossRef](#)]
16. Yuh, Y.G.; Tracz, W.; Matthews, H.D.; Turner, S.E. Application of machine learning approaches for land cover monitoring in northern Cameroon. *Ecol. Inform.* **2023**, *74*, 101955. [[CrossRef](#)]
17. Tsagkis, P.; Bakogiannis, E.; Nikitas, A. Analysing urban growth using machine learning and open data: An artificial neural network modelled case study of five Greek cities. *Sustain. Cities Soc.* **2023**, *89*, 104337. [[CrossRef](#)]
18. Li, F.; Yigitcanlar, T.; Nepal, M.; Nguyen, K.; Dur, F. Machine learning and remote sensing integration for leveraging urban sustainability: A review and framework. *Sustain. Cities Soc.* **2023**, *96*, 104653. [[CrossRef](#)]
19. Zheng, M.R.; Wang, H.Y.; Shang, Y.Q.; Zheng, X.Q. Identification and prediction of mixed-use functional areas supported by POI data in Jinan City of China. *Sci. Rep.* **2023**, *13*, 2913. [[CrossRef](#)]

20. Zheng, Y.; Lin, Y.; Zhao, L.; Wu, T.H.; Jin, D.P.; Li, Y. Spatial planning of urban communities via deep reinforcement learning. *Nat. Comput. Sci.* **2023**, *3*, 748–762. [[CrossRef](#)]
21. Wang, J.; Biljecki, F. Unsupervised machine learning in urban studies: A systematic review of applications. *Cities* **2022**, *129*, 103925. [[CrossRef](#)]
22. Cugurullo, F.; Caprotti, F.; Cook, M.; Karvonen, A.; McGuirk, P.; Marvin, S. The rise of AI urbanism in post-smart cities: A critical commentary on urban artificial intelligence. *Urban Stud.* **2024**, *61*, 1168–1182. [[CrossRef](#)]
23. Castaneda, J.; Cardona, J.F.; Martins, L.; Juan, A.A. Supervised machine learning algorithms for measuring and promoting sustainable transportation and green logistics. *Transp. Res. Procedia* **2021**, *58*, 455–462. [[CrossRef](#)]
24. Kalliola, J.; Kapočiūtė-Dzikiėnė, J.; Damaševičius, R. Neural network hyperparameter optimization for prediction of real estate prices in Helsinki. *PeerJ Comput. Sci.* **2021**, *7*, e444. [[CrossRef](#)] [[PubMed](#)]
25. Hjort, A.; Pensar, J.; Scheel, I.; Sommervoll, D.E. House price prediction with gradient boosted trees under different loss functions. *J. Prop. Res.* **2022**, *39*, 338–364. [[CrossRef](#)]
26. Calainho, F.D.; van de Minne, A.M.; Francke, M.K. A machine learning approach to price indices: Applications in commercial real estate. *J. Real Estate Financ. Econ.* **2024**, *88*, 624–653. [[CrossRef](#)]
27. Lorenz, F.; Willwersch, J.; Cajias, M.; Fuerst, F. Interpretable machine learning for real estate market analysis. *Real Estate Econ.* **2023**, *51*, 1178–1208. [[CrossRef](#)]
28. Heydari, Z.; Stillwell, A.S. Comparative analysis of supervised classification algorithms for residential water end uses. *Water Resour. Res.* **2024**, *60*, e2023WR036690. [[CrossRef](#)]
29. Quan, S.J. Comparing hyperparameter tuning methods in machine learning based urban building energy modeling: A study in Chicago. *Energy Build.* **2024**, 114353. [[CrossRef](#)]
30. Fan, T.H.; Ren, Y.; Chapman, A. Unveiling the carbon neutrality pathways of compact cities: A simulation-based scenario analysis from China. *Humanit. Soc. Sci.* **2025**, *12*, 1205. [[CrossRef](#)]
31. Jang, S.D.; Yoo, J.H.; Lee, Y.S.; Kim, B. Flood prediction in urban areas based on machine learning considering the statistical characteristics of rainfall. *Prog. Disaster Sci.* **2025**, *26*, 100415. [[CrossRef](#)]
32. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 2623–2631. [[CrossRef](#)]
33. Lenaers, I.; de Moor, L. Exploring XAI techniques for enhancing model transparency and interpretability in real estate rent prediction: A comparative study. *Financ. Res. Lett.* **2023**, *58*, 104306. [[CrossRef](#)]
