LSTM-Based Prediction of Airport Aircraft in and Outflow

Baoqiang Li, Jin Huang(✉), Yamei Duan, and Yuan Zhao

Civil Aviation Flight University of China, Guanghan 618300, China

Abstract. The prediction of airport inbound and outbound traffic is a hot research direction in civil aviation air traffic management. Using the historical data of air traffic as the data source of the traffic prediction model, the traffic data are processed and machine learning algorithm models such as Support Vector Machine (SVM) linear regression, LSTM (Long Short-Term Memory) recurrent neural network, and BP neural network are used to predict the air traffic. The experiments, analysis, and generalization of relevant machine learning algorithms for air traffic flow prediction are conducted. The experiments show that the prediction results are based on historical airspace traffic data, the LSTM model has the highest accuracy, the SVM linear regression has the second-highest prediction effect, and the BP neural network has a poor prediction effect and insufficient stability. The experimental results demonstrate that the LSTM-based inbound and outbound traffic prediction model can achieve airport traffic prediction based on historical traffic data. The usability and accuracy of the LSTM-based prediction results are illustrated by comparing different algorithms, proving that the LSTM model can be used for future urban air traffic flow prediction. It is demonstrated that the LSTM model can be used for future urban air traffic flow prediction. It provides a theoretical and reference basis for the future air traffic flow management of intelligent urban transportation systems.

Keywords: Airport traffic forecasting · Long and short-term memory networks (LSTM) · Support vector machines (SVM) · Machine learning

1 Introduction

In the coming years, airport construction will intensify, air routes will become more complex and airport traffic will continue to grow. However, the rapid development of China's civil aviation transport and the air traffic management system load-bearing limit of the conflict between the increasingly prominent, airspace congestion and extensive delays have become commonplace. The study of air traffic flow forecasting is of great importance.

Research on airspace traffic forecasting algorithms in China's national airspace began relatively late. The current airspace traffic forecasting algorithms in China's civil aviation industry are divided into short-term traffic forecasting algorithms and medium- and long-term traffic forecasting algorithms. Most of the research focuses on the prediction

X. Wang et al. (Eds.): AICON 2021, LNICST 396, pp. 42–54, 2021.
https://doi.org/10.1007/978-3-030-90196-7_4

of air cargo throughput and terminal passenger flow. For medium and long-term traffic forecasting, common models include multiple regression function fitting, artificial neural network, grey prediction, and other traffic forecasting methods. Short-term forecasts are generally chosen from autoregressive integrated moving average (ARIMA) forecasting models or models such as HoltWinters [1, 2].

Due to the intensive air traffic, the flow increases and causes delays. At the same time, air traffic flow changes are non-linear and time-series in nature. The prediction accuracy of traditional forecasting methods is low. The intelligent technology represented by the neural network has been more widely used in the field of prediction, but the method is influenced by the complexity of the network structure and the sample dimension, thus sometimes the learning or generalization ability is too low.

The core of the time series forecasting problem is to mine time-series data from the data for trends that keep changing over time and use these trend patterns to make predictions about future data [3]. Machine learning-based forecasting methods refer to the use of multiple machine learning methods for forecasting data in time series and are effective in dealing with non-linear time series data. Takashi K [4] used a deep belief network (DBN) consisting of a multilayer restricted Boltzmann machine (RBM) for feature capture of time series, which can be used for approximate or short-term forecasting. Rohitash Cl [5] used Elman neural networks to predict complex time series after decomposing them, which improved the prediction accuracy. Xiao Fan [6] used wavelet transform to decompose the time series and subsequently used a support vector machine to predict and fuse the sets of wavelet coefficients.

In this paper, the LSTM-based model for forecasting airport throughput time-series traffic data is proposed. After obtaining data such as population, short-term traffic data, annual total traffic data, and production values of each industry at the local airport, the data are pre-processed by correlation factor analysis, normalization, and other pre-processing operations, L2 regularisation terms are introduced into the LSTM network model, and the appropriate number of LSTM network layers and the number of hidden neurons in the feedforward network layers are explored experimentally, which can make accurate forecasts of airport throughput.

