

Article

Research on Short-Term Passenger Flow Prediction of LSTM Rail Transit Based on Wavelet Denoising

Qingliang Zhao, Xiaobin Feng, Liwen Zhang and Yiduo Wang *

School of Economics and Management, Beijing University of Chemical Technology, Beijing 100029, China; zhaoql@mail.buct.edu.cn (Q.Z.); 2021201301@buct.edu.cn (X.F.); 13163175122@163.com (L.Z.)

* Correspondence: wangyd@mail.buct.edu.cn

Abstract: Urban rail transit offers advantages such as high safety, energy efficiency, and environmental friendliness. With cities rapidly expanding, travelers are increasingly using rail systems, heightening demands for passenger capacity and efficiency while also pressuring these networks. Passenger flow forecasting is an essential part of transportation systems. Short-term passenger flow forecasting for rail transit can estimate future station volumes, providing valuable data to guide operations management and mitigate congestion. This paper investigates short-term forecasting for Suzhou's Shantang Street station. Shantang Street's high commercial presence and distinct weekday versus weekend ridership patterns make it an interesting test case, making it a representative subway station. Wavelet denoising and Long Short Term Memory (LSTM) were combined to predict short-term flows, comparing the results to those of standalone LSTM, Support Vector Regression (SVR), Artificial Neural Network (ANN), and Autoregressive Integrated Moving Average Model (ARIMA). This study illustrates that the algorithms adopted exhibit good performance for passenger prediction. The LSTM model with wavelet denoising proved most accurate, demonstrating applicability for short-term rail transit forecasting and practical significance. The research findings can provide fundamental recommendations for implementing appropriate passenger flow control measures at stations and offer effective references for predicting passenger flow and mitigating traffic pressure in various cities.

Keywords: rail transit short-term passenger flow prediction; LSTM; wavelet denoising analysis; SVR

MSC: 03-08; 03-11; 65J08; 68U99; 97N80; 97P40



Citation: Zhao, Q.; Feng, X.; Zhang, L.; Wang, Y. Research on Short-Term Passenger Flow Prediction of LSTM Rail Transit Based on Wavelet Denoising. *Mathematics* **2023**, *11*, 4204. <https://doi.org/10.3390/math11194204>

Academic Editor: Ioannis G. Tsoulos

Received: 15 August 2023

Revised: 27 September 2023

Accepted: 27 September 2023

Published: 9 October 2023



Copyright: © 2023 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

The modern development of the city constantly promotes the rapid growth of traffic demand [1]. Urban rail transit, as an important part of urban public transportation, is greatly significant for improving urban passenger flow and transportation efficiency, as well as alleviating traffic congestion. Scientific passenger flow prediction plays an extremely important role in feasibility studies for urban rail transit, layout planning of urban rail transit networks, and decision-making around urban rail transit construction scales and levels. On the one hand, it is conducive to dredging passengers, rationally arranging the flow lines of passengers in the station, and improving the quality of passenger flow organization. On the other hand, it helps the urban transportation system take timely response measures and ensure public safety [2,3].

Short-time passenger flow forecasting is a dynamic control method that mainly forecasts passenger flow in the future based on existing passenger flow data [4]. The research on the traditional forecasting model of rail transit passenger flow is quite mature. The passenger flow prediction models of rail transit mainly include time series models, regression models, and some related linear and nonlinear models. Roos [5] proposed a dynamic Bayesian network approach to forecast the short-term passenger flows of the urban rail network of Paris, which could deal with the incompleteness of the data caused by failures

or a lack of collection systems. Zhao [6] used a support vector machine to predict the passenger flow of Xinzhuang subway station and concluded that the nonlinear support vector machine model can predict the working day better. Anl [7] developed a long short-term memory-based (LSTM-based) deep learning model to predict short-term transit passenger volume on transport routes in Istanbul using a dataset that included the number of people who used different transit routes at a one-hour interval between January and December 2020 and compared that with popular models such as random forest (RF), support vector machines, autoregressive integrated moving average, multilayer perceptron, and convolutional neural networks. Taking the passenger flow of Chengdu East Railway Station as an example, Tan [8] verified the higher prediction accuracy and better prediction performance of the GRNN neural network model based on parameter optimization (GA) compared with other models. Pekel [9] developed two hybrid forecasting methods, POA-ANN and IWD-ANN, to forecast passenger demand, compared the forecasting results with GA-ANN, and concluded that the new algorithm had a good effect on passenger prediction.

In order to improve prediction accuracy, many scholars have studied the application of neural networks and combined models to short-term passenger flow prediction. Alghamdi [10] proposed an end-to-end deep learning-based framework with a novel architecture to predict multi-step-ahead real-time travel demand along with uncertainty estimation. Asce [11] presented a novel nonparametric dynamic time-delay recurrent wavelet neural network model for forecasting traffic flow that exploited the concept of wavelet in the model to provide flexibility and extra adaptable translation parameters in the traffic flow forecasting model. Nagaraj [12] used a greedy layer-wise algorithm to enter the processed cluster data into the long- and short-term memory models and a recurrent neural network to solve the passenger flow prediction problem in public transport. Ermagun [13] examined spatiotemporal dependency between traffic links, proposed a two-step algorithm to search and identify the best look-back time window for upstream links, and indicated the best look-back time window depends on the travel time between two study detectors. Dong [14] used a genetic algorithm to optimize the BP model, which significantly improved the prediction accuracy of short-term passenger flow on Beijing Line 4. Mirzahosseini [15] proposed a novel hybrid method based on deep learning to estimate short-term traffic volume at three adjacent intersections, combined with a time window and normal distribution of WND-LSTM for traffic flow prediction, and the MAPE obtained was 60–90% lower than that of ARIMA, LR, and other models.

The current research primarily focuses on the global prediction of passenger flow for all stations or the entire subway line. However, there is a lack of sufficient precise prediction research that takes into account the specific characteristics of individual stations. Furthermore, there are a limited number of prediction models available for comparison, and the existing models do not achieve a high level of accuracy. In general, the accuracy of passenger flow prediction is greatly influenced by the changing trends observed in previous data. Therefore, it is crucial to conduct a detailed analysis of the characteristics of subway stations and evaluate multiple prediction models to enhance the accuracy and effectiveness of the predictions [16]. The combination of wavelet denoising and the LSTM model in this study has several benefits and innovations. Wavelet denoising enhances data quality by reducing noise interference, while the LSTM model effectively handles the time-series relationships and dynamic characteristics of non-stationary data. For complex AFC data, by combining them and comparing them with other related models, the future trend of non-stationary data can be predicted more accurately, and the accuracy and stability of prediction results can be improved. In addition, this method is innovative and provides a new idea and solution for the prediction and analysis of non-stationary data related to subway passenger flow. Different suitable models are selected for different types of stations to predict. Based on the cluster analysis of subway stations, this paper carries out detailed prediction analysis for typical stations. This paper examines Shantang Street station in Suzhou, chosen for its high commercial nature and weekday/weekend passenger differences. Wavelet denoising processed the short-term flow data, which an LSTM model

used to predict volumes versus standalone LSTM, SVR, ANN, and ARIMA. The wavelet-denoised LSTM model [17–19] significantly improved accuracy, indicating effectiveness for real-world rail transit forecasting.

