electronics

MDPI

Article

Short-Term Demand Prediction of Shared Bikes Based on LSTM Network

Yi Shi [1,*], Liumei Zhang [1], Shengnan Lu [1] and Qiao Liu [2]

[1] College of Computer Science, Xi'an Shiyou University, Xi'an 710065, China
[2] School of Cyber Engineering, Xidian University, Xi'an 710071, China
[*] Correspondence: 200301@xsyu.edu.cn

Abstract: Shared transportation is widely used in current urban traffic. As a representative mode of transport, shared bikes have strong mobility and timeliness, so it is particularly critical to accurately predict the number of bikes used in an area every hour. In this paper, London bike-sharing data are selected as a data set to primarily analyze the impact of meteorological elements and time factors on bike-sharing demand. At the same time, it is important to use LSTM neural network models and popular machine learning models to predict demand for shared bikes at an hourly level. Through data analysis and visualization, the major elements affecting the bike-sharing demand are found to include humidity, peak hours, temperature, and other elements. The root mean squared error of the LSTM model is 314.17, the R^2 score is as high as 0.922, and the error is small in comparison to other machine learning models. Through the evaluation indicators, it can be seen that the LSTM model has the smallest error between the prediction results and the true values of the compared machine learning methods, and the change trend of the model prediction result curve is basically consistent with the actual result curve.

Keywords: shared bikes; transport; LSTM network; demand forecast; data analysis

check for updates

Citation: Shi, Y.; Zhang, L.; Lu, S.; Liu, Q. Short-Term Demand Prediction of Shared Bikes Based on LSTM Network. *Electronics* **2023**, *12*, 1381. https://doi.org/10.3390/electronics12061381

Academic Editor: Felipe Jiménez

Received: 11 January 2023
Revised: 26 February 2023
Accepted: 6 March 2023
Published: 14 March 2023

1. Introduction

With the increasing attention on environmental protection in different countries, environmentally friendly, pollution-free, and energy-saving transportation methods have been widely used and developed; shared transportation is one such method. Within the field of shared transportation, shared bikes are a popular mode of transportation in different cities that effectively solves the problem of commuting the last kilometer of a trip [1,2]. Originating in the Netherlands, the bike-sharing model became popular around the world in the mid-1990s. According to statistics, at least 535 cities in 49 countries have set up bike-sharing schemes. Therefore, it is an urgent problem to quickly and precisely predict the bike-sharing demand at a specific time and place by analyzing the data to reasonably determine the number of required bikes and plan their arrangement [3–5].

At present, the research status of bike-sharing demand forecasting is as follows. It was indicated by Matton et al. that meteorological conditions such as temperature, wind speed, and precipitation are the major components affecting the demand for shared bikes [6]. Faghih et al. suggested that time-related factors, including weekends, weekdays and peak hours, are the key variables affecting bike-sharing demand [7]. Bacciu et al. applied an SVM model and a random forest model in traditional machine learning algorithms for bike-sharing prediction but did not describe in detail how to predict bicycle usage changes over a short period of time [8]. Bajari et al. used several popular machine learning models for demand forecasting, among which the random forest model and stacking model obtained better prediction results. The ordinary least square linear model, the binary classification model, and the multi-classification logit model are the main methods applied to bike-sharing demand forecasting. These empirical models require a large number of observations and have obvious limitations, and the regression relationships generated do not match the actual demand situation well [9]. Cao et al. used support vector machine, extreme random tree, random forest, and other methods to perform short-term demand prediction of shared bicycles, and the extreme random tree obtained better results [10]. To predict the hourly demand of bike sharing more accurately, the trend in algorithm usage has shifted away from machine learning algorithms toward deep learning algorithms. The BP neural network has good adaptability; this network can solve the nonlinear problem by learning the data set itself, but it has a drawback in that the trained model is prone to local extremes when making predictions [11]. Gao X et al. chose rental data in the Washington area of the United States as a research object and proposed a time-based model and a hybrid method combining a backpropagation network and a genetic fuzzy c-means algorithm. The genetic algorithm provided better classification performance for the training data. The trained backpropagation network predictor was used to predict rental demand in the future [12]. Recurrent neural networks (RNN) are mainly applied to process sequence type data and are now widely used to process continuous sequence data for prediction [13,14]. However, due to imperfect processing of historical information, problems such as gradient disappearance and gradient explosion can easily occur in practical applications [15]. To overcome the shortcomings of the RNN model, the long short-term memory (LSTM) model emerged, and has been widely used in the study of sequence prediction problems so far and has achieved good prediction results [16–19].

Compared with previous forecasting methods, the innovation of the demand forecasting method used in this paper is as follows. (1) In the aforementioned related work, only one method is mainly used to predict demand for bicycles. In this paper, multiple machine learning methods and LSTM deep learning methods are used to predict demand, and the prediction effects and associated indicators of different methods are compared. (2) Compared with other related works, this paper introduces big data machine learning and deep learning methods into the "hourly" short-term demand prediction of the bike-sharing industry, which improves the prediction efficiency of the industry's immediate demand so as to assist enterprises in real-time scheduling and enhance the overall utilization level of bike resources.

The paper introduces an LSTM network algorithm to the shared transportation industry to predict the demand of bicycle sharing at the "hourly" level, improve the efficiency of immediate demand prediction, and help companies plan in real-time and determine the overall utilization of bicycle resources.

The rest of paper is organized as follows. In Section 2, the principles of the proposed method, data prepossessing, and influencing factors analysis are introduced. The details of the proposed model and the results are introduced in Section 3. In Section 4, future research directions are mainly discussed. Finally, in the conclusions section, the main significance of this study is expounded.

2. Theory and Methods

In general, public transportation methods exhibit a strong time dependency. Bike sharing is an important part of public transportation, and the short-time demand for bike sharing is also strongly influenced by the time. Usually, this demand pattern varies over time and has a certain pattern. Similarly, weather factor conditions have a significant impact on the short-time demand of shared bikes [20–22]. Therefore, in order to make bike-sharing planning timelier and more accurate, the following prediction model is applied in this paper.

