Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 16, No. 1, 2025

Prediction of Demand for Shared Bicycles Based on Machine Learning

Wenyu Jiang[1, a]

[1]London Brunel School, North China University of Technology, Beijing 100144, China
[a]jiangwenyu123@mail.ncut.edu.cn

Abstract: With the acceleration of urbanization, shared bicycles have become an indispensable component of modern urban transportation systems, playing an important role in improving social resource utilization, alleviating traffic pressure, and promoting green travel. However, a common problem of supply-demand imbalance exists in the spatial and temporal distribution of vehicles, which not only affects user experience but also increases operating costs. In this context, accurate prediction of shared bicycle rental demand is crucial for achieving refined operations. This paper presents a high-precision prediction model constructed using machine learning technology. The proposed methodology, structured into three distinct modules, begins with detailed exploratory data analysis and comprehensive feature engineering. This includes logarithmic transformation of target variables, encoding of categorical features, and critically, the construction of key interaction features to capture the complex relationship between variables like temperature and hour. On this basis, the XGBoost model is employed to evaluate feature importance and select an optimal feature subset. Subsequently, through a series of comparative experiments, a range of machine learning regression models, including linear, tree-based, and gradient boosting models, are systematically compared. Using a time-series cross-validation method, each model is trained and evaluated with Root Mean Square Logarithmic Error (RMSLE) and the coefficient of determination (R^2) as the main evaluation indicators. Finally, hyperparameters for the top-performing models are optimized to further improve prediction accuracy and generalization ability. The results indicate that the tree-based ensemble model LightGBM exhibits superior performance in the task of predicting shared bicycle demand, with its accuracy further improved after hyperparameter optimization. This study not only provides an effective demand forecasting solution for shared bicycle operators, but also offers a valuable reference for similar time series forecasting problems.

Keywords: shared bicycles; demand forecasting; machine learning; feature engineering; XGBoost; LightGBM; time series analysis.

1. Introduction

In the rapid development of shared bicycles, operational management faces many severe challenges, among which the most prominent is the extremely uneven use of vehicles in different regions and time periods. For example, during the morning rush hour on weekdays, a large number of citizens ride bicycles from residential areas and subway stations to commercial centers and office areas, resulting in the former having no cars to rent while the latter have a pile of vehicles; the opposite is true during the evening rush hour.

To address this, numerous studies have applied machine learning and deep learning models to forecast demand, with empirical evaluations comparing their relative performance [1]. These studies indicate that the optimal model often depends on the specific forecast horizon, with different models excelling at short-term versus long-term predictions [2]. Such approaches often involve extensive feature engineering and in-depth analysis of various ensemble techniques to build a robust predictive pipeline [3]. In addition to operational issues, research has also highlighted significant safety challenges within these systems, which must be addressed to ensure sustainable urban mobility [4].

Although these researchers have put forward excellent solutions for shared vehicle cost optimization and demand forecasting, they have not reached a fine level in data pre-processing. In addition, the above-mentioned papers have not further processed the optimization and processing methods for features, so that the model still has room for improvement. And many articles do not mention how to deal with features and super parameter optimization of the model. Beyond transportation, these systems are also being explored as mobile sensor platforms for smart city applications, such as using on- bike sensors and ontologies to detect urban events like illegal parking [5].

In order to solve the above problems, this paper proposes an integrated framework and data processing methods to maximize the relationship and use of mining features.First, after careful data analysis, the processing method of each feature is determined. After that, the missing values of each feature are processed by the capping method. Use feature engineering to disassemble, construct features.Second, because many features are generated in the process of constructing interactive features, this paper uses XGBoost model to screen features, and selects the best top 100 features for later input model. Finally, many models are constructed for evaluation and comparison, and it is found that the tree based model LightGBM has the best effect. The super parameter optimization such as grid search is carried out for multiple parameters of LightGBM, which further improves the performance of the model. The scope of these forecasting challenges is broad, ranging from short-term operational predictions to long-term time series analysis that must account for major external events like the COVID-19 pandemic [6]. Comprehensive data analytics systems are therefore essential, providing practical insights into usage trends for vendors and city planners alike [7].

In order to verify the effectiveness of our algorithm, we designed the following experiments:

1. Comparative Experiment: This paper will compare with

the algorithms used in other papers on the same data set and will use the same evaluation criteria such as RMSE, R2 to fairly show the advantages and disadvantages of all algorithms.

2. Ablation Experiment: In the ablation experiment, this paper will compare the changes of the model effect in the case of removing different modules, in order to get the best effect only when the three modules work together.

The contributions of this paper are as follows:

1. This paper analyzes the shortcomings of existing methods in data preprocessing and Feature Engineering in detail, and introduces the detailed data preprocessing methods and feature engineering processing, especially the interval division of different natural variables and the dynamic changes of natural variables in different time, which provides effective help for subsequent researchers.

2. This paper proposes a general and detailed framework of three modules to solve the problem of forecasting the demand for shared single vehicles. First, analysis and processing, second, feature selection, and finally, model construction and superparameter optimization. This framework process can provide clear ideas for other researchers.

3. The effectiveness of the results is proved by a large number of experiments, and the effectiveness and advancement of the framework algorithm are proved by comparative experiments. At the same time, the ablation experiment shows that the three modules in the framework are indispensable.

2. Related Work

[8] discussed the main influencing factors of short-term (hour based) demand forecasting for shared bicycles based on the multi-dimensional large sample data of shared bicycles, using machine learning models such as lasso regression, ridge regression, random forest and iterative decision tree, and compared the forecasting effects of different models. [8] have made important contributions to feature construction and processing by identifying key factors such as time, place and weather, and dividing dates into weekdays and weekends, which makes the feature division very delicate, but there is still a lack of explicit interactive feature construction. Their research relies on implicit learning of these relationships such as random forests, and may not be able to capture obvious interactive relationships, such as time and temperature. The first module of this paper can accurately solve these relationships. This paper not only constructs the random forest model to implicitly learn the nonlinear relationship, but also explicitly constructs the interactive characteristics, so that the model has a higher starting point at the beginning, and can learn the shallow and deep relationship at the same time. At the same time

[8] mentioned that 75 explanatory variables were selected for the OLS model, but no special screening was done, which may greatly increase the time complexity in general, while allowing the model to learn unimportant features and weaken the performance. Therefore, the second module of this paper can also perfectly solve this problem by using XGBoost model to screen the number of features. Similarly, other studies have also focused on short-term forecasting by combining machine learning with methods like spatial constraint clustering to group stations before prediction [9]. In contrast, some approaches argue for simplicity, demonstrating that incremental learning models with minimal features can achieve high accuracy for short-term predictions [10].