2 Traffic Data Pre-processing

The airport of a provincial capital city in China was selected as the object of the study by checking relevant information. Through the Tushar package that comes with python, a crawler program was used to select the historical data published in the airport production bulletin to crawl according to the prescribed format, combined with data from the ADS-B airport take-off and landing traffic statistics (Excel format) on the official website of flightradar24. Data for each industry was obtained by compiling information from visits to the annual economic development reports. The basic data include: GDP index, number of local people in passenger traffic, and industrial output as significant factors affecting passenger throughput, and the correlation between each influencing factor and passenger throughput is calculated using the grey correlation method.

Step 1: The significance factors were homogenized and the results were obtained in the following table. (Table 1)

Table 1. Results for each relevant factor

Year	Passenger traffic	GDP	Number of local people	Industrial output
2014	6271701	11218200	3946.91	6077587
2015	7339228	13830700	5250.42	7336821
2016	8746034	17103100	6344.21	9206999
2017	10472589	20854000	6820	11550000
2018	12525537	24975300	7730	14127000
2019	14598527	30265800	8749.18	17198500
Average value	9992269	19707850	6473.453	10916151

Step 2: Remove each of the original series from the above mean values to obtain the averaged series. The averaging is done separately for each indicator in each year.

Step 3: Using the mean value series derived in step 2, calculate the absolute difference between the passenger throughput and the number of passengers, the city's annual GDP, the local population, and the industrial output respectively for the same period. Similarly, the absolute difference between the indicators for each of the remaining years is calculated separately, and the maximum and minimum values are then found from the final results. The maximum value is $\Delta\text{max} = 0.287443784$ and the minimum value is $\Delta\text{min} = 0.001913079$.

Step 4: Calculate the correlation coefficient by taking the resolution factor $\rho = 0.5$ and calculating the formula as follows,

$$\xi_{\sigma}(t) = \frac{\Delta(\min) + 0.5\Delta(\max)}{\Delta_{oi}(t) + 0.5\Delta(\max)} = \frac{0.001913079 + 0.5 * 0.287443784}{\Delta_{oi}(t) + 0.5 * 0.287443784} = \frac{0.145634971}{\Delta_{oi}(t) + 0.14372189}$$

The results of calculating the correlation factors are as follows (Table 2):

Table 2. Relevance factor table

Year	ζ_{01}	ζ_{02}	ζ_{03}	ζ_{04}
2014	0.543969907	0.693844356	0.903049781	0.630949904
2015	0.818740287	0.55229239	0.932667747	0.680083667
2016	0.905959100	0.485204959	0.725522172	0.905162326
2017	0.679300508	0.414776189	0.583754289	0.733389278
2018	0.562927338	0.708439884	0.410303886	0.767843817
2019	0.513227692	0.845412192	0.661525638	0.898629578

3 The Inbound and Outbound Traffic Forecasting Model

3.1 LSTM Neural Network Prediction Model

RNNs (recurrent neural networks) are commonly used to analyze predictive sequence data [7], but research has shown that as time increases, RNNs forget information about previous states. Therefore, this paper introduces LSTM (Long-Short-Time Recurrent Neural Network). LSTM temporal recurrent neural networks have the property of being suitable for processing and predicting important events with long intervals and delays in time series and have been outstanding in many fields in recent years.

The LSTM model is an improved model based on an RNN model that can solve the correlation problem between short and long-term time series by using the hidden layer as a memory unit. Figure 1 gives a diagram of the structure of the memory unit, with the storage unit located at the core of the entire memory unit, indicated by a red circle. The input is the known data, while the output is the predicted result. There are three gates in the memory cell, the input gate, the forgetting gate, and the output gate, identified by the green circles in the diagram. The blue dots represent the convergence points and the dotted lines are the previous state functions. After being gated by different functions, the LSTM memory unit can capture complex correlation properties in both short and long-term time series, with significantly improved performance compared to the RNN model.

In this paper, LSTM networks are applied to data prediction. To optimize the LSTM network prediction effect, this paper focuses on exploring the appropriate number of layers of LSTM networks and the number of hidden neurons in its feedforward network layers to study an effective LSTM prediction network model. The LSTM network model with a different number of layers is used to analyze and forecast the traffic data of a domestic airport from 2014 to 2019 by comparing the real values with the predicted values. Through the experiments, we eventually arrive at the optimal parameters for the LSTM-based airport traffic prediction model.