2.1. Predictive Models Based on Machine Learning

2.1.1. XGBoost

XGBoost is a machine learning algorithm implemented in a gradient boosting framework and is efficient, flexible, and portable. The XGBoost algorithm is a combination of several known decision tree models with low prediction accuracy levels that combine to build a model with higher accuracy. The model is improved iteratively, with each iteration generating a tree that fits the residuals of the previous tree. For regression problems, XGBoost is an efficient implementation of the gradient boosting algorithm. It balances speed and efficiency, and this algorithm has shown excellent performance in major competition tasks since its release in 2015 [23].

2.1.2. Bagging

The full name of bagging is bootstrap aggregation, where bootstrap means that the training samples of the base learner are obtained by bootstrap sampling of the original data. For regression problems, aggregation means using simple averaging to obtain the final model output by arithmetically averaging the regression results obtained from the base learner. This algorithm is the most famous representative of parallel integrated learning methods. Repolho et al. used the bagging method algorithm in air transport demand forecasting and obtained better results compared to SARIMA and other methods [24].

2.1.3. Random Forest

Random forest is a model that uses multiple CART trees to train the samples of the training set and then performs regression prediction on the samples of the test set. It consists of several decision trees, and these decision trees are not related to each other. When using the random forest model to predict the demand for shared bicycles, each decision tree in the random forest appears each time a new sample is input, one regression prediction is made for each sample, and finally, all regression results are averaged to create the final model prediction [25,26].

2.1.4. Light Gradient Boosting (LightGBM)

LightGBM is an evolutionary version of the GBDT model [27]. LightGBM similar to XGBoost is in principle, but it is faster to train, consumes less memory, supports parallel learning, and can handle large amounts of data. The basic idea of LightGBM is to first discretize continuous floating-point feature values into k integers while constructing a histogram of width k. When constructing a decision tree for selecting features and feature splitting points, it is necessary to iterate through k discrete values to find the best splitting point. More importantly, LightGBM sets a maximum depth limit for the leaf growth strategy to keep the model efficient while preventing overfitting. Lu et al. applied this algorithm to obtain good results in dynamic bike-sharing distribution prediction [28,29].

2.1.5. Stacking Model

Stacking is the most popular method in the field of model fusion in recent years. It is not only one of the most common fusion methods used by competition champion teams, but also one of the solutions considered for practical applications of artificial intelligence in industry. As a strong learner fusion method, stacking combines the three advantages of good model effect, strong interpretation ability, and applicability to complex data and is one of the most practical pioneer methods in the field of fusion. According to Wolpert, model stacking can derive the bias in the model on a specific data set so that the bias in the meta-learner can be corrected [30].

2.2. Predictive Models Based on LSTM

LSTM was first described by Hochreiter et al. It was developed in 1997 and based on improvements to RNN networks [31]. The difference between LSTM neural networks and RNN is that it not only adds a storage unit that preserves previous information, but it also trains the data through a back propagation algorithm, eliminating the problem of the disappearance of the gradient, and the loss of long-term dependence in RNN networks is effectively mitigated [13]. LSTM networks are widely used in various fields such as machine translation, text generation, speech recognition, etc. They can also be applied to regression prediction and have good prediction results [18,19,32–34].

The LSTM model works mainly with a gating mechanism. It contains a memory cell and three control gates: input, output, and forget gates. The cell state is equivalent to the path that can transmit relevant information so that information can be passed down the sequence chain, and this part can be regarded as the ”memory” of the network. The input gate is used to update the cell state. The forget gate determines what information should be discarded or retained. The output gate determines the value of the next hidden state, which contains information about the previous input. In this process, LSTM uses a sigmoid function to determine which data to forget and which data to keep. The output of the sigmoid function is (0,1); when the output is 0, this part of the information is forgotten because any number multiplied by 0 is 0. Correspondingly, when the output of the sigmoid function is 1, the information is completely preserved, because any number multiplied by 1 is the same value. The network structure is shown in Figure 1.

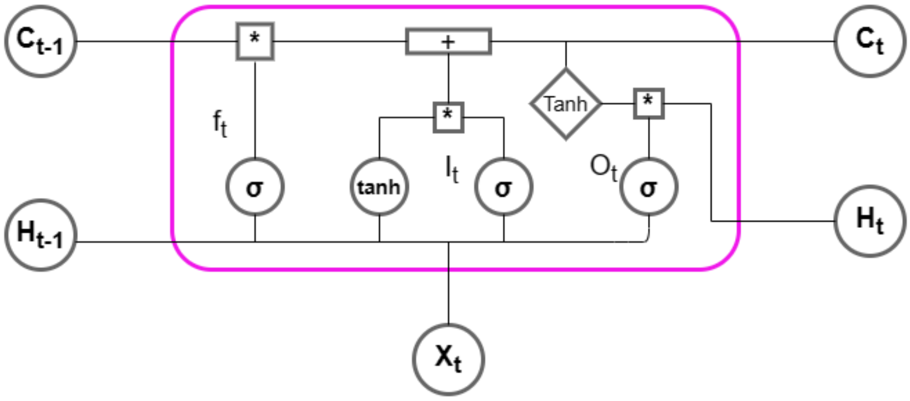

Figure 1. LSTM network structure.

In Figure 1, X_t is the input at time t, H_t is the state value of the cell at time t, and the small box with tanh is a feedforward network layer with activation function tanh. Gates are a way of conditionally letting information through, and they are accomplished through a sigmoid layer and dot multiplication operations. LSTM cells generally output two types of states to the next cell: the cell state and the hidden state. The individual gates of the LSTM work as follows. The first step in LSTM is to decide what information to discard from the unit state. The forget layer has two inputs, H_{t-1} and X_t, where H_{t-1} is the hidden state of the previous cell, and X_t is the input of the current time step. The calculation process of the forgetting layer is as follows. The value of the input gate and the state value of the input cell at time t are first calculated by the formula:

$$f_t = \delta[W_f(X_t, H_{t-1}) + b_f] \tag{1}$$

The next step is to decide on the new information to be stored in the cell state. First, the sigmoid layer of the input gate layer decides which values will be updated.

$$I_t = \delta[W_i(X_t, H_{t-1}) + b_i] \tag{2}$$

Then, a tanh layer creates a new candidate vector $\bar{C}_t$ that can be added to the state.

$$\bar{C}_t = tanh[W_c(X_t, H_{t-1}) + b_c] \tag{3}$$

Next, the cell state is updated in conjunction with the creation of these two gates, updating the old cell state C_{t-1} to the new cell state C_t.

$$C_t = I_t\bar{C}_t + f_tC_{t-1} \tag{4}$$

Last, the information to be output is finalized. The output will be based on the unit state but is actually a filtered version.

$$O_t = \delta[W_o(X_t, H_{t-1}) + b_o] \tag{5}$$

$$H_t = O_t tanh C_t \tag{6}$$

2.3. *Experiment Process*

The overall process of short-term demand forecasting for shared bicycles is shown in Figure 2. After loading the London shared bike data set, data pre-processing is performed on its database and feature engineering is performed. After that, the prediction models are built and the effects of the prediction models are compared.