[11] proved that within the neural network, the LSTM with more complex structure is superior to the traditional RNN and BP network, which is an excellent conclusion. However, it also has similar problems in the processing of features, that is, it simply carries out data standardization and missing value filling, and does not consider the relationship and details of feature time, which can be discussed in depth in this module. Similarly, [11] did not carry out feature screening work, but directly input 60 features into the model, which can also be solved by applying the second module of this paper.

[12] have made great breakthroughs in the innovation of model architecture. They use convolutional network (CNN) to extract local features, bidirectional long-term and short-term memory network (BiLSTM) to capture long-term time dependence, and attention mechanism to dynamically allocate feature weights. This is a very advanced combination of algorithms, but the above problems still exist in the processing of features and target variables. Beyond mainstream learning models, some research has applied advanced statistical methods, such as bimodal Gaussian inhomogeneous Poisson processes, to predict bike counts [13]. To tackle the prediction challenge at different spatial scales, hierarchical prediction methods have been proposed, which ensure consistency across various levels, from individual stations and clusters to the entire city [14], [15].

In addition to RNN-based architectures, other works have explored different approaches. For instance, to improve training efficiency, He et al. introduced a model integrating a Temporal Convolutional Network (TCN) with self-attention, utilizing periodic time signals for better feature representation [16]. To provide more operational value, Yang et al. developed a dual-branch neural network to simultaneously predict bike counts and classify the station's imbalance level, offering richer information for rebalancing [17]. Moving beyond pure prediction accuracy, some research aims for model interpretability. Gu et al. proposed a framework to predict bike flow patterns between regions by constructing interpretable base matrices that represent underlying traffic modes [18].

Other research has focused on optimizing the prediction task for operational efficiency. For example, Huang et al. proposed the concept of "central stations" to reduce computational and rebalancing costs by focusing prediction efforts only on high-demand locations [19]. Some works combine geographic information systems (GIS) with deep learning, such as GSTNet, which uses Gaussian Mixture Model clustering before applying a 3D-CNN to predict traffic flow between station groups [20]. The prediction horizon is also a key variable, with models like Bi-LSTM being specifically evaluated for very short-term forecasting (15-60 minutes), often incorporating explainability methods like SHAP to interpret feature importance [21]. A completely different perspective is to predict the destination of an individual trip rather than station-level demand, framing it as a classification problem based on user, time, and location features [22].

Further diversifying the research landscape, some works shift the focus from operator-centric demand prediction to user-centric services, such as developing machine learning systems that recommend specific bikes to users based on their travel needs [23]. Another critical challenge being addressed is the 'cold-start' problem, where Graph Neural Networks are

utilized to predict traffic flow for new city blocks that lack historical data, by modeling the relationships between them [24].

Graph Neural Networks (GNNs) have emerged as a particularly powerful tool for this domain. For instance, Guo et al. proposed BikeNet, a framework that uses a spatiotemporal GNN to predict demand, which is then directly used to optimize station rebalancing via integer linear programming [25]. Similarly, Qin et al. developed a two-step solution using a convolutional network to predict flows and an improved local search algorithm for multi-carrier path planning [26]. To further enrich the model's context, some works focus on fusing heterogeneous data, such as the multi-view network by Chai et al., which integrates spatial, temporal, and semantic information [27]. A more advanced approach involves cross-modal knowledge transfer, where domain-adversarial GNNs are used to leverage data from other transport systems like subways and ride-hailing to improve bike demand prediction, even when their demand patterns differ significantly [28]. Beyond integrated frameworks, research also investigates novel model architectures, such as modular deep learning designs that use separate components like CNNs and LSTMs to capture long-term and short-term patterns respectively [29].

A particularly novel approach tackles the 'cold-start' problem for new stations by learning 'place representations'. Instead of relying on historical demand, models like the one proposed by Zhou and Huang use large-scale movement data from other sources (e.g., taxi trips) to create functional embeddings for locations, thereby enabling demand prediction for stations with no prior activity records [30].

3. III. Methodology

A. Symbol Introduction

Let x represent the feature vectors. This is the input for all models, including all independent variables used for prediction such as perceived temperature, humidity, hour, and day of the week. y is the dependent variable output by the model; it is the desired target for the model to predict, which is the actual number of shared bike rentals. $\hat{y}$ represents the predicted value. This is the predicted value provided by machine learning model $M(x)$ based on input features. θ represents parameters such as model weights and biases. The model needs to be updated through training data learning. The value of θ directly determines how the model learns to map the input x to the output $\hat{y}$. $J(\theta)$ is the cost function. It measures the difference between all predicted values and the true values of the model. R^2 is the determination coefficient. It measures the percentage of variance in the model's predicted rental quantity. The closer R^2 is to 1, the better the fitting effect of the model. **RMSLE** stands for Root Mean Square Logarithmic Error. This is the main error metric used in this article. It calculates the root mean square of the difference between the logarithm of the predicted value and the true value. $P(t)$ is the local fitting polynomial, used by the Savitzky-Golay filter to approximate the data within a sliding window.

B. Module 1: Preprocess Module

This article's dataset is from the Bike Sharing Demands dataset on the Kaggle website, available at: https://www.kaggle.com/c/bike-sharing-demand/data. Working with such real-world public datasets often presents challenges related to data quality and consistency, necessitating robust preprocessing and exploratory analysis to distill useful patterns [31]. This initial analysis aligns with common practice, where studies investigate the influence of various factors, including the built environment and social-economic characteristics, on bike demand using data analytics before model construction [32]. With 12 attributes, this article focuses on the 10 most important attributes for research.

