# ETC Intelligent Navigation Path Planning Method

Jieren Cheng[1,2], Boyi Liu[1], Kuanqi Cai[3], Xiangyan Tang[1], Boyun Zhang[4]

[1] College of Information Science & Technology, Hainan University, China
[2] State Key Laboratory of Marine Resource Utilization in South China Sea, China
[3] Mechanical and Electrical Engineering College, Hainan University, China
[4] Hunan Police Academy, Department of information technology, China
cjr22@163.com, liuboyilby@163.com, 347735268@qq.com, tangxy36@163.com, 1052855713@qq.com

## Abstract

The efficacy of current path planning methods in anintelligent navigation system is compromised by poor self-adaptability and large errors in Big Data environments, because they only consider the original data in a road map and lack a comprehensive analysis of actual road conditions. In this paper, we report the details of research on the above problem. We defined the traffic transit coefficient (TTC) and traffic time-consuming index (TTCI), and then deduced formulas of for both. Based on the formulas, we designed a minimum time-consuming path planning method and desinated it the ETC (where E represents the Elman neural network, T the traffic transit coefficient, and C the traffic time-consuming index) path planning method. First, this method predicted the traffic flow on a road using the Elman neural network model. The TTCI of each section of the future unit time was calculated using the TTC. Finally, we used the Dijkstra algorithm to obtain the shortest path. Experiments and theoretical analysis showed that the ETC path planning method can adjust the parameters according to different road conditions. The method has high adaptability, high precision, and less time consumption. It has broad application prospects compared to the ordinary path planning algorithm in a Big Data environment.

**Keywords:** Vehicle navigation, Elman neural network, Traffic transit coefficient, Traffic time consuming index, Path planning

## 1 Introduction

With the rapid development of human society, increasingly more people use private vehicles, which has made urban traffic congestion an increasingly serious problem [1]. At present, the vehicle navigation system developed by Garmin Ltd. (Olathe, KS, USA) and Beijing UniStrong Science & Technology Co., Ltd. (Beijing, China), the navigator developed by Beijing Newman Ideal Digital Technology Co., Ltd. (Beijing, China), and some other systems already have the ability to conduct path planning [2]. However, most of them make navigation decisions for the user based on the principle of the shortest path and the minimum charge, which causes more vehicles to be on these paths, leading to even heavier traffic and more wasted time for users. Moreover, these methods perform poorly in terms of adaptability and accuracy. At present, since vehicle path planning [3] plays an increasingly important role in daily life, vehicle users need a more intelligent navigation system that can obtain the short-est time-consuming path to save travel time. Path planning has developed greatly in the field of transportation and communication [4-5].

Many scholars have carried out research on vehicle path planning methods. For example, Han et al. [6] proposed a reasonable path planning method for vehicles based on behavior coordination of particle swarm, which used the behavior dynamics model and particle swarm optimization algorithms to solve these problems for the optimization of behavior coordination parameters and to achieve vehicle navigation. It is able to achieve accurate vehicle path planning and navigation, but it cannot realize the identification and avoidance of road congestion. Some research teams proposed some optimal path planning algorithms for vehicle navigation systems, and implemented motion planning techniques. In these algorithms the available information, different motion planning and control techniques have been implemented to autonomously driving on complex environments. These algorithms can improve the timeliness of the shortest path algorithm effectively and, at the same time, make the executing strategies achieve maximum comfort, safety and energy savings. However, there still are problems that dynamic planning cannot properly solve at times, and the long time consumed during the process of driving a vehicle may lead to errors in urban dynamic environments [7]. Yu and Lu [8] proposed a genetic algorithm for multi-mode path planning that can be used for personalized path navigation. Researchers have also put forward numerous algorithms. The

*Corresponding Author: Boyi Liu; E-mail: liuboyilby@163.com
DOI: 10.3966/160792642018031902030

Dijkstra algorithm, which can find the shortest path to a node quickly, was proposed in Ref. [9], but its solution process is slow when there are too many nodes in the network, and it cannot be applied to a section of road with changing terrain. In 1983, Kirkpatrick, Gelatt, and Vecchi proposed the simulated annealing (SA) algorithm [10-12], imitating the material annealing process for road planning, which has the advantage of high efficiency, simplicity, and flexibility. However, it is not suitable for changing road environments due to low initial conditions. The fuzzy logic algorithm [13] can simulate the driving experience of the driver and add it into the path planning, so that the algorithm has high humanization. Once the fuzzy rule is established, however, it is difficult to adjust online, and it has poor resiliency and poor urgency-dealing capability. The grid method [14] employs search paths based on a barrier grid and a free grid to represent the map; the drawback of this method is that it is unable to handle cases in which the road environment is complex. In addition to the four typical algorithms described above, the generally used path planning algorithms also include the A* algorithm [15], the Ford algorithm [16], the augmented path algorithm [17], the queue optimization algorithm [18], etc.

In this paper, we propose an algorithm based on numerous factors, such as road condition and grade, vehicle type, and lane number. Unlike the other algorithms, which lack real-time adaptability, the proposed algorithm can predict traffic congestion based on the traffic flow data in a Big Data environment and can execute path planning based on the predicted traffic congestion. It mainly consists of a traffic flow prediction algorithm, a shortest path planning algorithm, and a traffic jam detection and traffic transit coefficient calculation model. This method can reduce the impact of transportation on the environment, gather vehicle trajectories of transportation, analyze the direction of transport vehicles, and provide convenient routes. Moreover, it can prevent traffic jams, divide vehicle flows to reduce traffic congestion, and provide eco-transportation plans for traffic managers. In addition, this method reduces pollutants and lowers the carbon dioxide emission level to the best extent possible, controls noise pollution effectively, and promotes the construction of green, low-carbon transportation modes.

## 2 ETC Path Planning Method Structure

The structure of the intelligent vehicle navigation system proposed in this paper is shown in Figure 1 (we designate the method used in this system ETC, where E represents the Elman neural network, T the traffic transit coefficient, and C the traffic time-consuming index).

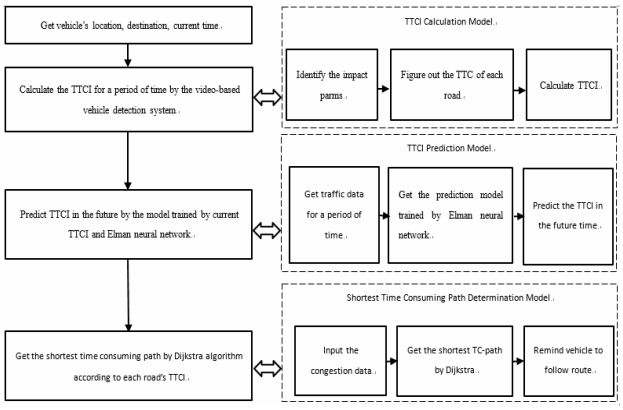



**Figure 1.** System structure diagram

## 3 Traffic Flow Prediction Based On Elman Neural Networks

In recent years, many traffic flow prediction algorithms have been proposed that can be roughly divided into two categories according to their forecasting basis: One prediction model is based on mathematical statistics and traditional mathematics such as calculus [19]; the other is developed by means mainly related to modern science, technology, and methods [20]. One of the representative results of the first category is Ahmed and Cook's first time-series model [21] in the traffic flow prediction field in 1979, which includes the autoregressive model (AR) [22], the moving average model (MA) [23], and the autoregressive moving average model (ARMA) [24-25], among others. The technology is mature and has high accuracy when the sample data are sufficient, and it needs to be used in relatively stable traffic because it has high requirements for data and needs a large amount of uninterrupted data. One of the representative traffic flow prediction algorithms in the second category is Davis and Nihan's nonparametric regressive model [26], which was applied to traffic flow prediction in 1991. Without prior knowledge, it can perform more accurately than parametric models with only sufficient historical data, but its complexity is very high. Dougherty proposed a neural network [27-28] for traffic flow prediction in 1995, which is suitable for complex and nonlinear conditions; it is also effective when the data is incomplete and inaccurate, but it requires a significant amount of learning data and the training process is complex. In addition, plenty of traffic flow forecasting methods based on the above methods, e.g., deep belief network models [29-30], support vector machines [31-34], Dynamic Multi-keyword Ranked Search Cloud Scheme [35], and wavelet neural network models [36-37], have been proposed in recent years.

