



# Trajectory prediction based on long short-term memory network and Kalman filter using hurricanes as an example

Wanting Qin<sup>1</sup> · Jun Tang<sup>1</sup> · Cong Lu<sup>1</sup> · Songyang Lao<sup>1</sup>

Received: 3 December 2019 / Accepted: 20 January 2021 / Published online: 5 March 2021  
© The Author(s) 2021

## Abstract

Trajectory data can objectively reflect the moving law of moving objects. Therefore, trajectory prediction has high application value. Hurricanes often cause incalculable losses of life and property, trajectory prediction can be an effective means to mitigate damage caused by hurricanes. With the popularization and wide application of artificial intelligence technology, from the perspective of machine learning, this paper trains a trajectory prediction model through historical trajectory data based on a long short-term memory (LSTM) network. An improved LSTM (ILSTM) trajectory prediction algorithm that improves the prediction of the simple LSTM is proposed, and the Kalman filter is used to filter the prediction results of the improved LSTM algorithm, which is called LSTM-KF. Through simulation experiments of Atlantic hurricane data from 1851 to 2016, compared to other LSTM and ILSTM algorithms, it is found that the LSTM-KF trajectory prediction algorithm has the lowest prediction error and the best prediction effect.

**Keywords** LSTM · Kalman filtering · One-hot · Trajectory prediction · Hurricane

## 1 Introduction

With the continuous development of satellite navigation, wireless communication and other technologies, mobile intelligent devices with positioning functions are currently widely used. When people use these devices, they also actively or passively record a large number of historical trajectories, leading to the formation of spatiotemporal trajectories [1, 2]. Spatiotemporal trajectory data can accurately record the activity of moving objects over a long period of time and objectively reflect the law of the activity of moving objects. Mobile communication equipment, animal migration, transportation and meteorological clouds are examples of moving objects in specific application fields. Therefore, mining the temporal and spatial patterns contained in the historical trajectories and predicting the

future position of a moving object have high application value. Trajectory prediction of a moving object has become a hot topic in many research fields. Hurricanes, for example, which is a strong and deep tropical cyclone produced in the Atlantic Ocean and the eastern part of the North Pacific Ocean and is also known as a typhoon or cyclone, are a potentially valuable application [3]. Hurricane trajectories are one type of spatiotemporal trajectory. As a common natural phenomenon, hurricanes often cause great losses of life and property. On October 12, 2019, the typhoon ‘Hagibis’ landed on the Izu Peninsula in Shizuoka-ken, Japan. This typhoon killed 88 people, and 7 people went missing. More than 3900 people were affected during this typhoon. In addition, the ‘Hagibis’ typhoon caused 102.73 billion yen of losses in Japan’s agriculture, forestry, fishery and other related industries. It is very important to monitor and record the trajectory of hurricanes and provide support for the analysis and forecast of hurricanes [4].

The moving object studied in this paper is the hurricane. Trajectory prediction, as the most important method to reduce the damage caused by a hurricane, has become a hot issue in the field of trajectory research. The common hurricane prediction involves climate persistence, integrated forecasting and probability forecasting [5, 6]. These

---

✉ Jun Tang  
tangjun06@nudt.edu.cn

<sup>1</sup> College of Systems Engineering, National University of Defense Technology, Changsha 41000, China

prediction methods are complex because of involving many factors such as phase transition, vertical advection and the influence of the boundary layer. DeMaria [7] introduced statistical methods to predict the intensity of hurricanes in the Atlantic Ocean and the East Pacific Ocean, delaying the forecast time from three days to five days. In [8], a new spatiotemporal multivariate function model was proposed to improve prediction. This model regarded the dimensions, accuracy and wind speed of hurricanes as time functions to understand the spatial and temporal trends of the path and intensity of a hurricane.

Machine learning is a part of artificial intelligence. In essence, machine learning enables computers to simulate human learning behaviour, automatically acquire knowledge and skills through learning, continuously improve performance and realize artificial intelligence [9]. The common algorithms of machine learning include regression algorithm, algorithm, support vector machine (SVM), clustering algorithm, dimension reduction algorithm and neural network [10]. Neural networks are currently the most influential algorithms in machine learning. With the popularization and application of new meteorological observation equipment, it is possible to obtain a large amount of ground and upper-air data from meteorological observation stations, as well as satellite and radar detection data. These data are in different formats, such as numbers, text and images, and have spatial and temporal characteristics [11]. Machine learning methods can provide new information regarding how to deeply excavate the physical mechanism of weather changes hidden in these massive data, explore signals that indicate the evolution law of weather and establish a new weather forecasting method. In recent years, it has become a hot spot in the fields of meteorology, mathematics or computer to study the effective weather forecasting methods combined with machine learning. The application of machine learning to hurricane prediction has also achieved some fruitful achievements. In [12], the method of fuzzy c-means clustering was used to classify typhoon trajectories and forecast them, which is suitable for moving objects with fuzzy trajectory boundaries. Song [13] combined big data and data mining technology, trained a long short-term memory (LSTM) neural network, and established a typhoon path prediction model based on machine learning. In [14], a sparse recurrent neural network (RNN) combined with a flexible topological structure was proposed to predict the trajectory of Atlantic hurricanes. The dynamic time warping (DTW) distance between the direction of target hurricanes and other hurricanes in the dataset was determined and compared to predict their future trajectories. Baik [15, 16] used a backpropagation (BP) network to forecast typhoon intensity and compared it with results from the regression method. The results showed that the error of the BP network's

typhoon intensity forecast model is lower than that of the regression method, demonstrating the application prospects of BP networks in typhoon intensity forecasting. The application of machine learning methods to hurricane prediction is still a new research field. These methods have a good ability to deal with nonlinear problems and is especially suitable for solving nonlinear problems with complex internal mechanisms. Therefore, the machine learning methods can be applied in meteorology.

LSTM as a neural network has been widely and used in text sequences or images for a long time. In this paper, an LSTM-KF trajectory prediction algorithm is proposed by combining LSTM and Kalman filter. The simulation results show that the LSTM-KF algorithm has a good effect. The rest of the paper is organized as follows: Section 2 introduces the correlation method of trajectory prediction and the basic mechanism of the neural network. The relevant methods and technologies used in trajectory prediction are described in Section 3. In Section 4, the improved LSTM trajectory prediction algorithm and a trajectory prediction algorithm based on LSTM and Kalman filter (LSTM-KF) are proposed. Finally, Atlantic hurricane data are used in a simulation experiment to verify the rationality of the improved LSTM and the LSTM-KF trajectory prediction algorithm in Section 5.

## 2 Related work

### 2.1 Trajectory prediction method

At present, the active space of moving objects can be divided into restricted movement [17, 18, 31] and free movement [19–22]. Current trajectory prediction methods mainly focus on restricted movement trajectory prediction with certain constraints (such as road networks). Because this kind of trajectory follows specific behaviour habits or motion patterns, useful patterns are easily found and the prediction results are satisfactory regardless of the accuracy or effectiveness. However, in nature, free movement is more frequent than limited movement and is more important to predict, especially for disaster prevention and mitigation. The sampling interval of free movement trajectory data are sparse, and it is difficult to accurately capture the moving direction, turning position and other moving characteristics of moving objects, which makes the prediction of the sparse trajectory in free space difficult in trajectory data research. Hurricane trajectory prediction is a kind of free movement trajectory prediction. According to the prediction cycle, trajectory prediction can be divided into short-term trajectory prediction [23] and long-term trajectory prediction [24].

Trajectory prediction methods can be divided into traditional prediction methods based on a construction motion model [25–28], methods based on frequent pattern mining technology [29–31] and methods based on a machine learning [32–36]. The prediction methods based on the construction of a motion model mainly consider the current movement of moving objects, such as their speed and direction, and matches the predicted movement with the constructed motion function for prediction. Wolfson [25] et al. proposed a spatiotemporal model of moving objects (MOST) that took location as a dynamic attribute and then predicted the location of a mobile object based on linear constraints. Junghans et al. [26] proposed a model and prediction method of a moving area that used the minimum closed box to model the moving area and used linear regression (LR) and recursive motion (RMF) functions to determine the evolution mode of the area.

Trajectory prediction based on frequent pattern mining technology achieves the goal of trajectory prediction by analysing historical data in a moving object database, processing the historical data and mining the matching pattern. Monreale [29] comprehensively considered the spatiotemporal information of moving objects to build a T-pattern tree and mined the motion law of the moving objects. Long [30] mined frequent paths based on an FP growth algorithm that introduced the speed of moving objects into the prediction and proposed a simple and effective trajectory prediction algorithm (E3TP). Kim et al. [31] combined the characteristics of road networks, calculated the similarity between trajectories to search for candidate trajectories, evaluated the historical trajectories stored in a historical mobile database to determine the sub-trajectories with similar trajectories, and finally, predicted a future path by analysing the direction of the candidate trajectories.

Machine learning methods mainly aim at mining the behaviour characteristics of moving objects in a historical trajectory and perform trajectory prediction by improving the Markov model, probability graph model, support vector machine model and neural network model. Based on the hidden Markov model, Qiao [32] proposed an adaptive parameter selection algorithm to improve the adaptability of trajectory prediction in a big data environment. Li [33] introduced the concept of fuzzy trajectory to solve the sharp boundary problem caused by fixed mesh generation and improved the traditional LSTM model to make full use of the proximity and periodic characteristics of the historical trajectory to improve the prediction accuracy of the trajectory position. Leege [34] proposed a path prediction method based on machine learning that combined and sorted the traffic flow of fixed arrival routes of airplanes by their actual flight path and meteorological data and trained the model by using historical data to perform time

prediction. Related technologies used for moving object trajectory prediction are listed in Table 1.

## 2.2 Overview of neural network

Neural networks (also known as artificial neural networks or ANNs) are among most powerful machine learning algorithms. The general idea of a neural network is to adopt learning and training. First, the input is calculated by processing the neurons in the hidden layer and the output layer, which are compared with the expected output. Then, a back-propagation learning algorithm is used to repeatedly correct the connection weight coefficient from the input layer to the hidden layer and from the hidden layer to the output layer. When the error between the output and the expected output of the neural network reaches the preset error convergence standard, the training is stopped to obtain a network model with an improved generalization ability [37].

Figure 1 shows the general structure of a neural network that consists of an input layer, hidden layer and output layer. The input layer is responsible for receiving the signal, the hidden layer is responsible for the decomposition and processing of data, and the final result is integrated into the output layer. The circles in each layer in Fig. 1 represent processing units that can be thought of as simulating a neuron. Neurons are connected by nerve wires, which can be assigned different weights. The core task of neural network is to train a set of nearly perfect weights. Several processing neuron make up a layer, and several layers make up a network, which is known as a neural network. When there is more than one hidden layer of the neural network is called a deep neural network.

The BP neural network was the first neural network described. Werbos [38] first proposed the BP neural network in 1974, and Rumelhart et al. [39] developed the theory of the network. The learning process of a BP neural network consists of the forward propagation of a signal and the backpropagation of error. The cell structure of a BP neural network is shown in Fig. 2. In Fig. 2,  $S_j = \sum_{i=1}^n w_{ij}x_i + b_j$  and  $y_j = f(S_j)$ .  $f(\cdot)$  represents the transfer function,  $W_{ij}$  represents weight,  $b_j$  represents bias. The weight and bias are acquired by the neural network through constant training. In forward propagation,  $y_j$  is the output of neuron. If the actual output does not match the expected output, then the error backpropagation stage is entered. The purpose of backpropagation process is to adjust weights and bias to reduce the error between the output value and the expected value. The backpropagation process is summarized in references [38, 39]. When there is more than one hidden layer of the neural network is called a deep neural network. At present, many advances in image recognition and speech recognition technology are derived from the development of deep neural network.