Table I
DATASET ATRIBUTE DESCRIPTION

Name	Meaning	Description
datetime	Date and time	Hourly date and time data
season	Season	1-4:Spring-Winter
holiday	Holiday	1=Holiday, 0=Not a holiday
workingday	Working day	1=Work-day, 0=Not Work-day
weather	Weather	1=Clear, 2=Cloudy, 3=snow, 4=Bad weather
temp	Actual temperature	Temperature(Celsius)
atemp	Apparent temperature	Apparent temperature(Celsius)
humidity	Humidity	Relative humidity (%)
windspeed	Wind speed	Wind speed (km/h)
count	Total number of rentals	Total number of hourly rentals

At the same time, this study also plotted data distribution graphs for 14 attributes (as timestamps are high cardinality timestamp sequences, the histograms of their original values usually do not directly reflect periodic or easily interpretable distribution patterns. It is better to break them down into more meaningful categories or periodic features such as year, month, day, and hour for analysis, so their histograms are not directly plotted).

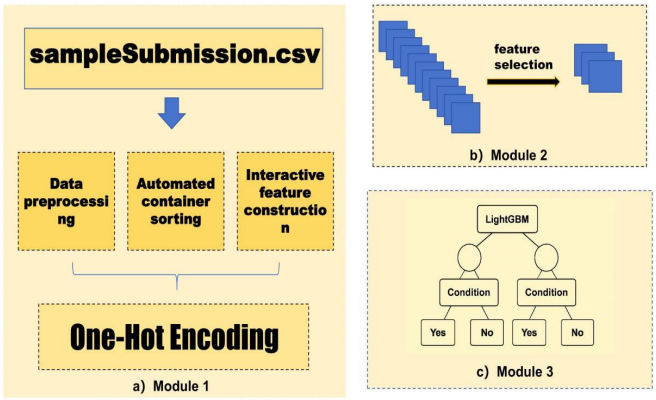

Figure 1. Overall Framework of the Proposed Model.

As shown in the figure, the target variable of this study, the hourly count of shared bicycle rentals (count), exhibits a significant right-skewed distribution. Such a skewed distribution can negatively impact the performance of many machine learning models, particularly those that assume a normal or symmetric distribution. To mitigate this issue, this study applies a logarithmic transformation to the target variable. Specifically, it adopts the form of a log-plus-one transformation, log(1 + x), as follows:

$$count_{\log} = \ln(1 + \text{count}) \tag{1}$$

All subsequent models are trained using the transformed rental count as the dependent variable. After a prediction is obtained from the model, denoted as y^log, it is restored to the original scale by applying the inverse transformation:

$$\hat{y} = e^{\hat{y}_{\log}} - 1 \quad (2)$$

As shown in Figure 3, the right subgraph displays the distribution of the target variable after applying the ln(1 +count) logarithmic transformation. By comparison, it is evident that the transformation substantially mitigates the right-skewness of the original data.

In order to investigate whether there is a linear relationship between different features, this article has drawn a thermal matrix diagram,as depicted in Fig. 4.

The features contained in the original dataset are not suitable for direct input into the model, and some features have little meaning to the model. Therefore, this article first extracted structured temporal features and preliminarily screened some of the original features.

1) Time feature extraction: This article extracted four variables, year, month, hour, and dayofweek, from the original timestamp features, and removed the original datatime variable.

2) Daily feature removal: Due to the relatively limited contribution of specific ”days” in the month to the hourly demand pattern and the introduction of unnecessary model complexity. Therefore, this article will remove daily features.

3) Actual temperature feature removal: There is a very high Pearson correlation coefficient between atemp and temp, indicating a serious collinearity issue between the two. Therefore, this article chooses to retain the perceived temperature that has a greater impact on the riding experience.

Through exploratory data analysis, it was found that there are many outliers in some numerical features. This article uses the cap method to correct these special observations.

1) **Firstly, the correction of humidity values:**

The lower bound of the outlier calculated by the anomaly detection method based on quartile range is 2%, and humidity observations below 2% are identified as potential statistical anomalies. The correction formula is h′ = max (h_{target_floor}, h_{min_valid}), where h_{target_floor} is a preset lower limit for target correction, set at 5% in this study. H_{min_valid} is the minimum value among all observations in the training dataset where the original humidity value is not lower than the threshold (i.e.,≥ 2%). Therefore, the final cap value h_{cap_value} is max(5%, h_{min_valid}).

2) **Wind speed processing:**

The dataset records some extremely high values. There are 1313 wind speed records in the original dataset with a value of 0, and a maximum wind speed value of 56.9969 km/h, while the 99% percentile is 35.0008 km/h. In order to reduce the impact of these outliers on the model, this paper has processed them. Based on the above two issues, the corrected wind speed w′ is defined as follows:

$$w' = \begin{cases} w_{1\%} & if\ w = 0\ (w_{1\%} \approx 6.0032\ km/h) \\ w_{max\_cap} & if\ w > w_{max\_cap}\ (w_{max\_cap} = 35\ km/h) \\ w & otherwise \end{cases} \quad (3)$$

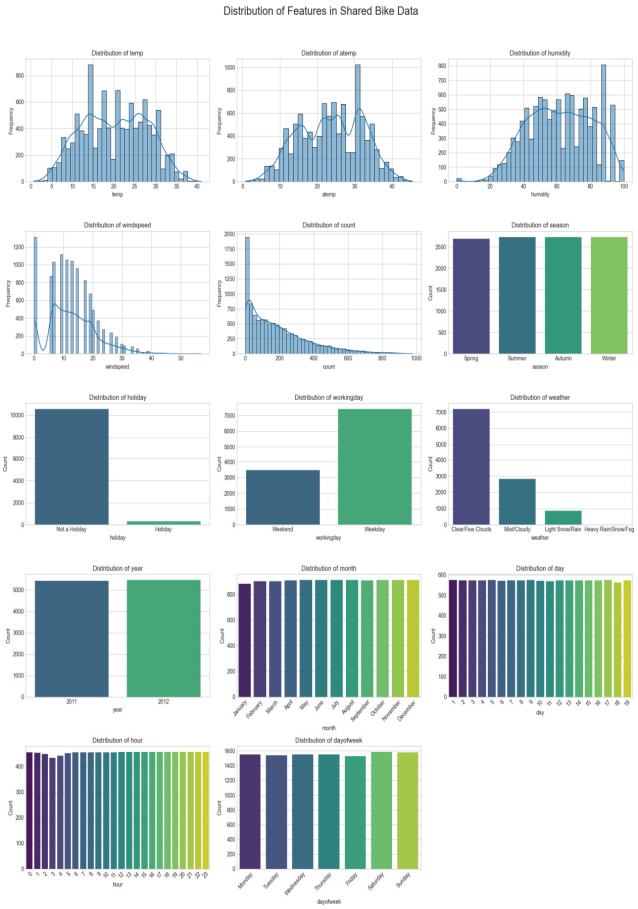

Figure 2. Distribution diagram of each variable.

