A Real Time Tracking Method for Intelligent Logistics Delivery Based on Recurrent Neural Network

Xunyan Bao[1](✉) and Dong'e Zhou[2]

[1] Zhejiang Changzheng Vocational and Technical College, Hangzhou 310012, China
baoxunyan@126.com

[2] Guangzhou Huashang Vocational College, Guangzhou 511300, China

Abstract. In order to further improve the real-time tracking effect of intelligent logistics distribution, this paper proposes a real-time tracking method for intelligent logistics distribution based on recurrent neural networks. Firstly, relevant analysis was conducted on real-time tracking of intelligent logistics delivery, and the planning process of order information and road information was determined. Secondly, considering the dynamic real-time traffic conditions and constantly updated customer orders analyzed above, an online target tracking model based on recurrent neural networks was established to predict the status of each node, and then correct the corresponding target status. Finally, Hungarian algorithm is used to solve the data association problem to reduce the detector error to the tracking algorithm. Finally, the Loss function is used to optimize the model performance and achieve accurate real-time tracking of intelligent logistics distribution. The results indicate the feasibility of the proposed method in practical applications, and by comparing it with other similar methods in solving the objective function, the advantages of this method in real-time tracking of intelligent logistics distribution are further verified, with high tracking accuracy.

Keywords: Smart logistics · Path planning · Real time tracking

1 Introduction

In recent years, with the continuous improvement of information technology and the rapid development of "online shopping era", the logistics industry has been an unprecedented spurt of development. Online shopping has many characteristics such as convenient transaction, low cost, rich profit and saving time, which makes the commercial institutions actively change their operation mode, while the real economy is also facing great challenges. After the online transaction is completed, the rest of the tasks are completed by the offline logistics. Therefore, the convenient service of e-commerce urgently needs a matching logistics system with low cost and high efficiency, so as to operate more stably [1, 2]. The compression of logistics costs and the diversified needs of consumers have become the bottleneck restricting the faster and more stable development of contemporary logistics. In this context, the concept of "intelligent logistics" came into being [3].

L. Yun et al. (Eds.): ADHIP 2023, LNICST 550, pp. 113–132, 2024.
https://doi.org/10.1007/978-3-031-50552-2_8

In recent years, China's logistics industry has undeniably achieved explosive breakthroughs, and these changes in smart logistics are entirely attributed to the rapid popularization and application of new generation advanced information technologies such as the global Internet of Things, cloud computing, and mobile internet in human life. Many advanced modern logistics facilities and equipment have advanced scientific and technological capabilities such as digitization, informatization, and intensification [4, 5]. Some logistics companies are increasingly using information technology in their logistics distribution business. However, there are still some problems that need to be solved. Firstly, due to the multiple links and participants involved in the process of logistics distribution, it is difficult to obtain and share information. Different logistics companies, suppliers and transporters use different tracking systems and data formats, resulting in information silos and data inconsistencies [6]. Second, existing tracking technologies also face challenges in complex urban environments. Factors such as tall buildings, traffic congestion, and signal interference may result in reduced positioning accuracy or inability to obtain accurate location information [7]. Therefore, in order to improve the visualization degree of logistics distribution, logistics companies and customers can understand the location and status of goods in real time, improve the efficiency and accuracy of distribution, and carry out real-time tracking of intelligent logistics distribution is of great significance.

Therefore, this paper proposes a real-time tracking method of intelligent logistics distribution based on recurrent neural network to achieve accurate distribution tracking. This method firstly analyzes the real-time tracking of intelligent logistics distribution, and determines the planning process of two types of information, order information and road information. Secondly, considering the dynamic real-time traffic conditions analyzed above and the constantly updated customer orders and other factors, an online target tracking model based on recursive neural network is established to predict the status of each node. Then modify the corresponding target state. Finally, the Hungarian algorithm is used to solve the data association problem to reduce the detector error to the tracking algorithm. Finally, the loss function is used to optimize the model performance and achieve accurate real-time tracking of intelligent logistics distribution.

2 Real Time Tracking Analysis of Intelligent Logistics Delivery

Real-time dynamic path planning of intelligent logistics refers to that when the real-time information that has a great influence on the path selection is conveyed to the system center, the system needs to make targeted responses to obtain the optimal path of the current situation. Real-time information here includes new order placing, order cancellation, order delay, road speed change, road construction, traffic congestion, etc., which can be roughly divided into two categories: order information and road information [8]. Although the real-time dynamic path planning system is said to be a re-optimization strategy, it does not completely re-plan from the beginning. But during the driving process, based on the real-time information received, new unserved customer point paths are added to the existing path planning to complete the overall path re planning [9]. Below is a detailed introduction to the process of re planning based on two categories of information: order information and road information:

(1) When the road information changes, the system will first automatically detect whether the planned delivery path passes through that road. If the system automatically captures the road speed RS (Road Speed) of the current driving section, then finds the previous customer point i_p that has completed the service, records its position j, and calculates the arrival time of j to all unassigned points. The point with the shortest time is assigned as the key point i_1, and gradually continues to rank the remaining customer points in descending order of time as the key points for the next service goal, Then, for each key point recorded, the system automatically removes it from the set of unassigned points until all final orders are re optimized; If not, the system will automatically ignore this information [10]. The reason for this arrangement is that the path before this customer point is already in the optimal state adjusted by the system, so there is no need to optimize it again. This not only saves computational time for the system, but also simplifies the complexity of model calculations. The specific calculation process is shown in Fig. 1:

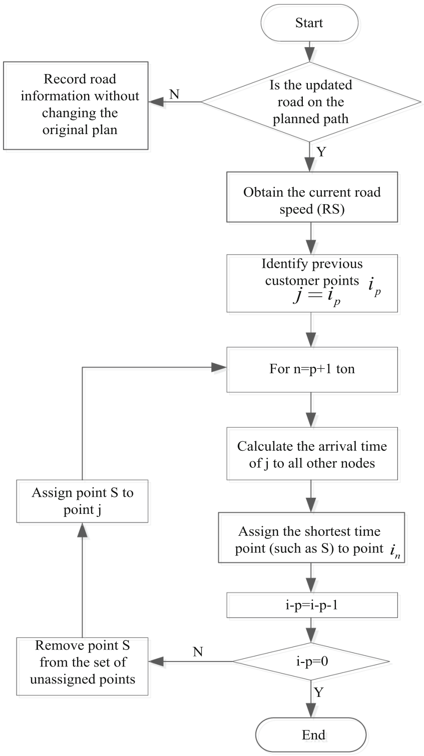


Fig. 1. Real time path planning flowchart for road information changes

(2) When the order information is changed, it is similar to the road information change strategy mentioned above. When A new order i_p arrives in the system, the first step is to determine whether the order is on the planned path. If yes, insert it between the customer points before and after the original path; If not, first calculate the distance d_{pn} between the closest customer points i_1 and i_p to the central system, and then calculate the distance $d_{n,n-1}$ between i_n and the previous customer point i_{n-1}, and then compare d_{pn} and $d_{n,n-1}$.

① If $d_{pn} > d_{n,n-1}$, then the new path is $i_{n-1} - i_p - i_n$, that is, the new order is inserted between i_{n-1} and i_n. The system has been optimized before i_{n-1} and after i_n, so it remains unchanged.

② If $d_{n,n-1} > d_{pn}$, the new path is $i_{n-1} - i_n - i_p$, that is, new orders are placed behind these two points. In this way, according to the non after effect of dynamic programming, the system needs to resort and optimize all customer points after the new order i_p.

Then, point i_{n-1} is regarded as the key point, marked as j, and the arrival time from point i_{n-1} to all other points is calculated. The points with the minimum value are selected and inserted after point i_n one by one until the set of all unallocated points is 0, namely, $i - p = 0$.

According to the above description, the calculation process when the order information changes is shown in Fig. 2:

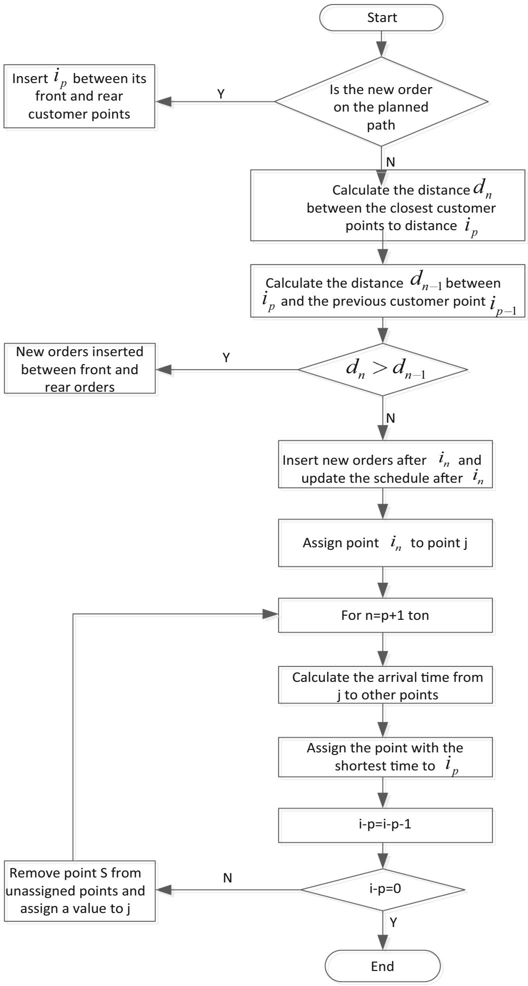


Fig. 2. Real-time path planning flow chart of order information change

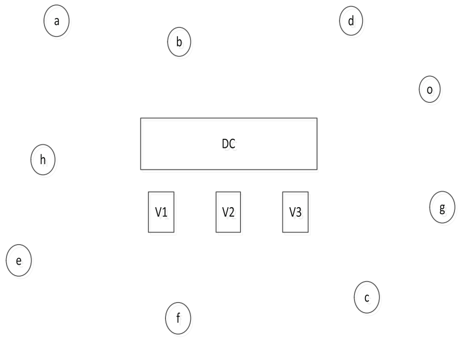
(3) An example of intelligent logistics distribution in a certain region is introduced here to demonstrate the characteristics of real-time dynamic path tracking. As shown in Fig. 3, the distribution center (DC) has 3 vehicles and 9 customer points in its initial state (a), and two vehicles are dispatched by the distribution center to complete the distribution task. The distribution scheme of the first car is DC-b-a-h-e-f-DC, and the distribution scheme of the second car is DC-d-o-g-c-DC (b). At some point, the distribution center received four new orders, which are q, k, l and r (c). Then the distribution center will readjust the route according to the current key points and real-time road traffic information. Figure (c) shows that the key points set $N_c(\varepsilon) = (b, o)$, the unallocated points set $N_U(\varepsilon) = (a, h, e, f, g, c, o)$, and the new demand points q, k, l and r. After route replanning, the distribution route of the first vehicle is DC-b-a-q-h-e-k-DC, and the route of the second vehicle is DC-o-l-g-m-c-f-DC (d). In

this way, the new demand is processed, and the path tracking adjustment caused by traffic information changes is similar.