Focusing on the nonlinear and nonstationary characteristics of traffic flow, this paper proposes an algorithm that uses the local recursive inner delay feedback neural network [38-39]. Compared to the

traditional Back Propagation (BP) neural network [40-41], the algorithm strengths the feedback signal. The network structure includes the input layer, the hidden layer, the output layer, and the receiver layer. Among these layers, the added receiver layer can be seen as a step delay, and can be used to store the output value of the hidden layer unit of previous time. If the input time series is a(t), the output of the feedback layer is $y_c(t)$, and the output of the network is $y(t)$, then the network can be described as

$$\begin{aligned} &\mathrm{x}(k) = f(w^1 x_c(k) + w^2 u(k-1)), \\ &x_c(k) = \alpha x_c(k-1) + x(k-1), \\ &y(k) = g(w^3 x(k)) \end{aligned} \tag{1}$$

In the above equation, $k$ represents the time scale, $x$ the one-dimensional output node vector, $y$ the n-dimensional input vector, $u$ the $n$-dimensional input vector, $x_c$ the m-dimensional feedback state vector, $w^1$ the connection weight matrix of the receiver layer to the hidden layer, $w^2$ the connection weight matrix of the input layer to the hidden layer, and $w^3$ the connection weight matrix of the hidden layer to the output layer. If the actual output of the system at step $k$ is $y_d(k)$, then the objective function (the error function) of the Elman neural network can be expressed as

$$E(k) = \tfrac{1}{2}(y_d(k) - y(k))^T (y_d(k) - y(k)) \tag{2}$$

The Elman neural network learning algorithm is based on the gradient descent method, which calculates the partial derivative of the weight and the update weight to let the partial derivative get close to zero. Accordingly, the following is obtained:

$$\begin{cases} \Delta w_{ij}^3 = \eta_3 \delta^0 xj(k) (i = 1,2,\ldots,m; \mathrm{j} = 1,2,\ldots,\mathrm{n}), \\ \Delta w_{iq}^2 = \eta_2 \delta_j^h uq(k\text{-}1) (j = 1,2,\cdots \mathrm{n}; \mathrm{q}=1,2,\ldots,\mathrm{r}), \\ \Delta w_{\mathrm{jl}}^1 = \eta_1 \sum_{\mathrm{i}=1}^{\mathrm{m}} (\delta^0 \mathrm{w}_{\mathrm{ij}}^3) \dfrac{\partial \mathrm{x}_\mathrm{j}(\mathrm{k})}{\partial \mathrm{w}_{\mathrm{jl}}^1} (i = 1,2,\ldots,\mathrm{n}; \mathrm{j} = 1,2,\ldots,\mathrm{n}) \end{cases} \tag{3}$$

$$\delta_{\mathrm{i}}^0 = (\mathrm{y}_{\mathrm{d,i}}(\mathrm{k}) - \mathrm{y}_{\mathrm{i}}(\mathrm{k})) \mathrm{g}_{\mathrm{i}}^{'}(\bullet) \tag{4}$$

$$\delta_{\mathrm{j}}^{\mathrm{h}} = \sum_{\mathrm{i}=1}^{\mathrm{m}} (\delta_{\mathrm{i}}^0 \mathrm{w}_{\mathrm{ij}}^3) \mathrm{f}_{\mathrm{i}}^{'}(\bullet) \tag{5}$$

$$\begin{aligned} &\frac{\partial \mathrm{x}_\mathrm{j}(\mathrm{k})}{\partial \mathrm{w}_{\mathrm{jl}}^1} = \mathrm{f}_{\mathrm{i}}^{'}(\bullet) \mathrm{x}_1(\mathrm{k}-1) + \alpha \frac{\partial \mathrm{x}_\mathrm{j}(\mathrm{k}-1)}{\partial \mathrm{w}_{\mathrm{jl}}^1}, \\ &(\mathrm{j} = 1,2,\cdots,\mathrm{n}; \ \mathrm{l}=1,2,\cdots,\mathrm{n}) \end{aligned} \tag{6}$$

$\eta_1$, $\eta_2$, and $\eta_3$ represent the learning steps of $w^1$, $w^2$, and $w^3$, respectively.

As shown in the network structure diagram presented in Figure 2, in the Elman neural network, the data are input through the input layer and linearly weighted by the output layer. The transfer function in the hidden layer unit can be either a linear function or a nonlinear function, and the main function of the receiver layer is to store the output memory of the hidden layer and pass it to the input layer. An Elman neural network possesses the function of short-term memory for the data, so it can reflect the trend of historical data to a certain extent. It also has the advantages of fast computation speed and real-time performance, compared to a traditional neural network. Therefore, it is suitable to apply to an intelligent navigation system.

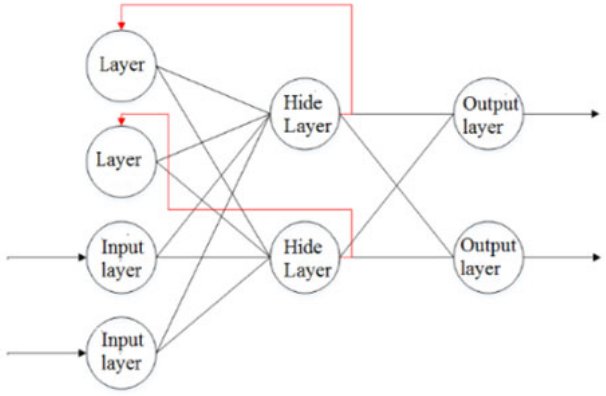



**Figure 2.** Elman neural network structure diagram

## 4 Traffic Transit Coefficient and Traffic Time-Consuming Index

Traffic flow is not only composed of drivers and vehicles, but the overall driving environment. Therefore, both the internal and external factors should be considered to increase the accuracy of prediction when predicting traffic flow. In this paper, we propose the concept of a traffic transit coefficient and introduce its calculation method, and then, based on this, we obtain the method for calculating the traffic time-consuming index. These components comprise a good foundation for final path planning.

### 4.1 Traffic Transit Coefficient

There are many traffic condition detection methods, including that which determines the traffic situation by calculating the number of vehicles in a certain area; for example, by detecting vehicles automatically and counting their number through the background subtraction method or its improved algorithm [42], virtual line analysis [43], the entropy weight method [44], etc. In addition, there are numerous methods of determining the road and traffic conditions by using various traffic indexes, such as speed, traffic density, and other factors. Chen et al. [45] determined the degree of traffic by using the virtual line analysis method. They built a model to calculate the maximum

traffic capacity of each road under the constraints of their service levels and factors according to the analysis of the different effect factors caused by different roads and traffic engineering theory. They then calculated the ratio of the maximum traffic volume and the difference between the maximum traffic volume and the actual traffic volume. The latter volume is used as the evaluation standard for traffic congestion detection. This standard's biggest advantage is that it has good adaptability to be able to adjust to different roads. Cheng et al. [46] proposed a method of traffic congestion detection in a harsh environment, but its time cost is very high and fluctuates.

The traffic transit coefficient (TTC) describedin this paper is a conceptual value that is reasonably determined by the combination of traffic flow, road grade, vehicle type, and drivers' driving technology.