In order to effectively utilize them by machine learning models, this study first encodes some basic categorical features:

1) Weather feature processing and encoding: The sample size of category 4 in the original weather features is extremely small (only 1 case appears in the training set). To avoid the interference that sparse categories may cause to model learning, category 4 is merged into category 3. Subsequently, the One Hot Encoding technique was used to convert it into multiple binary indicator variables.

2) Encoding of other time category features: Single hot encoding technology is also used for other time category features.

To precisely capture the complex, non-linear relation- ships between continuous features like apparent temperature (atemp) and humidity (humidity) and the rental counts, a data-driven binning approach was adopted. This method automates the discovery of meaningful thresholds by analyzing the trend of rental demand. The core of this approach is the Savitzky-Golay (SG) filter, which is used to smooth the noisy trend data.

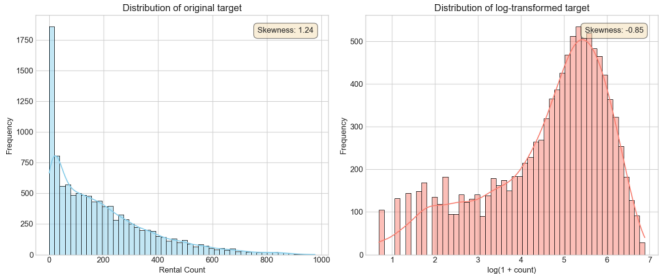

Figure 3. Comparison chart before and after count conversion

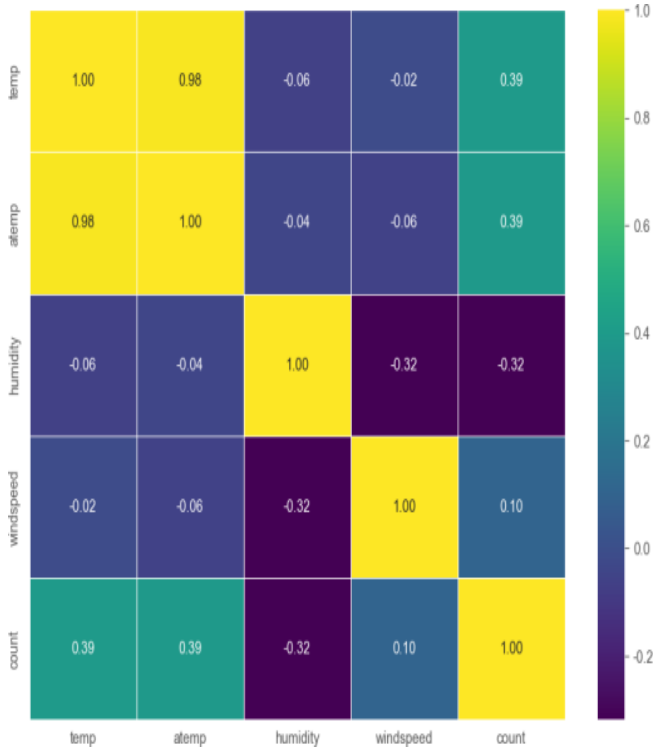

Figure 4. Heatmap of the Pearson correlation matrix.

The SG filter works by fitting a local polynomial regression to the data within a sliding window. For a given data point, its smoothed value is the value of the fitted polynomial at that point's location. This can be represented by fitting a polynomial of degree k:

$$P(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_k t^k \quad (4)$$

The key idea is to find the coefficients a_j that minimize the least-squares error between the polynomial P(t) and the raw data within the window. This effectively removes short-term noise while preserving significant features of the trend, such as peaks and turning points.

In this study, the process involved first creating a large number of fine-grained bins for atemp and humidity and calculating the average rental count for each. The SG filter was then applied to this noisy sequence. As shown in the debugging curves in Fig. 5, this smoothing step reveals the underlying trend clearly. Subsequently, a heuristic algorithm identified key turning points on these smoothed curves, which were then set as the final binning thresholds. This automated process allows the bins to adapt to the data's intrinsic structure, leading to more meaningful feature engineering.

Based on the thresholds automatically calculated, the apparent temperature and humidity features were binned. The statistical characteristics of the resulting new categories are summarized in Table II and Table III.

TABLE II
APPARENT TEMPERATURE (ATEMP) BINNED STATISTICS

atemp	Mean Rentals	Median Rentals	Records
Low (< 11.04°C)	73.72	50.0	743
Moderate (11.04–38.74°C)	197.57	153.0	9927
High (≥38.75°C)	321.42	289.0	216

To capture the complex synergistic effects between variables, this study constructed several interaction features. The core rationale is that the impact of a weather condition on bike rentals often depends on the time of day. For instance, a 'moderate' temperature during commute hours (e.g., 8 AM) likely stimulates more demand than the same temperature at midnight.

TABLE III
HUMIDITY BINNED STATISTICS

humidity	Mean Rentals	Median Rentals	Records
Very Low (< 28.8%)	275.78	242.0	367
Mid-Low (28.8–57.3%)	242.85	203.0	4327
Mid-High (57.4–77.7%)	178.91	132.0	3489
Very High (≥ 77.8%)	114.41	65.0	2703

This interaction was systematically created by performing an element-wise multiplication between the one-hot encoded vectors of the binned weather features and the one-hot encoded hourly features. For an interaction between the *k*-th temperature category and the *j*-th hour, the resulting feature is defined as:

$$F_{\text{interact\_temp}(k,j)} = \text{I}_{\text{temp\_k}} \times I_{\text{hour\_j}} \quad (5)$$

Where $\text{I}_{\text{temp\_k}}$ and $\text{I}_{\text{hour\_j}}$ are the binary indicator variables for the respective categories. A similar logic was applied to create interactions between humidity bins and hours.