**Table 1** Comparison of trajectory prediction methods

Method	Restricted movement	Free movement	Tectonic movement model	Frequent pattern mining	Machine learning
Wolfson [25]	✓	✓	✓		
Junghans [26]	✓		✓		
Prevost [27]		✓	✓		
Jeung [28]	✓	✓	✓		
Monreale [29]	✓			✓	
Long [30]	✓	✓		✓	
Kim [31]	✓			✓	
Qiao [32]	✓				✓
Li [33]	✓	✓			✓
Leege [34]		✓			✓

### 3 Trajectory definition and long short-term memory

Trajectory data store large amounts of position information of moving objects at different times. A collection of data in chronological order is called a trajectory  $Trj$ .

**Definition 1** Trajectory. Moving objects move in geospatial space, and their positions change with time. The ordered sequence of  $m$  discrete position points and related auxiliary information is defined as trajectories.

$$Trj = \{(P_1, I_1), (P_2, I_2), \dots, (P_m, I_m)\} \tag{1}$$

where  $P_m$  represents the spatial position information of the  $m$ -th trajectory point and  $I_m$  represents the relevant information of the  $m$ -th trajectory point. Taking hurricane data

as an example,  $I_m$  can represent the maximum sustained wind speed and status of system.

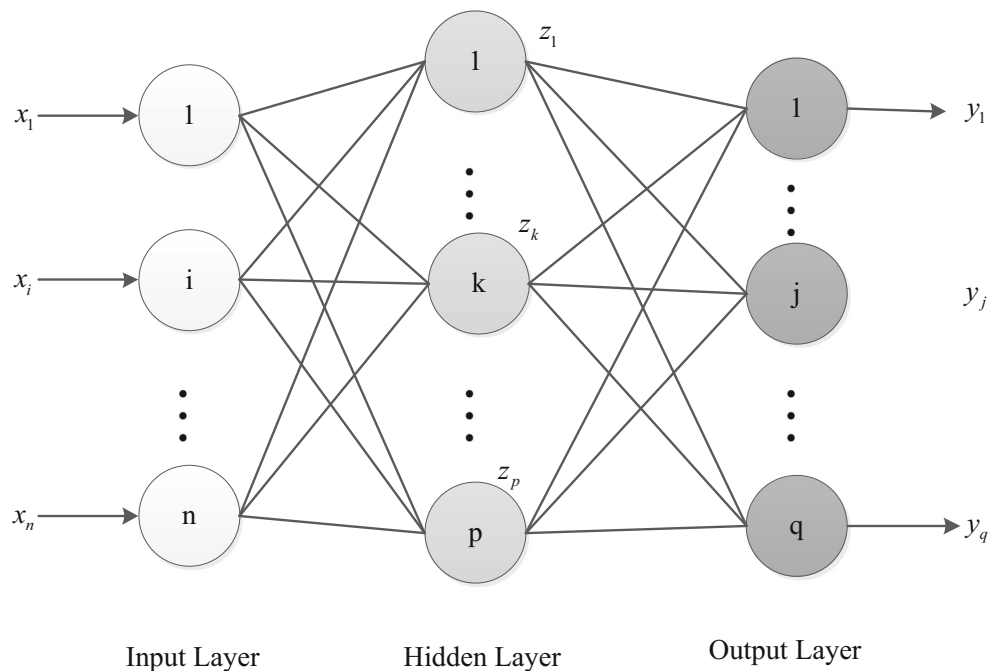
**Definition 2** Subtrajectory. A subtrajectory refers to the ordered set of trajectory points within a certain trajectory, as shown in Eq. 2.

$$Seq = \{(P_m, I_m), (P_{m+1}, I_{m+1}), \dots, (P_{m+k-1}, I_{m+k-1})\} \tag{2}$$

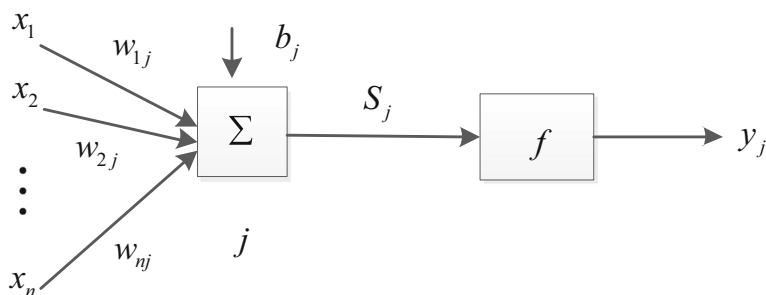
Where  $k$  is the length of subtrajectory.

**Definition 3** Grid trajectory. The geographic space of a moving object is divided into different space areas by fixed grids (such as squares, triangles or hexagons), and a single coordinate point of the trajectory is mapped to the space

**Fig. 1** General structure of a neural network



**Fig. 2** Cell structure of BP neural network



area. The representation of the grid trajectory of the original trajectory  $Trj$  after gridding is shown in Eq. 3:

$$G.Trj = \{(S_1, I_1), (S_2, I_2), \dots, (S_m, I_m)\} \tag{3}$$

where  $S_m$  represents the spatial position information of the  $m$ -th trajectory point mapped to the corresponding grid index on the grid space. As shown in Fig. 3, the corresponding grid trajectory is  $Trj = \{(P_1, I_1), (P_2, I_2), \dots, (P_5, I_5)\}$  is  $G.Trj = \{(S_1, I_1), (S_2, I_2), \dots, (S_5, I_5)\}$ .

The long short-term memory (LSTM) network [40] is a kind of time recurrent neural network (RNN) that is specially designed to solve the long-term dependence problem of the general RNN. LSTM is widely used in temporal sequences now. Trajectory data is also a kind of temporal data, so LSTM can be applied to the trajectory [13, 33]. The cell structure of LSTM is shown in Fig. 4. The cell generally includes a forget gate, an input gate and an output gate. And it is like a memory machine that constantly forgets some information, remembers some information, and takes into account previous input before output. In the cell structure of LSTM,  $x_t$  is the input at current moment,  $h_{t-1}$  is the output at last movement, and  $C_t$  generally referred to as the cell state, which is a cell state of long-term memory.

The forget gate, as the name implies, controls whether information should be forgotten. In LSTM, the forget gate controls how much information about the cell state of

long-term memory at last moment  $C_{t-1}$  can be retained until the present moment, as shown in Eq. 4.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{4}$$

The value of  $f_t$  in each dimension is in the range of (0, 1). The information will be forgotten when  $f_t$  is close to 0, and the information will be retained when  $f_t$  is close to 1. Where  $W_f$  represents weight matrix, and  $b_f$  represents bias.  $\bullet$  represents the matrix product.  $\sigma$  represents a Sigmoid function, as shown in Eq. 5.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

The input gate is responsible for processing the current input and updates new information selectively. The input gate has two steps. In Eq. 6,  $i_t$  can be used to control how much of the current input can be stored. The value of  $i_t$  in each dimension is in the range of (0, 1). In Eq. 7, the tanh function generates a new candidate vector  $\tilde{C}_t$ . The  $\tilde{C}_t$  represents cell state of short-term memory at current moment.

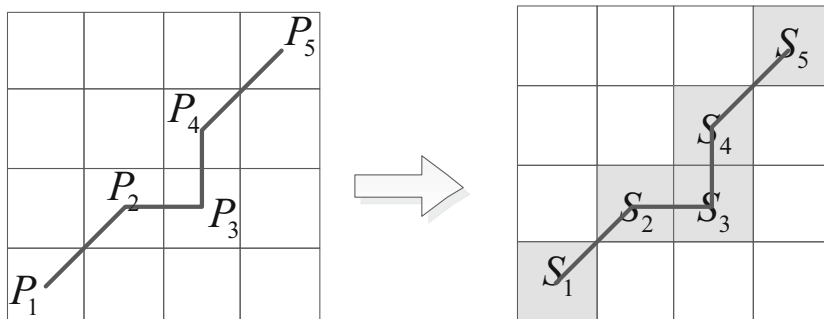
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{6}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{7}$$

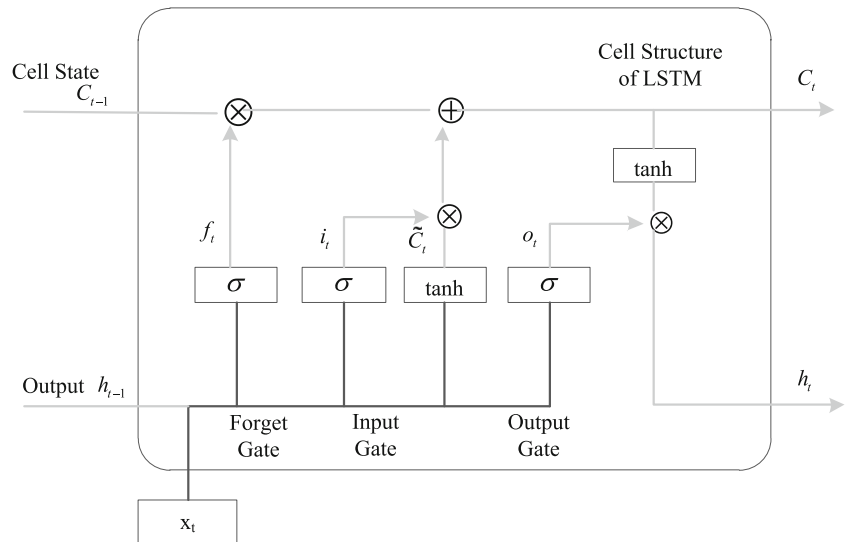
where  $W_i, W_C$  represent weight matrix and  $b_i, b_C$  represent bias. The  $\tanh$  function is as shown in Eq. 8.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{8}$$

**Fig. 3** Gridding of trajectory sequence



**Fig. 4** Cell structure of LSTM



Before output gate, the cell state of long-term memory needs to be updated, as shown in Eq. 9. The cell state of short-term memory at current movement  $\tilde{C}_t$  and cell state of long-term memory at last movement  $C_{t-1}$  are form new cell state.

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

where  $*$  represents the elementwise product.

The output gate is used to selectively output information from the cell. As shown in Eqs. 10 and 11,  $o_t$  is a probability that controls how much of  $C_t$  can be used as the current output. The value of  $o_t$  in each dimension is in the range of (0, 1).  $h_{t-1}$  is the output at time  $t$  and it is hidden state.

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

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

where  $W_o$  is weight matrix and  $b_o$  is bias.  $W_f, W_i, W_t, W_o$  and  $b_f, b_i, b_t, b_o$  represents respectively weights and bias of different gate. These two parameters are obtained by continuous training of LSTM network using historical training data. The notations referred to in Section 3 are listed in Table 2.