The advantage of the TTC calculation method described in this paper is that it can consider the differences in service level, lane width, and driving level caused by different roads and drivers simultaneously. Determining the road traffic situation using a variety of parameters can improve the accuracy and adaptability of the method, and thus we have proposed the TTC. Its calculation method is as follows:

$$\varepsilon = \frac{C_d}{\left| C_d - C_q \right|} \tag{7}$$

In formula (7), $C_d$ represents the passing ability of vehicles in a single lane; that is, the maximum traffic capacity among different service levels. Its measurement unit is *PCU/h* (where PCU represents passenger car unit). $C_q$ represents the transit capacity of the actual road, and its measurement unit is *PCU/h*. A high TTC value means that the traffic flow is large and the road is congested. In calculating the passing capacity of a single lane, a model is built considering two aspects of the transit capacity, that of the basic sections of the highway connecting cities and that of urban road sections.

## 4.2 TTC Calculation Method

**Transit capacity of highway basic sections.** Regarding highway road conditions, the single lane transit capacity includes the following formulas:

$$C_d = M_{sv_i} \cdot N \cdot f_w \cdot f_{HV} \cdot f_p \tag{8}$$

$$M_{sv_i} = C_B \cdot (V/C)_i \tag{9}$$

$$f_{HV} = \frac{1}{1 + P_{HV}(E_{HV} - 1)} \tag{10}$$

Substituting (9) into (8), we obtain the expression for the traffic capacity:

$$C_d = \frac{C_B \cdot (V/C)_i \cdot N \cdot f_w \cdot f_p}{f_{HV}} \tag{11}$$

In the above formulae, $N$ represents the lane number of the single lane, $f_w$ the correction coefficient on the transit capacity of lane width and lateral width, $M_{sv_i}$ themaximum service traffic volume [that is, the maximum service traffic volume at a certain service level, PCU/(h In)], $C_B$ the maximum traffic capacity of a lane under ideal conditions [PCU/(h In)); $V$ the maximum service traffic volume of the service level $i$ [PCU/(h In)]; $C$ rthe basic transit capacity of the service level $i$, $f_{HV}$ the correction coefficient of the transit capacity of large vehicles, $f_p$ the correction coefficient of the transit capacity of the driver's condition, $P_{HV}$ the percentage of the large vehicles' traffic volume of the total traffic volume, and $E_{HV}$ the conversion coefficient of the conversion from large vehicles to minibuses.

**Transit capacity of urban road sections.** The transit capacity of urban road sections is modified according to the theoretical transit capacity of a lane combined with the number of lanes, the width of lanes, and other factors; that is,

$$\mathrm{C_d} = \mathrm{N_0} \cdot \gamma \cdot \eta \cdot \mathrm{C} \cdot \mathrm{n'} \tag{12}$$

$$\gamma = \frac{0.8 - (\frac{Q_{bic}}{[Q_{bic}]} + 0.5 - W_2)}{W_1} \tag{13}$$

$$\mathrm{N_0} = 1000\mathrm{V/L} \tag{14}$$

$$\mathrm{L} = \mathrm{L_0} + \mathrm{L_1} + \mathrm{V} \cdot \mathrm{t_c} + \mathrm{I} \cdot \mathrm{V}^2 \tag{15}$$

$V$ represents running speed (km/h), $L$ the headway space of continuous traffic flow (m), $L_0$ thevehicle's safety distance in the parking state (m), $L_1$ represents the body length of a vehicle (m), $t_c$ the reaction time of braking (the general value is 1 s), and $I$ a parameter associated with the vehicle's weight, road resistance coefficient, adhesion coefficient, and slope. $\gamma$ represents the correction coefficient of the bicycle's influence, $\eta$ the correction coefficient of the lane width's influence, $C$ the correction coefficient of an intersection's influence, n' the correction coefficient of the lane number, $\mathrm{Q_{bic}}$ the traffic volume of bicycles (bicycles/h), $W_2$ the width of the one-way bicycle lane (m), and $W_1$ the width of the one-way motor lane (m). Substituting formulae (13)-(15) into formula 12, we can obtain the calculation model of the transit capacity of an urban road:

$$C_d = \frac{1000V \cdot 0.8 - (\frac{Q_{bic}}{[Q_{bic}]} + 0.5 - W_2) \cdot \eta \cdot C \cdot n'}{W_1 \cdot (L_0 + L_1 + V \cdot t_c + I \cdot V^2)} \quad (16)$$

According to the different widths of motor vehicles and the different distances between intersections, the correction coefficient of the lane width's influence and the correction coefficient of the intersection's influence can be divided into two types:

$$\eta = \begin{cases} 50(W_0 - 1.5)(\%) & W_0 \le 3.5m \\ -54 + 188W_0/3 - 16W_0^2/3 & W_0 > 3.5m \end{cases} \quad (17)$$

$$C = \begin{cases} C_0 & s \le 200m \\ C_0(0.0013s + 0.75) & s > 200m \end{cases} \quad (18)$$

$C_0$ in the formula 17 represents the width of motor lane (m), $C_0$ the green time ratio of the intersections, and $V = V_f(1 - \frac{K_j}{K})$ the space between intersections.

**Deduction of the relationship between a road's travel time and traffic flow.** Greenshields proposed a linear model of the velocity-density relation in 1963 [47]:

$$V = V_f(1 - \frac{K_j}{K}) \quad (19)$$

The obstruction density $K_j$ in the above formula represents the density in the situation in which the traffic flow is too dense for the vehicles to move ($V = 0$) and $V_f$ the average vehicle speed when the density approaches 0. The relationship of the traffic speed $V$, the average flow $Q$, and the density $K$ in the continuous flow traffic is deduced as follows:

$$V_f K_j \quad (20)$$

Substituting (20) into (19), the relationship between traffic volume and density after simplification is obtained:

$$V_f K_j \quad (21)$$

Assuming that the path between the two points is l, the time expression at the smooth speed is then:

$$V_f K_j \quad (22)$$

The function of the travel speed and the time spent passing a section of road is expressed as

$$V_f K_j \quad (23)$$

Substituting (22) and (23) into (21), we have

$$V_f K_j \quad (24)$$

For a fixed road, $V_f K_j$ is a constant, and setting Z $= V_f K_j$, then

$$\frac{t_0}{t} = \frac{1}{\frac{1}{2} + \sqrt{\frac{1}{4} - \frac{Q}{Z}}} \quad (25)$$

That is, $t_0/t$ on the unitary quadratic function, so the solution is:

$$\frac{t_0}{t} = \frac{1}{\frac{1}{2} + \sqrt{\frac{1}{4} - \frac{Q}{Z}}} \quad (26)$$

In order to obtain the trend graph between $Q/Z$ and $t_0/t$, MATLAB software was used to draw its function diagram, which is presented as Figure 3.

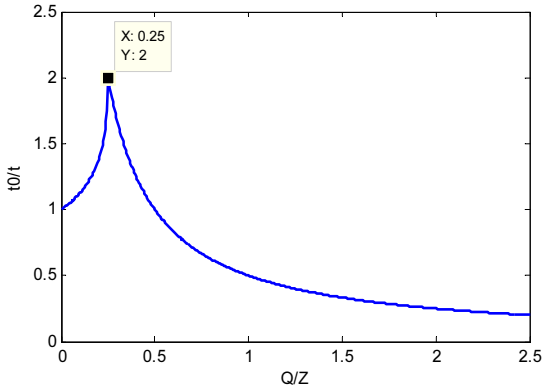



**Figure 3.** Trend graph between $Q/Z$ and $t_0/t$

The line in the figure does not represent congestion and the function described by the graph is $\frac{t_0}{t} = \frac{1}{\frac{1}{2} + \sqrt{\frac{1}{4} - \frac{Q}{Z}}}$; thus, we can see from the graph that $Q/Z$ increases from 0 to infinity. The time that traffic increased and the speed of the car decreased was observed to rise; when traffic flow reached the capacity of the road section, $Q/Z$=1/4, and the road reached maximum flow, and the traffic density and speed was optimal. When the density increased continuously, the traffic density in the crowded state was plotted as line B, and the speed of the car began to decrease. The time spent traveling on the road began to increase when the traffic density was $K_j$ and the road traffic was zero. The time passing through the road, in theory, was infinitely long.