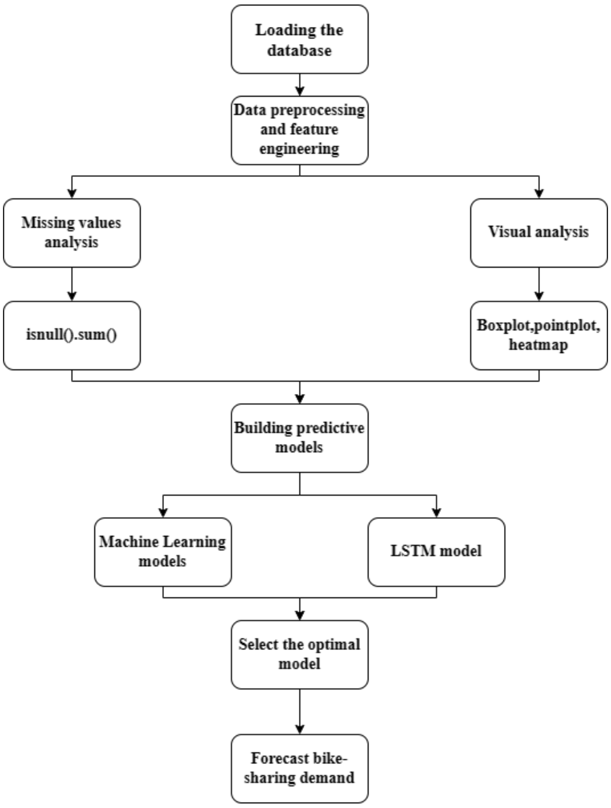

Figure 2. Flowchart of the experiment.

2.3.1. Experimental Environment

This project is conducted on PC with Intel(R) Core(TM) i7-10510U CPU @ 1.80 GHz 2.30 GHz and 16 GB memory and Windows 10 system. Anaconda Navigator3 (Jupyter notebook) and Python 3.8 served as experimental platform for simulation experiments. The integrated development environment (IDE) is Jupyter notebook and python libraries including Sklearn (1.1.3), Tensorflow (2.9.1), and Keras (2.9.0) are used to implement all the algorithmic models.

2.3.2. Acquisition and Introduction of Experimental Data Sets

The data set is from the website Kaggle and contains data on bike-sharing rides in the city of London. The data are hourly data from 4 January 2015 to 3 January 2017 (a total of 24 months), and the data set contains 17,414 data items. As can be seen from Table 1, the properties of the data set include date, hour, holiday, weekend, season, real temperature, apparent air temperature, wind speed, and weather. Additionally, the holiday and weekend values are represented by boolean fields.

Table 1. Properties description of data set.

Properties	Description and Value Range
timestamp	timestamp field for grouping the data [4/1/2015/00:00:00, 3/1/2017/23:00:00]
cnt	the count of bike shares [0, 7860]
t1	real temperature, unit: °C [−1.5, 34.0]
t2	apparent air temperature, unit: °C [−6.0, 34.0]
hum	humidity in percentage [20.5, 100.0]
windspeed	wind speed, unit: km/h [0.0, 56.5]
isholiday	0 = non holiday 1 = holiday
isweekend	0 = working day 1 = weekend
season	Seasonal Category 0 = spring 1 = summer 2 = fall 3 = winter
weathercode	Weather category 1 = clear/mostly clear but have some values with haze/fog/patches of fog/fog in vicinity 2 = scattered clouds/few clouds 3 = broken clouds 4 = cloudy 7 = rain/light rain shower/light rain 10 = rain with thunderstorm 26 = snowfall 94 = freezing fog

2.3.3. Experimental Data Preprocessing

It is necessary to detect if there are missing values in the received data set, which can be detected by using isnull().sum() in the pandas library. The isnull().sum() function is used to count the number of missing values. By inspection, there are no missing values in this data set.

From the box plot in Figure 3, the number of shared bikes used is mainly concentrated in the range of 0 to 2000. There are 17,414 entries in this data set. The data set is split into a training set (70%), a validation set (10%), and a test set (20%). The validation set is taken from the training set and set with validation_split = 0.1 before training.

To make training the model more simple, the continuous variables such as temperature, humidity, and wind speed were normalized to [0, 1] through the minimum–maximum normalization method. The conversion factors were also saved so that the converted data features can be recovered after the subsequent completion of prediction. The formula is as follows:

$$X_{ij} = \frac{x_{ij} - \underset{1 \le i \le N}{min}\, x_{ij}}{\underset{1 \le i \le N}{max}\, x_{ij} - \underset{1 \le i \le N}{min}\, x_{ij}} \tag{7}$$

where max is the maximum value in the feature data, and similarly, min represents the minimum value in the feature data. x_{ij} is the original data. X_{ij} is the normalized data.