## 4 Proposed LSTM-KF

### 4.1 Encoding of trajectory data

Hurricane trajectory data should be converted into data that can be received by the LSTM network through trajectory data encoding. In this paper, the latitude and longitude coordinates, maximum sustained wind speed and status of system of hurricane are selected. When the hurricane trajectory is meshed, the size of the mesh blocks is  $1 \times 1$

degree. Because of the spherical nature of the earth, the area of each grid block is not uniform across square miles, but since most points surround the earth’s equator, the size difference between each grid block is negligible. The grid trajectory, as shown in Fig. 3, is taken as the input of LSTM. The encoding of trajectory data can be divided into three steps:

Step 1: One-hot encoding for categorical data. The grid index  $S_m$  in the grid trajectory can be one-hot encoded. The status of system of hurricane data is also a kind of categorical data, which can also be one-hot encoded. The status of system has nine types, which can also be one-hot encoded, such as tropical cyclone of tropical depression intensity (TD), tropical cyclone of tropical storm

**Table 2** Nnotations in Section 3

Notation	Description
$Trj$	Trajectory
$Seq$	Subtrajectory
$G.Trj$	Grid trajectory
$\sigma$	Sigmoid function
$\tanh$	$\tanh$ function
$W_f, W_i, W_t, W_o$	Weight matrix
$b_f, b_i, b_t, b_o$	Bias
$f_t, i_t, o_t$	Output of the forget gate, input gate and output gate, between 0 and 1
$\bullet$	matrix product
$*$	elementwise product
$C_t$	Cell state of long-term memory
$\tilde{C}_t$	Cell state of short-term memory
$h_t$	Hidden state output at time $t$



intensity(TS), tropical cyclone of hurricane intensity(HU), extratropical cyclone(EX), subtropical cyclone of subtropical depression intensity(SD), subtropical cyclone of subtropical storm intensity(SS), a low that is neither a tropical cyclone (LO), tropical wave(WV) and disturbance(DB). One-hot encoding is the most commonly used and basic coding method to convert tags into vectors. One-hot encoding first requires categorical values to be mapped to integer values. Each integer value is then represented as a binary vector with zero values except for the index position, which has a value of 1. One-hot encoding diagram of grid index is shown in Fig. 5a and one-hot encoding diagram of status of system is shown in Fig. 5b.

Step 2: The numerical data are standardized so that the range of numerical data is between [0 1]. To eliminate the dimensional impact between indicators, data standardization is needed to solve the comparability between data indicators. After the original data are standardized, each index is on the same order of magnitude, which is suitable for a comprehensive comparative evaluation. The numerical data in  $I_m$  are standardized by Eq. 12. The maximum sustained wind speed of hurricane data is numerical data, so it can be normalized between [0,1].

$$z' = \frac{z - \min(z)}{\max(z) - \min(z)} \tag{12}$$

Step 3: Connect the categorical data and numerical data to form a complete trajectory vector. The transformation from grid trajectory to a trajectory vector is realized by one-hot encoding and standardization. The trajectory data coding process is shown in Fig. 6.

### 4.2 Improved LSTM (ILSTM)

The simple LSTM trajectory prediction algorithm selects the subtrajectory with a sequence length of  $K$ . Then, the subtrajectories are meshed and encoded into the LSTM network to predict the index of  $(K + 1)$ -th grid. The flow diagram of the simple LSTM trajectory prediction algorithm is shown in Fig. 7. The simple LSTM trajectory prediction algorithm is divided into three modules: data preprocessing, model training and prediction. The data preprocessing modules involves gridding and coding the trajectory data to produce an acceptable trajectory vector for LSTM. The training module is to finds the best LSTM prediction model. The prediction module predicts the next grid index according to the trained prediction model.

The LSTM network can learn historical knowledge, but it may not be suitable for learning new knowledge. For example, the area where the hurricane trajectory appears may not have occurred in history or occurred only a few times, so the LSTM network may produce a large error between the predicted position and the real position due to insufficient learning. At the same time, there are few feature data available for hurricane data, and the Atlantic hurricane data used in this paper only include coordinates, maximum sustained wind speed, status of system, so a lack of features is also an important factor that restricts neural network learning.

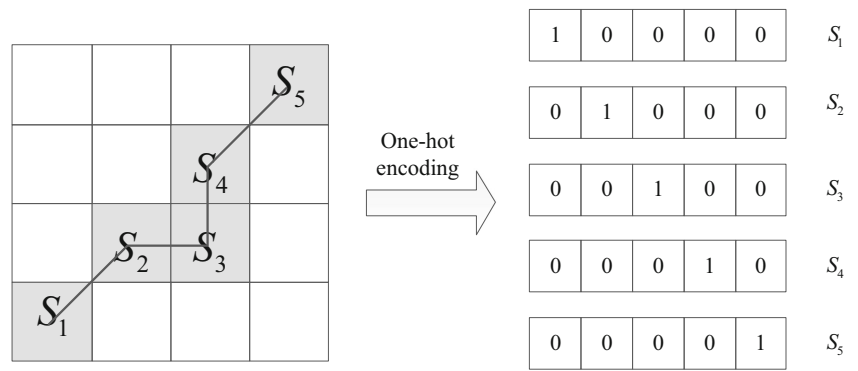
On average, hurricanes move at speeds ranging from 15 km/h to 20 km/h and can be as fast as 30 km/h according to hurricane trajectory analysis [41, 42]. With data intervals of 6 h, the distance a hurricane can travel is limited. The result of the simple LSTM trajectory prediction algorithm selects the grid corresponding to the highest probability value as the prediction position. The prediction of a single position may be take extreme case as the prediction result, that is, the predicted grid position significantly deviates from the normal position range. If the hurricane position predicted by the simple LSTM trajectory prediction algorithm exceeds a certain limit, then it needs to be corrected. In this case, we improve the simple LSTM trajectory prediction algorithm. We select a reasonable position from the candidate set by predicting multiple possible positions and filter out positions with a high probability of being extreme positions. The ILSTM trajectory prediction algorithm is an improvement of the prediction module of the simple LSTM algorithm. And it is consistent with the simple LSTM algorithm in regard to the data preprocessing module and model training module. First, the following definitions are given:

**Definition 4** Subtrajectory marker point *seq\_flag*. The last point in subtrajectory  $Seq = \{(P_m, I_m), (P_{m+1}, I_{m+1}), \dots, (P_{m+k-1}, I_{m+k-1})\}$  is taken as the mark point of the subtrajectory, that is,  $seq\_flag = (P_{m+k-1}, I_{m+k-1})$ .

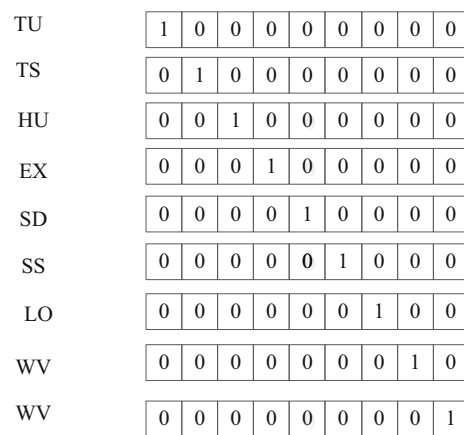
**Definition 5** Grid centre coordinate  $P_{s_i\_central}$ , represents the coordinate corresponding to the centre point of grid  $S_i$ , that is,  $(lon_{s_i\_central}, lat_{s_i\_central})$ .

Multiple grids are selected as a candidate set, and the order of the candidate set is sorted according to probability predicted by the simple LSTM. That is, the grid corresponding to the first *top\_N* probability value is selected as the candidate set  $C = \{S_1, S_2, \dots, S_{top\_N}\}$  of the prediction results. The first grid whose Euclidean distance (13) from the centre coordinates to the coordinates of *seq\_flag* is less than the threshold  $\tau$  is selected as predicted grid from candidate set  $C$ . Then, the centre coordinates of the grid selected

**Fig. 5** One-hot encoding diagram of a grid trajectory

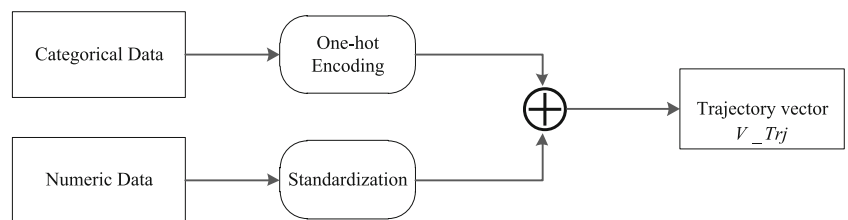


(a) One-hot encoding diagram of grid index

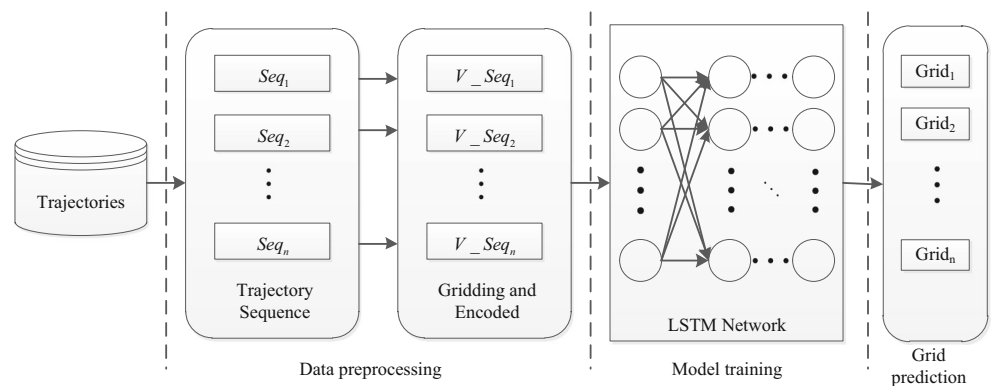


(b) One-hot encoding diagram of status of system

**Fig. 6** Schematic diagram of the trajectory data encoding process



**Fig. 7** Flow diagram of the simple LSTM trajectory prediction algorithm





from the candidate set are the predicted coordinates. If there is no grid located less than threshold  $\tau$  in candidate set  $C$ , then the grid centre coordinates of  $seq\_flag$  are taken as the predicted coordinates in this paper. That is,  $(lon_{seq\_flag\_central}, lat_{seq\_flag\_central})$  are used as the coordinates of the predicted trajectory. The selection of  $\tau$  is based on experience. The ILSTM trajectory prediction algorithm modifies the position of predicted extreme cases in a few cases. In most cases, the first grid in the candidate set satisfies the conditions. The improved LSTM trajectory prediction algorithm is shown in Fig. 8.

$$dist = \sqrt{(lon_{s_i\_central} - lon_{seq\_flag})^2 + (lat_{s_i\_central} - lat_{seq\_flag})^2} \tag{13}$$

### 4.3 Trajectory prediction based on LSTM and Kalman filter

The Kalman filter [43], as a data processing technology that removes noise and restores real data, has been applied in fields such as communication, navigation, guidance and control. Kalman filtering estimates the state of a system optimally through the observation of the input and output. Because the observation data include the influence of system noise and external interference, the optimal estimation can also be regarded as a filtering process.

**Definition 6** Prediction trajectory sequence  $Z$ . For a trajectory  $Trj$  with length  $n$ ,  $step$  is used as the extraction interval of the subtrajectory and  $k$  is the extraction length of the subtrajectory, and the coordinates

of the next point of each subtrajectory are predicted. If  $step = 1$ , then  $n - k$  subtrajectory sequences will be obtained, so  $n - k$  predicted coordinates will also be generated. These sequences generated by the predicted coordinates are called predicted trajectory sequences  $Z = \{(lon_1, lat_1), (lon_2, lat_2), \dots, (lon_{n-k}, lat_{n-k})\}$ .