The actual traffic on the roads is equal to the average flow rate; that is, $Q$=$C_q$. inserting formula (26) into the TTC, $\varepsilon = \frac{C_d}{|C_d - C_q|}$, the relationship between the available time and the traffic factors can then be obtained as follows:

$$t_2 = -\sqrt{\frac{t_0^2 C_d(\varepsilon - 1)}{Z\varepsilon} + \frac{t_0^2}{4}} + \frac{t_0}{2} \quad (27)$$

It can be deduced from (27) that

$$t_2 = -\sqrt{\frac{t_0^2 C_d(\varepsilon - 1)}{Z\varepsilon} + \frac{t_0^2}{4}} + \frac{t_0}{2} \quad \textbf{(28)}$$

Since $t$ is constantly greater than zero, then

$$t_2 = -\sqrt{\frac{t_0^2 C_d(\varepsilon - 1)}{Z\varepsilon} + \frac{t_0^2}{4}} + \frac{t_0}{2} \quad \textbf{(29)}$$

Since $t_2 = -\sqrt{\frac{t_0^2 C_d(\varepsilon - 1)}{Z\varepsilon} + \frac{t_0^2}{4}} + \frac{t_0}{2}$ and $\sqrt{\frac{t_0^2 C_d(\varepsilon - 1)}{Z\varepsilon} + \frac{t_0^2}{4}}$ is constantly greater than $\frac{t_0}{2}$, $t_2$ is constantly smaller than zero, and we thus terminate the calculation. According to the different $C_d$ values adopted for different roads, we obtained the function of TTC and time applied to a specific road. In order to obtain the trend graph of t and $\varepsilon$, we assumed that $C_d$ =200 PCU/h. The MATLAB function diagram is presented as Figure 4.

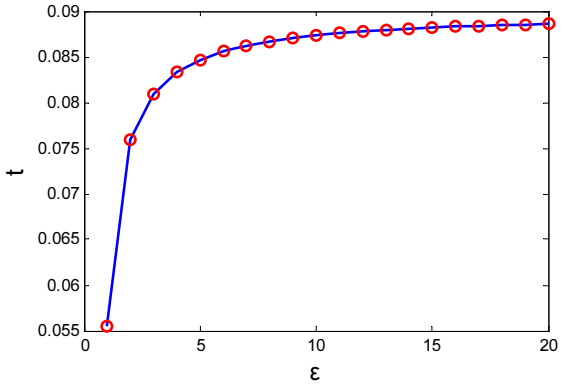



**Figure 4.** Function diagram of t and $\varepsilon$

It can be seen from the figure that $\varepsilon$ increases as time t increases, which is consistent with reality.

## 4.3 Deducing the Traffic Time-Consuming Index

The traffic time-consuming index (TTCI) proposed in this paper is the product of the average traffic flow speed on a certain road during a period of time and a basic algorithm of the TTCs, which are used as the weights of the direct path to evaluate the theoretical time cost of the road; that is, the TTCI.

For the shortest path planning problem, much research has been done and much progress achieved. Among this research, the Dijkstra algorithm is the most famous single-source shortest path planning algorithm. On the basis of this method, the shortest time-consuming path planning model is put forward in this paper.

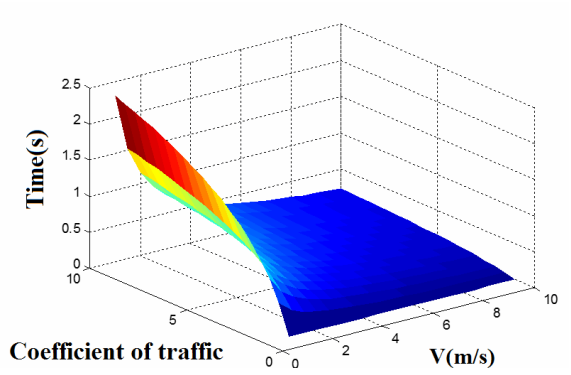
In the Dijkstra algorithm, the distance of roads is the only standard for selecting the vehicle and road. In this paper, we make use of the Dijkstra algorithm to propose the TTCI and obtain the relationship between the TTCI and the traffic transit coefficient of the road. The analysis and calculation process is as follows, in which $T$ denotes TTCI.

Regardless of the effect of traffic congestion, the time-consuming index $T$ is linearly related to the distance $L$ , and the travel time is determined by the distance and the travel speed.

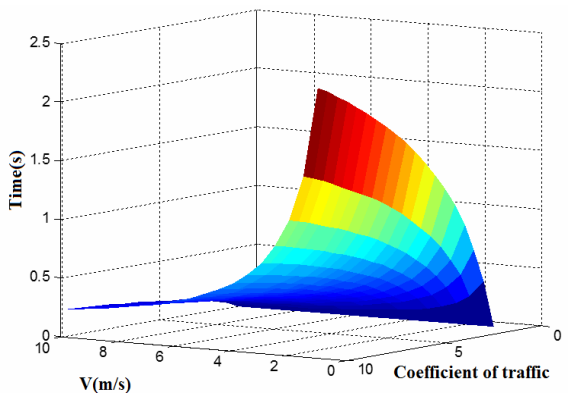
The time-consuming index $T$ is linearly related to the traffic transit coefficient when only the influence of the traffic transit coefficient is considered. The relationship is as shown in (29), and the reason is that when the traffic transit coefficient approaches 1, the travel time approaches $t_0$; that is, the speed of the car on the road approaches a smooth speed when the traffic transit coefficient continues to increase and the travel time increases as the congestion index increases.

Considering the traffic transit coefficient $\varepsilon$ and the effects of the road comprehensively, the algorithm obtains the relationship of the time-consuming index $T$, the distance $l$, traffic transit coefficient $\varepsilon$, and the vehicle's driving speed as shown in (30), and the function graphs are presented as Figure 5:

$$T = \frac{1}{V} \cdot \ln \varepsilon \quad \textbf{(30)}$$




(a)




(b)

**Figure 5.** Time, TTC, and speed graph

# 5 Shortest Time-Consuming Path Planning Algorithm

Murota [48] and other scholars believe that the Dijkstra algorithm has a great advantage in path planning when the vehicles are relatively concentrated because of its low time complexity and relatively simple calculation process. In order to decrease the reaction time by decreasing the calculation complexity, in this paper, based on the Dijkstra algorithm, we regarded each site as a node and the traffic roads as connection lines between nodes, and then calculated the shortest time-consuming index of each node to all other nodes. This approach is characterized by extending out from the starting point of the vehicle to the outer layer gradually until it extends to the end point.

## 5.1 Principle of the Algorithm

First, the algorithm regards each site as a node and the traffic roads as the connecting lines between nodes, and draws a weighted map of $G=(V,E)$. According to the actual situation of the road and the above-mentioned content, it calculates the time-consuming index of each road and divides the set of nodes $V$ into two groups. The first group is the set of nodes that the shortest time-consuming index has already calculated (denoted $P$, and there is only one starting point in the initial $P$; the shortest path will be inserted into $P$ once it is obtained, and the algorithm is finished after all of the nodes are added to $P$). The second group is the set of nodes that the shortest time-consuming index has not determined (denoted $Q$), and the nodes of the second group are added into $P$ successively according to the increasing order of the shortest time-consuming index. During the adding process, the shortest path length from the initial point $v$ to $P$ is maintained to be no greater than the length of the shortest path from the initial point $v$ to any node in $Q$. In addition, each vertex corresponds to a distance. The distance between the nodes in $P$ is the shortest path length from $v$ to the node, and the time-consuming index of nodes in $Q$ is the current shortest path length from $v$ to the point where only the nodes in $P$ are intermediate nodes.

