

Research Article

A Crash Severity Prediction Method Based on Improved Neural Network and Factor Analysis

Chen Zhang,^{1,2} Jie He ,¹ Yin Hai Wang,² Xintong Yan,¹ Changjian Zhang,¹ Yikai Chen,³ Ziyang Liu,¹ and Bojian Zhou¹

¹School of Transportation, Southeast University, Sipailou 2#, Nanjing 210018, Jiangsu, China

²Smart Transportation Applications and Research Laboratory, University of Washington, Seattle, WA 98195-2700, China

³School of Automobile and Traffic Engineering, Hefei University of Technology, Hefei 230009, China

Correspondence should be addressed to Jie He; hejie@seu.edu.cn

Received 25 April 2020; Accepted 4 June 2020; Published 30 June 2020

Guest Editor: Longzhuang Li

Copyright © 2020 Chen Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Crash severity prediction has been raised as a key problem in traffic accident studies. Thus, to progress in this area, in this study, a thorough artificial neural network combined with an improved metaheuristic algorithm was developed and tested in terms of its structure, training function, factor analysis, and comparative results. Data from I-5, an interstate highway in the Washington State during the period of 2011–2015, were used for fitting and prediction, and after setting the theoretical three-layer neural network (NN), an improved Particle Swarm Optimization (PSO) method with adaptive inertial weight was offered to optimize the NN, and finally, a comparison among different adaptive strategies was conducted. The results showed that although the algorithms produced almost the same accuracy in their predictions, a backpropagation method combined with a nonlinear inertial weight setting in PSO produced fast global and accurate local optimal searching, thereby demonstrating a better understanding of the entire model explanation, which could best fit the model, and at last, the factor analysis showed that non-road-related factors, particularly vehicle-related factors, are more important than road-related variables. The method developed in this study can be applied to a big data analysis of traffic accidents and be used as a fast-useful tool for policy makers and traffic safety researchers.

1. Introduction

1.1. Crash Severity. Traffic safety is a challenging task to be accomplished and has been identified as crash hotspots around the world. The total number of fatal crashes in the U.S. increased to around 35,000 in 2016. In addition, according to the Washington State Collision Summary report, a total of 117,053 crashes were identified in the Washington State, including 499 fatal collisions, 36,531 injury collisions, and 77,358 property-damage-only collisions, indicating a crash occurred every 4.5 min and a person died in a crash every 16 hours [1]. Figure 1 shows the traffic fatality rates between the U.S. and Washington State [1]. Billions of dollars in personal and property damage are wasted in traffic crashes each year around the world [2].

To achieve the intrinsic goal of exploring numerous factors that trigger the crash, crash severity is often used for

crash analyses to represent the degree of injury. A KABCO scale was proposed by WSDOT to represent the level of the injury: K—fatal injury; A—incapacitating-injury; B—nonincapacitating injury; C—minor injury; and O—property-damage-only injury. The KABCO scale has been widely adopted and adapted by many scholars (e.g., [3–5]). In this paper, the predicted targets were regrouped into three categories based on the combination of the KABCO scale and crash severity category of 5-year data (2011–2015) obtained from the HISI data system, namely, incapacitating-injuries, injuries, and noninjuries.

1.2. Crash Severity Prediction. Crash prediction problems have long been a popular area of study around the world. Numerous studies conducted the prediction analyses based on classic statistical models, e.g., the linear, nonlinear,

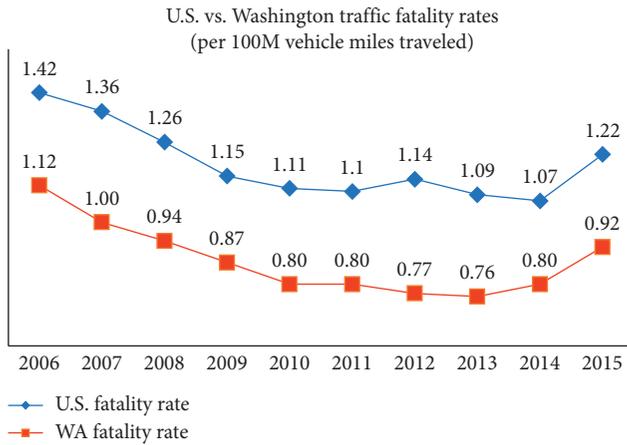


FIGURE 1: Traffic fatality rates in the U.S. vs. Washington State (per 100M vehicle miles traveled). Source: CLAS (WSDOT) and FARS (WTSC).

generalized linear model (GLM), generalized estimating equation (GEE), nominal binary (NB), and Poisson regression models, which are regarded as a good attempt at thoroughly formulating the relationship between tens or hundreds of explanative variables. However, it should be noted that the traditional statistical method has its limitation. The artificial intelligence (AI) technique, particularly, the deep learning methodology [6], as an emerging but promising tool for addressing the problems faced by the traditional statistical domain, deserves more attention and exploration.