This data-driven approach automatically generated a total of 168 new interaction features. These features enable the model to learn context-dependent patterns, significantly enhancing its predictive power compared to using the base features alone.

After completing the conversion and encoding of all features, in order to eliminate the adverse effects that may arise from differences in intrinsic dimensions and numerical ranges between different numerical input features on model training, this paper standardized them using Z-score. The conversion formula is as follows:

$$X_{\text{std}} = \frac{X - \mu_{\text{train}}}{\sigma_{\text{train}}} \quad (6)$$

Among them, $X$is the original feature value, while μ_{train} and σ_{train} are the mean and standard deviation of the feature calculated on the **training set**, respectively

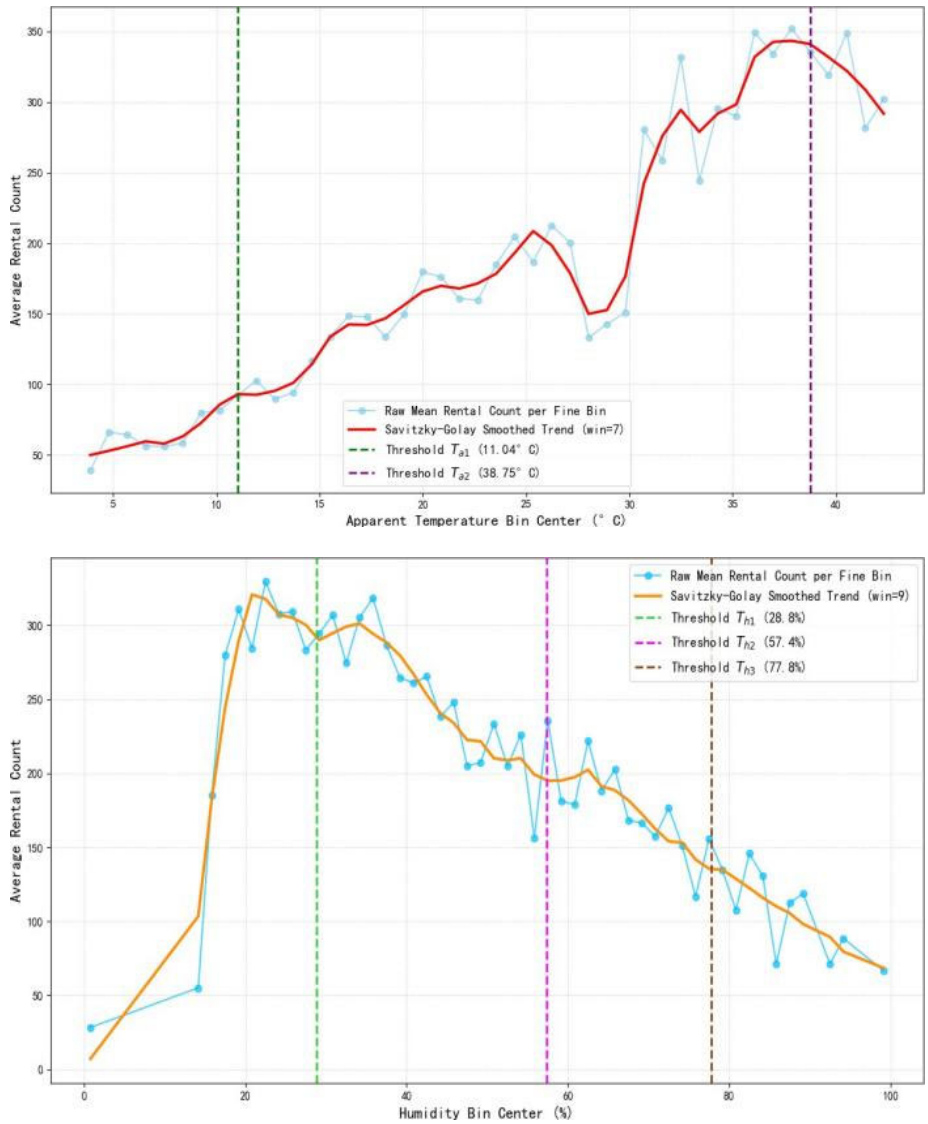

Figure 5. Sensory temperature and humidity box division results.

C. Module 2: Feature Selection

After preprocessing and feature engineering, the dimension of the new feature set significantly increased to 232. Although high-dimensional feature spaces may contain richer information, they can also introduce noise, increase the computational complexity of model training, and increase the risk of overfitting. Therefore, it is necessary to perform feature screening.

This article uses the feature importance evaluation function built into the XGBoost model for feature selection. We chose Gain as the core evaluation metric, which measures the reduction of the objective function when each feature is used as a splitting node in all decision trees of the model.

After evaluating all 232 features, we found that their importance scores exhibited a typical long tail distribution: a few features contributed the vast majority of predictive power, while the contribution of a large number of features was negligible. As shown in Table IV, some interaction features and temporal features occupy the top positions in importance ranking. Specifically, we observed that 80 features had an importance score of 0. Based on the above analysis, in order to strike a balance between retaining key information and improving model efficiency, we ultimately chose to retain the top 152 most important features for subsequent model training.

TABLE IV
FEATURE IMPORTANCE RANKING (PARTIAL EXAMPLE - TOP 10)

Ranking	Feature Names	Importance (Gain)
1	atemp interval atemp moderate x hour 1	0.246892
2	atemp interval atemp moderate x hour 4	0.095076
3	hour 4	0.083281
4	hour 3	0.059399
5	atemp interval atemp moderate x hour 3	0.055435
6	hour 5	0.048859
7	hour 2	0.047714
8	atemp interval atemp low x hour 8	0.039121
9	hour 1	0.034995
10	hour 0	0.016326

D. Module 3: Model Building

The construction, training, and evaluation of all machine learning models in this paper are based on the Python programming language and its powerful scientific computing ecosystem. Specifically, it primarily relies on scikit-learn, a widely used machine learning library. The feature set for model construction, X_{train}, consists of 152 feature columns determined after feature selection, with a dimension of $m \times 152$, where m is the number of training samples, 10886. The target variable, $y_{\text{train\_log}}$, is the log-transformed rental count, with a dimension of $m \times 1$. For the subsequent validation of the model's generalization ability, a test set

feature matrix, X_{test}, which has undergone consistent processing and feature selection, was also prepared, with a dimension of 6493 ×152. To comprehensively explore the potential of various algorithms for this bike-sharing demand prediction task, this study selected a range of mainstream regression algorithms, including linear models and tree-based ensemble models, to construct a suite of predictive models, as shown in Table V. All non-deterministic models were configured with a fixed random seed (random_state=42) to ensure the reproducibility of the experimental results.