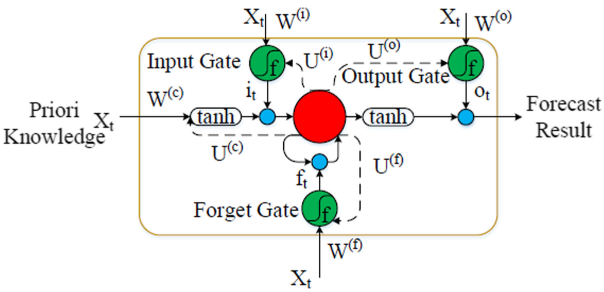

Fig. 1. LSTM model schematic diagram

3.2 SVM Linear Regression Prediction Model

In the 1990s scientists, Vapnik et al. proposed a support vector machine (SVM) algorithm, based on structural risk minimization to find inductive statistical design models to achieve minimum risk generalization, using kernel functions to map data from low-dimensional to high-dimensional space. Dimensional catastrophe and computational complexity are reduced [8].

SVM model input linearly separable training set. $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_M, y_M)\} y_i \in \{-1, 1\}, i = 1, 2, \ldots, M$

SVM model construction steps.

Step 1, constructing the constrained optimization problem

$$\min_{\alpha} \frac{1}{2} \cdot \sum_{i=1}^{M} \alpha_i \alpha_j y_i y_j \left(x_i \cdot x_j\right) - \sum_{i=1}^{M} \alpha_i \tag{1}$$

$$s.t. \sum_{i=1}^{M} \alpha_i y_i = 0$$

$$0 \le \alpha_i \le c, i = 1, 2, \ldots M$$

In step 2, the SMO algorithm is used to solve the optimization problem above to obtain values of the vectors α*.

In step 3, the value w* of the w vector is then calculated using the following equation.

$$w^* = \sum_{i=1}^{M} \alpha_i^* y_i x_i \tag{2}$$

In step 4, find the support vector point $(x_S, y_S), s = 1, 2, \ldots, S$ that satisfies $0 < \alpha_S^* < c$ the corresponding support-vector, and use the following equation to calculate b the value b^*.

$$b^* = \frac{1}{S} \sum_{s=1}^{S} \left[y_s - w^* \cdot x_s\right] \tag{3}$$

In step 5, the segmentation hyperplane $w^* \cdot x + b^* = 0$ and the classification decision function are obtained from $w^* b^*$:

$$h(x) = sign\left(w^* \cdot x + b^*\right) \tag{4}$$

Output segmentation hyperplane $w^* x + b^* = 0$ and classification decision function:

$$h(x) = sign\left(w^* x + b^*\right) \tag{5}$$

Traditional time-series prediction methods based on mathematical statistics do not have self-learning, self-organizing and self-adaptive capabilities, especially for data types with multiple feature dimensions that cannot be effectively fitted and functionally expressed [8, 9]. Support vector machine SVM, as a machine learning method based on statistical learning theory [10, 11], is mainly based on VC dimensional theory and structural risk minimization principles [12] and is also a machine learning algorithm based on geometric distance. Therefore SVM models are sensitive to missing data in the dataset when the sample size is very large. SVMs are prone to lag when the computational volume is too large. The experiments in this paper use the sklearn. SVM module in the Python package library to implement the SVM linear regression model.

3.3 Flow Prediction Model with BP Neural Network

In 1986, Rumelhart first proposed an error backward-corrected multilayer feedback network based on BP neural network, which has good pattern classification ability and multidimensional function mapping ability, and has wide application prospects. BP neural network is mainly divided into three layers: input layer, implicit layer, and output layer, and uses empirical risk minimization and gradient descent methods to calculate the optimal value of the objective function in the form of an approximate function representation.

The BP neural network structure is shown in Fig. 2:

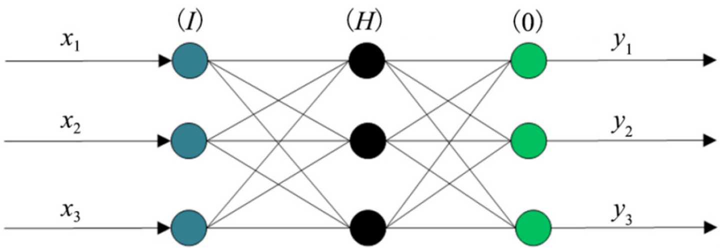

Fig. 2. BP neural network structure

where (I) is the input layer, (H) is the implicit layer, and (0) is the output layer. $\{x_1,x_2,x_3 \ldots x_n\}$ represent the n-dimensional model input units, $\{y_1,y_2,y_3 \ldots y_n\}$ represent the model outputs, and the forward and backward propagation parameters are adjusted with the corresponding weights w and bias terms b of the inputs and outputs. BP neural network as a machine learning algorithm has the advantages of self-learning, self-adaptive and scalable. However, BP neural networks face the complex problem of optimizing the output of the neuron of the objective function to approximate the true value. It is easy to fall into local optimality, and its network structure and neuron requirements are artificially set, so the predictive ability and scalability of BP neural networks need to be further improved.

4 Experimental Studies

The overall experimental procedure was as follows.

(1) Data download: Historical statistics of air traffic on the FR24 website using ADS-B observation-based statistics.
(2) Data pre-processing: The raw data obtained may have disordered and missing values, which need to be interpolated and sorted to obtain regular time-series data.
(3) Data noise reduction: Since the data contains noise due to unstable observation methods, the pywt library in python is used to remove the noise from the data.
(4) Data normalization: As multiple parameters with too large values are fed into the model as feature values at the same time, it is not possible to show a large proportion of influence on the prediction results just because the values of certain indicators are too large, so the feature series need to be normalized.

(5) Fine-tuning of parameters: The network framework structure and the values of the regularisation term parameters in the LSTM layer are continuously adjusted during the training process of the model until the best prediction results are achieved.

4.1 Experimental Procedure

4.1.1 LSTM Neural Network Construction

The GPU version of the Keras framework was built under the Linux operating system. Keras framework is highly encapsulated, modular, simple, easy to extend, and fine-tuning steps, etc. Its core data structure contains two models: one is the Sequential model, and the other is called the Model model. The Sequential model is a series of network layers in a sequential stack, with single input and a single output, and only adjacency between layers.

This experiment uses the Model model to build single-layer and two-layer LSTM network models to analyze and predict future traffic respectively. The model's prediction performance evaluation metrics are used to compare the experimental results using root mean square error (RMSE), mean absolute error (MAE), and model prediction accuracy (accuracy). The RMSE and MAE are calculated by the following equations.

$$RMSE = \sqrt{\frac{1}{N}\sum_{\mathrm{t}=1}^{N}(X_{pridiction,t} - X_{real,t})^2} \tag{6}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|X_{pridiction,i} - X_{real,i}\right| \tag{7}$$

For the predicted LSTM model, to improve the model generalization, this experiment uses the following two ways to avoid overfitting.

First: the experiment uses dropout regularization, in which a portion of the units are randomly selected to be deactivated at a certain deactivation rate in each update of the network training, including input connections and recursive connections, which can effectively prevent overfitting. If a deep LSTM network is used, deactivation regularization can be applied between each layer at the same time, so there are three deactivation parameters in each layer of the network.

Second: the experiment uses the early stop method, dividing the training samples into training and validation sets. In each iteration, the loss values of the training and validation sets are calculated separately, and if the loss value of the validation set no longer decreases within step k, the training is stopped and the model parameters with the lowest validation loss value are returned.

In this paper, two layers of the LSTM neural network are used, so there are six deactivation parameters. In the training set, 80% of the samples are used as the training set, 20% of the samples are used as the validation set, and the number of steps k is set to 50. To observe the prediction effects of different prediction methods for the short, medium, and long term, the last 20, last 60, and last 250 parity data of the overall data set of each index are taken as the test set for short, medium, and long term prediction respectively in this paper, and the test set is excluded as the corresponding training set.

Experiment 1 used the data from 2014–2016 as the training set and the data from 2017–2018 as the test set. After several tests, we set the number of hidden neurons in the feedforward network layer to 10 after weighing the amount of computation against the prediction accuracy of the model. To avoid the overfitting phenomenon, the experiments used L2 regularization terms and the dropout mechanism to improve the generalization ability of the model.

Experiment 2 built a two-layer LSTM network and a fully connected layer model, where the number of hidden neurons in the first LSTM layer was the same as in Experiment 1. Experiment 2 used the same input values and test values as in Experiment 1, and finally, it was compared and analyzed with the prediction results of Experiment 1, which led to good or bad prediction performance.