## 5.2 Algorithm Steps

The algorithm collects the information of roads between each node, and calculates the time-consuming index for each road using the method presented above.

Initially, $P$ only contains the starting point of the vehicle; that is, $P=\{v\}$, where $v$'s time-consuming index is 0. $U$ includes all other points except $v$; that is, $U=\{\text{other points}\}$. If there is a road $u$ between $U$ and $v$, then $\langle u,v\rangle$ has a normal weight as the time-consuming index; otherwise, if there is no road between $u$ and $v$, then the $\langle u,v\rangle$ time-consuming index is $\infty$.

The algorithm selects a minimum time-consuming point $L$ from $U$, and then adds $L$ to $P$ (the selected distance is the shortest path length of $v$ to $L$)

It then sets $L$ as the intermediate point of the new consideration, and then modifies each vertex's time-consuming index in $U$. If the time-consuming index from the source point $v$ to the vertex $u$ (passing through vertex $L$) is shorter than the original time-consuming index (not passing through vertex $L$), then the algorithm modifies the time-consuming index value of vertex $u$. The modified value is the traffic time-consuming index value of vertex distance $L$ added to the weight of the side.

Repeat steps before until all vertexes included in $P$ are intermediate nodes.

# 6 Experimental Verification

## 6.1 Experimental Data and Environment

The experimental data used in this article was obtained from the traffic flow dataset published by the official website of the province of British Columbia, Canada [45]. Experiments were carried out using the data collected at monitoring points located at Royal Oak, Sooke, Six Mile Road, Burnside, and Colwood in British Columbia. The road conditions at the monitoring points are shown in Figure 6 to Figure 10. A total of 8760 h of data from each monitoring point in 2001 were used.


**Figure 6.** Road conditions at Royal Oak


**Figure 7.** Road conditions at Colwood


**Figure 8.** Road conditions at Sooke

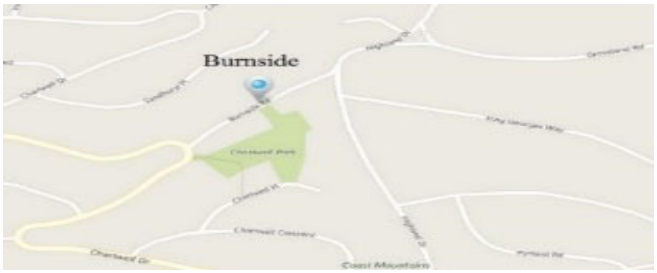



**Figure 9.** Road conditions at Burnside


**Figure 10.** Road conditions at Six Mile Road

The experiment was carried out in MATLAB on a two-core, four-thread, 2.5-GHz computer with 8 G RAM running the Linux operating system (Ubuntu 16.04 64-bit). The traffic flow was collected once every hour. The data collected from one of the monitoring points in a week are shown in Figure 11.

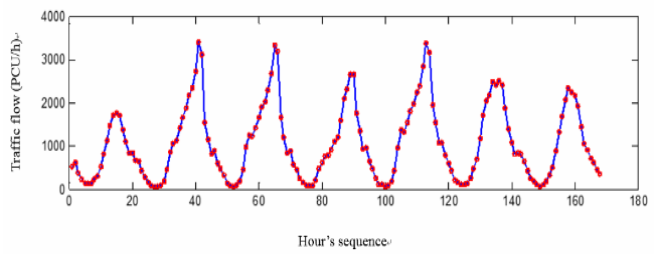



**Figure 11.** Thedata from one monitoring point in a week

## 6.2 Experiment and Analysis of Traffic Flow Prediction

Using the traffic flow data given in this paper and the Elman neural network prediction method of traffic flow, we trained and predicted five monitoring nodes, using the first 1000 h of data as the training data and the last 120 h as the test data. The prediction results from one monitoring station are shown in Figure 12.

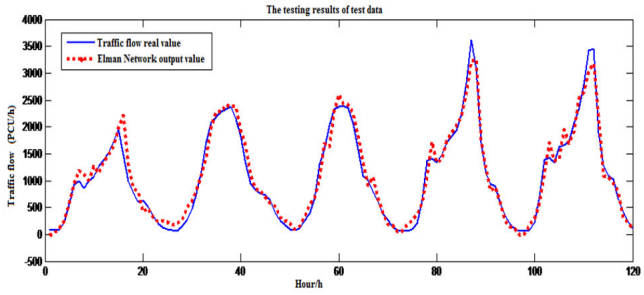



**Figure 12.** Elman neural network traffic flow prediction

Figure 13 presents the test data and results of the test's residual error distribution. Figure 14 shows the training status, where the top row is the change of gradient, the second is adopted to avoid the overfitting problem of the network, and the third is the learning rate. After 2000 rounds of training, the learning rate reached 0.08699 in Figure 14. Figure 15 is the error distribution histogram and Figure 16 shows the linear regression. The data obtained from other monitoring points can also be presented using the above method

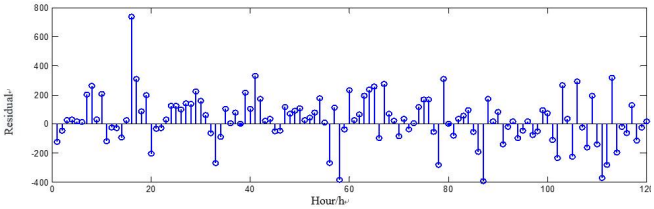



**Figure 13.** Test data and results of the test's residual error

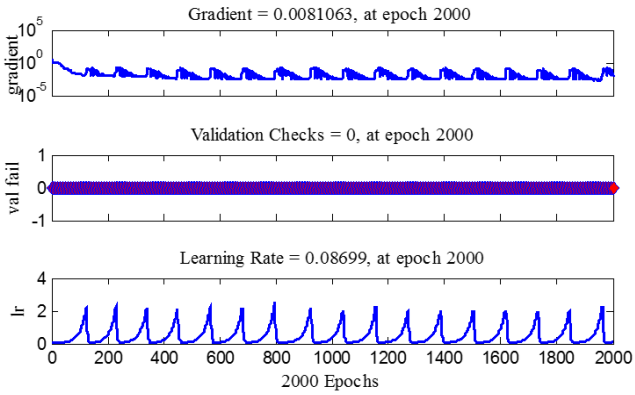



**Figure 14.** Training status

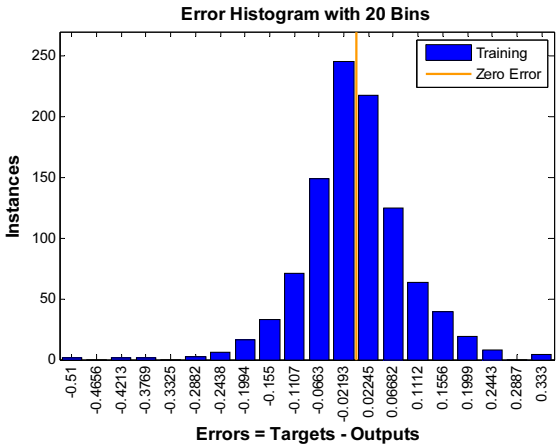



**Figure 15.** Error distribution histogram

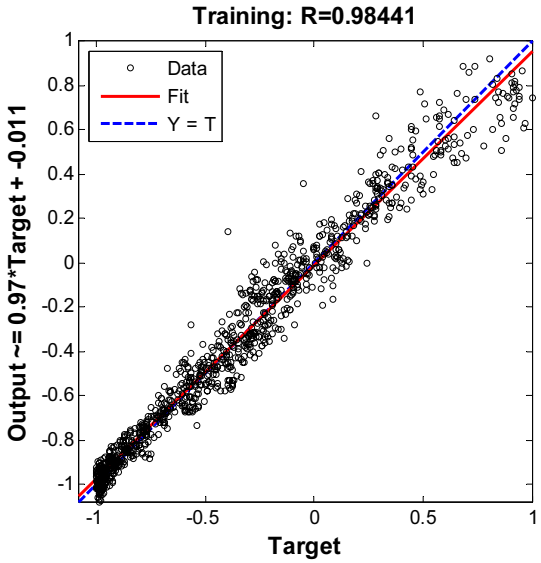



**Figure 16.** Regression chart

As can be seen from the diagrams above, by using the Elman neural network method for traffic flow forecasting, we achieved high accuracy and efficiency.