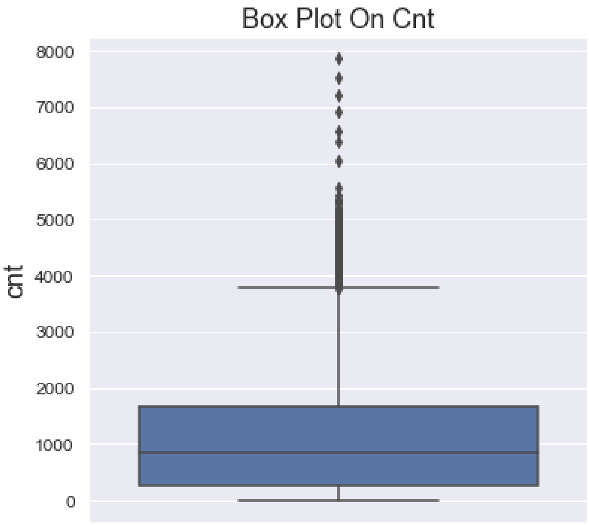

Figure 3. Target variable cnt.

2.3.4. Analysis of the Influencing Factors

Bike sharing is a mode of transport that is heavily influenced by meteorological factors [20]. As can be seen from Figure 4, when the real temperature is greater than 10 degrees, the temperature change has little effect on shared bike usage. When the actual temperature is below 10 degrees, there is a significant reduction in shared bike usage. The effect of apparent air temperature on the shared bike usage is consistent with the effect of actual temperature. When apparent air temperature is greater than 10 degrees, the temperature change has little effect on the shared bike usage. When the apparent air temperature is lower than 10 degrees, the shared usage is significantly reduced. Overall, the influence of humidity on the use of shared bikes is not very large, and use is reduced when humidity is higher than 90%, probably because it is already raining when humidity reaches 90%. The effect on bike-sharing usage was also not significant when the wind speed was in the range of 0 km/h to 40 km/h, but when the wind speed approached 40 km/h and above, bike-sharing usage decreased significantly.

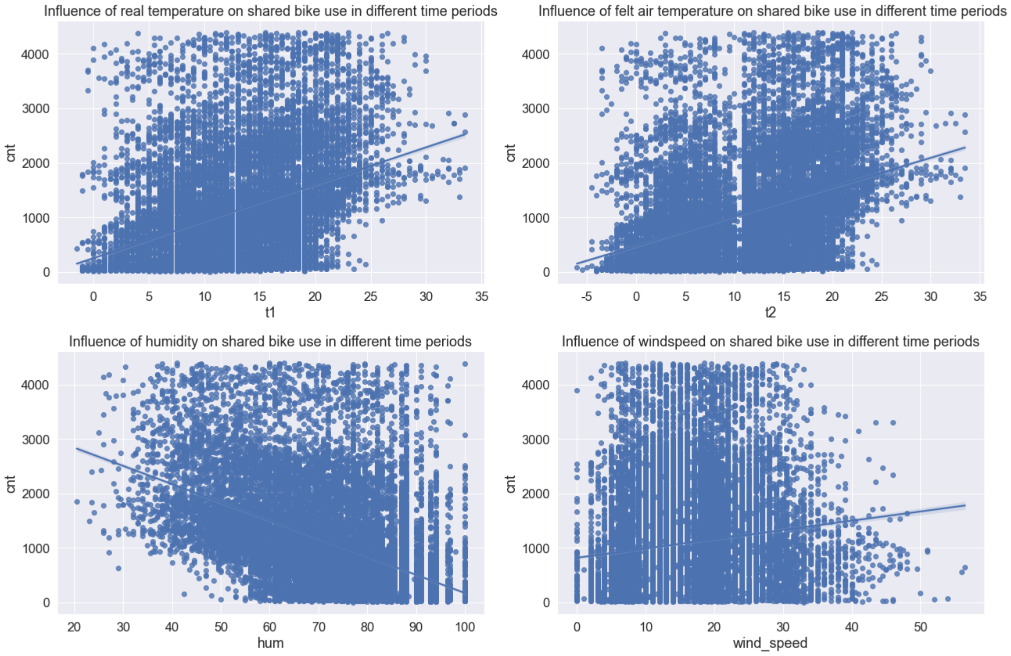

Figure 4. The influence of meteorological factors on the short-time demand of shared bikes.

As shown in Figure 5, the heat distribution plot shows the correlation matrix between the total bike-share demand and the four meteorological factors in London from January 2015 to January 2017. From Figure 5, we can see that there is a correlation between demand for shared bikes and four meteorological factors. In addition, there is a strong positive correlation between both real and apparent air temperature and the demand for bike sharing, with cold suppressing the demand for bike sharing. The correlation coefficient between 't1' and 'cnt' and the correlation coefficient between 't2' and 'cnt' is very close, which are 0.39 and 0.37, respectively. Humidity is negatively correlated with the number of rental bikes, which is −0.46. Rain and snow dampened demand for rental bikes. The correlation between the demand for shared bikes and temperature and humidity is the highest. The demand for shared bikes has little correlation with wind speed and weather, which have correlation values of 0.12 and −0.17, respectively. Based on Figures 4 and 5, there is a strong correlation between real temperature and apparent air temperature. Including both features in the model will cause multicollinearity issues; therefore, it is required to removed on of the features. In this paper, the apparent air temperature feature is removed because the correlation between it and the count is weaker than that of the real temperature.

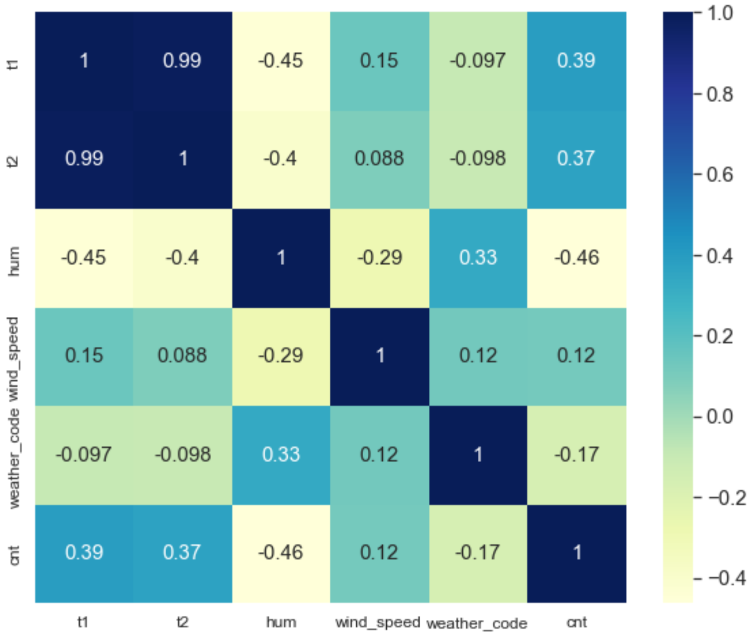

Figure 5. Correlation analysis figure between meteorological factors and demand for shared bikes.