2. Research Methods

2.1. Wavelet Denoising Analysis

2.1.1. Principle of Wavelet Denoising Analysis

Wavelet denoising analysis [20–22] has been successfully utilized in many fields. Due to the irregularity of short-term passenger flow data at stations, the prediction error for short-term rail transit passenger flow may be substantial.

The short-term passenger flow data of rail transit stations fluctuates constantly, with a certain level of noise. High-frequency signals can be denoised through threshold values, and then data can be reconstructed to achieve denoising. The traffic signal for short-term traffic volume containing noise can be formulated as follows:

$$S(x) = f(x) + \sigma e(x) \tag{1}$$

$f(x)$: data after noise removal

$e(x)$: contained noise

σ : noise intensity

$S(x)$: short-term passenger flow data of rail transit with noise signal

2.1.2. Wavelet Denoising Process

The basic process of wavelet denoising analysis is shown in Figure 1 below:

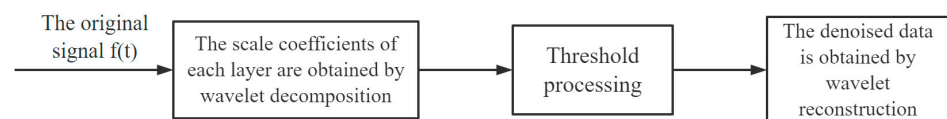


Figure 1. Flow chart of wavelet denoising.

Therefore, when utilizing wavelet denoising to analyze short-term passenger flow data for rail transit, it can be simplified into five processes: selecting the wavelet function, wavelet base order, threshold function, decomposition layer, and wavelet reconstruction.

2.2. Basic Principles of Long Term Memory Networks

2.2.1. LSTM Process

The LSTM neural network [23] has four structures: forgetting gate, input gate, output gate, and memory unit. The cell structure of the unit is controlled through the forgetting and input gates. The LSTM process is (Figure 2):

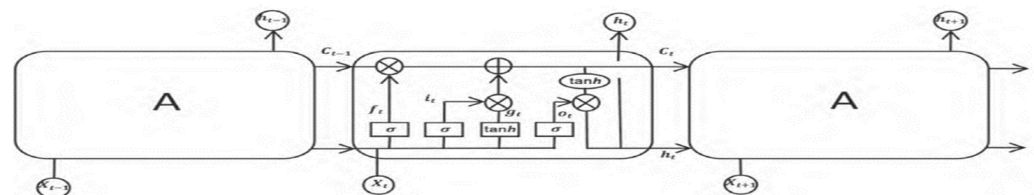


Figure 2. LSTM process structure diagram.

The arrows in the figure above represent vectors, showing input from the previous node to the node the arrows point to. LSTM controls information flow through three gate structures, consisting of sigmoid activation functions and a multiplicative structure with an

output of 0 or 1. The sigmoid activation function in the gate is Equations (2) and (3), with the tanh functions being (4) and (5).

$$\sigma(z) = y = \frac{1}{1 + e^{-z}} \tag{2}$$

$$\sigma'(z) = y(1 - y) \tag{3}$$

$$\tanh(z) = y = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{4}$$

$$\tanh'(z) = 1 - y^2 \tag{5}$$

C_{t-1} : The cell state passed in at the previous time.

x_t : The new value of information that is read at the present moment causes the module to generate a new memory.

h_{t-1} : The output value of the previously hidden neuron module.

C_t : Belongs to the current time output information, to the next time transmitted unit state.

h_t : New output at the current time.

2.2.2. Calculation of LSTM Forward Propagation

The LSTM forward propagation calculation process is from the forgetting gate to the input gate, updating the unit state, and finally to the output gate [24].

The forgetting gate determines how much information can be retained from the previous moment to the current one. After h_{t-1} and x_t are activated by activation function, f_t is obtained, representing the degree of retention of the previous hidden neuron state. The activation function is sigma, with the f_t expression being:

$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f) \tag{6}$$

W_f : Represents the weight of the input forgetting gate of the previous hidden neuron module.

U_f : The information value of the input layer flows into the weight of the forgetting gate.

b_f : Calculate the bias parameters of the forgetting door.

The input gate determines how much information will be received and can determine the new information generated and what percentage of the new information will be used. The calculation process is as follows:

$$i_t = \sigma(W_i [h_{t-1}, x_t] + b_i) \tag{7}$$

$$\tilde{C}_t = \tanh(W_c [h_{t-1}, x_t] + b_c) \tag{8}$$

After passing through the input gate, the output of the input gate is: $i_t \times C_t$.

The updating of memory cell state means that the output f_t of the forgetting gate is multiplied by the cell state C_{t-1} at the previous time and combined with the output of the input gate to obtain a new cell state C_t . The expression of C_t is as follows:

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{9}$$

Finally, we need to go through the output door, which is composed of two parts of calculation, partly by the current information combined with short-term memory thus calculated, and o_t , another part is calculated combined with long-term memory and concluded h_t, o_t by the module of a hidden neurons on the output value of h_{t-1} combined

with the current input value x_t and activated by sigma function. The calculation process is as follows:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{10}$$

The final output after LSTM model is as follows:

$$h_t = o_t \circ (\tanh((C_t))) \tag{11}$$

2.2.3. Reverse Calculation of LSTM

After LSTM model forward propagation, the weight set and relevant bias terms must be updated; therefore, reverse calculation can be performed by propagating the error up the layer.

2.3. Principles of Support Vector Machine Regression

The training samples of the SVR model [25–27] are $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, with $y_i \in \mathbb{R}$. The goal is to learn a model $f(x)$ with a value close to y . When model $f(x)$ exactly matches y , the final loss is 0. In the SVR model, the deviation between $f(x)$ and y is set at most to ϵ . When the difference between $f(x)$ and y is greater than ϵ , the loss is calculated; otherwise, the loss is ignored. This is equivalent to establishing a 2ϵ wide tolerance band centered on $f(x)$. The red horizontal line is the standard data cable. The two dashed lines in the figure represent the soft interval. The data between the soft interval is represented by a blue dot, and the data outside the soft interval is represented by a white dot. If sample data falls within the tolerance bands, the prediction is accurate, as shown (Figure 3):

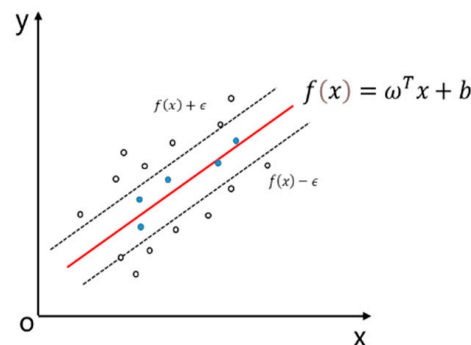


Figure 3. Support vector machine regression display.