In this paper, it is assumed that the generated predicted trajectory sequence  $Z$  by the ILSTM method is a kind of observation data. Then, the Kalman filter will filter the predicted trajectory sequence to produce a more accurate and optimal estimation.

**Definition 7** Trajectory marker point  $Trj\_flag$ . The subtrajectory marker point  $seq\_flag$  of the first subtrajectory sequence extracted from each trajectory is taken as the trajectory marker point  $Trj\_flag$  of the whole trajectory.

The Kalman filter is used to filter the predicted trajectory sequence. First, the state vector of the hurricane movement is defined. In this paper, the state vector of a hurricane, as shown in Eq. 14, is used.

$$X(m) = [lon_m, lat_m, V_{lon}, V_{lat}]^T \tag{14}$$

where  $V_{lon}, V_{lat}$  represent velocity components in longitude and latitude, respectively. In hurricane trajectory data, there is no hurricane velocity information at the current moment, so for convenience we set the initial moment  $V_{lon}, V_{lat}$  to 0. In this paper, when the Kalman filter is used, the system state vector of the trajectory mark point  $Trj\_flag$  is input into the system as the initial value, that is,  $X(0) = [lon_{Trj\_flag}, lat_{Trj\_flag}, 0, 0]^T$ .

Fig. 8 ILSTM trajectory prediction algorithm

---

**Algorithm: Improved LSTM algorithm (ILSTM)**

---

```

Input: Subtrajectory sequence to be predicted seq ;
Trained LSTM model model_lstm ;
Subtrajectory marker point seq_flag ;
top_N ,
Output: P Predicted coordinates
01: V_seq = Gridding_Encoding(seq )
/*Gridding and encoding the subtrajectory sequence*/
02: C = model_lstm.predict(V_seq , top_N )
/*Select the top_N grids with the highest probability as the candidate set C */
03: for grid in C:
04: P = get_centralCoord(grid)
/* Obtain the coordinates of the centre point corresponding to the grid */
05: if distance(P , seq_flag) <  $\tau$ 
06: return P
07: return the centre coordinate of grid seq_flag
    
```

---

Kalman filtering requires a discrete control process system. The system state equation (15) and observation equation (16) of hurricane movement are as follows:

$$X(m+1) = AX(m) + W(m) \quad (15)$$

$$Z(m) = HX(m) + V(m) \quad (16)$$

where  $X(m)$  represents the system state vector and describes the state vector of the hurricane at time  $m$  (14).  $A$  represents the state transition matrix, which is used to describe the motion state transition mode from the previous time to the current time.  $W(m)$  represents the system noise, whose statistical characteristics are similar to white noise or Gaussian noise.  $Z(m) = [lon_m, lat_m]^T$  is the observation vector. In this paper, the prediction trajectory sequence  $Z$  is used to represent the observation sequence, which represents  $m$ -th predicted value.  $H$  is the observation matrix, and  $V(m)$  is the observation noise.

The core of the Kalman filter algorithm uses a recursive algorithm to achieve the optimal state estimation model and update the current state variables by using the previous estimated value and the current observed value. The state equation of the system is used to estimate the state of the system at the next time point. If the current time of system is  $m$ , then it can be estimated to the current state based on the previous state according to the state equation of the system.

$$x(m|m-1) = AX(m-1) + W(m) \quad (17)$$

where  $X(m|m-1)$  is the previous state estimation and  $X(m-1|m-1)$  is the previous state optimal estimation. The system result has been updated, but the covariance corresponding to  $X(m|m-1)$  has not been updated. Equation 18 represents the covariance.

$$\rho(m|m-1) = A\rho(m-1|m-1)A^T + Q \quad (18)$$

where  $\rho(m|m-1)$  is the covariance of  $X(m|m-1)$ .  $\rho(m-1|m-1)$  is the covariance of  $X(m-1|m-1)$ .  $A^T$  is the transpose matrix of  $A$ .

The present state  $X(m|m-1)$  has been estimated. Then, we obtain the measured values of the current state, that is, the trajectory coordinates  $Z(m)$  predicted by ILSTM algorithm. Combining the estimated value and the measured value, we can obtain the optimal estimated value  $X(m|m)$  at time  $m$ , as shown in Eq. 19.

$$X(m|m) = X(m|m-1) + K(Z(m) - HX(m|m-1)) \quad (19)$$

where  $K$  is the Kalman gain matrix that is shown in Eq. 20. In this paper,  $K$  is a  $4 \times 2$  matrix.

$$K = \rho(m|m-1)H^T(H\rho(m|m-1)H^T + R)^{-1} \quad (20)$$

Until now, we produced the best estimate value  $X(m|m)$  at time  $m$ . However, to keep the Kalman filter running

until the end of the system process, we need to update the covariance of  $X(m|m)$  at time  $m$ .

$$\rho(m|m) = (I - KH)\rho(m|m-1) \quad (21)$$

So far, the entire Kalman filtering process of trajectory prediction has been completed. The function of the Kalman filter in this paper is to estimate the optimal trajectory coordinates predicted by ILSTM algorithm. The notations referred to in Section 4 are listed in Table 3.

## 5 Experimental results and discussion

### 5.1 Description of Data set

Atlantic hurricane data released by the National Hurricane Centre of the United States ([www.nhc.noaa.gov/data](http://www.nhc.noaa.gov/data)) is used in the experiment. The dataset includes the longitude and latitude, maximum sustained wind speed, minimum central pressure, time and other information of hurricanes. The sampling interval of this data set is 6 h. This experiment mainly deals with the latitude and longitude, maximum sustained wind speed and the status of the system. The 165 years of hurricane data from 1851 to 2015 are selected as training sets to train the LSTM prediction model. The hurricane data from 1851 to 2015 include 1780 hurricane trajectories and 49139 sampling points. First, the dataset is preprocessed, and the missing data are filled with the linear interpolation method. The trajectory map of Atlantic hurricanes from 1851 to 2015 is shown in Fig. 9 (East longitude is represented by positive values and west longitude is represented by negative values).

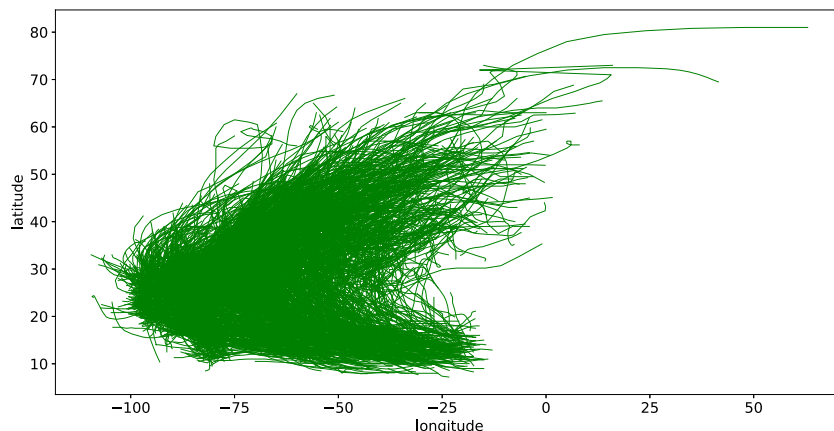
### 5.2 Prediction error comparison experiment

The real trajectories are  $T = \{Trj_1, Trj_2, \dots, Trj_n\}$ . The predicted trajectories are  $T = \{Trj'_1, Trj'_2, \dots, Trj'_n\}$ .

**Table 3** Notations in Section 4

Notation	Description
$seq\_flag$	Subtrajectory marker point
$P_{s\_central}$	Grid centre coordinate
$\tau$	Distance threshold
$Z$	Prediction trajectory sequence
$Trj\_flag$	Trajectory marker point
$X(m)$	State vector of hurricane movement
$K$	Kalman gain matrix
$A$	State transition matrix
$H$	Observation matrix
$Q$	System noise covariance matrix
$R$	Observation noise covariance matrix

**Fig. 9** Atlantic hurricane trajectories from 1851 to 2015



The geometric space error between the predicted trajectory point and the actual trajectory point is used as the prediction error. As shown in Eq. 22, the root mean square error (*RMSE*) unit is degrees.

$$RMSE = \frac{1}{n} \sum_{i=1}^n \frac{\sum_{k=1}^m \sqrt{(lon_{ik} - lon'_{ik})^2 + (lat_{ik} - lat'_{ik})^2}}{m} \tag{22}$$

Here, *n* represents the number of predicted trajectories and *m* represents the number of predicted trajectory points contained in each predicted trajectory.

In this experiment, hurricanes in 2016 are used as the verification set. The prediction errors of the simple LSTM algorithm, ILSTM algorithm and LSTM-KF algorithm are compared. The setting of experimental parameters is shown in Table 4. Figures 10 and 11 show the change of RSME value with the neural network training epochs. Figure 10 indicates that *Q* is [1,0,0,0; 0,1,0,0; 0,0,1,0; 0,0,0,1], and Fig. 11 indicates that *Q* is [2,0,0,0; 0,2,0,0; 0,0,2,0; 0,0,0,2].

**Table 4** Parameter settings of the prediction error comparison experiment

Parameter	Value
Network structure of LSTM	128*128*256; 128*256*512; 256*256; 256*256*256
Distance threshold $\tau$	6
<i>top_N</i>	20
State transition matrix <i>A</i>	[1,0,1,0; 0,1,0,1; 0,0,1,0; 0,0,0,1]
Observation matrix <i>H</i>	[1,0,0,0; 0,1,0,0]
System noise covariance matrix <i>Q</i>	[1,0,0,0; 0,1,0,0; 0,0,1,0; 0,0,0,1] or [2,0,0,0; 0,2,0,0; 0,0,2,0; 0,0,0,2]
Observation noise covariance matrix <i>R</i>	[0.3,0; 0, 0.3]

The structure of LSTM network shown in Figs. 10a and 11a is 128 \* 128 \* 256, the structure of LSTM network shown in Figs. 10b and 11b is 128 \* 256 \* 512, the structure of LSTM network shown in Figs. 10c and 11c is 256 \* 256, and the structure of LSTM network shown in Figs. 10d and 11d is 256 \* 256. It can be seen from Figs. 10 and 11 that the RMSE of the simple LSTM trajectory prediction algorithm is initially large. The LSTM network does not learn enough at the initial moment, and the prediction error of simple LSTM algorithm is initially large. However, the RMSE of the ILSTM and LSTM-KF trajectory prediction algorithms are significantly lower than those of the simple LSTM. With an increase in the LSTM network training epochs, the RMSE of the simple LSTM algorithm gradually decreases and becomes stable. However, the prediction error of the simple LSTM is also significantly higher than that of the ILSTM and LSTM-KF trajectory prediction algorithms.