Time of day has a large impact on the short-term demand for bike sharing, which includes not only long-term time of day such as months, weekends, and weekdays but also includes short-term time of day such as different times of the day. Data from the bike-sharing project in London, UK, were analyzed regularly, and the results are shown in Figure 6.

As can be seen from Figure 6, the demand for shared bikes gradually increases between January and July, reaching a peak in July followed by a slow decline from July to October and a sharp drop after October, which is clearly related to the seasons. After analyzing the data in the Figure 6, the month feature also has a more obvious influence on the number of demand for shared bikes. Because the month and season features have a consistent impact on demand, and the month feature is more detailed, the month feature was retained and the season feature was deleted.

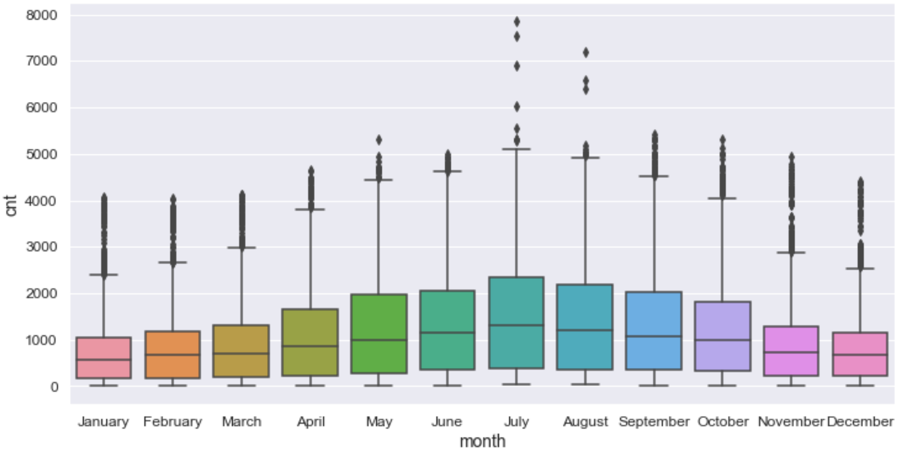

Figure 6. Average shared bike usage per mouth.

As can be seen from Figure 7, the shared bike usage level is relatively large from Monday to Friday, and there are two peak periods every day, which are at 7:00–8:00 and 17:00–18:00; these are the peak traffic times on weekdays. In addition, during the weekend, the demand for shared bikes is higher during the hours of 12:00–4:00, so it can be seen that the time of day is a significant factor influencing the demand for shared bikes. The trend in demand for shared bikes on different days of the week shows that whether it is a weekend or not also has an impact on the number of shared bikes used.

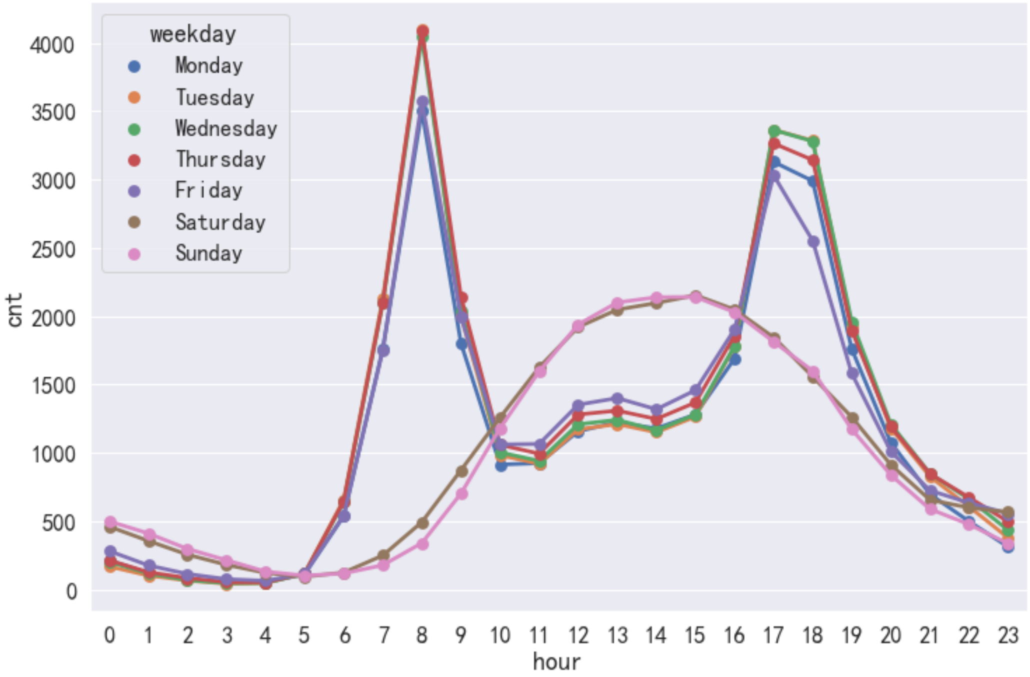

Figure 7. Shared bike usage per hour of weekday.

2.3.5. Predictive Model Evaluation Metrics

Through an in-depth analysis of the target research problem, the short-time demand forecasting problem of bike-sharing can be considered as a regression problem. Therefore, the root mean squared error (RMSE), mean squared error (MSE), and R^2 score can be used to evaluate the prediction model. The MSE calculates the sum of squares of the prediction residuals, and a smaller value indicates better prediction of the model. Mean squared error is sensitive to discrete points because it has a larger penalty for large error samples, but the outliers have been deleted in the preprocessing stage, so it has little effect. RMSE takes the square root such that the vector stiffness and y are equal. R^2 score takes into account both the difference between the predicted and true values and the difference between the true values of the problem itself. Its maximum value is 1, but it can also be negative. A value of 0 indicates that the model is approximately as good as a random estimate. A value of 1 means the model is error free, and a value closer to 1 means the model is better. The calculation formula is as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \tag{8}$$

$$SS_{res} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{9}$$

$$SS_{tot} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2 \tag{10}$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{11}$$

where y_i and $\hat{y}_i$ are the actual value and the predicted value, respectively, and N is the number of data items. In this paper, we mainly refer to the RMSE and R^2 score value to train the neural network and also refer to the changes in the MSE values.