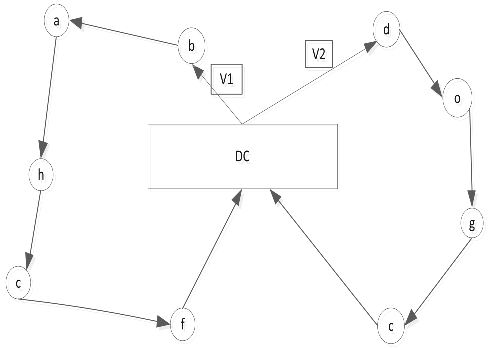
It should be emphasized that during system operation, DC can accept new demand orders at any time. When a new order arrives, the system only considers unassigned nodes, and customer nodes that have completed the service will be automatically deleted. The method to distinguish these points is to determine key points, which can be determined by capturing real-time vehicle information through GPS equipment in the distribution center [11]. In addition, after the issuance of new orders, vehicles in transit must depart from key points, while newly dispatched vehicles must depart from the distribution center. At this point, the key point can also be imagined as a virtual distribution center, which reflects the Markov nature of the platform problem, that is, the planning of future paths only relies on current information and is independent of past solutions.

As this paper studies smart logistics distribution, customers have strict requirements on the timeliness of logistics distribution. In recent years, the timeliness of smart logistics is also a major factor in the competition of various logistics enterprises, and has become an important aspect for more and more customers to report complaints. Therefore, the model in this paper adds a limit on the hard time window [12]. Vehicles are required to complete the service within a given time range. If the delivery path beyond this range is not only not the optimal solution of the objective function, but also causes the loss of waiting or delay. However, the final goods will be delivered to the customer beyond the time limit in advance, so the time window in this case is a soft time window problem. This does not usually occur, so the model does not consider this type for the time being.

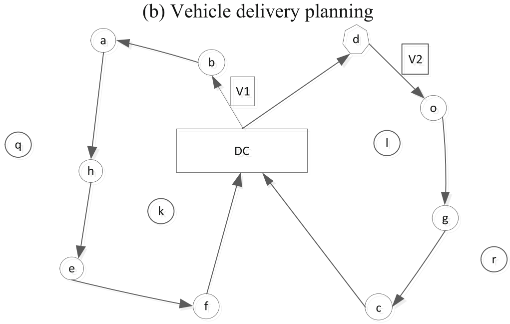
In real life, on the one hand, the service areas of smart logistics delivery are determined in advance based on market research, so situations beyond time limits are rare; On the other hand, even if the travel time exceeds the time limit, other emergency vehicles can arrive in a timely manner due to being served in a specific local area, and only need to plan the order of customer points again [13]. Unless there are occasional large orders and the current vehicle cannot meet the current time window limit, the distribution center will dispatch additional vehicles based on actual needs. The intelligent logistics delivery real-time tracking model based on recurrent neural networks mainly considers two factors: random orders and real-time traffic information. These two uncertain factors can make it more difficult to grasp the delivery time of delivery vehicles, so the calculation process is slightly more complex than the static platform problem [14]. The system dispatch center must provide a pruning plan in a short period of time after receiving real-time information, with strict timeliness requirements, which increasingly require high calculation speed and accurate and appropriate results. Therefore, the next step is to conduct research on the intelligent logistics delivery real-time tracking model based on recurrent neural networks, in order to achieve precise intelligent logistics delivery real-time tracking and improve service response speed.



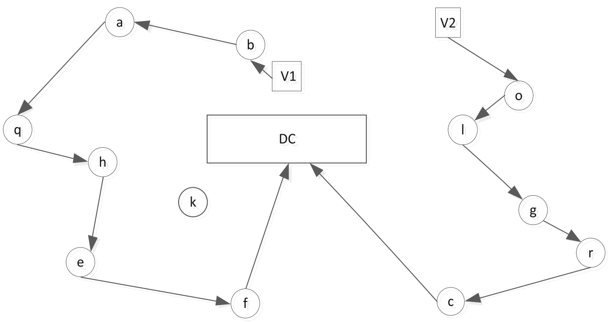
(a) Initial state



(b) Vehicle delivery planning



(c) Key point retrieval



(d) Re planning

Fig. 3. Schematic diagram of real-time path re planning

3 Design of Intelligent Logistics Distribution Real-Time Tracking Model

This article implements an online target tracking model based on recurrent neural network (RNN) using Bayesian filtering ideas. It mainly consists of two parts: one is prediction, which learns the real-time dynamic tracking model of the target through a temporal RNN network and predicts the target state; The second is to update, obtain the latest observation values, calculate their matching relationship with the target, and correct the corresponding target state. Integrate the above two parts to achieve accurate intelligent logistics delivery real-time tracking. The structure of the target tracking network based on RNN is shown in Fig. 4.

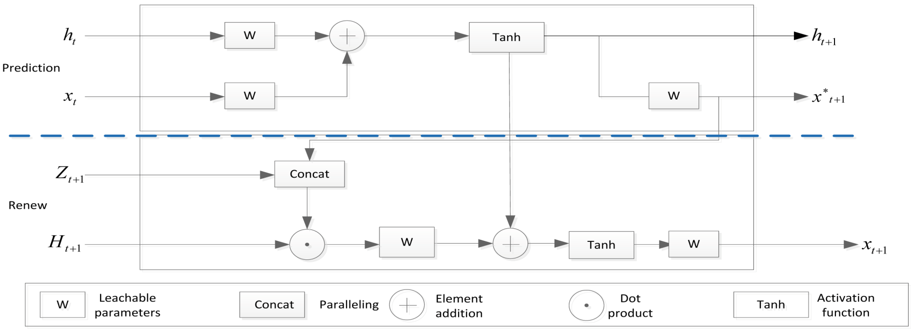


Fig. 4. Structure of RNN based target tracking network

The target tracking model based on RNN has four input values. $h_t \in v^n$ is defined as the state of the hidden layer of RNN in frame t, and n represents the size of the hidden layer. $x_t \in v^{N,D}$ is the state vector of all targets in frame t; N is the maximum number of targets in each frame; D is the dimension of the state vector. Bounding box (x, y, w, h) of the target is selected as the state vector x_t, where (x, y) is the center position of bounding box and w, h is the width and height of bounding box respectively, so $D = 4$. $Z_t \in v^{M,D}$ is defined as all observation vectors in the t frame, and M represents the maximum number of detection responses in each frame. In the model, the Hungarian algorithm is used to calculate the correlation between the network prediction state x^*_{t+1} and the detection response z_{t+1} between two adjacent frames, and H is used to represent the Hungarian correlation matrix, whose size is $N \times (M + 1)$

$$H_{nm} = \begin{cases} 1, & n \text{ and } m \\ 0, & \text{other} \end{cases} \tag{1}$$

$H_{nm} = 1$ when the target n of frame t is associated with the detection response m of frame $t + 1$, and for $\forall n$, $\sum_m H_{nm} = 1$. The $M + 1$ column in the matrix H represents the case of missing detection in the detection response.