4.1.2 Tuning of Linear SVM Model Parameters

According to the prediction principle of the support vector machine model, it is known that SVM model parameters mainly include kernel function type, penalty coefficient c, and insensitivity coefficient ε. The determination of this parameter has a great impact on the accuracy of the SVM model, which is a difficult and hot issue in the current support vector machine model research, and this experiment uses grid search which is an exhaustive idea based on the specified parameter value search method.

As the primary means of machine learning models, grid search is optimized by cross-validating the parameters of the estimation function. After selecting the grid search to train each machine learning model for fitting all possible parameter combinations, the phenotype of the model is evaluated using cross-validation to finally obtain the combination of parameters under optimal performance.

In this experiment, comparing the optimization results of the grid search, the efficiency performance of the model is not significantly improved under the optimization of the random search method. Therefore, the grid search method, which is faster in computation, was finally chosen in this paper. The experiments in this paper use the GridSearchCV module in the Python package library sklearn.model_selection to implement grid search.

4.1.3 Parameter Selection for the BP Neural Network Prediction Model

The pre-processed data were divided into training samples p-train and T-train and input to the three-layer BP neural network model, with the transfer functions “transit”, “login”, “purely”, “trail” and “learned”. “purely”, the training function is “trail” and the learning function is “learned”. After repeated training of the BP neural network model, it was found that the best results were obtained when the number of nodes in the hidden layer was 5, the learning rate was 0.8, the number of training times was 500, and the training target error was 0.01.

4.2 Experimental Results and Analysis

4.2.1 Experimental Analysis

The accuracy of the LSTM predicted models varies due to the number of different iterations set. The figure below shows the box plots of the RMSE errors for the LSTM models at 50, 100, 150, 200, 250, 300, 350, and 400 iterations in the experiment. The box plot shows that the mean value of the RMSE at 350 iterations is 19.991, which sets this parameter as the optimal parameter for the LSTM model (Table 3) (Fig.3).

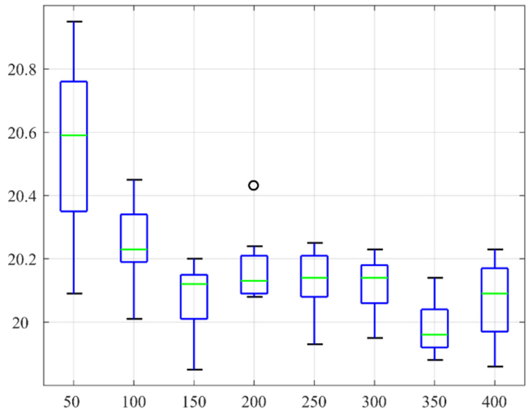

Fig. 3. Box plot of error values for different iteration times

Table 3. Comparison of the results of the improved LSTM algorithm for flight traffic prediction with the traditional single-layer LSTM regression algorithm

Actual flow/sortie	Predicted value		Prediction error/%		Actual flow/sortie	Predicted value	
	Traditional single-layer LSTM	Improved model	Traditional single-layer LSTM	Improved model		Traditional single-layer LSTM	Improved model
156	176	152	12.8	2.5	156	176	152
134	114	136	14.9	1.5	134	114	136
185	195	182	5.4	1.6	185	195	182
247	207	240	16.2	2.8	247	207	240
278	248	287	10.8	3.2	278	248	287
290	261	285	10	1.7	290	261	285

The single-layer LSTM network and fully-connected layer model built-in Experiment 1 to predict throughput, based on separate 2014 and 2015 annual data, with 2014 data as the training set and 2015 data as the test set, was debugged and tested several times in the experiments, and the number of hidden neurons in the feedforward network layer was set to 10 after weighing the computational effort against the prediction accuracy of the model. In Experiment 1, it was mainly demonstrated that the single-layer LSTM network is capable of making predictions, and the model error performance results of its experiments are shown below (Table 4) (Fig. 4)