3. Predictive Model Analysis

3.1. Model Structure

Regarding the construction of machine learning models, this paper uses the python(3.8) sklearn library to complete the construction of decision tree regression, linear regression, kernel ridge regression, support vector regression, extra tree regressor, Adaboost, gradient boosting, XGBoost, light gradient boosting, random forest, bagging, etc. Among them, linear regression, kernel ridge regression, and support vector regression prediction results are not ideal and the R^2 score is less than 0.5, so no parameter description is given. On the other hand, XGBoost, bagging, random forest, and light gradient boosting have good prediction results, so they are used as the first layer of stacking model. The second layer of the stacking model is set as gradient boosting regressor.

The first step is to build the CPU version of the Tensorflow(2.9.1) and Keras(2.9.0) framework in a Windows 10 operating environment. This project uses the "tf.keras.layer.LSTM" module provided by Keras, a deep learning framework, to complete the construction of LSTM models. This module encapsulates the basic structure of LSTM in Keras, which mainly contains three gate structures: the input gate, the forget gate, and the output gate structures. The dropout mechanism is used so that reducing the weights makes the network more robust to the loss of specific neuronal connections and avoids overfitting. Previously, the process of building deep learning models was laid out in terms of neural network nodes, and the current process of building LSTM framework using Keras is to build the model in terms of network layers. The LSTM class in the Keras deep learning framework contains multiple input and output layers of network nodes, which are represented by vectors. In simple terms, the number of nodes in each layer is the length of the vector.

This paper uses the LSTM neural network model to predict the hourly demand for bike-sharing in London. The prediction of hourly bike demand was accomplished using the deep learning framework Tensorflow and the LSTM class module in Keras. Multiple network layers are linearly stacked by "add" functions. The final network model is continuously debugged to determine the following network structure parameters.

1. Layer number settings: build the LSTM model structure, set the number of LSTM layers to 4 and the feature size to 12, the input of LSTM is the [time_steps, feature], and the output layer is 1.
2. Model parameter settings: when debugging the LSTM neural network, we tried to change the batch size of training samples, the number of neurons, and the step size.

The batch sizes were set to 32, 64, and 128; the time_steps were set to 10, 12, 24; and the numbers of neurons were set to 32, 64, and 128. Its results with respect to these parameters are given in the following tables. The learning_rate is set to 0.0005, the optimizer chooses Adam, the loss function is set as mse, and the LSTM model is trained for 100 rounds (epochs). In order to prevent overfitting in the training process, the dropout of each layer is set to 0.2. For the activation function, the ReLU activation function is chosen. The main purpose is to reduce the interdependence of the parameters and alleviate the overfitting problem.

3. Dimension transformation: when inputting the features into the prediction model, the tensor needs to be transformed into a two-dimensional matrix to use its computed results as inputs to the hidden layer. Finally, the tensor is transformed into three dimensions as the input to the LSTM class. In addition, batch processing of data is performed via the get_batches function.

3.2. Model Prediction Results

3.2.1. Prediction Results of LSTM Neural Network Model

According to Tables 2–4, there are three main parameters, time_step, batch_size, and number of neurons, that have impacts on the LSTM model performance. As can be seen from Table 2, when the batch size is 64 and the number of neurons is 64, the optimal result of RMSE is 329.49 and the R^2 score is 0.915. The optimal result, as shown in Table 3, indicates that RMSE is 336.37 and the R^2 score is 0.911 with a time_step of 12, a batch size of 64, and 64 neurons. In the below Tables 2–4, a comparison of the LSTM model performance with different parameters is presented. An RMSE of 314.17 and a R^2 score of 0.922 are the optimal overall experimental results.

Table 2. Experimental results (time_step = 10).

Batch_Size	Number of Neurons	RMSE	R^2 Score
32	32	335.68	0.912
	64	345.73	0.906
	128	357.58	0.899
64	32	373.94	0.890
	64	**329.49**	**0.915**
	128	354.90	0.901
128	32	332.15	0.914
	64	342.07	0.908
	128	342.03	0.907

Table 3. Experimental results (time_step = 12).

Batch_Size	Number of Neurons	RMSE	R^2 Score
32	32	352.02	0.903
	64	355.72	0.901
	128	350.33	0.904
64	32	344.29	0.907
	64	**336.37**	**0.911**
	128	361.55	0.897
128	32	353.83	0.902
	64	368.25	0.892
	128	373.38	0.891

In addition, the trained model was used to test the 1000 h data, the predicted value of the model was compared with the true value, and the comparison results are shown in Figure 8. Figure 8 shows that the LSTM model has a good prediction performed. The trend of the shared bicycle usage curve predicted by the model is basically consistent with the real usage curve, and there are errors only in some peak parts. In general, the fitting effect of the model is very good, which meets the empirical error requirements in the regression prediction process. It is further illustrated that the short-time demand forecasting problem of bike-sharing can be solved by using LSTM forecasting model.

Table 4. Experimental results (time_step = 24).