The target tracking model based on RNN can be divided into two parts: one is prediction, learning a complex dynamic model to predict the state of the target at the next moment; The second is updating, measuring the matching relationship between the target and the observed value of the next frame, and correcting the target state according to the corresponding observed value.

(1) Prediction stage. Assuming that the initial state x_0 of the target and the initial state h_0 of the network hidden layer are known. Firstly, the RNN network model is used to predict the possible state of the target. At time t, given the state h_t of the hidden layer and the target state x_t, the hidden layer state at time $t+1$ can be obtained

$$h_{t+1}^{l} = \tanh\left(W_h h_t^l + W_x x_t\right) \tag{2}$$

where, W_h and W_x represent learnable parameters in the full connection layer.

From this, it can get the predicted state of the target

$$x_{t+1}^{*} = W_{x^*} \tanh\left(W_h h_t^l + W_x x_t\right) \tag{3}$$

From the above equation, it can be seen that the predicted value x_{t+1}^{*} of the target state at time $t+1$ only depends on the hidden layer state h_t and target state x_t at time t. That is to say, the RNN model only predicts the possible state of the target at the next moment based on its current state.

(2) Update stage. After the latest observed value z_{t+1} is obtained, the predicted value of the network can be updated. It is noted that when tracking multiple targets, multiple detection responses will be obtained in each frame. At this time, it is necessary to determine which detection response should update the corresponding target, which is a data association problem. In this paper, the Hungarian algorithm is used to solve this problem [15]. By constructing the cost matrix, the solution matrix that minimizes the correlation cost between data is found, so as to obtain the optimal allocation.

At moment t, the target state x_t is known and the latest observation value z_{t+1} is obtained. The euclidene distance d between the predicted location (x_n, y_n) of each target bounding box and the bounding box position (x_m, y_m) of all observations in the prediction stage is calculated successively.

$$d_{nm} = \sqrt{(x_n - x_m)^2 - (y_n - y_m)^2} \tag{4}$$

Build a cost matrix C with size $N \times (M+1)$, the expression is as follows:

$$C = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1j} & \sigma \\ d_{21} & d_{22} & \cdots & d_{2j} & \sigma \\ \cdots & & \cdots\cdots & & \cdots\sigma \\ d_{i1} & d_{i2} & \cdots & d_{ij} & \sigma \end{bmatrix} \tag{5}$$

where d_{nm} represents the distance measurement between target n and all detection responses m, the number of targets is N, and the number of detection responses is M. The distance threshold σ is artificially set. If the distance between the target n and all detection responses is greater than σ, it is determined that the target n is missed and column $M + 1$ in the solution matrix H is marked as 1.

With the cost matrix C, the matching relation between the target and the detection response can be obtained by finding the solution matrix H_{t+1} which minimizes the correlation cost through the Hungarian algorithm. If column m of row n in H_{t+1} is marked as 1, it means that the m detection response z_{t+1}^{m} is assigned to the n target x_t^n, and after obtaining H_{t+1}, the target state x_t^n can be updated by the corresponding observed value z_{t+1}^{m} according to the matching relation.

Then connect the predicted value O of the network in parallel with the observed value P, denoted as Q. The update of the target state at time R can be calculated as follows:

$$x_{t+1} = W_{x_{t+1}} \tanh\left(h_{t+1} + W_H\left(H_{t+1} \bullet z_{t+1}^{*}\right)\right) \tag{6}$$

Next, let's do the dot product between H_{t+1} and z_{t+1}^{*}. The state update in the model relies on the predicted state x_{t+1}^{*} at the current time, the latest observation vector z_{t+1}, and the matching relationship H_{t+1} between the state and observation to correct the target's state.

Because the association algorithm is added, the algorithm itself has a certain robustness to the error caused by the detector. In case of missing detection, the undetected target n will be marked as 1 in the n row and $M + 1$ column of the incidence matrix H, that is, it will be judged as invalid detection, and the target state will not be updated, so as to minimize the impact of the detector error on the tracking algorithm.

After constructing the tracking model, we need to calculate the optimal model through the loss function, which measures the error between the predicted value and the real solution. The model performance can be optimized by minimizing the loss. For tracking algorithms, loss functions are defined mainly by measuring their tracking performance. However, the emphasis of tracking performance evaluation varies with different application scenarios. For example, in football matches, we tend to pay more attention to the correct identification of athletes and try to avoid the situation of identity ID conversion, while in vehicle assistance systems, tracking algorithms are required to have a high accuracy and recall rate to avoid accidents. For this algorithm, we hope that it can have a good performance in tracking accuracy, so the loss is defined as follows:

$$L\left(\tilde{x}, x^{*}, x\right) = \frac{\lambda}{ND} \sum \left\| x^{*} - \tilde{x} \right\|^{2} + \frac{\kappa}{ND} \sum \| x - \tilde{x} \|^{2} \tag{7}$$

Among them, $\tilde{x}$ represents the true value, x^{*} and x represent the predicted and updated values, N represents the number of targets, D represents the feature dimension, and λ and κ are constants. Note that the loss here is the sum of the losses of the training samples in all frames of the video sequence. From the definition of loss, it can be seen that we hope the trajectory predicted by the network can be as close to the real trajectory as possible. Therefore, the model is optimized by minimizing the Mean squared error between the predicted value and the updated value of the target state and the actual value to improve the tracking accuracy.