## 6.3 Experiment and Analysis of ETC Path Planning Algorithm

Finally, the data of the six nodes mentioned in Section 2.1 was tested using the intelligent navigation method proposed in this paper. As shown in Figure 17, our method selects monitoring points of the road as follows:

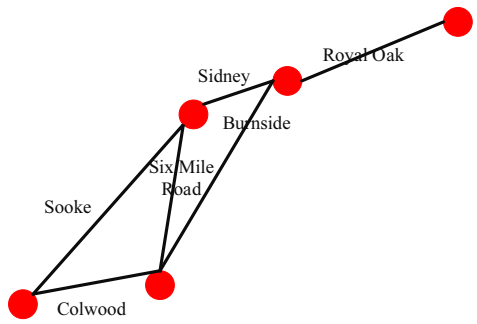



**Figure 17.** Path graph

**Step 1.** Obtain the traffic flow statistics from the monitoring points in each path.
**Step 2.** Predict the traffice flow in next unit of time based on the Elman neural network. The results are presented in Table 1.

**Table 1.** Results of Elman neural network prediction for the six roads used in experimental verification of the proposed method

| Road names | Colwood | Royal Oak | Sooke | Six Mile Road | Burnside | Sidney |
|---|---|---|---|---|---|---|
| Traffic flow | 99 | 93 | 5 | 2 | 6 | 1026 |

**Step 3.** Use a calculation model of the coefficient for the traffic to calculate the traffic transit coefficient.

The data for the six roads used in experimental verification of the proposed method are presented in Table 2. From the table, $C_B$ [PCU/(h In)] denotes the largest amount of traffic in the next lane under ideal conditions, $(V/C)_i$ is the ratio of the maximum service volume to the basic capacity of the *i*-class service level, $N$ is the number of unidirectional carriageway lanes, $f_w$ is the capacity of the correction factor of lane width and lateral net width, $f_p$ is the correction factor of the driver's condition for the traffic capacity; and $f_{HV}$ is the correction factor of traffic capacity of oversize vehicles. Substituting the above data into the Formula 8, you can get the basic section of the highway traffic capacity in Table 3.

**Table 2.** Index data for the six roads used in experimental verification of the proposed method

| | $C_B$ | $(V/C)_i$ | $N$ | $f_w$ | $f_p$ | $f_{HV}$ |
|---|---|---|---|---|---|---|
| Colwood | 2000 | 0.71 | 4 | 0.98 | 1 | 0.71 |
| Royal Oak | 2000 | 0.71 | 4 | 0.99 | 1 | 0.63 |
| Sooke | 1900 | 0.83 | 4 | 1 | 1 | 0.71 |
| Six Mile Road | 1900 | 0.51 | 4 | 0.97 | 1 | 0.67 |
| Burnside | 2000 | 0.71 | 4 | 0.96 | 1 | 0.71 |
| Sidney | 1900 | 0.71 | 4 | 1 | 1 | 0.63 |

**Table 3.** Basic capacities of the six roads used in experimental verification of the proposed method

| Road names | Colwood | Royal Oak | Sooke | Six Mile Road | Burnside | Sidney |
|---|---|---|---|---|---|---|
| $\varepsilon$ | 1.03 | 1.03 | 1.00 | 1.00 | 1.00 | 1.43 |

**Step 4.** Take the highest speed limit instead of the vehicle's speed, and then make the assumption that per unit length is a. The lengths of the six roads used in experimental verification of the proposed method are shown in Table 4.

Inserting the data into the formula, the traffic transit coefficients are obtained and presented in Table 5.

The results of the normalized processing of $\varepsilon$ is shown in Table 6.

**Table 4.** Ground speed and distance travelled of six roads used in experimental verification of the proposed method

| Road Names | Colwood | Royal Oak | Sooke | Six Mile Road | Burnside | Sidney |
|---|---|---|---|---|---|---|
| *V*(km/h) | 100 | 100 | 80 | 100 | 100 | 80 |
| T | 4*a* | 5.6*a* | 6.4*a* | 4.7*a* | 7.2*a* | 3.1*a* |

**Table 5.** Traffic transit coefficients of the six roads used in experimental verification of the proposed method

| Road names | Colwood | Royal Oak | Sook e | Six Mile Road | Burnside | Sidney |
|---|---|---|---|---|---|---|
| $\varepsilon$ | 1.03 | 1.03 | 1.00 | 1.00 | 1.00 | 1.43 |

**Table 6.** Uniformization of traffic transit coefficient of six roads used in experimental verification of the proposed method

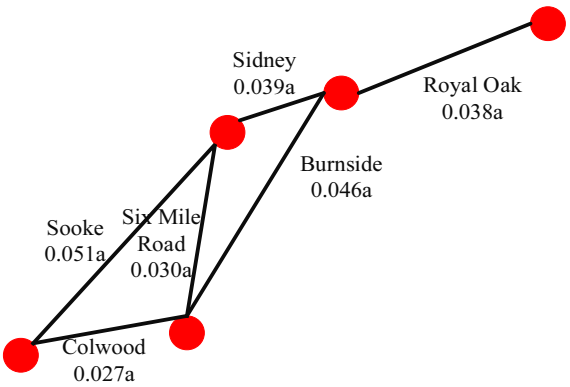
| Road names | Colwood | Royal Oak | Sooke | Six Mile Road | Burnside | Sidney |
|---|---|---|---|---|---|---|
| T | 1.96 | 1.96 | 1.9 | 1.9 | 1.9 | 2.72 |

**Step 5.** After the journey, and after the normalized processing of the traffic coefficient, insert it and the speed of the vehicle into formula (16), thetransit time-consuming indexes for the six roads used in experimental verification of the pro-posed method are obtained and shown in Table 7.

**Table 7.** Traffic time-consuming indexes for the six roads used in experimental verification of the proposed method

| Road names | Colwood | Royal Oak | Sooke | Six Mile Road | Burnside | Sidney |
|---|---|---|---|---|---|---|
| T | 0.027*a* | 0.038*a* | 0.051*a* | 0.030*a* | 0.046*a* | 0.039*a* |

**Step 6.** The time costs are obtained as presented in Figure 18. Then, inserting them into the Dijkstra algorithm, the shortest path is determined to be Colwood→Burnside→Royal Oak. Compared to the algorithm that only considers path length, the proposed method is less time-consuming.




**Figire 18.** Model of weights

## 6.4 Algorithm for Real-time Inspection

Real-time inspection is an important factor in intelligent vehicle navigation systems. Therefore, we tested the real-time performance of the algorithm. We used the detector mode of the MATLAB software to examine the algorithm and totaled the running time; the results are presented in Figure 19.