Batch_Size	Number of Neurons	RMSE	R^2 Score
32	32	353.75	0.902
	64	**314.17**	**0.922**
	128	333.82	0.912
64	32	337.36	0.911
	64	348.93	0.904
	128	367.07	0.894
128	32	328.38	0.915
	64	334.48	0.912
	128	359.23	0.899

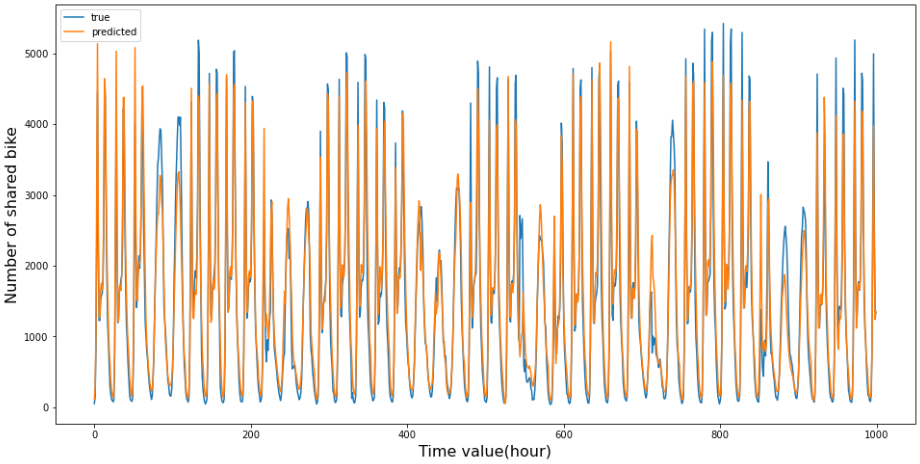

Figure 8. The prediction results of LSTM model.

Figure 9 displays the variation of the loss function on the training and validation sets during the training period of the LSTM model. During training, the "validation_split" of the module is set to 0.1. This means that 10% of the data in the training set are validation data. Furthermore, for each completed round of training, a round of validation is performed on the validation set. From Figure 9, we can see that the value of the loss function of the model on the training and validation sets gradually decreases with the increase in the number of training rounds and stabilizes after 40 rounds (epochs), where the value is close to 0.

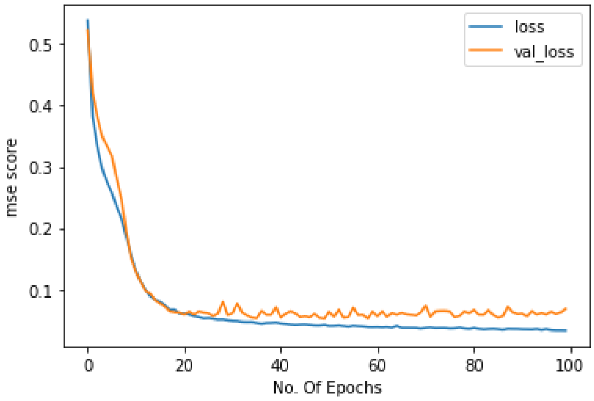

Figure 9. Train set and validation set loss curve.

3.2.2. Predictive Model Comparison

The RMSE and R^2 score obtained by the above eight models after fitting all the data are displayed in Table 5. According performance indicator data of each model in the table, it can be seen that the smallest RMSE value and the highest R^2 score are generated when using the LSTM model for prediction, with values of 314.17 (RMSE) and 0.922 (R^2 score). The OLS model has the worst performance, the lowest R^2 score, and the largest RMSE value. In addition, the fusion model achieves good prediction results in terms of performance metrics, because it incorporates several machine learning models with good prediction results, but there is still a small gap in prediction error compared to the LSTM model. Therefore, the LSTM model is a better choice for short-time demand prediction of shared bikes.

Table 5. Comparison of prediction results of different forecasting models.

Predictive Model	LSTM	Stacking Model	Light GBM	Random Forest	Bagging	XGBoost	Extra Tree Regressor	OLS Model
RMSE	**314.17**	351.47	356.57	358.13	366.13	367.19	487.95	881.62
R^2 score	**0.922**	0.857	0.853	0.805	0.805	0.843	0.724	0.099

4. Conclusions

In the short-term demand forecasting of shared transportation, to address the problem of effective prediction of the hourly demand of shared bikes in a region, this paper uses a public data set of shared bikes in London and analyzes the impact of each of its characteristic variables on the total demand of shared bikes. Finally, multiple prediction models are used to predict the hourly demand of bike sharing in London and compare the prediction results. The following conclusions are obtained through experiments.

The main factors affecting the demand for shared bikes are temperature, public holidays, seasons, morning and evening rush hours, etc. The most important factors are temperature and morning and evening peak hours (7:00–8:00 and 17:00–18:00).

Compared with machine learning models, the LSTM model has the minimum RMSE of 314.17, the highest R^2 score (0.922), the prediction error is small, the change curve of the prediction result is basically consistent with the real result, and only some of the extreme value regions have error. This model is suitable for short-term demand forecasting of shared bikes. The use of LSTM neural network models does not differ much from machine learning algorithms in terms of algorithm running speed.

Therefore, the LSTM neural network model can be used in the actual city bike-sharing service to predict the demand for bike sharing at the hourly level to assist bike scheduling and better serve users.

In further research, we need to explore other relevant factors such as the locations of buses and subway stations and population distribution. By analyzing these factors and optimizing the short-term demand prediction model of shared bikes, the scheduling of shared bikes can be executed more effectively. Finally, it provides a more effective implementation scheme for scientific planning of public transport.

Author Contributions: Conceptualization, Q.L.; investigation, L.Z.; methodology, Y.S.; software, L.Z.; visualization, S.L.; writing—original draft, Y.S.; writing— review and editing, Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant 61902297; the Fundamental Research Funds for the Central Universities Grant No. XJS211509; Open Project Funds of Shaanxi Key Laboratory for Network Computing and Security Technology (NCST2021YB-04).

Data Availability Statement: Not applicable.