TABLE V
BASELINE MODELS AND THEIR GENERAL SETUP

Model Name	General Setup Description
Linear Regression	Standard implementation with no regularization.
Lasso	Regularized with a small L1 penalty.
Ridge	Regularized with a standard L2 penalty.
Elastic Net	A combination of L1 and L2 regularization.
Decision Tree	A single tree model with baseline parameters.
Random Forest	An ensemble of 100 trees with baseline settings.
XGBoost	Gradient boosting with baseline learning rate and depth.
LightGBM	Gradient boosting with baseline learning rate and leaves.

Each of the selected baseline models will be trained on the prepared training data.

E. Experiment

1) Comparative Experiment: In order to systematically and scientifically evaluate the effectiveness and progressiveness of the shared bicycle demand forecasting model proposed in this paper, this paper designs a series of strict contrast experiments. Fair evaluation principle: All models run on the same dataset and hardware environment. The evaluation indicators are uniformly RMSLE and R2 to ensure horizontal comparison under the same standard. Adopting rigorous Time Series Cross Validation for model training and evaluation. For a robust performance evaluation, the model training will incorporate a time-series cross-validation strategy. The sklearn.mode_selection.TimeSeriesSplit method will be used to divide the training data into 5 (n_splits=5) training-validation set pairs, ensuring that the validation set always follows the training set in chronological order. At the same time, in order to ensure that the starting point of all model algorithms is the same, this article sets the hyperparameters of the following experimental models as random hyperparameters.

Experimental group 1: This is the model algorithm proposed in this article, which is also the core comparative experimental algorithm. The evaluation results of this experiment will be compared in depth with the remaining three experimental groups.

The table clearly displays the results of group1, where the performance of LightGBM is particularly outstanding.

TABLE VI
MODEL PERFORMANCE

Model Name	CV Mean RMSLE	CV Mean R^2
LightGBM	0.4653	0.8881
XGBoost	0.4756	0.8788
Random Forest	0.5236	0.8463
Linear Regression	0.6189	0.7934
Ridge	0.6196	0.7930
Lasso	0.6300	0.7867
Elastic Net	0.6308	0.7861
Decision Tree	0.7081	0.7259

Note: Both metrics are calculated on the log-transformed target variable. The CV Mean RMSLE is equivalent to the RMSE of the transformed values.

Experimental group 2: Compared to the method in [8].

The evaluation results are shown in the following figure and table:

TABLE VII
PERFORMANCE OF THE MODEL PROPOSED BY [8]

Model	CV Mean RMSLE	CV Mean R^2
Random Forest	0.524117	0.845876
Gradient Boosting	0.608922	0.798118
Linear Regression	0.619973	0.792532
Ridge	0.620408	0.792309
Decision Tree	0.660715	0.757237
Elastic Net	1.261171	0.144826
Lasso	1.267578	0.136411

From the results table, even the best performing random forest model 0.524117 did not perform as well as Lightgbm's 0.4653 in this paper, which indirectly reflects the effectiveness of module 3 in this paper.

Experimental group 3: LSTM deep learning model based on [11].

This model strictly replicates the method proposed by this group in their paper, using a single-layer long short-term memory network (LSTM) for prediction. Data preprocessing follows the original text, using minimum maximum normalization and dividing the training and testing sets in chronological order.

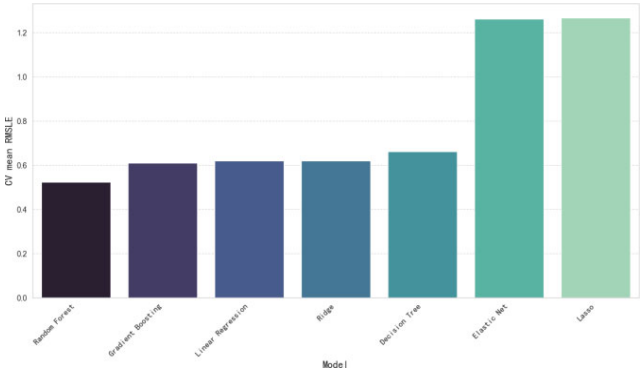

Figure 6. CV Mean RMSLE

Although the results of each training may differ due to differences in algorithm settings and datasets, overall, the training results using Module 1 are better than the original algorithm without Module 1. This proves that the data preprocessing and feature engineering in Module 1 have indeed excavated the value of the data.

Experimental group 4: Based on the CNN BiLSTM Attention deep learning benchmark proposed by [12].

This model replicates the more cutting-edge and complex deep learning architecture proposed by [12] in their paper, namely CNN-BiLSTM-Attention. Data preprocessing and model hyperparameters strictly follow the original text.

The final evaluation result of the baseline model is as follows: R^2: 0.9228; RMSLE: 0.3891. Although the ultimate performance of the model proposed in this article might not surpass that of this baseline model, it does not invalidate the effectiveness of the modules proposed herein.

In order to verify that Modules 1 and 2 can also enhance the

performance of other models, this paper integrates Modules 1 and 2 with the algorithm proposed by [12], combining our data processing and feature engineering techniques into their approach. The final evaluation results for this combined approach are presented below, along with the corresponding prediction plot shown in Figure 7.

- R^2: 0.9324
- RMSLE: 0.3765

As observed, R^2 has significantly improved and RMSLE has notably decreased, which directly indicates the significant role of these modules.

2) Ablation Experiment: In order to rigorously verify that different modules do indeed help improve the performance of the model, this paper designed three different ablation experiments, removing different modules to observe the predicted results.