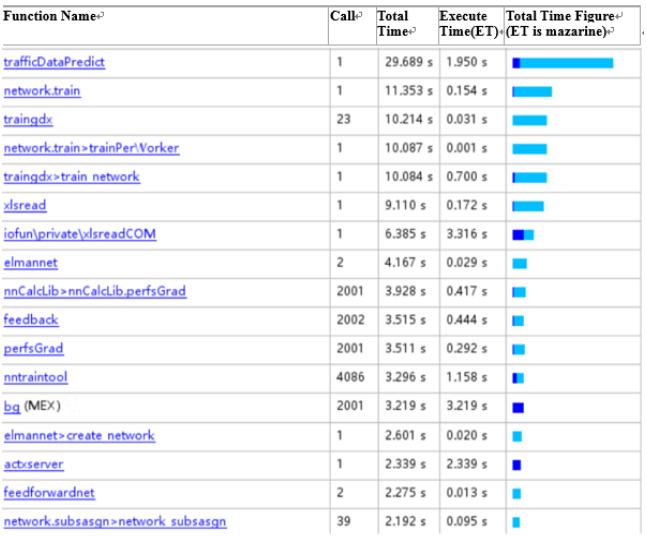



**Figure 19.** Algorithm running times

As can be seen from Figure 19, the main time cost occurred in making the traffic flow forecast, which consumed approximately 29s. The network training time and data reading time was approximately 20s, while the actual path travel time used in the traffic navigation path planning was generally more than 5 min. Therefore, the ETC path planning method proposed in this paper has a better instantaneity performance, which can meet the real-time requirements of intelligent traffic navigation systems, and its fluctuation in the time cost of the traffic flow forecast is within tolerances.

Meanwhile, compared to an ordinary traffic navigation system, the proposed intelligent traffic navigation system based on the ETC path planning method takes the time cost of a vehicle's driving process into account, which means it is more in line with actual situations. It solves the untimeliness problem in navigation caused by the fluctuation of the degree of traffic congestion at a node due to changes in traffic flow. While ensuring the instantaneity of the algorithm, this method further improves the instantaneity of navigation results in a Big Data environment.

## 7 Conclusion

We have designed an intelligent vehicle navigation path planning system based on the vehicle automatic identification method, forecasting method, the shortest path determination algorithm, and the cloud model,

which can save a user time in reaching a destination and relieve the degree of urban traffic congestion in a Big Data environment. Meanwhile, the proposed methods of calculating both the traffic transit coefficient of the video-based vehicle identification system and the traffic time-consuming index considering various factors comprehensively have high reference value in the field of traffic detection. Moreover, they also be used to alleviate traffic congestion, and can be applied to the detection of traffic congestion in a specific environment. The proposed method, in addition, has low time cost and space complexity, and it can be analyzed in the cloud platform for processing of very large amounts of data.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Project No. 61762033, 61363071, 61471169); the National Natural Science Foundation of Hainan (Project No. 617048, 2018CXTD333); Hunan Province Education Science Planning Funds (Project No. XJK011BXJ004); Hainan University Doctor Start Fund Project (Project No. kyqd1328); Hainan University Youth Fund Project (Project No. qnjj1444); State Key Laboratory of Marine Resource Utilization in the South China Sea, Hainan University; College of Information Science & Technology, Hainan University; Nanjing University of Information Science & Technology (NUIST); A Project Funded by the Priority Academic Program Development of Jiangsu Higer Education Institutions; and the Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology.

## References

[1] F. Stefanello, L. S. Buriol, M. J. Hirsch, P. M. Pardalos, T. Querido, M. G. C. Resende, M. Ritt, On the Minimization of Traffic Congestion in Road Networks with Tolls, *Annals of Operations Research*, Vol. 249, No. 1-2, pp. 119-139, February, 2017.

[2] Y. Li, J. H. Park, B. S. Shin, A Shortest Path Planning Algorithm for Cloud Computing Environment Based on Multi-access Point Topology Analysis for Complex Indoor Spaces, *The Journal of Supercomputing*, Vol. 73, No. 7, pp. 2867-2880, July, 2017.

[3] S. Yokoyama, K. Morimoto, A. Nanba, K. Katoh, K. Kuroda, H. Kishi, T. Ito, *Vehicle Navigation System*, Patent US5343399A, August, 1994.

[4] K. Curran, M. Mulvenna, C. Nugent, A. Galis, Challenges and Research Directions in Autonomic Communications, *International Journal of Internet Protocol Technology*, Vol. 2, No. 1, pp. 3-17, January, 2007.

[5] B. Cousin, J. C. Adépo, S. Oumtanaga, M. Babri, Tree Reconfiguration without Lightpath Interruption in WDM Optical Networks, *International Journal of Internet Protocol Technology*, Vol. 7, No. 2, pp. 85-95, November, 2012.

[6] S. Yang, *Study of Intelligent Vehicle Navigation System*, University of Electronic, 2014.

[7] G. J. Sheng, *The Research of Optimal Path planning Algorithms in Vehicle Navigation System*, Dalian University of Technology, 2013.

[8] H. Yu, F. Lu, A Multi-modal Multi-criteria Route Planning Method Based on Genetic Algorithm, *Acta Geodaetica Et Cartographica Sinica*, Vol. 43, No. 1, pp. 89-96, January, 2014.

[9] S. Fan, Y. Yan, S. Zhang, Z. Tian, The Optimal Route Design for Self-driving Travel Based on the Dijkstra Algorithm, *Journal of Xian University of Posts & Telecommunications*, Vol. 19, No. 1, pp. 121-124, January, 2014.

[10] Z. Du, G. Liu, Path Planning of Mobile Robot Based on Genetically Simulated Annealing Algorithm, *Computer Simulation*, Vol. 26, No. 12, pp. 118-121, December, 2009.

[11] D. Bertsimas, J. Tsitsiklis, Simulated Annealing, *Statistical Science*, Vol. 8, No. 1, pp. 10-15, February, 1993.

[12] M. S. Ganeshmurthy, G. R. Suresh, Path Planning Algorithm for Autonomous Mobile Robot in Dynamic Environment, *International Conference on Signal Processing, Communication and Networking*, Chennai, India, 2015, pp. 1-6.

[13] Q. Li, C. Zhang, C. Han, Y. Xu, Y. Yin, W. Zhang, Path Planning Based on Fuzzy Logic Algorithm for Mobile Robots in Dynamic Environments, *2013 25th Chinese Control and Decision Conference*, Guiyang, China, 2013, pp. 2866-2871.

[14] M. A. Mansor, A. S. Morris, Path Planning in Unknown Environment with Obstacles Using Virtual Window, *Journal of Intelligent & Robotic Systems*, Vol. 24, No. 3, pp. 235-251, March, 1999.

[15] W. Zhan, W. Wang, N. Chen, C. Wang, Path Planning Strategies for UAV Based on Improved A* Algorithm, *Geomatics & Information Science of Wuhan University*, Vol. 40, No. 3, pp. 315-320, March, 2015.

[16] W.-Y. Han, Improvement and Experimental Evaluation on Classical Bellman-Ford Algorithm, *Journal of Harbin Institute of Technology*, Vol. 44, No. 7, pp. 74-77, July, 2012.

[17] R. K. Ahuja, J. B. Orlin, Distance-directed Augmenting Path Algorithms for Maximum Flow and Parametric Maximum Flow Problems, *Naval Research Logistics*, Vol. 38, No. 3, pp. 413-430, June, 1991.

[18] F. Lu, Shortest Path Algorithms: Taxonomy and Advance in Research, *Acta Geodaetica Et Cartographic Sinica*, Vol. 30, No. 3, pp. 269-275, August, 2001.

[19] D. Romero, N. Rico, M. I. Garcia-Arenas, Modellation and Forecast of Traffic Series by a Stochastic Process, *Time Series Analysis and Forecasting*, Granada, Granada, Spain, 2016, pp. 279-292.