Parameters involved in support vector machine (SVM) regression include ϵ and C . ϵ is the loss function and affects model precision and training speed. Parameter C is a penalty factor, aiming to balance the model. The smaller C means a lower model complexity and penalty. The choice of C should not be too large or small; otherwise, overfitting or underfitting may occur.

2.4. Principles of Artificial Neural Networks

An Artificial Neural Network (ANN) consists of an input layer, a hidden layer(s), and an output layer [28,29]. The state of the hidden layer remains unaffected by external factors; however, its state changes can lead to variations in the output. The back propagation algorithm is commonly utilized in ANN. It involves forward propagation, where the input layer is sequentially propagated through each layer, followed by back propagation, which adjusts the weights and related thresholds. This iterative process aims to minimize the error until the desired outcome is achieved. The calculation process is illustrated in the diagram [30] (Figure 4).

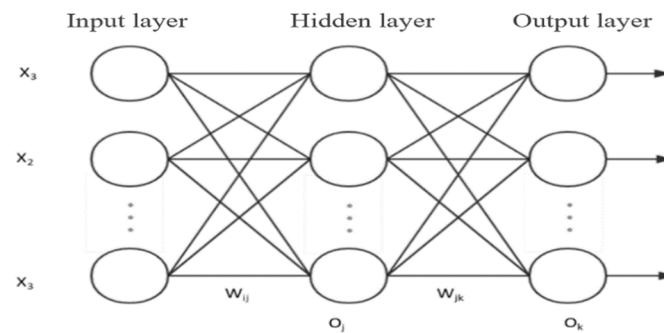


Figure 4. Flowchart of artificial neural network.

- (1) Forward propagation process The output value of the input layer is denoted as O . The connection weights between the input layer and the hidden layer are represented as w_{ij} . The output value of the input layer is multiplied by the corresponding weights w_{ij} . The resulting values are then passed through an activation function, typically the sigmoid function, to obtain the output values of the hidden layer. This process is repeated for each subsequent layer until the output layer is reached [31].
- (2) Backward propagation process The backward propagation process involves adjusting the parameters of the artificial neural network model to optimize its performance. The connection weights and cell thresholds are then modified accordingly to minimize the error. This adjustment is performed iteratively to refine the model’s performance.
- (3) Training termination conditions The training process can be terminated based on certain conditions. Once these conditions are met, the training process is concluded.

2.5. Basic Principles of Time Series Model

The ARIMA model is a statistical model used for analyzing and predicting time series data [32,33]. It is particularly effective in forecasting future values based on past observations and the autocorrelation within the series. The model consists of three main components: the autoregressive (AR) part, the differencing (I) part, and the moving average (MA) part. These components work together to capture the patterns and trends in the data, allowing for accurate predictions [34].

- (1) Autoregressive Model (AR) The autoregressive model utilizes historical data to construct a predictive model for its own data. It is important to note that the autoregressive model assumes the data to be stationary. The formula for a p -order autoregressive model is as follows:

$$y_t = \mu + \sum_{m=1}^p r_m y_{t-m} + \varepsilon_t \tag{12}$$

y_t : The current value of the variable μ : Constant term p : Order γ_m : Autocorrelation coefficient ε_t : Residual

- (2) Moving Average Model (MA) The moving average model utilizes the past values of the residual to represent the linear relationship, aiming to observe the magnitude of its fluctuations. The formula for a p -order moving average model is as follows:

$$y_t = \mu + \sum_{m=1}^q \theta_m \varepsilon_{t-m} + \varepsilon_t \tag{13}$$

- (3) Auto-Regression and Moving Average Model (ARMA) The ARMA model combines the autoregressive (AR) model and the moving average (MA) model. It expresses the relationship between the current value and both past values and past residuals. The formula for the ARMA model is as follows:

$$y_t = \mu + \sum_{m=1}^p r_m y_{t-m} + \varepsilon_t + \sum_{m=1}^q \theta_m \varepsilon_{t-m} + \varepsilon_t \tag{14}$$

- (4) Integrated (I) Before determining the parameters p and q in the ARIMA model, it is necessary to conduct a stationarity test on the data. If the data fails the test, differencing is performed. After differencing, the data should meet the stationarity condition.

3. Empirical Study

3.1. Data Preprocessing

This article examines short-term traffic predictions for the Shantang Street station of the Suzhou Rail Transit system. The AFC system data used in this article is collected from the automatic ticket machines at various stations in the rail transit system, which record the card swipes of people entering and exiting the stations. The data utilized in this study were obtained from the Suzhou Rail Transit AFC system and include transaction time, ticket ID and type, inbound and outbound station codes and names, and inbound and outbound times. The experiments in this article were conducted on a Windows 10 64-bit operating system. The hardware used includes an AMD Ryzen 7 5800H with a Radeon Graphics 3.20 GHz processor and 16 GB of memory. The programming language used is Python 3.7, and the Matplotlib 3.0.2 plotting tool was utilized for generating plots.

MySQL was used to clean the raw data. Relevant database rules were applied to extract the required information, resulting in over 14 million data points for the month of July that were used in this analysis. Given the high commercial nature and distinct weekday versus weekend ridership patterns, outbound passenger traffic from Shantang Street station was selected as the prediction target. The 1–27 July inbound passenger flow was used as the training set, while the 28–29 July (Monday and Sunday) data were held out as the test set. Based on the extracted swipe card data from Shantang Street station in MySQL, the inbound passenger flow at Shantang Street is calculated with a time interval of 1 h. Each column represents the inbound passenger flow at Shantang Street every hour throughout the day. The processed Shantang Street passenger flow data are summarized, and a portion of the hourly passenger flow data are shown in Table 1 below:

Table 1. Hourly passenger flow data for Shantang Street station (person/hour).

Time	Station	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
5:00:00	Shantang Street	4	2	4	12	3	9	10
6:00:00	Shantang Street	220	235	203	250	237	198	133
7:00:00	Shantang Street	472	471	495	470	471	432	410
8:00:00	Shantang Street	519	491	467	543	491	601	513
9:00:00	Shantang Street	497	525	595	596	552	655	572
10:00:00	Shantang Street	461	538	537	531	516	583	656
18:00:00	Shantang Street	415	360	317	382	409	527	621
19:00:00	Shantang Street	391	396	400	425	466	640	636
20:00:00	Shantang Street	407	479	536	497	494	845	772
21:00:00	Shantang Street	306	365	371	431	463	703	551
22:00:00	Shantang Street	77	81	129	94	149	100	171

3.2. LSTM Model Construction and Prediction Analysis

The training dataset for the LSTM network consists of the inbound passenger flow data at Shantang Street for the month of July. The objective is to predict the outbound passenger flow at Shantang Street. The training set includes the inbound passenger flow data from 1–27 July, while the test set comprises the inbound passenger flow data from 28–29 July (which corresponds to Monday and Sunday). The LSTM model is trained on each subset, and the validation subset is used to evaluate the model’s performance. According to the size of the passenger flow data set and the limitation of computing resources, $K = 5$ was determined, 5-fold cross-validation was selected, and the training set was divided into 5 subsets, of which 4 subsets were used to train the model and the remaining 1 subset was used to verify the model, and the performance index of each fold number was recorded.