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

References

1. Wang, S.; Zhang, J.; Liu, L.; Duan, Z.-Y. Bike-Sharing-A new public transportation mode: State of the practice and prospects. In Proceedings of the IEEE International Conference on Emergency, Beijing, China, 8–10 August 2010.
2. Jiang, H.; Song, S.; Zou, X.; Lu, L. *How Bike-Sharing Affects Cities*; World Resources Institute (WRI): Washington, DC, USA, 2020.
3. Campbell, A.A.; Cherry, C.R.; Ryerson, M.S.; Yang, X. Factors influencing the choice of shared bicycles and shared electric bikes in Beijing. *Transp. Res. Part C Emerg. Technol.* **2016**, *67*, 399–414. [CrossRef]
4. Yoshida, N.; Ye, W. Commuting travel behavior focusing on the role of shared transportation in the wake of the COVID-19 pandemic and the Tokyo Olympics. *TIATSS Res.* **2021**, *45*, 405–416. [CrossRef]
5. Peng, Y.; Liang, T.; Yang, Y.; Yin, H.; Li, P.; Deng, J. A Key Node Optimization Scheme for Public Bicycles Based on Wavefront Theory. *Int. J. Artif. Intell. Tools* **2020**, *29*, 2040016. [CrossRef]
6. Mattson, J.; Godavarthy, R. Bike Share in Fargo, North Dakota: Keys to Success and Factors Affecting Ridership. *Sustain. Cities Soc.* **2017**, *34*, 172–182. [CrossRef]
7. Faghih-Imani, A.; Eluru, N.; El-Geneidy, A.M.; Rabbat, M.; Haq, U. How land-use and urban form impact bicycle flows: Evidence from the bicycle-sharing system (BIXI) in Montreal. *J. Transp. Geogr.* **2014**, *41*, 306–314. [CrossRef]
8. Bacciu, D.; Carta, A.; Gnesi, S.; Semini, L. An experience in using machine learning for short-term predictions in smart transportation systems. *J. Log. Algebr. Methods Program.* **2016**, *87*, 52–66. [CrossRef]
9. Ryan; Stephen, P.; Bajari; Patrick. Machine Learning Methods for Demand Estimation. *Am. Econ. Rev.* **2015**, *105*, 481–485. [CrossRef]
10. Cao, D.D.; Fan, S.R.; Xia, K.W. Comparison of machine learning methods for short-term demand forecasting of shared bicycles. *Comput. Simul.* **2021**, *38*, 92–97.
11. Liu, M.; Shi, J. A cellular automata traffic flow model combined with a BP neural network based microscopic lane changing decision model. *J. Intell. Transp. Syst.* **2019**, *23*, 309–318. [CrossRef]
12. Gao, X.; Lee, G.M. Moment, based rental prediction for bicycle-sharing transportation systems using a hybrid genetic algorithm and machine learning. *Comput. Ind. Eng.* **2019**, *128*, 60–69. [CrossRef]
13. Connor, J.; Martin, R.; Atlas, L. Recurrent neural networks and robust time series prediction. *IEEE Trans. Neural Netw.* **1994**, *5*, 240–254. [CrossRef]
14. Qiu, X.; Ren, Y.; Suganthan, P.N.; Amaratunga, G.A.J. Empirical mode decomposition based ensemble deep learning for load demand time series forecasting. *Appl. Soft Comput.* **2017**, *54*, 246–255. [CrossRef]
15. Tian, Y.; Zhang, K.; Li, J.; Lin, X.; Yang, B. LSTM-based traffic flow prediction with missing data. *Neuro Comput.* **2018**, *318*, 297–305. [CrossRef]
16. Liu, M.; Shi, J. Short-Term Traffic Flow Prediction Based on KNN-LSTM. In Proceedings of the 2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems, Xi'an, China, 7–8 November 2021.
17. Viadinugroho, R.A.A.; Rosadi, D. Long Short-Term Memory Neural Network Model for Time Series Forecasting: Case Study of Forecasting IHSG during COVID-19 Outbreak. *J. Phys. Conf. Ser.* **2021**, *1863*, 012016. [CrossRef]
18. Ma, X.; Tao, Z.; Wang, Y.; Yu, H.; Wang, Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp. Res. Part Emerg. Technol.* **2015**, *54*, 187–197. [CrossRef]
19. Sundermeyer, M.; Schlüter, R.; Ney, H. LSTM Neural Networks for Language Modeling. In Proceedings of the Interspeech 2012, ISCA's 13th Annual Conference, Portland, OR, USA, 9–13 September 2012.
20. Wang, X.; Sun, H.; Zhang, S.; Lv, Y.; Li, T. Bike sharing rebalancing problem with variable demand. *Phys. A Stat. Mech. Its Appl.* **2022**, *591*, 1266–1267. [CrossRef]
21. Zhang, Y.; Mi, Z. Environmental benefits of bike sharing: A big data-based analysis. *Appl. Energy* **2018**, *220*, 296–301. [CrossRef]
22. Zhang, D.; Yu, C.; Desai, J.; Lau, H.Y.K.; Srivathsan, S. A time-space network flow approach to dynamic repositioning in bicycle sharing systems. *Transp. Res. Part B Methodol.* **2017**, *103*, 188–207. [CrossRef]
23. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016.
24. Dantas, T.M.; Oliveira, F.L.C.; Repolho, H.M.V. Air transportation demand forecast through Bagging Holt Winters methods. *J. Air Transp. Manag.* **2017**, *59*, 116–123. [CrossRef]
25. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
26. Zhang, L.; Suganthan, P.N. Random Forests with ensemble of feature spaces. *Pattern Recognit.* **2014**, *47*, 3429–3437. [CrossRef]
27. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Lin, T.-Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Neural Inf. Process. Syst.* **2017**, *30*, 52.
28. Han, L.; Yu, C.; Chen, Y.; Tang, X. Shared bicycle dynamic distribution model based on Boosting algorithm. In Proceedings of the 2019 Chinese Control and Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 3972–3977.
29. Li, L.; Lin, Y.; Yu, D.; Liu, Z.; Gao, Y.; Qiao, J. A Multi-Organ Fusion and LightGBM Based Radiomics Algorithm for High-Risk Esophageal Varices Prediction in Cirrhotic Patients. *IEEE Access* **2021**, *9*, 15041–15052. [CrossRef]
30. Li, M.; Yan, C.; Liu, W. The network loan risk prediction model based on Convolutional neural network and Stacking fusion model. *Appl. Soft Comput.* **2021**, *113*, 107961. [CrossRef]
31. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]

32. Tai, K.S.; Socher, R.; Manning, C.D. Improved Semantic Representations from Tree-Structured Long Short-Term Memory Networks. *Comput. Sci.* **2015**, *5*, 1–36.
33. Kaltenbrunner, A.; Meza, R.; Grivolla, J.; Codina, J.; Banchs, R. Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system. *Pervasive Mob. Comput.* **2010**, *6*, 455–466. [CrossRef]
34. Siami-Namini, S.; Tavakoli, N.; Siami Namin, A. A Comparison of ARIMA and LSTM in Forecasting Time Series. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; pp. 1394–1401.

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.