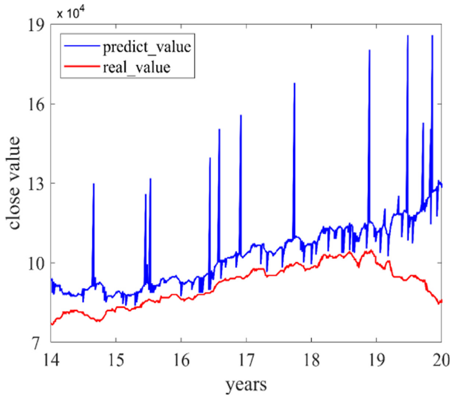

Fig. 4. Prediction diagram of traditional single-layer LSTM network model

Table 4. Prediction results of single-layer LSTM network models

Performance	RMSE	MAE	Accuracy
Numerical values	2.58	1.74	0.44

From the comparative validation in Table 5, it can be intuitively seen that the prediction effect of the single-layer LSTM is not satisfactory, even though the flow prediction results are consistent with the actual trend, but only the trend is judged, and the gap with the actual flow still exists, the prediction value is generally high, and it is necessary to further improve its prediction performance.

Experiment 2 built a two-layer LSTM network and a fully connected layer model, where the number of hidden neurons in the first LSTM layer was the same as in Experiment 1, and the same input and test values were used as in Experiment 1. After pushing to the model for training and testing, the prediction performance was compared and analyzed with that of Experiment 1, and the experimental results are shown in Fig. 5:

By analyzing the results of Experiment 2, it can be found that the two-layer LSTM network model has been greatly improved in terms of prediction performance, and from the above graph comparing the LSTM prediction model's 2018–2020 annual traffic forecast curve with the real airport traffic curve, there is a high degree of fit between the predicted and actual values. The RMSE and MAE values are reduced by 2.12 and 1.632 respectively for the two-layer LSTM compared to the single-layer model, and the prediction accuracy of Experiment 2 is improved by approximately 30% compared to Experiment 1. This shows that the LSTM network model has a clear advantage in the processing of traffic sequence data.

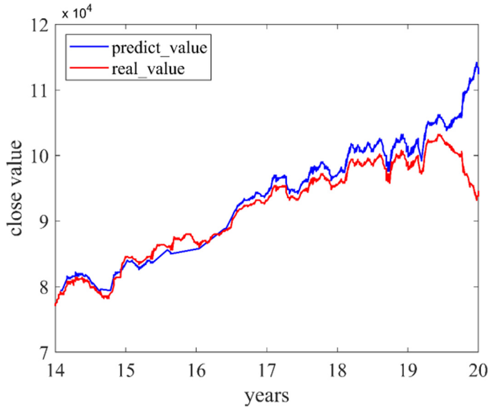

Fig. 5. Two-layer LSTM network model prediction graph

Table 5. Two-layer LSTM network model prediction results

Performance	RMSE	MAE	Accuracy
Numerical values	0.46	0.108	0.78

Furthermore, analysis of the literature shows that as the number of layers increases, the effectiveness of feature extraction improves, as does the prediction accuracy [13]. However, the three-layer LSTM network is not significant in improving the prediction accuracy, only 0.002%, indicating that a continuous increase in the training depth of the network does not consistently improve the prediction performance and may even lead to computational redundancy. Because of the above findings, prediction performance and computational effort must be considered together. Based on the comparative analysis, this paper concludes that the two-layer LSTM network model is suitable for the prediction of airport inbound and outbound traffic.

4.2.2 Comparison of Experimental Results

A comparison of the graphs below shows that there is a significant lag in the prediction results of the SVM linear prediction model. the SVM linear model is unstable and can be partially good or bad, and the BP neural network is worse than the other two algorithms due to the tendency of the BP neural network to over-fit the training set, lose generalization and also under-fit to achieve good prediction results. The blue line shows the LSTM prediction model with relatively good results and a good fit to the true values.

Among the three machine learning prediction models, BP neural network prediction SVM support vector machine prediction LSTM, LSTM neural network prediction model is the most effective, SVM support vector machine prediction is the second most effective, while the traditional BP neural network prediction is the least effective. This shows that LSTM has a clear advantage in air traffic flow prediction (Table 6) (Figs. 6, 7).