Root Mean Squared Error (RMSE) was selected as the evaluation index, and the parameters were constantly evaluated and optimized. Ultimately, the output layer is set to have a dimension of 1, the hidden layer is set to 4, the number of iterations is set to 1000, and the historical time step length is set to 30. To capture traffic patterns, a historical time step of 30 was used. A batch size of 10 and dropout layers were incorporated to improve accuracy and prevent overfitting. Sigmoid activation functions were utilized for all fully connected layers during training; the prediction results are compared in the final summary figure.

Compare the forecast results with the chart, and analyze the result index values. This model configuration was trained and used to predict the test set ridership. The results were compared to the actual values using the RMSE, MAE, and MAPE metrics for both weekdays and weekends. As seen in the figure, the LSTM model predictions did not match the true values very closely, indicating poor performance that needs improvement across all accuracy metrics.

3.3. LSTM Model Construction and Prediction Analysis of Wavelet Denoising

3.3.1. Steps of Model Construction and Prediction

To address these limitations, a wavelet denoising approach was applied prior to LSTM modeling. The key steps were:

1. Perform a 3-level discrete wavelet transform on the time series data using the db6 wavelet.
2. Decompose the signal into low- and high-frequency components.
3. Apply soft thresholding denoising to the three high-frequency signals.
4. Reconstruct the denoised signal.
5. Split the data into training and test sets.
6. Train the LSTM model on denoised training data.
7. Validate model performance on denoised test data.

The visualizations below depict the original noisy data versus the smoothed denoised signal after wavelet decomposition and thresholding.

3.3.2. Predictive Analysis

Based on the aforementioned basic prediction steps, continuous validation is performed to determine the wavelet base function and conduct wavelet decomposition. First, the db6 wavelet basis function is selected to decompose the three-layer wavelet of July inbound passenger flow data of Shantang Street station with a time interval of 10 min, and the results are shown in Figure 5 below.

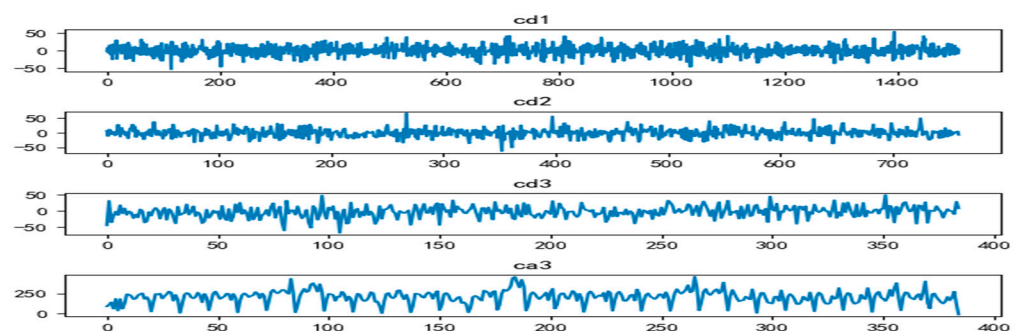


Figure 5. Data display after wavelet decomposition.

After wavelet decomposition and soft threshold denoising, the denoised data and the original data are visualized, as shown in Figure 6. It can be seen that the denoised data are smoother. The blue curve in the Figure 6 represents the original data, and the orange curve represents the data after noise removal.

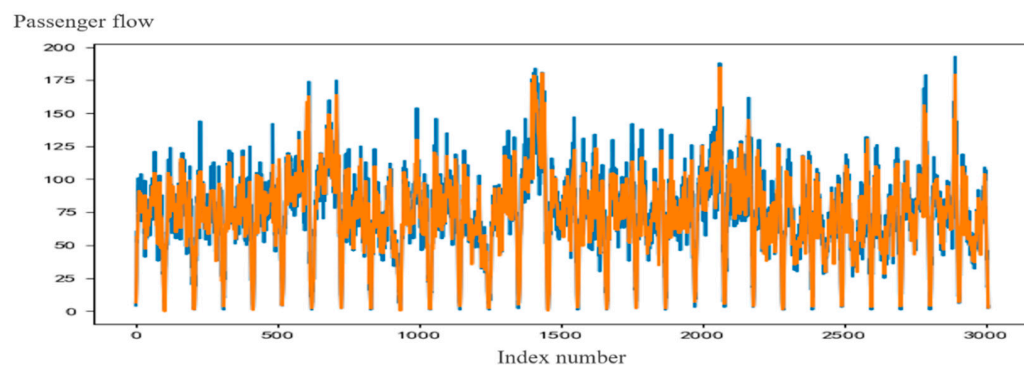


Figure 6. Comparison between denoised data and original data.

The inbound short-time passenger flow training set data of Shantang Street station after noise removal is used as the input of the LSTM network. Considering the data features and model performance after wavelet decomposition, RMSE is again selected as the evaluation metric. Cross-validation is employed to continuously assess and optimize the parameters. Ultimately, the following parameter settings are determined: the input layer has a dimension of 1, the time step length is set to 1, the output layer has a dimension of 1, the hidden layer is set to 8, the number of iterations is set to 3000, and the historical time step length is set to 30. For accurate training, the batch_size is set to 10, and the dropout layer is added. It is better to set the probability to 0.1. Based on the above settings, the training of the model is expanded, and the prediction results are compared in the final summary figure.

Compare the forecast results with the chart, and analyze the result index values. It can be seen that there is almost no difference between the prediction results of the test set data and the data after noise removal, and the prediction model effect of the processed data are more significant.

With the denoised data, the LSTM model was re-trained using the same configuration described previously. As evident in the figure, the predictions closely matched the denoised test set values, demonstrating significantly improved model performance compared to the non-denoised data. The RMSE, MAE, and MAPE were substantially lower for both weekday and weekend results, confirming the benefits of preprocessing with wavelets prior to LSTM modeling for this application. It has great significance for forecasting.

3.4. SVR Model Construction and Prediction Analysis

According to the existing experimental results, it can be concluded that the SVR model has good fitting ability and has a good effect on solving some complex nonlinear problems. The short-term passenger flow of rail transit has the characteristic of complexity; therefore, a support vector machine model can be used to deal with the problem of short-term passenger flow prediction.

3.4.1. Steps of SVR Model Construction and Prediction

1. Separate the data into training (1–27 July) and test sets (28–29 July).
2. Train support vector machines (SVMs) with different kernels, selecting RBF based on best fit.
3. Initialize hyperparameter values for penalty factors C and gamma.
4. Refine hyperparameters via grid search cross-validation to minimize MSE.
5. Assess the model on test data.