Safety performance functions (SPFs) are frequently utilized to demonstrate the relationships between different crashes and crash impact parameters. Such functions usually use the crash frequency as the targeting variable. The Highway Safety Manual [7] has several chapters demonstrating the average crash frequency of an entire network, facility, or individual site. Elvik et al. [8] developed six power functions to demonstrate the relationship between speed and road safety; these six equations are slightly adjusted according to the road characteristics under the following categories: fatal injuries, fatal and serious injuries, all injuries, fatal accidents, fatal and serious accidents, and all injury accidents. Russo et al. [9] used a negative binomial regression model to develop four sets of SPFs based on the crash frequency and conducted a residual analysis to prove the accuracy. Park and Abdel-Aty [10] developed several crash modification factors (CMFs) for a combination of traffic and roadway cross-sectional elements on noncurved and curved roadway sections using a cross-sectional method and found that CMFs for increasing the lane and shoulder width were decreasing as the annual average daily traffic (AADT) level increased. Using a Bayesian ranking technique, Ahmed et al. [11] examined the safety effects of the roadway geometry on the crash occurrence rate along a freeway section that features mountainous terrain and an adverse weather condition and confirmed that segments with steep downgrades are more crash-prone along the studied section.

A number of researchers are eager to dig into the use of statistical analysis for traffic safety research, such as linear, nonlinear, GLM, GEE, NB, or Poisson regression models. Such models have performed well when the number of explanatory variables is constrained. Debrabant et al. [12] applied an autoregressive Poisson–Tweedie model to mine spatially and temporally aggregated hospital records of traffic accidents, and it was confirmed that this method is very accurate when applied to a black spot identification problem. Pei et al. [13] developed a joint probability model combined with a Markov chain, Monte Carlo (MCMC) approach, and full Bayesian to estimate the effects of explanatory factors, and their results indicated that the model achieves a good statistical fit and provides an accurate analysis of the influences of the explanatory factors. El-Basyouny et al. [14] applied a multivariate model based on a MCMC simulation to address the impact of weather elements, and their results showed that temperature and snowfall are statistically significant with intuitive signs for all crash types, whereas rainfall is mostly insignificant, as is the maximum wind gust with a few exceptions that are positively related to the crash type.

To address the shortcomings of the traditional statistical model, scholars in the crash analysis field are more willing to use AI methods at present. Karlaftis and Vlahogianni [15] discussed the differences and similarities between a statistical method and neural networks (NNs), and their results showed that the goals of the analysis are more important than the tools used, and that there are always assumptions to all modeling approaches. Specific to a traffic accident analysis, although classic statistical models such as NB, Poisson, and a Bayesian network can achieve a good identification of a broad range of risk factors, they are also limited to a finite factor assumption as compared with a deep-learning method [16]. Aided by the powerful hardware and software of modern computers, deep-learning methods are becoming powerful tools for many aspects of our daily lives [6]. For example, in a crash analysis, Zeng and Huang [17] used a pruned NN for the crash severity, adopting a convex combination (CC) training algorithm and a NN pruning for a function approximation (N2PFA) structure optimization method, and found that the CC outperforms the backpropagation (BP) method in both convergence ability and training speed; in addition, simplification of the nodes in an NN structure can obtain a better performance. Huang et al. [18] developed an optimized radial basis function neural network (RBFNN) model to analyze the relationships between crash frequency and the relevant risk factors, and their comparative work showed that RBFNN models outperform negative binomial models and backpropagation neural network (BPNN) models. Li et al. [5] developed a data-driven method combining the non-dominated sorting genetic algorithm (NSGA-II) with an NN to identify the key factors in a fatal highway crash analysis. All of the abovementioned studies have focused more on the NN structure itself, using complicated mathematical equations to illustrate their abstract concepts; nevertheless, they have seldom given a thoughtful, detailed, and general

procedure to deal with a complete traffic severity analysis problem.

However, in short, both statistical and AI methods in the previous studies are still facing some challenges, and especially for neural networks, the traditional NNs are more easily stuck in the local optimum in accordance with its random weights initialization at the very first beginning. Although some studies [5, 18] were carried out to address the abovementioned problems, the efficiency of the global optimum search for particle swarm optimization (PSO) and local optimum search for some other optimization methods is still to be improved.

To solve the aforementioned problems, the purpose of this paper is as follows: (1) to provide a BPNN algorithm integrating the PSO with adaptive inertial weights for the establishment of the crash severity prediction model; (2) to conduct a detailed factor analysis (FA) based on the refined model to quantify the internal relationship and heterogeneity of different variables that trigger the crash of distinguished severity. The prediction target in the first phase is the crash severity. It should be noted that the novelty of the paper lies in the fact that an integrated method incorporating an emerging AI technique and a traditional statistical model was provided for crash severity analysis. Besides, it is marginally original to use FA and PSO for the calculation of the parameters of the crash triggers.