Table 6. Forecast comparison table

Year	Actual value	BP neural networks		Support vector machines		LSTM model	
		Predicted value	RE (%)	Predicted value	RE (%)	Predicted value	RE (%)
2017	101076	119613	18.34	119593	18.32	112568	11.37
2018	107930	126375	17.09	117611	8.97	121863	12.91
2019	108275	125003	15.45	119221	10.11	121116	11.86
Average relative error			15.09		12.98		10.75

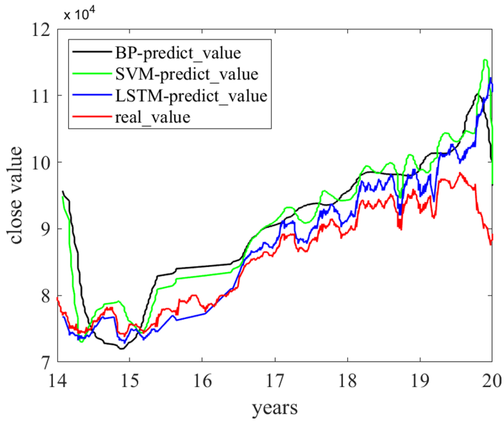

Fig. 6. Comparison of prediction results

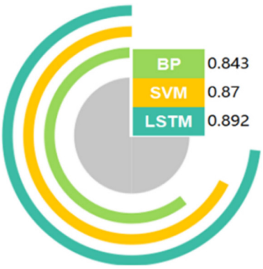

Fig. 7. Prediction accuracy analysis based on machine learning algorithms

5 Conclusion

In this paper, an optimized LSTM deep learning neural network is developed to predict airport traffic by analyzing the intrinsic causal links between airport traffic and its influencing factors. By collecting data related to the throughput of an international airport in China, the local GDP index of a city, and the production value index of various industries into the established LSTM model, the feasibility and accuracy of the LSTM model in airport traffic forecasting is verified by comparing the real values of airport traffic with the traffic forecasting values of various machine learning models such as linear SVM and LSTM. The establishment of the traffic prediction model is affected by various factors, so in future work, instead of simply considering the influence of multiple factors on the traffic prediction results, the interrelationship between various factors should be considered to build a more perfect and accurate traffic prediction model. In 2020, the predicted values deviated significantly from the true values due to the occurrence of a worldwide epidemic 2020, which the authors were unable to take into account due to their limited level.

This study provides example proofs and a basis for ideas for subsequent studies of airport traffic forecasting models. It has important research implications for modeling and simulation of aircraft traffic forecasting and controller and airspace load forecasting.

References

1. Jing, L., Lulu, X.: Comparison and analysis of research hot trend prediction models based on machine learning algorithms-BP neural network, support vector machine, and LSTM model. Modern Inf. **39**(4), 23–33 (2019)
2. Yuanhui, L., Xia, L., Zhipeng, O.: ARIMA model-based passenger traffic forecasting for Sanya airport. Software **7**, 50–55 (2018)
3. Dantas, T.M., Oliveira, F.L.C., Repolho, H.M.V.: Air transportation demand forecast through Bagging Holt-Winters methods. J. Air Transp. Manag. **3**, 116–123 (2017)
4. Takashi, K., Shinsuke, K., Kunikazu, K., et al.: Time series forecasting using a deep belief network with restricted Boltzmann machines. Neurocomputing **137**, 47–56 (2014)
5. Rohitash, C., Mengjie, Z.: Cooperative coevolution of Elman recurrent neural networks for chaotic time series prediction. Neurocomputing **86**, 116–123 (2012)
6. Fan, X., Jiezhong, M., Lankun, R.: Time series prediction based on wavelet analysis and support vector machines. Aviation Computing Technology **41**(6), 49–52 (2011)
7. Li, J., Lin, Y.F.: Time-series data prediction based on multi-timescale RNN. Computer Applications and Software **35**(7), 33–37 (2018)
8. Cristianini, N.: Introduction to Support Vector Machines = An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Electronic Industry Press, Cambridge (2004)
9. Xuegong Z.: Introduction to statistical learning theory and support vector machines. J. Automation (2000)
10. Vapnik, V.N.: The Nature of Statistic Learning Theory. Springer, New York (2000)
11. Wapnik, X.J., Zhang, X.G.: Statistical Learning Theory: Statistical Learning Theory. Electronic Industry Press, Cambridge (2009)
12. An Introduction to Support Vector Machines. Recent Advances and Trends in Nonparametric Statistics **32**(8), 3–17 (2002)
13. Chen, K., Zhou, Y., Dai, F.: A LSTM- based method for stock returns prediction: a case study of China stock market. In: IEEE International Conference on Big Data, pp. 2823–2824 (2015)