3.4.2. Predictive Analysis

The step of prediction is set to 1. It is proven that the first 30 data points are used to predict the next data points, and the calculated error is relatively small. Firstly, the penalty factor parameter C was set as 1, 5, 10, 30, 100, and the parameter gamma was set

as 0.1, 0.12, 0.01, 0.05, 0.001, 1, 0.5, and 0.9. The rbf function was selected as the kernel function. In the prediction, RMSE is chosen as the evaluation metric to continuously assess the model's performance using cross-validation and evaluate the model's generalization ability. The parameters are continuously evaluated and optimized to select the optimal hyperparameters. Finally, $C = 5$ and $\gamma = 0.1$ were determined to predict the test set data based on the parameters, and the prediction results are compared in the final summary figure.

Compare the forecast results with the chart, and analyze the result index values. By calculating the predicted results, it can be seen that the predicted results do not deviate much from the actual values; however, there is still a certain gap compared with the LSTM model of wavelet denoising. However, in general, the SVR model is relatively reasonable for the prediction of short-term passenger flow.

3.5. ANN Model Construction and Prediction Analysis

Using the sklearn library in Python, an artificial neural network model was called to implement the backpropagation algorithm. The model was evaluated using the Root Mean Squared Error (RMSE) metric. Cross-validation was performed to determine the relevant parameters, and parameter tuning was conducted to optimize the model. After experimentation, it was found that setting the number of iterations to 100 and the batch size to 1 yielded relatively ideal results. The prediction step was set to 1, the network layer had 12 neurons, and the activation function used was sigmoid. The prediction results are compared in the final summary figure.

The forecast results were evaluated using RMSE, MAE, and MAPE. Compare the forecast results with the chart, and analyze the result index values. However, upon analyzing the prediction result graph, it was observed that the artificial neural network model did not perform exceptionally well. It failed to accurately predict sudden fluctuations in passenger flow and exhibited lower prediction accuracy. The predicted values were generally higher than the actual values

3.6. ARIMA Model Construction and Prediction Analysis

3.6.1. Steps of ARIMA Model Construction and Prediction

Step 1: The Augmented Dickey-Fuller test (ADF test) can be utilized to test for stationarity [35,36]. The ADF test examines the presence of a unit root in the model, which implies that $b = 1$ in an autoregressive equation $y_t = by_{t-1} + c + \epsilon_t$. This phenomenon can create spurious relationships between independent and dependent variables. The ADF test assumes the existence of a unit root and evaluates the significance test statistic at three confidence levels (1%, 5%, and 10%).

White noise [37,38] is characterized by data that lacks any discernible patterns, with mean values fluctuating around zero and no clear trend. It follows a normal distribution with a mean of 0 and a variance of σ^2 . If the data contains white noise after testing, it indicates that there is no useful information, and modeling would be meaningless. Conversely, if there is no white noise, it suggests that the data can be modeled.

Step 2: Determine the values of p_{max} and q_{max} . This can be achieved by examining the autocorrelation and partial autocorrelation plots of the original time series data. Table 2 can be used as a guide to determine the appropriate values for p_{max} and q_{max} in the ARIMA model.

Step 3: Determine the final values of p and q by considering the maximum likelihood function value and the minimum number of parameters. The higher the likelihood function value, the better the model is. Additionally, a model with fewer parameters has lower complexity and computational requirements. The optimal values of p and q can be determined by calculating the Bayesian Information Criterion (BIC) [39,40].

The BIC is a criterion based on Bayesian theory that provides a more accurate judgment, particularly for large sample sizes, compared to the Akaike Information Criterion (AIC) [41].

The BIC is calculated as $BIC = \ln(n)$ (number of parameters in the model) $- 2\ln$ (maximum likelihood function value of the model).

Table 2. pmax, qmax judgment table.

Model	ACF	PACF
AR	After decay approaches 0	p-order truncation (after a value greater than a rapid approach to 0)
MA	q order back truncated	After decay approaches 0
ARMA	Attenuation tends to 0 after order q	Attenuation tends to 0 after order p

Step 4: Test the model’s validity using the Durbin-Watson (DW) test [42,43] and the QQ plot test [44]. The DW test assesses the autocorrelation of a dataset by calculating the DW value of the residual from the established model. A DW value close to 0 or 4 indicates the presence of autocorrelation in the residual, while a value approaching 2 suggests no autocorrelation.

3.6.2. Predictive Analysis

First of all, we need to check the short-time passenger flow data series of Shantang Street to judge whether the time series data of the inbound passenger flow of Shantang Street is stable (Figure 7).

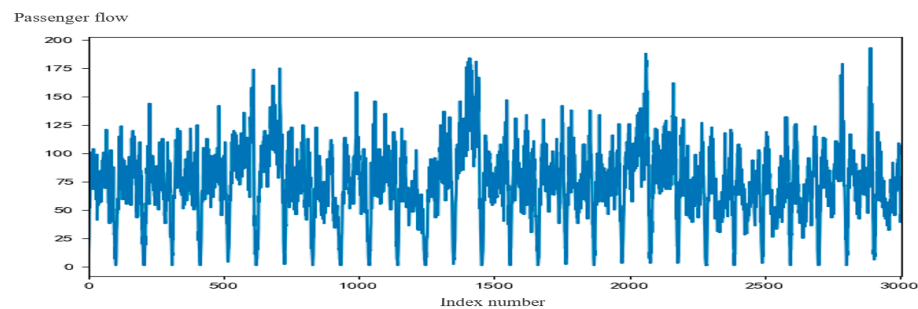


Figure 7. Time series diagram of short-term passenger flow in Shantang Street.

It can be seen from the figure above that the data are basically stable, and then the unit root and stationarity tests of the data are carried out. By calculating ADF, the test results are as follows: $(-9.19, 2.11e^{-15}, 18, 2986, \{ '1\%': -3.43, '5\%': -2.86, '10\%': -2.57 \}, 25241.09)$. All of these calculations are reserved for two decimal places. The statistical value is lower than the original hypothesis at the 1%, 5%, and 10% significance levels, indicating that the data does not have a unit root and there is no white noise present. Therefore, the data are stable and suitable for ARIMA modeling analysis. Additionally, the calculated p -value of $2.11e^{-15}$ is less than 0.05, further supporting the conclusion that the data does not have a unit root.

The autocorrelation and partial autocorrelation plots of the original sequence data for the Shantang Street inbound passenger flow training set were used to determine the values of p and q in the ARIMA model (Figure 8).

From the autocorrelation plot, it can be observed that the values approach 0 after the 10th order. Similarly, the partial autocorrelation plot shows that the values mostly approach 0 after the 4th order. Based on these observations, $p_{max} = 10$ and $q_{max} = 5$ were selected. The final values of p and q were determined using the Bayesian Information Criterion (BIC). The BIC calculation indicated that the smallest value was obtained when $p = 3$ and $q = 3$. Therefore, the ARIMA (3,0,3) model was established.