To better display the RMSE of the three methods, the RMSEs of epoch 71 to epoch 80 shown in Figs. 10b and 11b are listed in Table 5. The system noise covariance matrix *Q* in Figs. 10b and 11b are [1,0,0,0; 0,1,0,0; 0,0,1,0; 0,0,0,1] and [2,0,0,0; 0,2,0,0; 0,0,2,0; 0,0,0,2], respectively. The Network structure of LSTM in Figs. 10b and 11b are both 128\*256\*512. The observation matrix *H* in Figs. 10b and 11b are both [1,0,0,0; 0,1,0,0]. The observation noise covariance matrix *R* in Figs. 10b and 11b are both [0.3, 0, 0, 0.3]. It can be seen from Table 5 that the RMSE of each epoch of LSTM-KF is smaller than that of ILSTM and simple LSTM. In other words, the prediction effect of LSTM-KF should be the best of the three methods. In Fig. 10b, the average prediction error of the simple LSTM from epoch 71 to epoch 80 is 2.1065°, that of the ILSTM is 1.4690°, and that of the LSTM-KF is 1.4605°. In Fig. 11b, the average prediction error of the simple LSTM from epoch 71 to epoch 80 is 2.3865°, that of the ILSTM is 1.5141°, and that of the LSTM-KF is 1.5017°. Therefore, the prediction

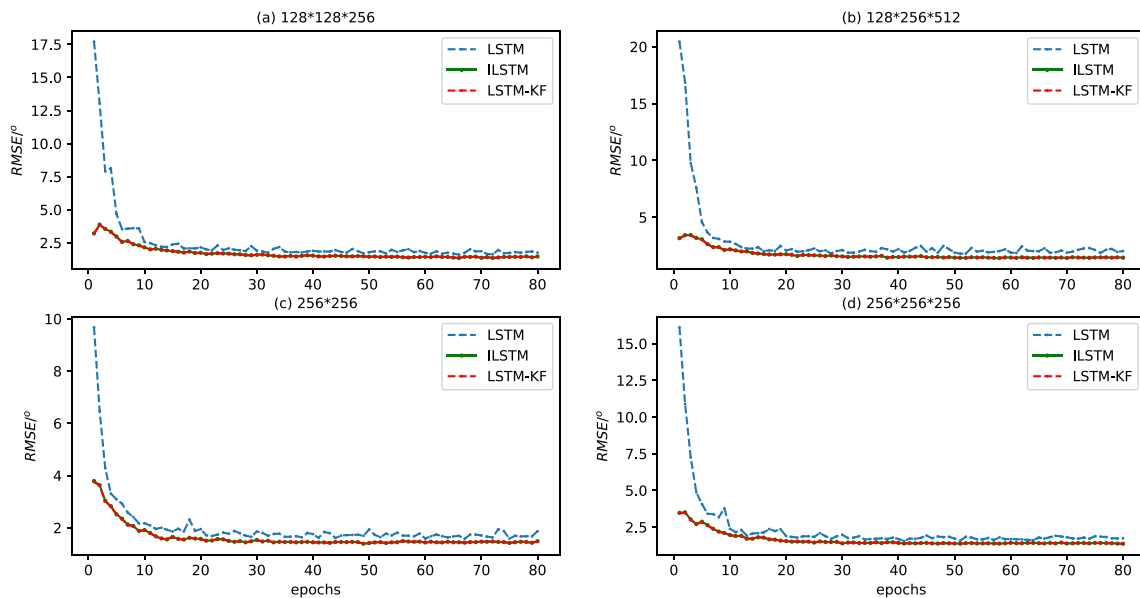


Fig. 10 Change of the RMSE with LSTM network training epochs when  $Q = [1,0,0,0; 0,1,0,0; 0,0,1,0; 0,0,0,1]$

error of LSTM-KF is the smallest of the three methods. The prediction error of the simple LSTM trajectory prediction algorithm is approximately  $2^\circ$ , and its effect is the worst of the three methods. However, the prediction error of LSTM-KF is slightly smaller than that of ILSTM, which shows that the trajectory coordinates predicted by LSTM-KF are more precise than those predicted by ILSTM. It can be concluded that LSTM-KF has the best prediction effect of the three methods.

### 5.3 Comparison experiment of single hurricane trajectory prediction

In this experiment, five hurricanes named ‘KARL’, ‘MATTHEW’, ‘NICOLE’, ‘BONNIE’ and ‘ALEX’ are selected from hurricanes that occurred in 2016. The simple LSTM, ILSTM and LSTM-KF algorithms are used to predict the trajectories of these hurricanes. The experimental parameters are set as shown in Table 6.

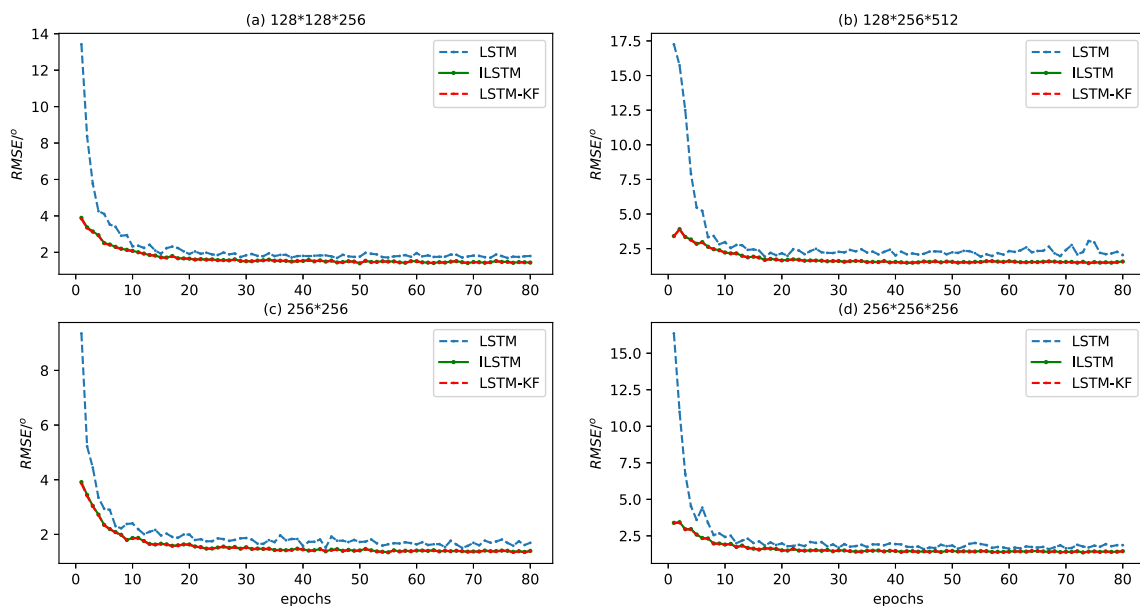


Fig. 11 Change of the RMSE with LSTM network training epochs when  $Q = [2,0,0,0; 0,2,0,0; 0,0,2,0; 0,0,0,2]$

**Table 5** RMSEs of epoch 71 to epoch 80 in Figs. 10b and 11b (4 decimal places are reserved)

Epochs	Fig. 10b			Fig. 11b		
	LSTM	ILSTM	LSTM-KF	LSTM	ILSTM	LSTM-KF
71	1.9795	1.4897	1.4854	2.7606	1.5401	1.5268
72	2.1406	1.4681	1.4584	2.0849	1.4859	1.4807
73	2.2656	1.4557	1.4515	2.2552	1.5472	1.5331
74	2.3020	1.4398	1.4359	3.0540	1.4551	1.4370
75	2.1322	1.4748	1.4669	2.9492	1.5269	1.5167
76	1.8742	1.4817	1.4717	2.1975	1.4902	1.4782
77	2.1561	1.4827	1.4731	2.1008	1.5131	1.4955
78	2.2247	1.4572	1.4523	2.1516	1.4858	1.4751
79	1.9575	1.4850	1.4772	2.2730	1.5291	1.5167
80	2.0322	1.4554	1.4373	2.0382	1.5680	1.5569
Averaging	2.1065	1.4690	1.4610	2.3865	1.5141	1.5017

First, the simple LSTM, ILSTM and LSTM-KF trajectory prediction algorithms are used to predict hurricanes KARL and MATTHEW. The results are compared in Figs. 12 and 13 respectively (the trajectory sequence used for training is not shown in the figures.). In Figs. 12 and 13, the blue dotted line represents the original trajectory, the black dotted line represents the trajectory predicted by the simple LSTM algorithm, the green dotted line represents the trajectory predicted by the ILSTM algorithm, and the red line represents the trajectory predicted by the LSTM-KF algorithm. From Figs. 12 and 13, the simple LSTM, ILSTM and LSTM-KF all show good prediction results, almost according to the change trend of the original trajectory, which shows that according to the learning of the history track, the neural network still has great advantages. While LSTM is effective, ILSTM and LSTM-KF produce better prediction results.

Then, the simple LSTM, ILSTM and LSTM-KF trajectory prediction algorithms are used to predict hurricanes NICOLE, BONNIE and ALEX. The results are shown in Figs. 14, 15 and 16, respectively. From Figs. 14, 15 and

16, some predicted points of the simple LSTM algorithm deviate significantly from the original trajectory. Figure 15 shows the last two predicted points of the elliptical circle in hurricane NICOLE, and Fig. 16 shows the predicted points of the elliptical circle in hurricane BONNIE. These predicted trajectories deviated significantly from the original hurricane orbit. In Fig. 16, the parts marked by the boxes ‘Contrast 1’ and ‘Contrast 2’ indicate that the predicted trajectory points of the simple LSTM algorithm show obvious confusion, while the predicted trajectory points of ILSTM and LSTM-KF basically follow the original trajectory. Therefore, there are some defects in the simple LSTM trajectory prediction. The prediction accuracy of the simple LSTM trajectory prediction algorithm is greatly reduced without adequate learning. The simple LSTM trajectory prediction algorithm has a good advantage for learning historical knowledge, but for time-series data with fewer sample data and high mutation, the simple LSTM algorithm easily produces a large range of prediction errors. The ILSTM and LSTM-KF algorithm, which modify the simple LSTM algorithm, show better robustness and can reduce the errors of the simple LSTM algorithm. It can be seen from Table 5 that ILSTM performs a rough prediction of trajectory coordinates, while LSTM-KF performs a more precise prediction of trajectory coordinates.

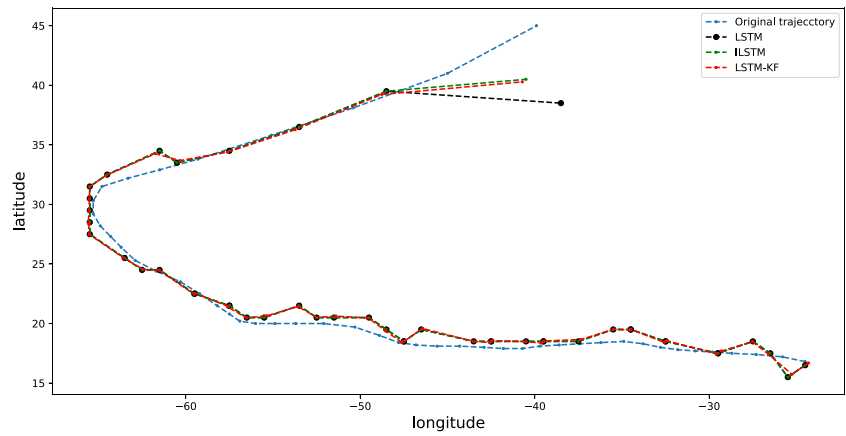
**Table 6** Parameter settings of comparison experiment of single hurricane trajectory prediction

Parameter	Value
Network structure of LSTM	256*256*256
Distance threshold $\tau$	6
$top\_N$	20
State transition matrix $A$	[1,0,1,0; 0,1,0,1; 0,0,1,0; 0,0,0,1]
Observation matrix $H$	[1,0,0,0; 0,1,0,0]
System noise covariance matrix $Q$	[1,0,0,0; 0,1,0,0; 0,0,1,0; 0,0,0,1]
Observation noise covariance matrix $R$	[0.3,0; 0, 0.3]