4 Experimental Analysis

4.1 Example Parameter Setting and Evaluation Index

In this model, weight setting and parameter setting are two important factors for solving the optimal solution of the objective function. Their values directly affect the accuracy of the model's calculation results, and thus affect the overall efficiency of real-time tracking of intelligent logistics distribution. So in order to avoid errors in arbitrary guesses, this article provides the following reference data:

(1) The weight of delivery time for vehicles in transit $\alpha = 0.6$
(2) Weight of waiting time before service $\beta = 0.2$
(3) The new order must be completed within two hours after the sub acceptance is issued; The initial order is completed within one and a half hours after the vehicle leaves the distribution center.
(4) The weight of the waiting time between the completion of the service and the time when the vehicle leaves the customer point is $\gamma = 0.1$
(5) Random number weight $\theta = 1.3$ for new orders.
(6) The conditions for determining the deployment of additional vehicles: unserved customer points exceed 0.5% of the total number.
(7) Set the interval for collecting real-time information to 10 min.
(8) Jitter process:

$$P_{insert} = 0.3 \quad P_{cross} = 0.2 \quad P_{incross} = 0.15$$

Domain search operator probability:$P_{or-opt} = 0.5 \quad P_{2-opt} = 1 - P_{or-opt} = 0.5$.

In order to improve the timeliness of the smart logistics distribution real-time tracking model, this paper defaults that the orders of the system come from the initial time of the planning period, and the other 50% of the orders are received by the distribution center later, randomly generated during the operation of the system. The random number determination method is as follows:$\max(0, S_i - \theta d_{oi} - r)$.

The maximum payload of each vehicle is 7.5 tons, and the maximum distance for each delivery must be within 30 km. According to the company's order system, we have obtained the weight of the goods required for each customer point and the vertical distance from the distribution center, as shown in Table 1:

Table 1. Customer point distance and demand statistics

Information	Cargo weight kg	Distance km
Customer points		
8	2	6.5
15	4.5	10
19	2.6	18
34	4	27
36	5.1	21

In this paper, the advantages and disadvantages of the calculation method considering the platform problem are mainly based on: the length of distribution time, the total time used to adjust the scheme, the number of cars dispatched, the waiting time of customers and the loss statistics caused by the number of orders refused. But the objective function only considers the timeliness of the scheme from the perspective of time.

4.2 Path Planning for Real-Time Tracking of Intelligent Logistics Distribution

According to the Road Traffic Management Bureau, based on the calculation software of highway operating speed, after inputting a street in a certain area into an electronic scanning measuring instrument, the software measures the average speed and real-time vehicle speed of each road segment in the distribution of all customer points in the area. The specific test results are shown in Table 2:

Table 2. Real time road speed statistics

Road type	Average speed	Vehicle speed	Covering road sections
Primary road	60 km/h	45 km/h	22⇔23, 22⇔34, 24⇔25
Secondary road	40 km/h	30 km/h	1⇔2, 2⇔3, 1⇔7, 7⇔9
Third level road	30 km/h	20 km/h	All others

The initial position of the vehicle is at 1:00. According to the company's order system, the current customer orders requiring service are: 8, 15, 19, 34, 36. According to the actual road conditions, we stipulate that direction 34 → 37 is a one-way street. According to the shortest path method, the nodes of the initial path are sorted as 1–8-19–15-34–36. The delivery time and shortest path data of each customer point provided according to the equipped on-board recorder are shown in Table 3: Time (min).

Table 3. Statistics of Shortest Distance Nodes and Shortest Delivery Times for Real Time Tracking of Intelligent Logistics Delivery Paths

Node	1-8	8-19	19-15	15-34	34-36
Route	1-7-8	8-12-19	19-16-15	15-24-23-22-34	34-35-36
Delivery time	1.24	1.85	1.47	2.91	1.63

According to the table above, the shortest initial path is:
1-7-8-12-19-16-15-24-23-22-34-35-36.
The expected shortest delivery time is $1.24 + 185 + 1.47 + 2.91 + 163 = 9.1$ min.
The specific real-time tracking route is shown in the figure (Fig. 5):

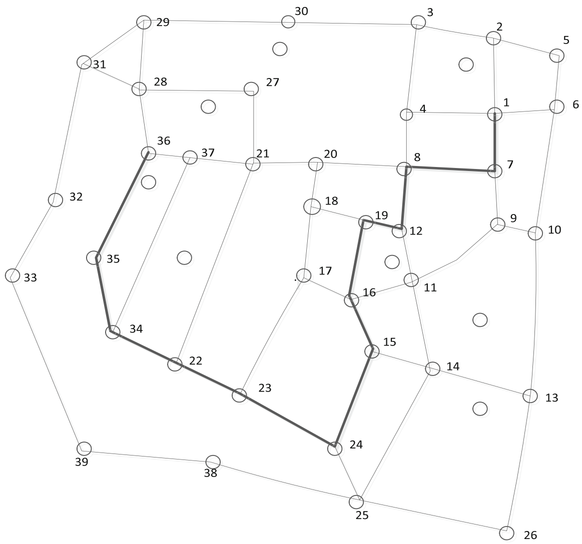


Fig. 5. Path planning diagram for real-time tracking of smart logistics delivery

However, when the distribution system started to operate, the real-time information collected by the central system for the first time was displayed 10 min later, and the new order information and road condition information were received after 8:00. Then 8 is the key point. All nodes before 8 o'clock and 8 o'clock are not considered in the replanning. Two new orders appear at 20 o'clock and 27 o'clock. 24 → 22 direction due to rush in the rush hour, the road traffic has sustained congestion, so that the traffic is severely paralyzed, all vehicles can not walk, at this time the traffic speed on the road is almost zero.