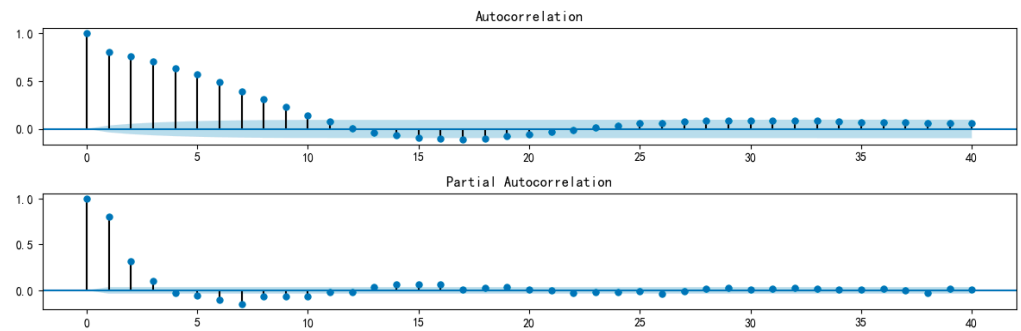


Figure 8. Autocorrelation and partial autocorrelation diagram.

The residuals of the ARIMA (3,0,3) model were tested using the Durbin-Watson (DW) test and QQ plot. The calculated DW test value was 2.008262490962122, which is close to 2. The red line is the standard data cable. The QQ plot also showed that the data points were approximately on a straight line. The blue dots in the figure are the index values of the test results. These results indicate that the model is reasonable (Figure 9).

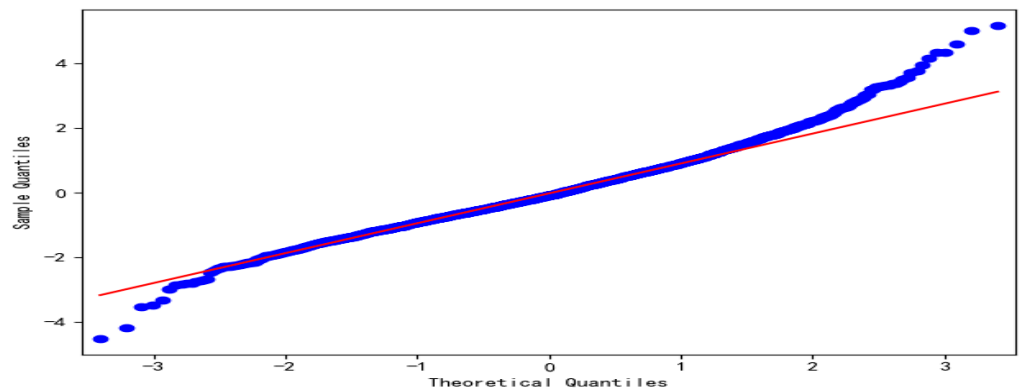


Figure 9. QQ diagram test results.

After conducting the DW and QQ graph tests, it can be concluded that the established model is reasonable. The trained model is then utilized to predict the short-term passenger flow of Suzhou Metro on 28–29 July, and the results are compared in the final summary figure. Compare the forecast results with the chart, and analyze the result index values. Based on the calculated MAE index, it can be observed that the error in predicting short-term passenger flow on weekdays is relatively small, with an average MAPE value of around 20%. Overall, the ARIMA model demonstrates relatively good prediction performance.

3.7. Comparison of Results

By utilizing various models to forecast the short-term passenger flow of Shantang Street station, the prediction results can be visually displayed, and the corresponding prediction indices can be summarized. First, a visual representation of the predicted results is presented. Subsequently, the RMSE, MAE, and MAPE indices of the predicted results for Sunday (28 July) and Monday (29 July) are calculated based on different feature days, allowing for an analysis of the accuracy of each model’s predictions. The comparison of the prediction results of different models on 28 July is shown in Figure 10. And the comparison of the prediction results of different models on 29 July is shown in Figure 11. Xiaobo_predict in the figure below is a Chinese noun for the prediction results of the LSTM model based on wavelet denoising.

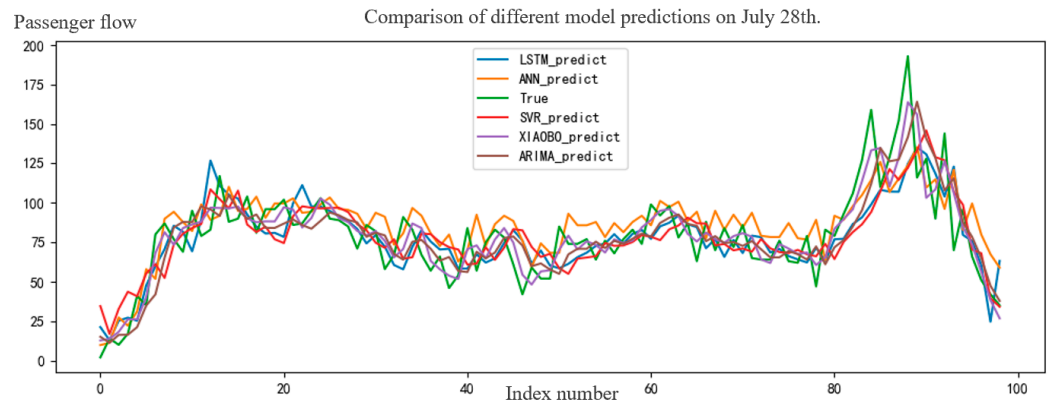


Figure 10. Comparison of prediction results of different models at Shantang Street station on 28 July.

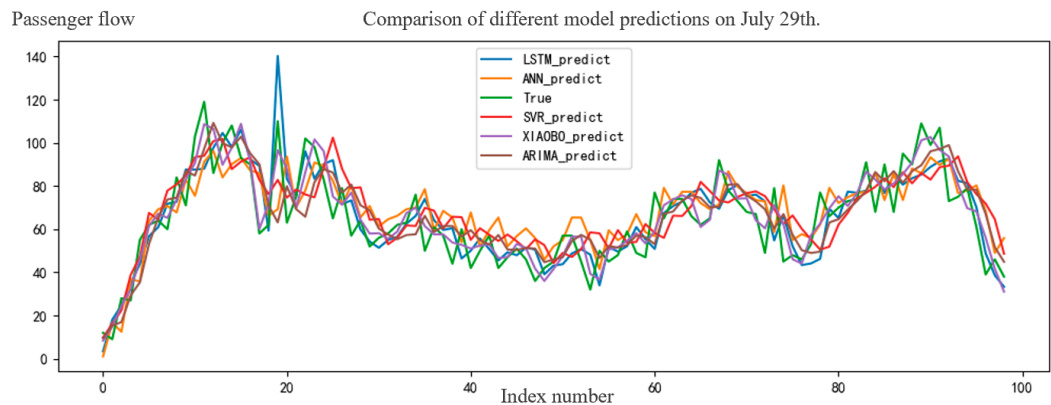


Figure 11. Comparison of prediction results of different models at Shantang Street Station on 29 July.

The forecast result chart displays the number of input indicators on the x-axis and the passenger flow, measured in terms of the number of people, on the y-axis. The green curve represents the true value, and the other curves in different colors represent the predicted value of the different models. This visual representation effectively demonstrates the model’s prediction accuracy and effectiveness.