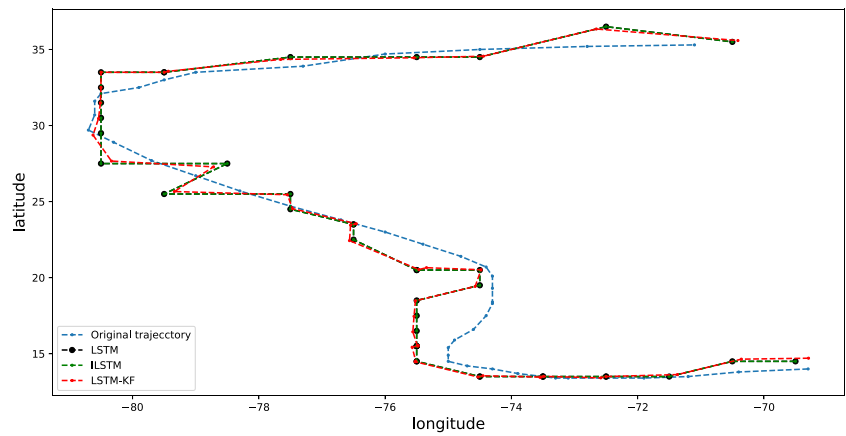
### 5.4 Parameter sensitivity experiment and time performance analysis

In order to verify the influence of different system noise covariance matrix  $Q$  and observation noise covariance matrix  $R$  on the prediction results of LSTM-KF algorithm, different experimental parameters  $Q$  and  $R$  are selected for the experiment. The  $RMSE$  difference between ILSTM and LSTM-KF is used to measure the impact of  $Q$  and  $R$  on the

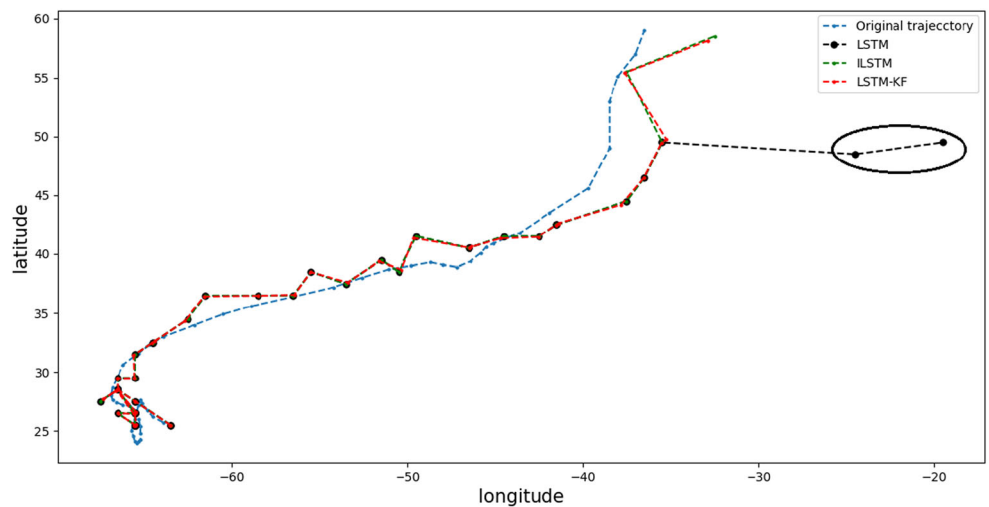
**Fig. 12** Comparison of the trajectory prediction results of the simple LSTM, ILSTM and LSTM-KF trajectory prediction algorithms for hurricane KARL



**Fig. 13** Comparison of the trajectory prediction results of the simple LSTM, ILSTM and LSTM-KF trajectory prediction algorithms for hurricane MATTHEW

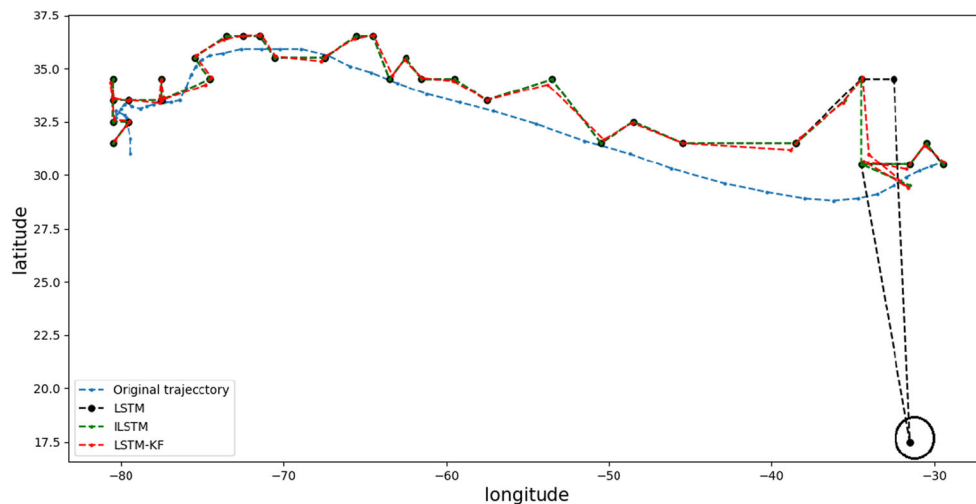


**Fig. 14** Comparison of the trajectory prediction results of the simple LSTM, ILSTM and LSTM-KF trajectory prediction algorithms for hurricane NICOLE





**Fig. 15** Comparison of the trajectory prediction results of the simple LSTM, ILSTM and LSTM-KF trajectory prediction algorithms for hurricane BONNIE



prediction results, as shown in Eq. 23. This neural network structure is set to  $256 * 256 * 256$ , the state transition matrix  $A = [1,0,1,0; 0,1,0,1; 0,0,1,0; 0,0,0,1]$ , the observation matrix  $H = [1,0,0,0; 0,1,0,0]$ . We select hurricane data from 1895–2015 for training and forecast hurricanes in 2016. When the system noise covariance matrix  $Q$  are set to  $[1,0,0,0; 0,1,0,0; 0,0,1,0; 0,0,0,1]$  and  $[2,0,0,0; 0,2,0,0; 0,0,2,0; 0,0,0,2]$ , the observation noise covariance matrix  $R$ , respectively, using the following parameters are shown in Table 7.

$$\overline{\Delta RMSE} = \frac{1}{epochs} \sum_{i=1}^{epochs} RMSE_{ILSTM} - RMSE_{LSTM-KF} \quad (23)$$

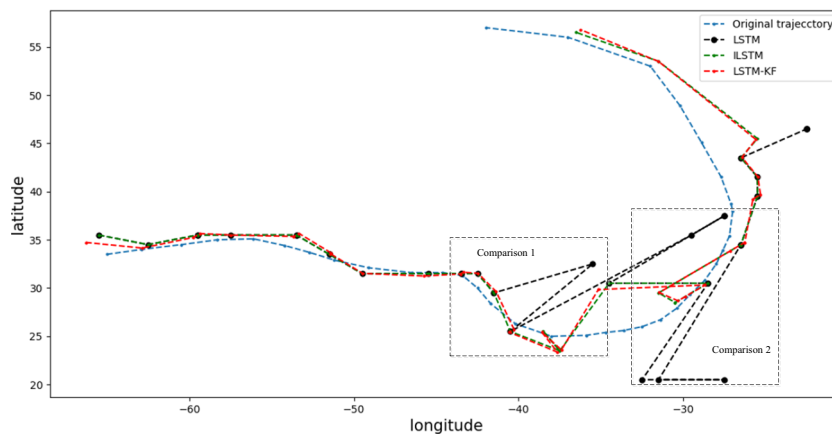
Where  $RMSE_{LSTM-KF}$  represents the root mean square error of LSTM-KF algorithm.  $RMSE_{ILSTM}$  represents the root mean square error of LSTM-KF algorithm.  $epochs$  represent the training rounds, and  $epochs$  are set to 100.

When  $Q = [1,0,0,0; 0,1,0,0; 0,0,1,0; 0,0,0,1]$ , the result of  $\overline{\Delta RMSE}$  changes with  $R$  is shown in Fig. 17a. When

$Q = [2,0,0,0; 0,2,0,0; 0,0,2,0; 0,0,0,2]$ , the result of  $\overline{\Delta RMSE}$  changes with  $R$  is shown in Fig. 17b. It can be seen from Fig. 17 that the observation noise covariance matrix  $R$  will have an impact on the prediction accuracy of LSTM-KF algorithm. Figure 17a and b show that  $\overline{\Delta RMSE}$  tend to increase and then decrease rather than increase indefinitely as  $R$  increases. In Fig. 17a, the maximum  $\overline{\Delta RMSE}$  is 0.0419 when  $R = R_2$ , and the minimum  $\overline{\Delta RMSE}$  is -0.0033 when  $R = R_{10}$ . In Fig. 17b, the maximum  $\overline{\Delta RMSE}$  is 0.0404 when  $R = R_4$ , and the minimum  $\overline{\Delta RMSE}$  is 0.0077 when  $R = R_{0.1}$ . Therefore, different  $R$  will have different influences on the accuracy of the predicted trajectory, and if the selected  $R$  is not appropriate, it may also have a negative effect on the predicted results. For example,  $\overline{\Delta RMSE}$  is negative when  $R = R_{10}$  in Fig. 17a.

Then, the observation noise  $R$  is set as  $[1, 0; 0, 1]$ ,  $[2, 0; 0, 2]$ ,  $[3, 0; 0, 3]$ ,  $[4, 0; 0, 4]$ , and the system noise covariance matrix  $Q$  is set as the parameters shown in Table 8. The result of  $\overline{\Delta RMSE}$  changes with system noise covariance matrix  $Q$  as shown in Fig. 18.

**Fig. 16** Comparison of the trajectory prediction results of the simple LSTM, ILSTM and LSTM-KF trajectory prediction algorithms for hurricane ALEX



**Table 7** Parameter Settings of observation noise covariance matrix  $R$

$R$	Value	$R$	Value
$R_{0,1}$	[0.1,0; 0, 0.1]	$R_1$	[1, 0; 0, 1]
$R_{0,2}$	[0.2,0; 0, 0.2]	$R_2$	[2, 0; 0, 2]
$R_{0,3}$	[0.3,0; 0, 0.3]	$R_3$	[3, 0; 0, 3]
$R_{0,4}$	[0.4,0; 0, 0.4]	$R_4$	[4, 0; 0, 4]
$R_{0,5}$	[0.5,0; 0, 0.5]	$R_5$	[5, 0; 0, 5]
$R_{0,6}$	[0.6,0; 0, 0.6]	$R_6$	[6, 0; 0, 6]
$R_{0,7}$	[0.7,0; 0, 0.7]	$R_7$	[7, 0; 0, 7]
$R_{0,8}$	[0.8,0; 0, 0.8]	$R_8$	[8, 0; 0, 8]
$R_{0,9}$	[0.9,0; 0, 0.9]	$R_9$	[9, 0; 0, 9]
		$R_{10}$	[10, 0; 0, 10]

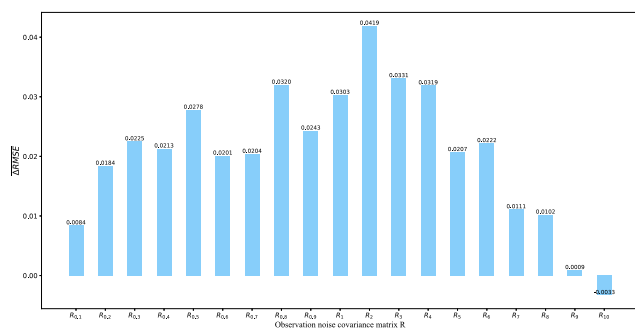
**Table 8** Parameter Settings of system noise covariance matrix  $Q$