4.3 Algorithm Analysis

According to real-time information and initial order (except 8), the next six customer points to be served are 15, 19, 20, 27, 34 and 36.

The program based on recursive neural network algorithm was programmed into the MATLAB platform, all parameters were set, the recorded data collected in the input stage was clicked to start calculation, and the output calculation results were shown in Table 4: (Time: min).

Table 4. Adjusted inter node delivery time statistics for road sections

Road section	Target value	Delivery time	Wait time before service	Wait time after service
8 → 20	1.9	2.1	0.3	0.24
20 → 19	2.4	1.7	0.18	0.245
19 → 15	3.8	2.03	0.28	0.125
15 → 27	4.8	5.34	0.31	0.19
27 → 36	3.4	2.68	0.194	0.38
36 → 34	3.2	1.41	0.11	0.17

As can be seen from the above table, the general direction of the route after the redistribution of the central distribution system is: 8-20-19-15-27-36-34.

The model described in the article obtains real-time information from the traffic information center, taking into account the congestion of actual road conditions. In order to improve the timeliness of smart logistics delivery and minimize delivery time to meet customers' requirements for time window services, it is not the traditional solution that aims to have the shortest distance, but rather to allocate it more flexibly and randomly. So congestion occurs on 27-21; In the case of a ban on traffic from 37 to 34, the optimal lane for the next route was not used. Instead, the delivery continued on the road, catering to the timeliness of customers' hard time window services. However, due to severe congestion on 24 to 22, the three customer points 22, 23, and 24 in the initial path had to be temporarily abandoned.

Table 5. Statistics of objective function and actual value after improvement scheme

Road section	Target value	Actual value	Relative drop
8 → 20	1.9	2.64	−0.74
20 → 19	2.4	2.125	0.275
19 → 15	3.8	2.435	1.365
15 → 27	4.8	5.84	−1.04
27 → 36	3.4	3.254	0.146
36 → 34	3.2	4.89	−1.69

According to the statistical results in Table 5, it can be seen that the actual running time value of the distribution route after adjustment is still lower than the target value in the initial plan. However, due to the congestion of individual sections, the lag of updating information, and the response buffer time of receiving new orders, the overall service time of three sections is still lower than the target value.

Figure 6 shows the planned paths of all nodes after system readjustment.

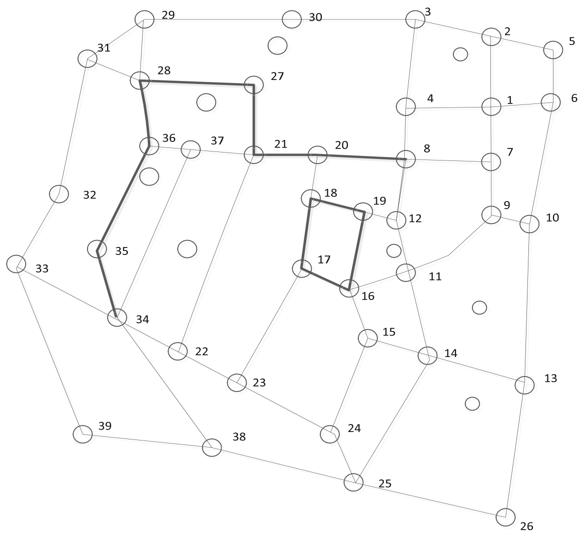


Fig. 6. Path planning scheme diagram based on real-time tracking information

It should be noted that in the figure, a double line indicates that the driving vehicle has passed through the road section twice, but the direction is opposite. Therefore, according to the above figure, the adjusted final vehicle planning path scheme is: 8-20-18-19-16-15-16-17-18-20-21-21-28-28-36-35-34.

The final delivery time used is $2.1 + 1.7 + 2.03 + 5.34 + 1.68 + 1.41 = 14.26$ min.

On the basis of the above case study, this article conducted an experiment to increase the number of regional customer points of M company in a certain location to 100. In order to prove the feasibility of the algorithm and model, and expand the research scope, a relatively complex platform problem study was conducted for single yard and multi vehicle models. The shortest path method and recursive neural network algorithm were used to calculate 6 times, and the average value was taken using MATLAB, The maximum number of iterations for the search algorithm set in the program is 5. The final result analysis is shown in Table 6: (Time: min).

Among them, the initial path creation time refers to the time taken by the two algorithms in the first stage of the planned path. Since both algorithms use Dijkstra method, the time used is the same. The running time of path optimization refers to the total time after the planned path is changed according to dynamic real-time information until the delivery plan is completed. The last column of path readjustment time refers to the sum of the first two columns, that is, the total time of the whole scheme implementation.

Table 6. Results analysis of 100 customer points processed based on recursive neural network algorithm and Dijkstra method

Project	Recurrent neural networks	Shortest path method
Target value	1.789	2.635
Aggregate demand	84	75
New demand	14	8
Number of vehicles	4	6
Number of rejected customers	2	0
Initial path build time	0.67	0.67
Path optimization run time	1.4	2.6
Path readjustment time	2.07	3.27

From the above table, it can be seen that under the same conditions, a total of 98 requirements were effectively processed using the recursive neural network algorithm, with a calculation time of 2.07 min and an average processing time of 0.021 min for each requirement; The shortest path method processed a total of 83 requirements, with a total calculation time of 3.27 min. The average processing time for one requirement was 0.039 min, and the processing level of the former was about 46.15% higher than that of the latter. Not only is the former more than the latter in terms of the number of processing requirements during the same period, but the former is far less than the latter in terms of the number of vehicles used. In addition, the former considers that due to real-time information, the calculation results are more realistic, thus having greater advantages in dealing with practical problems. When dealing with large-scale implementation path optimization problems, algorithm runtime is an important consideration. This article has also done a lot of work in this area, adopting the 2-opt positioning operator and repeated positioning operator to synchronize, and in the local search stage, a portion of customer points are first located. This greatly promotes convergence speed, and the two operators synchronize the search domain structure, successfully increasing the accuracy of the results.