The Shantang Street station predictions reveal that the estimated volumes from all methods stayed relatively close to their true values. This suggests the selected techniques were appropriate for modeling this station’s ridership. Across both weekday and weekend results, the denoised LSTM predictions aligned most tightly with the real data. The index values of prediction results of different models are shown in Table 3 below.

Upon analyzing the calculated results in the aforementioned table, it can be observed that the LSTM method, which incorporates wavelet analysis for denoising, yields lower RMSE and MAE indices compared to other methods. Furthermore, the MAPE index is also significantly reduced. Consequently, it can be concluded that this method exhibits certain advantages in terms of prediction accuracy. In the absence of wavelet denoising, the LSTM model demonstrates superior performance, followed by SVR and ARIMA, while the ANN model exhibits relatively poorer performance when predicting short-term passenger flow on weekdays. When it comes to predicting short-term passenger flow on Sundays, both the LSTM and ARIMA models outperform the ANN model. As Sundays typically experience higher station traffic compared to Mondays, it is expected that the prediction errors will be higher on Sundays compared to Mondays. Considering both predictive power and practicality, integrated wavelet denoising with LSTM emerges as the superior methodology, demonstrating its applicability to real-world forecasting.

Table 3. Index values of prediction results under different models.

Method	Index	Monday (29 July)	Sunday (28 July)
LSTM	RMSE	12.86	19.78
	MAE	10.27	15.35
	MAPE	18%	31%
The wavelet + LSTM	RMSE	8.94	12.32
	MAE	7.22	9.88
	MAPE	12%	19%
SVR	RMSE	14.15	19.25
	MAE	11.78	14.72
	MAPE	21%	38%
ANN	RMSE	15.29	20.82
	MAE	12.40	16.62
	MAPE	22%	29%
ARIMA	RMSE	14.04	17.68
	MAE	10.92	13.52
	MAPE	18%	24%

4. Conclusions

The focus of this study is to apply different short-term forecasting techniques to predict the passenger flow at Shantang Street Station of Suzhou Rail Transit. The goal was to analyze whether the proposed denoised LSTM method provided higher accuracy and effectiveness.

This paper examines Shantang Street station in Suzhou, chosen for its high commercial nature and weekday/weekend passenger differences. For short-term prediction research, wavelet denoising processed the time series data before LSTM modeling. Based on signal-to-noise ratios and rail transit passenger flow characteristics, 3-level decomposition via soft thresholding and the db6 wavelet filtered out noise. This denoised data were used to train the LSTM model and compare its forecasts against the original noisy LSTM, SVR, ANN, and ARIMA results. This study confirms the necessity of selecting appropriate methods for predicting rail transit passenger flow. The wavelet-enhanced LSTM significantly improved prediction quality, providing a new perspective for rail transit volume forecasting. Leveraging big data and scientific modeling in this manner can produce practical gains, demonstrating the value of this integrated approach.

In this paper, single-step prediction is adopted when using the model to forecast short-term passenger flow, and multi-step prediction can be carried out in future research, which may save the time of model calculation. When forecasting short-term passenger flow, only the time series data of passenger flow is used in this paper. In the next forecasting study, features such as weather and geographical location can be added so that the factors considered will be more comprehensive, which will be helpful in improving the forecasting accuracy.

Author Contributions: Conceptualization, Q.Z. and L.Z.; methodology, X.F. and Y.W.; software, X.F.; data curation, Q.Z.; writing—original draft preparation, Q.Z. and X.F.; writing—review and editing, Q.Z. and Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number NFC52075030.

Data Availability Statement: Not applicable.

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

References

1. Li, H.; Wang, Y.; Xu, X.; Qin, L.; Zhang, H. Short-term passenger flow prediction under passenger flow control using a dynamic radial basis function network. *Appl. Soft Comput.* **2019**, *83*, 105620. [[CrossRef](#)]
2. Wang, J.S.; Ou, X.W.; Chen, J.Y. Short-time passenger flow classification prediction of urban rail stations based on Combined model. *J. Railw. Sci. Eng.* **2023**, *20*, 2004–2012.
3. Lei, J.; He, M.; Shuai, C. A comparison study of short-term passenger flow forecast model of rail transit. In Proceedings of the 19th COTA International Conference of Transportation Professionals (CICTP 2019), Nanjing, China, 6–8 July 2019; pp. 1776–1787.
4. Ni, M.; He, Q.; Gao, J. Forecasting the Subway Passenger Flow under Event Occurrences with Social Media. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 1623–1632. [[CrossRef](#)]
5. Roos, J.; Bonnevey, S.; Gavin, G. Short-term rail passenger flow forecasting: A Dynamic Bayesian network approach. In Proceedings of the 15th IEEE International Conference on Machine Learning and Application, Anaheim, CA, USA, 18–20 December 2016.
6. Zhao, Y.T.; Yang, X.F.; Yang, K. Subway passenger flow prediction based on support vector Machine. *Urban Rapid Transit* **2014**, *27*, 35–38. (In Chinese)
7. Anl, U.; Sema, K.K. New deep learning-based passenger flow prediction model. *Transp. Res. Rec. J. Transp. Res. Board* **2023**, 2677, 1–17.
8. Tan, Y.; Liu, H.; Pu, Y.; Wu, X.; Jiao, Y. Passenger Flow Prediction of Integrated Passenger Terminal Based on K-Means–GRNN. *J. Adv. Transp.* **2021**, *2021*, 1055910. [[CrossRef](#)]
9. Pekel, E.; Kara, S.S. Passenger flow prediction based on newly adopted algorithms. *Appl. Artif. Intell.* **2017**, *31*, 64–79.
10. Alghamdi, D.; Basulaiman, K.; Rajgopal, J. Multi-stage deep probabilistic prediction for travel demand. *Appl. Intell.* **2022**, *52*, 11214–11231. [[CrossRef](#)]
11. Jiang, X.M.; Adeli, H.; Asce, H. Dynamic wavelet neural network model for traffic flow forecasting. *J. Transp. Eng.* **2005**, *131*, 771–779. [[CrossRef](#)]
12. Nagaraj, N.; Gururaj, H.L.; Swathi, B.H.; Hu, Y.C. Passenger flow prediction in bus transportation system using deep learning. *Multimed. Tools Appl.* **2022**, *81*, 12519–12542. [[CrossRef](#)]
13. Ermagun, A.; Levinson, D. Spatiotemporal short-term traffic forecasting using the network weight matrix and systematic detrending. *Transp. Res. Part C Emerg. Technol.* **2019**, *104*, 38–52. [[CrossRef](#)]
14. Dong, S.W. Research on Short-Term Passenger Flow Prediction Method of Rail Transit Based on Improved BP Neural Network. Master's Thesis, Beijing Jiaotong University, Beijing, China, 2013. (In Chinese)
15. Mirzahosseini, H.; Gholampour, I.; Sajadi, S.R.; Zamani, A.H. A hybrid deep and machine learning model for short-term traffic volume forecasting of adjacent intersections. *IET Intell. Transp. Syst.* **2022**, *16*, 1648–1663. [[CrossRef](#)]
16. Liu, Y.; Liu, Z.Y.; Ruo, J. DeepPF: A deep learning based architecture for metro passenger flow prediction. *Transp. Res. Part C Emerg. Technol.* **2019**, *101*, 18–34. [[CrossRef](#)]
17. Chi, D. Research on electricity consumption forecasting model based on wavelet transform and multi-layer LSTM model. *Energy Rep.* **2022**, *8*, 220–228. [[CrossRef](#)]
18. Chi, Y.; Cai, C.; Ren, J.; Xue, Y.; Zhang, N. Damage location diagnosis of frame structure based on wavelet denoising and convolution neural network implanted with Inception module and LSTM. *Struct. Health Monit.* **2023**, 14759217231163777. [[CrossRef](#)]
19. Peng, S.; Chen, R.; Yu, B.; Xiang, M.; Lin, X.; Liu, E. Daily natural gas load forecasting based on the combination of long short term memory, local mean decomposition, and wavelet threshold denoising algorithm. *J. Nat. Gas Sci. Eng.* **2021**, *95*, 104175. [[CrossRef](#)]
20. Goyal, B.; Dogra, A.; Agrawal, S.; Sohi, B.S.; Sharma, A. Image denoising review: From classical to state-of-the-art approaches. *Inf. Fusion* **2020**, *55*, 220–244. [[CrossRef](#)]
21. Gilles, J. Empirical wavelet transform. *IEEE Trans. Signal Process.* **2013**, *61*, 3999–4010. [[CrossRef](#)]
22. Sardy, S.; Tseng, P.; Bruce, A. Robust wavelet denoising. *IEEE Trans. Signal Process.* **2001**, *49*, 1146–1152. [[CrossRef](#)]
23. Houdt, G.V.; Mosquera, C.; Gonzalo, N. A review on the long short-term memory model. *Artif. Intell. Rev.* **2020**, *53*, 5929–5955. [[CrossRef](#)]
24. Al-Musaylh, M.S.; Deo, R.C.; Adamowski, J.F.; Li, Y. Short-term electricity demand forecasting with MARS, SVR and ARIMA models using aggregated demand data in Queensland, Australia. *Adv. Eng. Inform.* **2018**, *35*, 1–16. [[CrossRef](#)]
25. Ceperic, E.; Ceperic, V.; Baric, A. A strategy for short-term load forecasting by support vector regression Machines. *IEEE Trans. Power Syst.* **2013**, *28*, 4356–4364. [[CrossRef](#)]
26. Nguyen, H.; Vu, T.; Vo, T.P.; Thai, H.T. Efficient machine learning models for prediction of concrete strengths. *Constr. Build. Mater.* **2021**, *266*, 120950. [[CrossRef](#)]
27. Barbu, T. CNN-based temporal video segmentation using a nonlinear hyperbolic PDE-based multi-scale analysis. *Mathematics* **2023**, *11*, 245. [[CrossRef](#)]
28. Mohiddin, M.B.; Mallikarjuna, P.; Dandagala, S. Applications of Artificial Neural Network for Streamflow Forecasting—A Review. *Artif. Intell. Syst. Mach. Learn.* **2018**, *10*, 25–29.
29. Rehman, K.U.; Shatanawi, W.; Çolak, A.B. Computational Analysis on Magnetized and Non-Magnetized Boundary Layer Flow of Casson Fluid Past a Cylindrical Surface by Using Artificial Neural Networking. *Mathematics* **2023**, *11*, 326. [[CrossRef](#)]

30. Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Umar, A.M.; Linus, O.U.; Arshad, H.; Kazaure, A.A.; Gana, U.; Kiru, M.U. Comprehensive review of artificial neural network applications to pattern recognition. *IEEE Access* **2019**, *7*, 158820–158846. [[CrossRef](#)]
31. Wu, W.Y.; Dandy, G.C.; Maier, H.R. Protocol for developing ANN models and its application to the assessment of the quality of the ANN model development process in drinking water quality modelling. *Environ. Model. Softw.* **2014**, *54*, 108–127. [[CrossRef](#)]
32. Kumar, S.V.; Vanajakshi, L. Short-term traffic flow prediction using seasonal ARIMA model with limited input data. *Eur. Transp. Res. Rev.* **2015**, *7*, 21. [[CrossRef](#)]
33. Singh, S.N.; Mohapatra, A. Repeated wavelet transform based ARIMA model for very short-term wind speed forecasting. *Renew. Energy* **2019**, *136*, 758–768.
34. Jandoc, R.; Burden, A.M.; Mamdani, M.; Lévesque, L.E.; Cadarette, S.M. Interrupted time series analysis in drug utilization research is increasing: Systematic review and recommendations. *J. Clin. Epidemiol.* **2015**, *68*, 950–956. [[CrossRef](#)] [[PubMed](#)]
35. Azam, M.; Khan, A.Q.; Zafeiriou, E.; Arabatzis, G. Socio-economic determinants of energy consumption: An empirical survey for Greece. *Renew. Sustain. Energy Rev.* **2016**, *57*, 1556–1567. [[CrossRef](#)]
36. Wang, X.Q.; Su, C.W.; Tao, R.; Lobonț, O.R. When will food price bubbles burst? A review. *Agric. Econ. (Zemědělská Ekon.)* **2018**, *64*, 566–573. [[CrossRef](#)]
37. Sakai, H.M.; Ken-Ichi, N.; Korenberg, M.J. White-noise analysis in visual neuroscience. *Vis. Neurosci.* **1988**, *1*, 287–296. [[CrossRef](#)] [[PubMed](#)]
38. Riedy, S.M.; Smith, M.G.; Rocha, S.; Basner, M. Noise as a sleep aid: A systematic review. *Sleep Med. Rev.* **2021**, *55*, 101385. [[CrossRef](#)] [[PubMed](#)]
39. Rescorla, M. Bayesian modeling of the mind: From norms to neurons. *Wiley Interdiscip. Rev. Cogn. Sci.* **2021**, *12*, e1540. [[CrossRef](#)] [[PubMed](#)]
40. Stephens, M. The Bayesian lens and Bayesian blinkers. *Philos. Trans. R. Soc. A* **2023**, *381*, 20220144. [[CrossRef](#)]
41. Symonds, M.R.E.; Moussalli, A. A brief guide to model selection, multimodel inference and model averaging in behavioural ecology using Akaike’s information criterion. *Behav. Ecol. Sociobiol.* **2011**, *65*, 13–21. [[CrossRef](#)]
42. Guzmán-Santiago, J.C.; Aguirre-Calderón, O.A.; Vargas-Larreta, B. Forest volume estimation techniques with special emphasis on the tropics. *Rev. Chapingo Ser. Cienc. For. Y Del Ambiente* **2020**, *26*, 291–306. [[CrossRef](#)]
43. Tong, H. A personal journey through time series in Biometrika. *Biometrika* **2001**, *88*, 195–218. [[CrossRef](#)]
44. Li, R.Z.; Fang, K.T.; Zhu, L.X. Some QQ probability plots to test spherical and elliptical symmetry. *J. Comput. Graph. Stat.* **1997**, *6*, 435–450.

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.