$Q$	Value	$Q$	Value
$Q_{0,1}$	[0.1,0; 0, 0.1]	$Q_1$	[1, 0; 0, 1]
$Q_{0,2}$	[0.2,0; 0, 0.2]	$Q_2$	[2, 0; 0, 2]
$Q_{0,3}$	[0.3,0; 0, 0.3]	$Q_3$	[3, 0; 0, 3]
$Q_{0,4}$	[0.4,0; 0, 0.4]	$Q_4$	[4, 0; 0, 4]
$Q_{0,5}$	[0.5,0; 0, 0.5]	$Q_5$	[5, 0; 0, 5]
$Q_{0,6}$	[0.6,0; 0, 0.6]	$Q_6$	[6, 0; 0, 6]
$Q_{0,7}$	[0.7,0; 0, 0.7]	$Q_7$	[7, 0; 0, 7]
$Q_{0,8}$	[0.8,0; 0, 0.8]	$Q_8$	[8, 0; 0, 8]
$Q_{0,9}$	[0.9,0; 0, 0.9]	$Q_9$	[9, 0; 0, 9]
		$Q_{10}$	[10, 0; 0, 10]

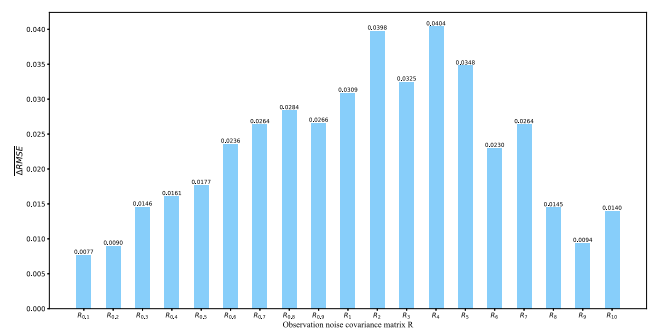
When  $R = [1, 0; 0, 1]$ , the result of  $\overline{\Delta RMSE}$  changes with  $Q$  are shown in Fig. 18a. When  $R = [2, 0; 0, 2]$ , the result of  $\overline{\Delta RMSE}$  changes with  $Q$  are shown in Fig. 18b. When  $R = [3, 0; 0, 3]$ , the result of  $\overline{\Delta RMSE}$  changes with  $Q$  are shown in Fig. 18c, and When  $R = [4, 0; 0, 4]$ , the result of  $\overline{\Delta RMSE}$  changes with  $Q$  are shown in Fig. 18d. As shown in Fig. 18a, b, c, and d,  $\overline{\Delta RMSE}$  also approximate a trend of increasing and then decreasing. Therefore, different system noise covariance matrix  $Q$  will have an impact on the prediction accuracy of LSTM-KF algorithm, and inappropriate values may have a negative impact. For example, in Fig. 18b, c, and d,  $\overline{\Delta RMSE}$  is negative when  $Q$  is compared at the beginning. However, it is worth noting that the better effect of  $Q$  and  $R$  is around 0.05. As shown in Fig. 17a, the maximum  $\overline{\Delta RMSE}$  is 0.0419 when  $R = R_2$ . As shown in Fig. 18a, the maximum  $\overline{\Delta RMSE}$  is 0.0535 when  $Q = Q_{0,6}$ . As shown in Fig. 18b, the maximum  $\overline{\Delta RMSE}$  is 0.0556 when  $Q = Q_{0,9}$ . As shown in Fig. 18d, the maximum  $\overline{\Delta RMSE}$  is 0.0512 when  $Q = Q_5$ . Selecting the appropriate value of  $Q$  and  $R$  will produce better optimization effect on the predicted

results, but the effect of prediction optimization may be in a small range.

In order to test the time performance of LSTM-KF algorithm, this experiment tested the time performance by counting model training time and predicted response time of different network structures. The hardware platform of this experiment is CPU Intel(R) Core(TM) i7-9850h, 2.60 Ghz, 16G memory, GPU Quadro RTX 3000. We select hurricane data from 1895 to 2015 for training and statistic the model training time, as shown in Fig. 19. Forecast the hurricane trajectory in 2016 and statistic their predicted response time, as shown in Fig. 20. It can be seen from Fig. 19 that the neural network structure is  $128*128*256$ ,  $256*256*256$ ,  $128*256*512$ ,  $256*256$ , and the training time of the model is 870s, 929s, 913s and 817s, respectively. As shown in Fig. 20, the structure of the neural network is  $128*128*256$ ,  $256*256*256$ ,  $128*256*512$ ,  $256*256$ , and the response time of model is 5s, 6s, 6s and 4s respectively. The predicted response time of the whole year of 2016 has remained at a few seconds. Although LSTM-KF algorithm takes more time in model training, the time



(a)  $Q = [1,0,0,0; 0,1,0,0; 0,0,0,1; 0,0,0,1]$



(b)  $Q = [2,0,0,0; 0,2,0,0; 0,0,2,0; 0,0,0,2]$

**Fig. 17** Result of  $\overline{\Delta RMSE}$  changes with observation noise covariance matrix  $R$

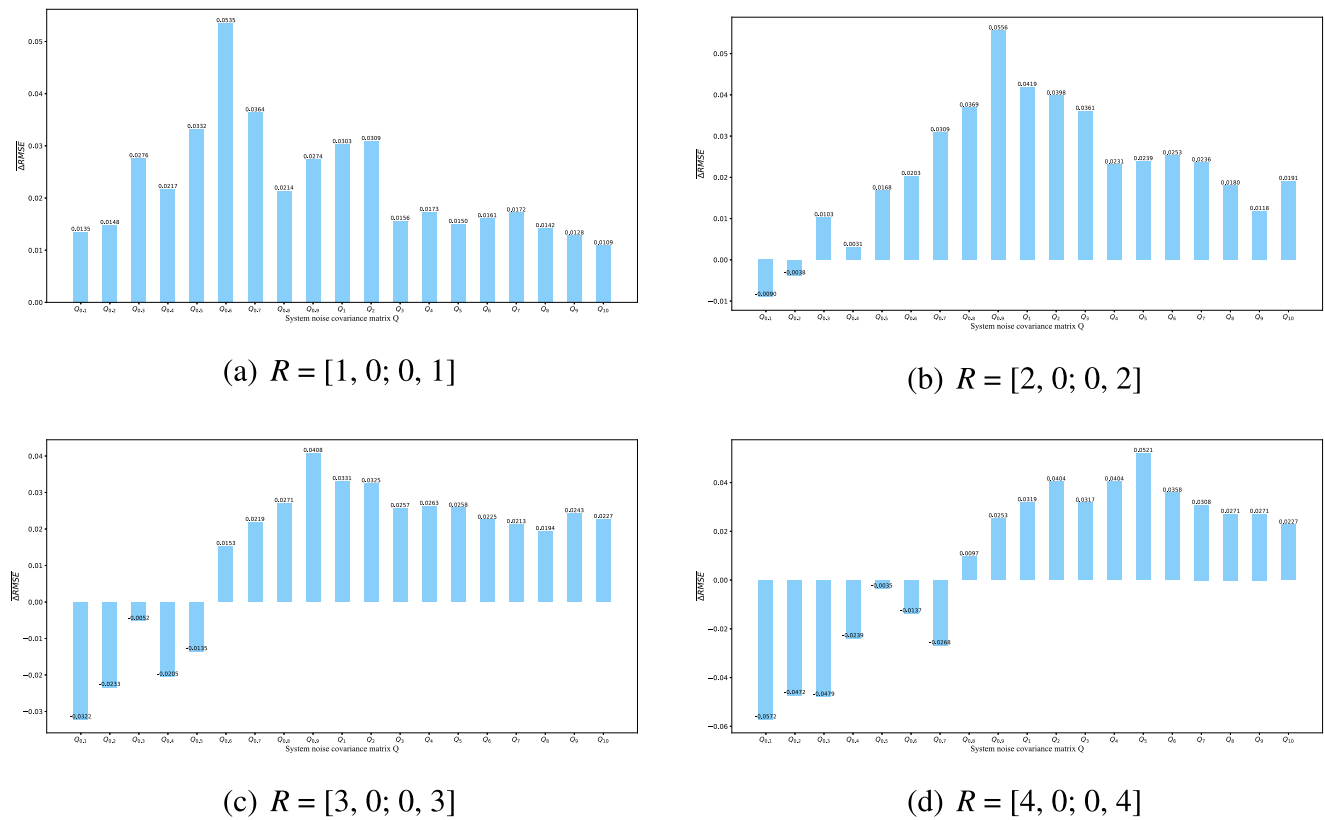
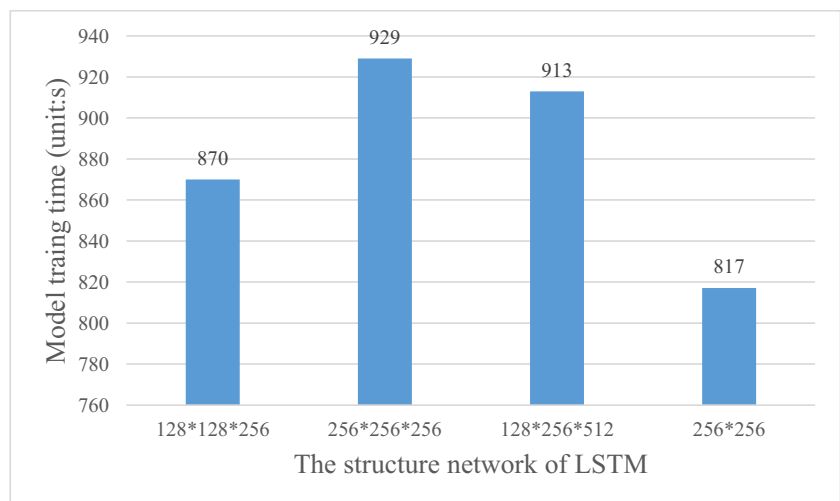


Fig. 18 Result of  $\Delta RMSE$  changes with system noise covariance matrix  $Q$

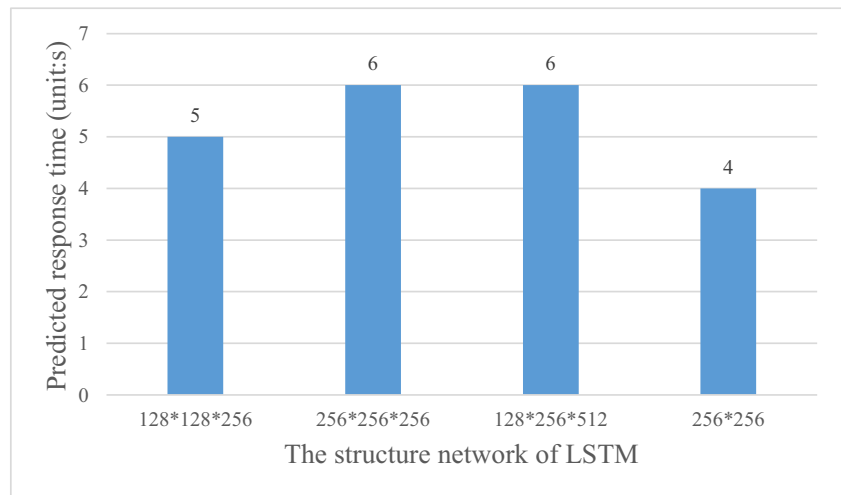
interval of hurricane trajectory prediction is 6 h, so the model training time is much lower than the time interval between the sampling points. Meanwhile, the LSTM-KF algorithm only needs to train the trajectory prediction model before each hurricane arrives and it is not necessary to train the model for each prediction. The time spent in the

prediction process of trajectory is response time. As shown in Fig. 20, the predicted response time of the whole year of 2016 is very short, only within a few seconds. And the better the hardware platform, the shorter the time. Therefore, LSTM-KF algorithm can meet the requirements of real-time hurricane prediction.