Experimental group 1: Remove Module 1

In this experiment, I will not perform any additional processing on the initial data, but will only perform basic feature splitting and logarithmic transformation to demonstrate the effectiveness of interactive feature construction and automated binning operations.

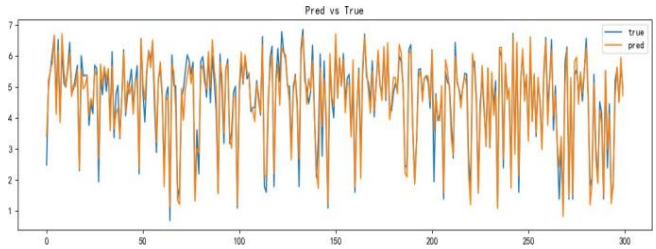

Figure 7. Prediction vs. True Values for the Combined Model

TABLE VIII
MODEL PERFORMANCE WITH BASELINE FEATURES

Model	CV Mean RMSLE	CV Mean R2
LightGBM	0.474516	0.872277
XGBoost	0.478621	0.872030
Random Forest	0.522100	0.846892
Linear Regression	0.621773	0.791246
Ridge	0.622055	0.791140
Elastic Net	0.628228	0.788013
Lasso	0.630166	0.786675
Decision Tree	0.707639	0.728073

1) Significant Improvement for LightGBM Model: Among all models, LightGBM showed the most promi- nent performance improvement after introducing optimized feature engineering. Its CV Mean RMSLE decreased from 0.4745 with baseline feature engineering to 0.4653. This strongly demonstrates the positive impact of the advanced feature engineering strategy proposed in this paper on LightGBM's model performance.

Slight Improvement for Linear Models: For linear models, both RMSLE and R^2 showed slight improvements after introducing optimized feature engineering, although the magnitude of improvement was relatively small, there was still an improvement.

Conclusion: The experimental results indicate that the Module 1 operation proposed in this paper is most effective for improving LightGBM's performance, capable of effectively reducing prediction errors and enhancing model interpretability.

Experimental group 2: Remove Module 2

In Module 2, we removed the feature filtering part, and the final evaluation value showed the same as the unfiltered result. Although the value did not change, the training time was increased by 1.2 seconds, which is equivalent to a 2.3 percentage reduction in training time. Therefore, it can also prove that Module 2 is effective.

Experimental group 3: Verify the effectiveness of Module 3 According to the evaluation results in Table VI, the tree based ensemble learning models LightGbm and XGBoost have significantly better performance than other models, which can significantly verify the effectiveness of Module 3.

3) Hyperparameter optimization: In order to optimize the performance and prediction accuracy of the model, this article has decided to choose the LightGBM model for hyperparameter tuning. The tuning method used is Bayesian tuning, and the results are shown in the table below.

TABLE IX
LIGHTGBM MODEL PERFORMANCE AFTER BAYESIAN TUNING

Metric	Value
RMSE	0.4109 ± 0.0839
R^2	0.9048 ± 0.0415

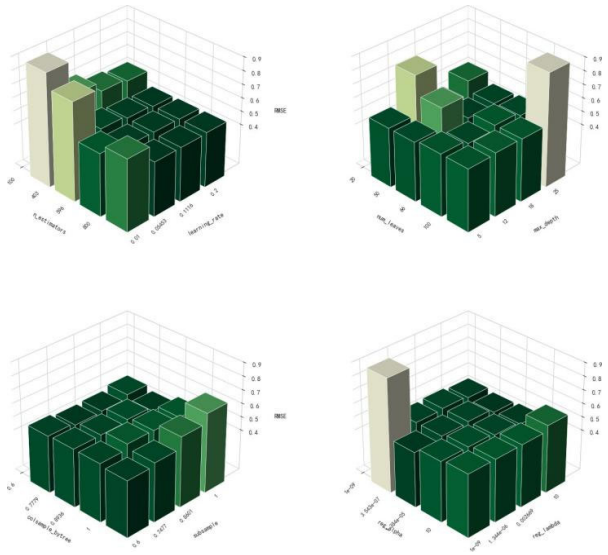

Figure 8. 3D Bar Charts of RMSE for LightGBM Hyperparameter Combina- tions.

From the Figure 8, it can be seen that the RMSE score varies with different hyperparameters, but there is a lowest point, which is the locally optimal hyperparameter

F. Conclusion

This article aims to solve the problem of hourly level prediction of shared bicycle demand by using a framework that combines data processing, feature engineering, feature filtering, tree based model construction, and hyperparameter optimization. Comparative and ablation experiments have been conducted to demonstrate the effectiveness of the frameworks in this module. The framework presented here can serve as the core prediction engine for a real-time demand-supply tracking system, providing operators with a dashboard to monitor and manage station-level demand effectively [33].

References

[1] S. H. Choi and M. K. Han, "The empirical evaluation of models predicting bike sharing demand," in 2020 International Conference on Information and Communication Technology Convergence (ICTC), 2020, pp. 1560–1562.

[2] M. M. Isalm, M. E. Biswas, M. Shahzamal, M. D. Haque, and M. S. Hos- sain, "An effective data driven approach to predict bike rental demand," in 2023 5th International Conference on Sustainable Technologies for Industry 5.0 (STI), 2023, pp. 1–5.

[3] M. J. A. Shanto, R. Akter, D. S. Kim, and T. Jun, "Predicting bike- sharing demand: A machine learning approach for urban mobility analysis," in 2023 14th International Conference on Information and Communication Technology Convergence (ICTC), 2023, pp. 1079–1081.

[4] A. Kealy and J. Wu, "Safety challenges and solutions in bike-sharing systems," in 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS), 2021, pp. 651–656.

[5] A. S. Patel, M. Ojha, M. Rani, A. Khare, O. P. Vyas, and R. Vyas, "Ontology-based multi-agent smart bike sharing system (sbss)," in 2018 IEEE International Conference on Smart Computing (SMARTCOMP), 2018, pp. 417–422.

[6] A. Jaber, B. Csonka, and J. Juha´sz, "Long term time series prediction of bike sharing trips: A cast study of budapest city," in 2022 Smart City Symposium Prague (SCSP), 2022, pp. 1–5.