[20] P. Lopez-Garcia, E. Onieva, E. Osaba, A. Masegosa, A. Perallos, A Hybrid Method for Short-term Traffic Congestion Forecasting Using Genetic Algorithms and Cross Entropy, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 17, No. 2, pp. 557-569, February, 2016.

[21] N. Fan, X. Zhao, M. Dai, Y.-S. An, Short-term Traffic Flow

Prediction Model, *Journal of Traffic and Transportation Engineering*, Vol. 2, No. 4, pp. 114-119, August, 2012.

[22] S. V. Kumar, L. Vanajakshi, Short-term Traffic Flow Prediction Using Seasonal ARIMA Model with Limited Input Data, *European Transport Research Review*, Vol. 7, No. 3, pp. 1-9, September, 2015.

[23] C. K. Moorthy, B. G. Ratcliffe, Short Term Traffic Forecasting Using Time Series Methods, *Transportation Planning and Technology*, Vol. 12, No. 1, July, 1988, pp. 45-56.

[24] A. Wiesel, O. Bibi, A. Globerson, Time Varying Autoregressive Moving Average Models for Covariance Estimation, *IEEE Transactions on Signal Processing*, Vol. 61, No. 11, pp. 2791-2801, June, 2013.

[25] B. Williams, P. Durvasula, D. Brown, Urban Freeway Traffic Flow Prediction: Application of Seasonal Autoregressive Integrated Moving Average and Exponential Smoothing Models, *Journal of the Transportation Research Board*, Vol. 1644, No. 1, pp. 132-141, January, 1998.

[26] G. A. Davis, N. L. Nihan, Nonparametric Regression and Short-term Freeway Traffic Forecasting, *Journal of Transportation Engineering*, Vol. 117, No. 2, pp. 178-188, February, 1991.

[27] A. Nazemi, B. Abbasi, F. Omidi, Solving Portfolio Selection Models with Uncertain Returns Using an Artificial Neural Network Scheme, *Applied Intelligence*, Vol. 42, No. 4, pp. 609-621, June, 2015.

[28] M. Dougherty, A Review of Neural Networks Applied to Transport, *Transportation Research Part C: Emerging Technologies*, Vol. 3, No. 4, pp. 247-260, August, 1995.

[29] H. Zhang, F. Zhou, W. Zhang, X. Yuan, Z. Chen, Real-time Action Recognition Based on a Modified Deep Belief Network Model, *IEEE International Conference on Information and Automation*, Hailar, Nei Menggu, 2014, pp. 225-228.

[30] F. Liu, B. Liu, C. Sun, M. Liu, X. Wang, Deep Belief Network-based Approaches for Link Prediction in Signed Social Networks, *Entropy*, Vol. 17, No. 4, pp. 2140-2169, April, 2015.

[31] B. Gu, V. S. Sheng, A Robust Regularization Path Algorithm for ν-Support Vector Classification, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 1, No. 5, pp. 1-8, March, 2016.

[32] C. Bai, Z. R. Peng, Q. C. Lu, J. Sun, Dynamic Bus Travel Time Prediction Models on Road with Multiple Bus Routes, *Computational Intelligence and Neuroscience*, Vol. 2015, No. 63, pp. 1-9, July, 2015.

[33] Z. Fu, K. Ren, J. Shu, X. Sun, F. Huang, Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 27, No. 9, pp. 2546-2559, September, 2016.

[34] B. Gu, V. S. Sheng, A Robust Regularization Path Algorithm for ν-Support Vector Classification, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 28, No. 5, pp. 1241-1248, February, 2016.

[35] Z. Xia, X. Wang, X. Sun, Q. Wang, A Secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Data, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 27, No. 2, pp. 340-352, February, 2015.

[36] S. Bahri, Applied Multiresolution B-Spline Wavelet to Neural Network Model and Its Application to Predict Some Economics Data, *Journal of Chromatography*, Vol. 514, No. 2, pp. 241-251, March, 2016.

[37] Y. Chai, D. Huang, L. Zhao, A Short-term Traffic Flow Prediction Method Based on Wavelet Aalysis and Neural Network, *Control and Decision Conference*, Las Vegas, NV, 2016, pp. 7030-1034.

[38] W. Jia, D. Zhao, T. Shen, S. Ding, Y. Zhao, C. Hu, An Optimized Classification Algorithm by BP Neural Network Based on PLS and HCA, *Applied Intelligence*, Vol. 43, No. 1, pp. 176-191, July, 2015.

[39] X. Z. Gao, X. M. Gao, S. J. Ovaska, A Modified Elman Neural Network Model with Application to Dynamical Systems Identification, *IEEE International Conference on Systems, Man and Cybernetics*, Beijing, China, 1996, pp. 1376-1381.

[40] X. H. Han, X. Xiong, F. Duan, A New Method for Image Segmentation Based on BP Neural Network and Gravitational Search Algorithm Enhanced by Cat Chaotic Mapping, *Applied Intelligence*, Vol. 43, No. 4, pp. 855-873, December, 2015.

[41] Z. Li, X. Zhao, BP Artificial Neural Network Based Wave front Correction for Sensor-less Free Space Optics Communication, *Optics Communications*, Vol. 385, No. 15, pp. 219-228, February, 2017.

[42] J. J. Qu, Y. H. Xin, Combined Continuous Frame Difference with Background Difference Method for Moving Object Detection, *Acta Photonica Sinica*, Vol. 43, No. 7, 2014.

[43] T. Kryjak, M. Komorkiewicz, M. Gorgon, Hardware-software Implementation of Vehicle Detection and Counting Using Virtual Detection Lines, *Conference on Design and Architectures for Signal and Image Processing*, Madrid, Spain, 2014, pp. 1-8.

[44] B. Liu, J. Cheng, X. Tang, et al., Moving Vehicles Recognition in Complex Dynamic Environment, *Frontiers of Computer Science & Technology*, Vol. 11, No. 1, pp. 134-143, May, 2017.

[45] H.-T. Chen, L.-W. Tsai, H.-Z. Gu, B.-S. P. Lin, Traffic Congestion Classification for Nighttime Surveillance Videos, *IEEE International Conference on Multimedia and Expo Workshops*, Melbourne, Australia, 2012, pp. 169-174.

[46] J. Cheng, B. Liu, X. Tang, A Traffic-Congestion Detection Method for Bad Weather Based on Traffic Video, *International Symposium on Intelligence Computation and Applications*, Guangzhou, China, 2015, pp. 21-22.

[47] M. Slinn, P. Matthews, P. Guest, *Traffic Engineering Design*, Butterworth-Heinemann, 2005.

[48] K. Murota, A. Shioura, Dijkstra's Algorithm and L-concave Function Maximization, *Mathematical Programming*, Vol. 145, No. 1, pp. 163-177, June, 2014.

## Biographies

**Jieren Cheng** was born in 1974. He received the Ph.D. degree from School of Computer, National University of Defense Technology in 2010. Now he is a professor and graduate supervisor at Hainan University. His research interests include cloud computing, artificial intelligence, network security and intelligent transportation, etc.

**Boyi Liu** was born in 1995. He is Master graduate student. His research interests include intelligent transportation, robot vision and so on.

**Kuanqi Cai** was born in 1996. He is Master graduate student. His research interests artificial intelligence, robot design and so on.

**Xiangyan Tang** was born in 1981. She received the M.S. degree from School of Computer, Hunan Agricultural University in 2011. Now she is a lecturer at Hainan University. Her research interests include artificial intelligence, network security and intelligent transportation, etc.

**Boyun Zhang** was born in 1972. He received his Ph.D. degree from the National University of Defense Technology in 2007. He is currently a professor of computer science in Hunan Police Academy, Changsha, China. His research interests include pattern recognition and information security.