34. Neves, F.T.; Aparicio, M.; de Castro Neto, M. The impacts of open data and eXplainable AI on real estate price predictions in smart cities. *Appl. Sci.* **2024**, *14*, 2209. [[CrossRef](#)]
35. Jin, S.X.; Zheng, H.X.; Marantz, N.; Roy, A. Understanding the Effects of Socioeconomic Factors on Housing Price Appreciation Using Explainable AI. *Appl. Geogr.* **2024**, *169*, 103339. [[CrossRef](#)]
36. Khan, A.; Debnath, P.; Sayeed, A.A.; Sumon, F.I.; Rahman, A.; Khan, T.; Pant, L. Explainable AI and Machine Learning Model for California House Price Predictions: Intelligent Model for Homebuyers and Policymakers. *J. Bus. Manag. Stud.* **2024**, *6*, 73–84. [[CrossRef](#)]
37. Renaud, J.; Karam, R.; Salomon, M.; Couturier, R. Deep learning and gradient boosting for urban environmental noise monitoring in smart cities. *Expert Syst. Appl.* **2023**, *218*, 119568. [[CrossRef](#)]
38. Ho, W.K.O.; Tang, B.S.; Wong, S.W. Predict property prices with machine learning algorithms. *J. Prop. Res.* **2021**, *38*, 48–70. [[CrossRef](#)]
39. Long, Y.S.; Zeng, L.Q.; Wang, J.; Long, X.C.; Wu, L. A gradient boosting approach to estimating tail risk interconnectedness. *Appl. Econ.* **2022**, *54*, 862–879. [[CrossRef](#)]
40. Yin, H.; Sharma, B.; Hu, H.; Liu, F.; Kaur, M.; Cohen, G.; McConnell, R.; Eckel, S. Predicting the climate impact of healthcare facilities using Gradient Boosting Machines. *Clean. Environ. Syst.* **2024**, *12*, 100155. [[CrossRef](#)]
41. Sanchez, T.W.; Shumway, H.; Gordner, T.; Lim, T. The prospects of artificial intelligence in urban planning. *Int. J. Urban Sci.* **2022**, *27*, 179–194. [[CrossRef](#)]
42. Jun, H.J.; Jung, S.; Kang, S.; Kim, T.; Cho, C.H.; Jhoo, W.Y.; Heo, J.P. Factors associated with pedestrian-vehicle collision hotspots involving seniors and children: A deep learning analysis of street-level images. *Int. J. Urban Sci.* **2023**, *28*, 359–377. [[CrossRef](#)]
43. Sideris, N.; Bardis, G.; Voulodimos, A.; Miaoulis, G.; Ghazanfarpour, D. Using Random Forests on Real-World City Data for Urban Planning in a Visual Semantic Decision Support System. *Sensors* **2019**, *19*, 2266. [[CrossRef](#)] [[PubMed](#)] [[PubMed Central](#)]
44. Choy, L.H.T.; Ho, W.K.O. On the use of machine learning in real estate research. *Land* **2023**, *12*, 740. [[CrossRef](#)]
45. Aysan, A.F.; Ciftler, B.S.; Unal, I.M. Predictive power of Random Forests in analyzing risk management in Islamic banking. *J. Risk Financ. Manag.* **2024**, *17*, 104. [[CrossRef](#)]

46. Loef, B.; Wong, A.; Janssen, N.A.H.; Strak, M.; Hoekstra, J.; Picavet, H.S.J.; Boshuizen, H.C.H.; Verschuren, W.M.M.; Herber, G.-C.M. Using Random Forest to identify longitudinal predictors of health in a 30-year cohort study. *Sci. Rep.* **2022**, *12*, 10372. [[CrossRef](#)] [[PubMed](#)]
47. Friedman, J. Greedy boosting approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [[CrossRef](#)]
48. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
49. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [[CrossRef](#)]
50. Breiman, L. *Arcing the Edge*; Technical Report 486; Statistics Department, University of California: Oakland, CA, USA, 1999.
51. Bergstra, J.; Bengio, Y. Random Search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
52. OPTUNA. Optuna: A Hyperparameter Optimization Framework. 2018. Available online: <https://optuna.readthedocs.io/en/stable/> (accessed on 5 April 2024).

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.