[7] X. Han, P. Wang, J. Gao, M. Shah, R. K. M. Ambulgekar, A. Jarandikar, and S. Dhar, "Bike sharing data analytics for silicon valley in usa," in 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), 2017, pp. 1–9.

[8] Z. Jiao, H. Jin, B. Liu, and Z. Zhang, "Short-term demand forecasting of shared bicycles driven by big data: A comparative analysis of machine learning models," Journal of Business Economics, vol. 322, no. 8, pp. 16–25, 2018, [In Chinese].

[9] H. Yang, X. Zhang, L. Zhong, S. Li, X. Zhang, and J. Hu, "Short-term demand forecasting for bike sharing system based on machine learning," in 2019 5th International Conference on Transportation Information and Safety (ICTIS), 2019, pp. 1295–1300.

[10] M. H. Almannaa, M. Elhenawy, F. Guo, and H. A. Rakha, "Incremental learning models of bike counts at bike sharing systems," in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 3712–3717.

[11] D. Cao, S. Fan, Y. Zhang, and K. Xia, "Short-term demand forecasting of shared bicycles based on long short-term memory neural network model," Science Technology and Engineering, vol. 20, no. 20, pp. 8344–8349, 2020, [In Chinese].

[12] X. Xing, Z. Yin, and L. Fang, "Shared bicycle demand prediction incorporating multivariate meteorological factors," Intelligent Computer and Applications, vol. 15, no. 1, pp. 178–186, 2025, [In Chinese].

[13] F. Huang, S. Qiao, J. Peng, and B. Guo, "A bimodal gaussian inhomogeneous poisson algorithm for bike number prediction in a bike- sharing system," IEEE Transactions on Intelligent Transportation Sys- tems, vol. 20, no. 8, pp. 2848–2857, 2019.

[14] Y. Li and Y. Zheng, "Citywide bike usage prediction in a bike sharing system," IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 6, pp. 1079–1091, 2020.

[15] S. Feng, H. Chen, C. Du, J. Li, and N. Jing, "A hierarchical demand prediction method with station clustering for bike sharing system," in 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), 2018, pp. 829–836.

[16] M. He, X. Xue, X. Zhang, and C. Zhou, "A bike-sharing demand predicting model with integrating temporal convolutional network and self-attention," in 2021 International Conference on Electronic Information Engineering and Computer Science (EIECS), 2021, pp. 278–281.

[17] H. Yang, S. M. Raza, D. T. Le, D. S. Kim, and H. Choo, "Dual-branch neural networks for predicting shared bikes," in 2023 17th International Conference on Ubiquitous Information Management and Communication (IMCOM), 2023, pp. 1–4.

[18] J. Gu, Q. Zhou, J. Yang, Y. Liu, F. Zhuang, Y. Zhao, and H. Xiong, "Exploiting interpretable patterns for flow prediction in dockless bike sharing systems," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 2, pp. 640–652, 2022.

[19] J. Huang, X. Wang, and H. Sun, "Central station based demand prediction in a bike sharing system," in 2019 20th IEEE International Conference on Mobile Data Management (MDM), 2019, pp. 346–348.

[20] W. Zheng, H. Deng, and F. Han, "Gst-net: A gis-based hybrid prediction model for shared bike traffic flow," in 2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C), 2021, pp. 941–946.

[21] E. Collini, P. Nesi, and G. Pantaleo, "Deep learning for short-term prediction of available bikes on bike-sharing stations," IEEE Access, vol. 9, pp. 124337–124347, 2021.

[22] Y. Du, B. Xiao, W. Xu, D. Cui, Q. Xu, and L. Yan, "Destination prediction for sharing-bikes' trips," in 2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC), 2018, pp. 198–202.

[23] A. K. Das, A. M. Joshi, and S. Dhal, "A machine learning based bike recommendation system catering to user's travel needs," in 2020 IEEE 17th India Council International Conference (INDICON), 2020, pp. 1–6.

[24] M. Jiang, C. Li, K. Li, Z. Yang, and H. Liu, "Interblock flow prediction with relation graph network for cold start on bike-sharing system," IEEE Internet of Things Journal, vol. 9, no. 15, pp. 13390–13404, 2022.

[25] R. Guo, Z. Jiang, J. Huang, J. Tao, C. Wang, J. Li, and L. Chen, "Bikenet: Accurate bike demand prediction using graph neural net- works for station rebalancing," in 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Comput- ing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (Smart-World/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), 2019, pp. 686–693.

[26] R. Qin, L. Kong, M. Guo, B. Yao, and M. Guizani, "Rebalance modern bike sharing system: Spatio-temporal data prediction and path planning for multiple carriers," in 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), 2018, pp. 1081–1086.

[27] J. Chai, J. Song, H. Fan, Y. Xu, L. Zhang, B. Guo, and Y. Xu, "St-bikes: Predicting travel-behaviors of sharing-bikes exploiting urban big data," IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 7, pp. 7676–7686, 2023.

[28] Y. Liang, G. Huang, and Z. Zhao, "Cross-mode knowledge adaptation for bike sharing demand prediction using domain-adversarial graph neural networks," IEEE Transactions on Intelligent Transportation Systems, vol. 25, no. 5, pp. 3642–3653, 2024.

[29] M. Tabandeh, C. Antoniou, and G. Cantelmo, “Long-term & short-term bike sharing demand predictions using contextual data,” in 2023 8th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2023, pp. 1–6.

[30] Y. Zhou and Y. Huang, “Place representation based bike demand prediction,” in 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 1577–1586.

[31] S. S. Chawathe, “Mining bike-share data,” in 2020 IEEE International Smart Cities Conference (ISC2), 2020, pp. 1–8.

[32] M. Bencekri, A. Founoun, A. Haqiq, and A. Hayar, “Investigation of shared-bike demand using data analytics,” in 2022 IEEE International Smart Cities Conference (ISC2), 2022, pp. 1–4.

[33] A. A. Ramesh, S. P. Nagisetti, N. Sridhar, K. Avery, and D. Bein, “Station-level demand prediction for bike-sharing system,” in 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), 2021, pp. 0916–0921.