Fig. 19 Model training time for different network structures



**Fig. 20** The predicted response time of different network structures



## 6 Conclusion

Trajectory prediction has become a research hotspot in many fields. Hurricanes are serious threats to people's lives and cause significant economic losses. Effective prediction of the trajectory of a hurricane has good application value. In this paper, from the perspective of machine learning, using real Atlantic hurricane data, an LSTM network is applied to hurricane trajectory prediction, and the prediction model is trained using historical hurricane data. The main contributions of this paper are as follows: 1. In the stage of data preprocessing, the trajectories are gridded and encoded, and a trajectory vector is generated by combining numerical data with categorical data to input into the LSTM. 2. Based on the simple LSTM trajectory prediction algorithm, this paper improves the prediction module and proposes an improved LSTM trajectory prediction algorithm. 3. Combined with a Kalman filter, the predicted coordinates of the improved LSTM trajectory prediction algorithm are filtered, and the LSTM-KF trajectory prediction algorithm is proposed. 4. Real Atlantic hurricane data from 1851 to 2016 are used in simulation experiments. The prediction results of the LSTM-KF trajectory prediction algorithm are better than those of the improved LSTM algorithm and the simple LSTM algorithm. The trajectory prediction algorithm proposed in this paper only considers several factors, such as latitude and longitude, maximum sustained wind speed and the system state, but its prediction error is still very large. It is hoped that more meteorological factors can be considered in future studies to establish a more complete prediction model.

**Acknowledgements** This work was supported by the National Natural Science Foundation of China (62073330), Natural Science Foundation of Hunan Province (2019JJ20021), NUDT Scientific Research Project (ZK18-02-12), Huxiang Young Talents (2018RS3079) and Innovation project for graduate students in Hunan Province (CX20190046).

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Zheng, Y.: Trajectory data mining: an overview. *ACM Trans. Intell. Syst. Technol. (TIST)* **6**(3), 1–41 (2015)
- Tang, J.: Conflict detection and resolution for civil aviation: a literature survey. *IEEE Aerosp. Electron. Syst. Mag.* **34**(10), 20–35 (2019)
- Meuel, T., Prado, G., Seychelles, F., Bessafi, M., Kellay, H.: Hurricane track forecast cones from fluctuations. *Sci. Rep.* **2**, 446 (2012)
- Lee, R.S., Liu, J.N.: Tropical cyclone identification and tracking system using integrated neural oscillatory elastic graph matching and hybrid RBF network track mining techniques. *IEEE Trans. Neural Netw.* **11**(3), 680–689 (2000)
- Rajasekaran, S., Lee, T., Jeng, D.-S.: Tidal level forecasting during typhoon surge using functional and sequential learning neural networks. *J. Waterway Port Coastal Ocean Eng.* **131**(6), 321–324 (2005)
- Rožanova, O.S., Yu, J.-L., Hu, C.-K.: Typhoon eye trajectory based on a mathematical model: comparing with observational data. *Nonlinear Anal.: Real World Appl.* **11**(3), 1847–1861 (2010)
- DeMaria, M., Mainelli, M., Shay, L.K., Knaff, J.A., Kaplan, J.: Further improvements to the statistical hurricane intensity

- prediction scheme (SHIPS). *Weather Forecast.* **20**(4), 531–543 (2005)
8. Rekadarkolae, H.M., Krut, C., Fuentes, M., Reich, B.J.: A Bayesian multivariate functional model with spatially varying coefficient approach for modeling hurricane track data. *Spatial Stat.* **29**, 351–365 (2019)
  9. Mitchell, R., Michalski, J., Carbonell, T.: *An Artificial Intelligence Approach*. Springer, Berlin (2013)
  10. King, D.E.: Dlib-ml: a machine learning toolkit. *J. Mach. Learn. Res.* **10**, 1755–1758 (2009)
  11. Alemany, S., Beltran, J., Perez, A., Ganzfried, S.: Predicting hurricane trajectories using a recurrent neural network. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 468–475 (2019)
  12. Kim, H.-S., Kim, J.-H., Ho, C.-H., Chu, P.-S.: Pattern classification of typhoon tracks using the fuzzy c-means clustering method. *J. Clim.* **24**(2), 488–508 (2011)
  13. Gao, S., Zhao, P., Pan, B., Li, Y., Zhou, M., Xu, J., Zhong, S., Shi, Z.: A nowcasting model for the prediction of typhoon tracks based on a long short term memory neural network. *Acta Oceanol. Sin.* **37**(5), 8–12 (2018)
  14. Kordmahalleh, M.M., Sefidmazgi, M.G., Homaifar, A., Liess, S.: Hurricane trajectory prediction via a sparse recurrent neural network. In: *Proceedings of the 5th International Workshop on Climate Informatics*, pp. 2–3. ACM, Denver (2015)
  15. Baik, J.-J., Hwang, H.-S.: Tropical cyclone intensity prediction using regression method and neural network. *J. Meteorol. Soc. Jpn. Ser. II* **76**(5), 711–717 (1998)
  16. Baik, J.-J., Paek, J.-S.: A neural network model for predicting typhoon intensity. *J. Meteorol. Soc. Jpn. Ser. II* **78**(6), 857–869 (2000)
  17. Formentin, S., Bianchessi, A.G., Savaresi, S.M.: On the prediction of future vehicle locations in free-floating car sharing systems. In: *2015 IEEE Intelligent Vehicles Symposium (iv)*, pp. 1006–1011. IEEE (2015)
  18. Qiao, S., Han, N., Zhu, W., Gutierrez, L.A.: TraPlan: an effective three-in-one trajectory-prediction model in transportation networks. *IEEE Trans. Intell. Transp. Syst.* **16**(3), 1188–1198 (2014)
  19. Chen, M., Liu, Y., Yu, X.: Nlpm: a next location predictor with markov modeling. In: *Pacific-Asia conference on knowledge discovery and data mining*, pp. 186–197. Springer, Cham (2014)
  20. Figueiredo, F., Ribeiro, B., Almeida, J.M., Faloutsos, C.: TribeFlow: mining & predicting user trajectories. In: *Proceedings of the 25th International Conference on World Wide Web*, pp. 695–706 (2016)
  21. Tang, J., Piera, M.A., Guasch, T.: Coloured Petri net-based traffic collision avoidance system encounter model for the analysis of potential induced collisions. *Transp. Res. Part C: Emerg. Technol.* **67**, 357–377 (2016)
  22. Tang, J., Zhu, F., Piera, M.A.: A causal encounter model of traffic collision avoidance system operations for safety assessment and advisory optimization in high-density airspace. *Transp. Res. Part C: Emerg. Technol.* **96**, 347–365 (2018)
  23. Wiest, J., Höffken, M., Kreßel, U., Dietmayer, K.: Probabilistic trajectory prediction with gaussian mixture models. In: *2012 IEEE Intelligent Vehicles Symposium*, pp. 141–146. IEEE, Alcalá (2012)
  24. Chapuis, B., Moro, A., Kulkarni, V., Garbinato, B.: Capturing complex behaviour for predicting distant future trajectories. In: *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, pp. 64–73. ACM, California (2016)
  25. Wolfson, O., Xu, B., Chamberlain, S., Jiang, L.: Moving objects databases: issues and solutions. In: *Proceedings. Tenth International Conference on Scientific and Statistical Database Management (Cat. No. 98TB100243)*, pp. 111–122. IEEE, DC (1998)
  26. Junghans, C., Gertz, M.: Modeling and prediction of moving region trajectories. In: *Proceedings of the ACM SIGSPATIAL International Workshop on GeoStreaming 2010*, pp. 23–30. ACM, New York (2010)
  27. Prevost, C.G., Desbiens, A., Gagnon, E.: Extended Kalman filter for state estimation and trajectory prediction of a moving object detected by an unmanned aerial vehicle. In: *2007 American Control Conference*, pp. 1805–1810. IEEE, New York (2007)
  28. Jeung, H., Liu, Q., Shen, H.T., Zhou, X.: A hybrid prediction model for moving objects. In: *2008 IEEE 24th International Conference on Data Engineering*, pp. 70–79. IEEE, Cancun (2008)
  29. Monreale, A., Pinelli, F., Trasarti, R., Giannotti, F.: Wherenext: a location predictor on trajectory pattern mining. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 637–646. ACM, Paris (2009)
  30. Long, T., Qiao, S., Tang, C., Liu, L., Li, T., Wu, J.: E 3 TP: a novel trajectory prediction algorithm in moving objects databases. In: *Pacific-Asia Workshop on Intelligence and Security Informatics*, pp. 76–88. Springer, Heidelberg (2009)
  31. Kim, S.-W., Won, J.-I., Kim, J.-D., Shin, M., Lee, J., Kim, H.: Path prediction of moving objects on road networks through analyzing past trajectories. In: *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pp. 379–389. Springer, Heidelberg (2007)
  32. Qiao, S., Shen, D., Wang, X., Han, N., Zhu, W.: A self-adaptive parameter selection trajectory prediction approach via hidden Markov models. *IEEE Trans. Intell. Transp. Syst.* **16**(1), 284–296 (2014)
  33. Li, M., Lu, F., Zhang, H., Chen, J.: Predicting future locations of moving objects with deep fuzzy-LSTM networks. *Transportmetrica A: Transp. Sci.* **16**(1), 119–136 (2020)
  34. De Leege, A., van Paassen, M., Mulder, M.: A machine learning approach to trajectory prediction. In: *AIAA Guidance, Navigation, and Control (GNC) Conference 2013*, p. 4782. AIAA, MA (2013)
  35. Chen, M., Liu, Y., Yu, X.: Predicting next locations with object clustering and trajectory clustering. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 344–356. Springer, Heidelberg (2015)
  36. Shah, R., Romijnders, R.: Applying deep learning to basketball trajectories. Applying deep learning to basketball trajectories. Last revised 16 Aug 2016 (2016)
  37. Plaut, D.C.: *Experiments on Learning by Back Propagation*. Technical Report CMU-CS-86-126. Department of Computer Science Carnegie-Mellon University (1986)
  38. Werbos, P.: *Beyond regression: new tools for prediction and analysis in the behavioral sciences*. Ph.D. dissertation, Harvard University (1974)
  39. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. *Nature* **323**(6088), 533–536 (1986)
  40. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
  41. Mann, M.E., Woodruff, J.D., Donnelly, J.P., Zhang, Z.: Atlantic hurricanes and climate over the past 1,500 years. *Nature* **460**(7257), 880–883 (2009)
  42. Elsner, J.B.: Tracking hurricanes. *Bull. Am. Meteorol. Soc.* **84**(3), 353–356 (2003)
  43. Kalman, R.E.: A new approach to linear filtering and prediction problems. *J. Basic Eng.* **82D**(1), 35–45 (1960)

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.