It can be seen from the previous introduction that the variable domain search algorithm is essentially an iterative process, so the reasonable design of convergence criteria is also a key factor to determine the computational workload. However, the setting of the number of iterations determines the amount of computation and further affects the time of scheme adjustment, so the reasonable setting of the number of iterations is also very important. In this paper, the number of iterations is set as 5. In order to verify the rationality of such setting, the sensitivity analysis of the iterations of the target value is also made, as shown in Fig. 7: Time (s).

According to the sensitivity analysis results, it can be seen that after the function is calculated 4 times, the target value gradually stabilizes, so it is reasonable to choose 5 iterations in the article.

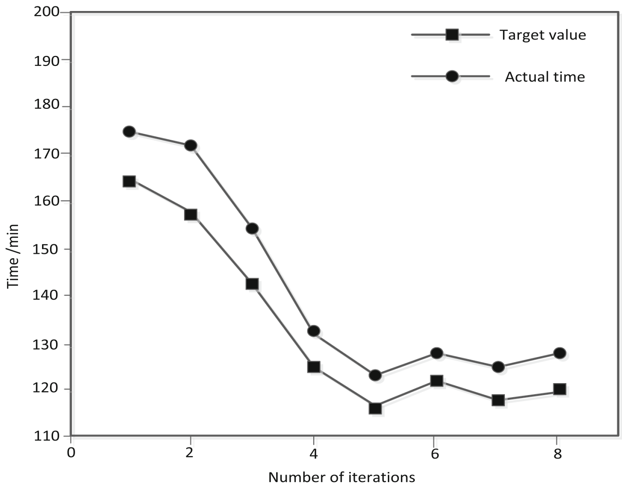


Fig. 7. Sensitivity Analysis of Target Value - Iteration Calculation Times

In order to prove that the optimized recursive neural network algorithm has more advantages than the Tabu search method and ant colony algorithm in solving such platforms under the same conditions, this paper, on the basis of unchanged previous parameter settings, conducts experiments respectively on 39 customer point cases, 60 customer points and 120 customer points in this area. The system takes the average value through eight calculations. The final comparison results are shown in Table 7: (Time: min).

Table 7. Comparison analysis of calculation results between recursive neural network algorithm and other methods

	RNN algorithm		Tabu search method		Ant colony algorithm	
	Average time	Use a car	Average time	Use a car	Average time	Use a car
39	1.426	2	1.26	3	1.41	3
60	1.624	3	1.65	5	1.503	5
120	2.29	5	2.45	6	2.06	6

Three methods can be calculated to handle the average order of one car: 2117, 15, 15. The average time of each order processed by the four methods is 1.78 min, 1.80 min and 1.85 min respectively. Obviously, it can be seen that the recursive neural network algorithm is the most efficient algorithm.

In order to more intuitively compare the advantages of recursion-based neural network algorithm compared with other algorithms of the same kind, origin software is used in this paper to make a drawing, and a simple analysis is shown in Fig. 8:

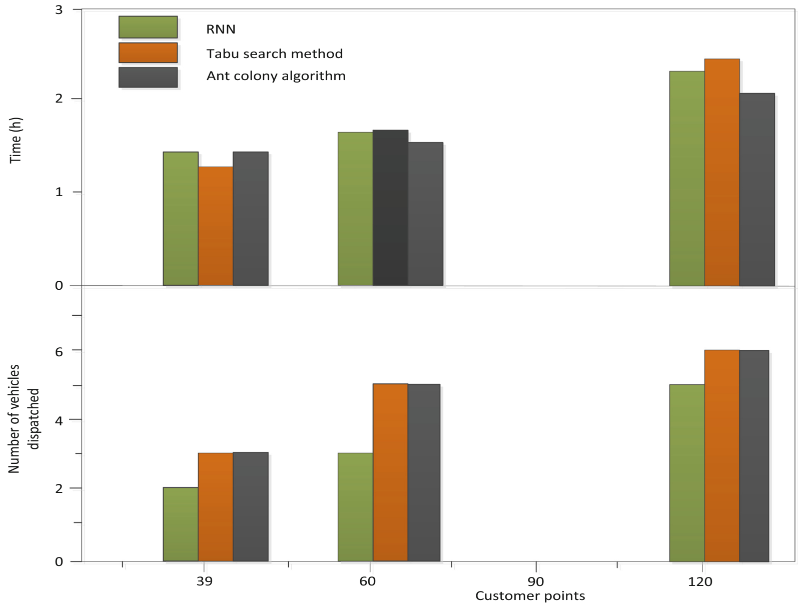


Fig. 8. Comparison of three algorithms

As can be seen from the figure above, under the same conditions, the algorithm based on recursive neural network has obvious advantages in the average time and the number of vehicles, but compared with tabu search method and ant colony algorithm, the calculation time is slightly longer than the latter two, but the number of vehicles used is less than the latter two. These differences are caused by the constant occurrence of uncertain information, which is inevitable in real life. However, considering all the factors, the algorithm based on recursive neural network is an ideal choice. In fact, when processing orders of less than 100 customer points based on recursive neural network algorithm, the response speed is lower than 0.5min, but due to the hysteresis of the actual operation, feedback and information transmission of various parts of the system, the calculation result has to be lower than the expected level. Therefore, with the increasing popularity of smart logistics in the future, this problem will soon be improved.

In order to further verify the advantages of the proposed method, the real-time tracking effects of RNN algorithm, Tabu search algorithm and Ant colony optimization algorithms were tested respectively. 500 nodes were tracked for distribution, and their tracking accuracy was counted. The comparison results are shown in Fig. 9 below.