This paper is organized as follows: in the first part, the background and severity levels are discussed to demonstrate the reason for conducting this research. In the second part, classic statistical models and NN models dealing with crash analysis are reviewed to illustrate their advantages and limitations. The following section demonstrates the processing of the dataset, including its description and simplification. In the fourth part, the entire methodology for the developed model and data incorporation are presented to highlight the process of conducting a severity prediction process. Finally, the results are presented with the conclusion.

2. Data Description

The dataset used in this study consisted of the data acquired from the Highway Safety Information System (HSIS); in this system, data from nine states (California, Washington, Minnesota, Michigan, Maine, Ohio, North Carolina, and Illinois) in the U.S. are available. Considering the author's time studying in the Washington State during the period of 2016 to 2017, the crash data from this were selected as the target data.

The HSIS data contained, roughly, two tables for the variables. The first one is related to Accident, Vehicle, and Occupant files, which involve TIME, ENVIRONMENT, ACCIDENT-RELATED INFORMATION, VEHICLE INFORMATION, DRIVER INFORMATION, OCCUPANT, ROADWAY ELEMENTS, and PEDESTRIAN/BICYCLIST INFORMATION, whereas the second table was more concerned with the roadway containing LOCATION/LINKAGE ELEMENTS, ROADWAY CLASSIFICATION, ROAD ALIGNMENT, CROSS SECTION, ROAD

FEATURES, TRAFFIC CONTROL/OPERATIONS, and TRAFFIC DATA. In addition, there were tens of sub-variables for both the tables.

The crash data from I5 in the Washington State covering the years 2011–2015 were extracted. The data from the first four years, 2011–2014, were used to fit the model, and the data from the later year were used as the prediction validation set. The total crashes were 9926, 10083, 10127, 11628, and 12804 from 2011–2015. Based on these raw samples, the following steps need to be conducted before digging into the model input procedure:

- (1) Exclude apparently irrelevant variables. More than 40 features were requested from the HSIS database system, and some features, such as "CASENO" (accident case number), "MILEPOST" (milepost), "RD_INV" (a linkage variable on the Accident file which is used in the merging operation), and "RTE_NBR" (route number) are not related to the crash severity and were omitted for simplicity.
- (2) Samples with features such as "LIGHT," "WEATHER," and "ACCTYPE" (accident type) which have values such as "UNKNOWN," "NAN," "UNSTATED," and "NULL" were also omitted for simplicity.
- (3) Some nominal variables which cannot be denoted by the continuous number such as "DIR_CURV" (the horizontal curve direction) and "DIR_GRAD" (the vertical curve grade direction), both representing the relative direction of left or right, were transformed into discrete scale values ("1" or "0").
- (4) The vehicle-related and driver-related variables such as "DRV_AGE" (driver age) and "DRV_SEX" (driver sex) were incorporated with accident-related data files through the "CASENO" label, whereas the grad/curve-related variables were incorporated with accident-related data through "MILEPOST"; here, a data process computer program written in MATLAB was developed to locate the "MILEPOST" between "BEGPOST" and "ENDPOST" in the grad/curve files.
- (5) "VEHYR," which indicates the vehicle model year, was transformed into the vehicle operation year through the following formula:

$$\text{Vehyr}_m^i = \begin{cases} 100 - \text{Vehyr}_{\text{raw}}^i + \text{val}(\text{year}^i) \\ (\text{Vehyr}_{\text{raw}}^i > \text{val}(\text{year}^i)), \\ \text{val}(\text{year}^i) - \text{Vehyr}_{\text{raw}}^i \\ (\text{Vehyr}_{\text{raw}}^i \leq \text{val}(\text{year}^i)), \end{cases} \quad (1)$$

where Vehyr_m^i refers to the vehicle operation year in the year of i , $\text{Vehyr}_{\text{raw}}^i$ refers to the vehicle model year i in the raw file (for example, "11" represents the year 2011 and "98" represents the year "1998"), and $\text{val}(\text{year}^i)$ refers to the value of the year. For example,

if a car was produced in the year 2011 and a crash occurred in the year 2013, $Vehyr_m^{2013}$ should be $(year^i) - Vehyr_{raw}^i$, indicating $13 - 11 = 2$.