According to the results obtained in Fig. 9, it can be seen that the tracking accuracy of the proposed algorithm does not show a decreasing trend with the increase of nodes, always maintaining around 99%. However, the real-time tracking accuracy of the other two algorithms shows a decreasing trend with the increase of nodes, and when the delivery real-time tracking node reaches 500, its tracking accuracy is 79% and 74%, respectively. Compared with the three algorithms, the proposed algorithm has better real-time tracking performance and more practical applicability.

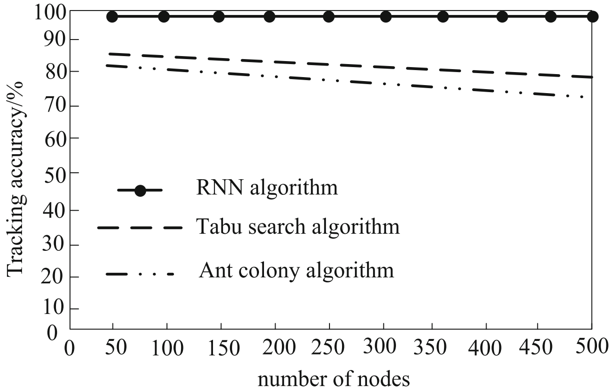


Fig. 9. Comparison Results of Tracking Accuracy

5 Conclusion

The proposal of intelligent logistics conforms to the new trend of the development of logistics industry in various countries, and modern logistics enterprises basically have various cutting-edge technologies of information and intelligence. Intelligent logistics distribution route based on real-time information is a hot topic in many academic circles. Therefore, in order to solve the problem of poor real-time tracking effect, this paper proposes a real-time tracking method of intelligent logistics distribution based on recurrent neural network. Firstly, the real-time tracking of intelligent logistics distribution is analyzed, and the planning process of two types of information, order information and road information, is determined. Secondly, considering the dynamic real-time traffic conditions analyzed above and the constantly updated customer orders and other factors, an online target tracking model based on recursive neural network is established to predict the state of each node, and then correct the corresponding target state. Finally, the Hungarian algorithm is used to solve the data association problem to reduce the detector error to the tracking algorithm. Finally, the loss function is used to optimize the model performance and achieve accurate real-time tracking of intelligent logistics distribution. The results show that the proposed method is feasible in practical application, and by comparing with other similar methods in solving the objective function, the advantages of the proposed method in real-time tracking of intelligent logistics distribution are further verified, and it has high tracking accuracy. In the future, how to design more efficient jitter methods and operators is a direction of our future research. In addition, how to quickly obtain real-time information in a short period of time, and match it with the real road conditions and electronic maps, and then quickly process and complete the planning of new schemes is also a problem that we need to continue to in-depth study. Finally, more accurate, faster and more comprehensive optimization is needed in the research of the solution algorithm.

References

1. Xie, S., Ren, J.: Recurrent-neural-network-based predictive control of piezo actuators for precision trajectory tracking. In: Proceedings of the 2019 American Control Conference (ACC). IEEE, pp. 016–023 (2019)
2. Hu, M.: Logistics vehicle tracking method based on intelligent vision. Int. J. Comput. Appl.Comput. Appl. **41**(3–4), 276–282 (2019)
3. Yang, S., Chen, Z., Ma, X., et al.: Real-time high-precision pedestrian tracking: a detection-tracking-correction strategy based on improved SSD and Cascade R-CNN. J. Real-Time Image Process. **19**(2), 287–302 (2022)
4. Wang, S.: Artificial intelligence applications in the new model of logistics development based on wireless communication technology. Sci. Program. **2021**(9), 1–5 (2021)
5. Liang, Z., Wang, J., Xiao, G., et al.: FAANet: feature-aligned attention network for real-time multiple object tracking in UAV videos. Chin. Opt. Lett. **20**(8), 8–17 (2022)
6. Voigt, S., Kuhn, H.: Crowdsourced logistics: The pickup and delivery problem with transshipments and occasional drivers. Networks **79**(3), 403–426 (2021)
7. Wang, Y., Peng, S., Guan, X., et al.: Collaborative logistics pickup and delivery problem with eco-packages based on time-space network. Expert Syst. Appl. **170**(3), 1–24 (2021)
8. Han, Q.H.: Research on the construction of cold chain logistics intelligent system based on 5G ubiquitous internet of things. J. Sens. **11**(6), 1–11 (2021)
9. Ma, L., Zhang, Y., Du, Y., et al.: Research on the framework of full-process condition monitoring and evaluation method for express logistics based on multi-information fusion and intelligent identification. IOP Conf. Ser. Mater. Sci. Eng. **740**(1), 8–19 (2020)
10. Chen, Y.T., Sun, E.W., Chang, M.F., et al.: Pragmatic real-time logistics management with traffic IoT infrastructure: Big data predictive analytics of freight travel time for Logistics 4.0. Int. J. Prod. Econ. **238**(8), 1–27 (2021)
11. Gao, D.: Design and development of intelligent logistics tracking system based on computer algorithm. J. Phys. Conf. Ser. **2074**(1), 2–12 (2021)
12. Feng, W., Wu, Y., Fan, Y.: A new method for the prediction of network security situations based on recurrent neural network with gated recurrent unit. Int. J. Intell. Comput. Cybern. **13**(1), 25–39 (2020)
13. Teng, S.: Route planning method for cross-border e-commerce logistics of agricultural products based on recurrent neural network. Soft. Comput.Comput. **15**(18), 12107–12116 (2021)
14. Xu, J., Wang, K., Lin, C., et al.: FM-GRU: a time series prediction method for water quality based on seq2seq framework. Water **13**(8), 1031 (2021)
15. Gan, H., Ou, M., Zhao, F., et al.: Automated piglet tracking using a single convolutional neural network. Biosyst. Eng.. Eng. **205**(1), 48–63 (2021)