- (6) The output file contains the vectors derived from the “SEVERITY” variable in the raw dataset. In detail, noninjury was derived from “1, No Injury,” injury was derived from “6, Nondisabling Injury; 7, Possible Injury,” and incapacitating injury was derived from the remaining. The vectors are described through the following formula:

$$Out_i = (B_{Injury}, B_{Non-Injury}, B_{Incapacitating-injury}), \quad (2)$$

where Out_i refers to the output of the crash item i , and $B_{Injury}, B_{Non-Injury}, B_{Incapacitating-injury}$ refer to the Boolean index of the crash severity for an injury, noninjury, and incapacitating injury.

After the processing was conducted through the abovementioned steps, a total of 4310, 4494, 4436, 4666, and 4984 samples from 2011 to 2015 were used for model fitting and validation; glancing at the data, crashes seemed to be slightly more prone to occur during the winter (cold season) and on work days, whereas younger drivers contribute to a significant number of accidents (Figure 2).

After the selection of 20 features (Table 1) available from the raw file, the next step for the data cleaning and processing work was variable standardization which was carried out using the min-max method through the following formula:

$$x_n = \frac{(x - \text{MinValue})}{(\text{MaxValue} - \text{MinValue})}. \quad (3)$$

From Table 1, we can see that, among all 20 features, there are 15 categorical and five continuous variables. In addition, for a research perspective, we divided all variables into two large categories: road-related and nonroad-related.

3. Methodology

3.1. NN with the BP Method. An artificial neural network uses information technology to mimic the human neurons and can process complicated connections between the input, hidden, and output layers. Among the multiple-layer neural networks, a three-layer simple NN has been proven to be most adopted and effective in a previous research [19].

In the forward propagation three-layer NN, the input variables can be defined as an input vector X :

$$X = (x_1, x_2, x_3, \dots, x_i, \dots, x_I)^T, \quad (4)$$

where x_i refers to the i th input variable, $I = 20$, and T refers to a transpose in the matrix calculation.

Similarly, the expectation of the crash severity level output vectors Ψ can be

$$\Psi = (\psi_{Injury}, \psi_{Non-Injury}, \psi_{Incapacitating-Injury})^T, \quad (5)$$

where ψ_{Injury} refers to the Boolean index for an injury-related crash, $\psi_{Noninjury}$ refers to a Boolean index for a noninjury-related crash, and $\psi_{Incapacitating-Injury}$ refers to a Boolean index for an incapacitating injury.

In addition, the weight matrix between the input and hidden layers, W^1 , should be

$$W_{j,i}^{(1)} (j = 2, \dots, J; i = 1, \dots, I), \quad (6)$$

where $W_{j,i}^{(1)}$ denotes the weight between the i th input node and j th hidden node and J refers to the total number of the hidden layers, $I = 20$.

The weight matrix between the hidden and output layers, W^2 , should be

$$W_{k,j}^{(2)} (k = \{\text{Injury, Non - injury, Incapacitating - Injury}\}; j = 1, \dots, J), \quad (7)$$

where $W_{k,j}^{(2)}$ denotes the weight between the k th output node and the j th hidden node, J refers to the total number of hidden layers, and the k vector has values indicating injuries, noninjuries, and incapacitating injuries.

The structure of a general three-layer neural network is shown in Figure 3.

Using the forward calculation method [17], the outputs of the nodes in the hidden layer $H_j(m)$ can be depicted as

$$H_j(m) = g_j \left(\sum_{i=1}^I W_{j,i}^{(1)} \cdot x_i(m) + \beta_j \right), \quad (8)$$

where m is the number of the output neurons, $W_{j,i}^{(1)}$ denotes the weight between the i th input node and the j th hidden node, J refers to the total number of nodes in hidden layer, g_j is the activation function between the input and hidden layers, and β_j is the bias term.

Similarly, the outputs calculated from the hidden layer $\psi_k(m)$ are shown as

$$\psi_k(m) = g_k \left(\sum_{j=1}^J W_{k,j}^{(2)} \cdot H_j(m) + \theta_k \right), \quad (9)$$

where $W_{k,j}^{(2)}$ denotes the weight between the k th output node and the j th hidden node, J refers to the total number of nodes in the hidden layer, the k vector has values including injuries, noninjuries, and incapacitating injuries, and g_k is the activation function between the output layer and hidden layers. θ_k is the bias term.

Generally, the activation functions such as sigmoid or \tanh are selected and have the ability to transform the input signal into a certain range. If the network adopts sigmoid function as the output activation function, the output can be narrowed into a small scale as $(0, 1)$; however, between the hidden and input layers, usually, the \tanh function is adopted because it usually can converge fast.

Another aspect used for building an NN is the definition of the number of nodes in the hidden layer, and there is no easy and complete mathematical way of defining this number; however, based on the experience from former research [20], the empirical equation used is as follows:

$$n = \sqrt{n_i + n_o} + \alpha, \quad (10)$$

where n is the number of hidden nodes, n_i is the number of input nodes, n_o is the number of outputs, and α is a constant varying from 1 to 10.

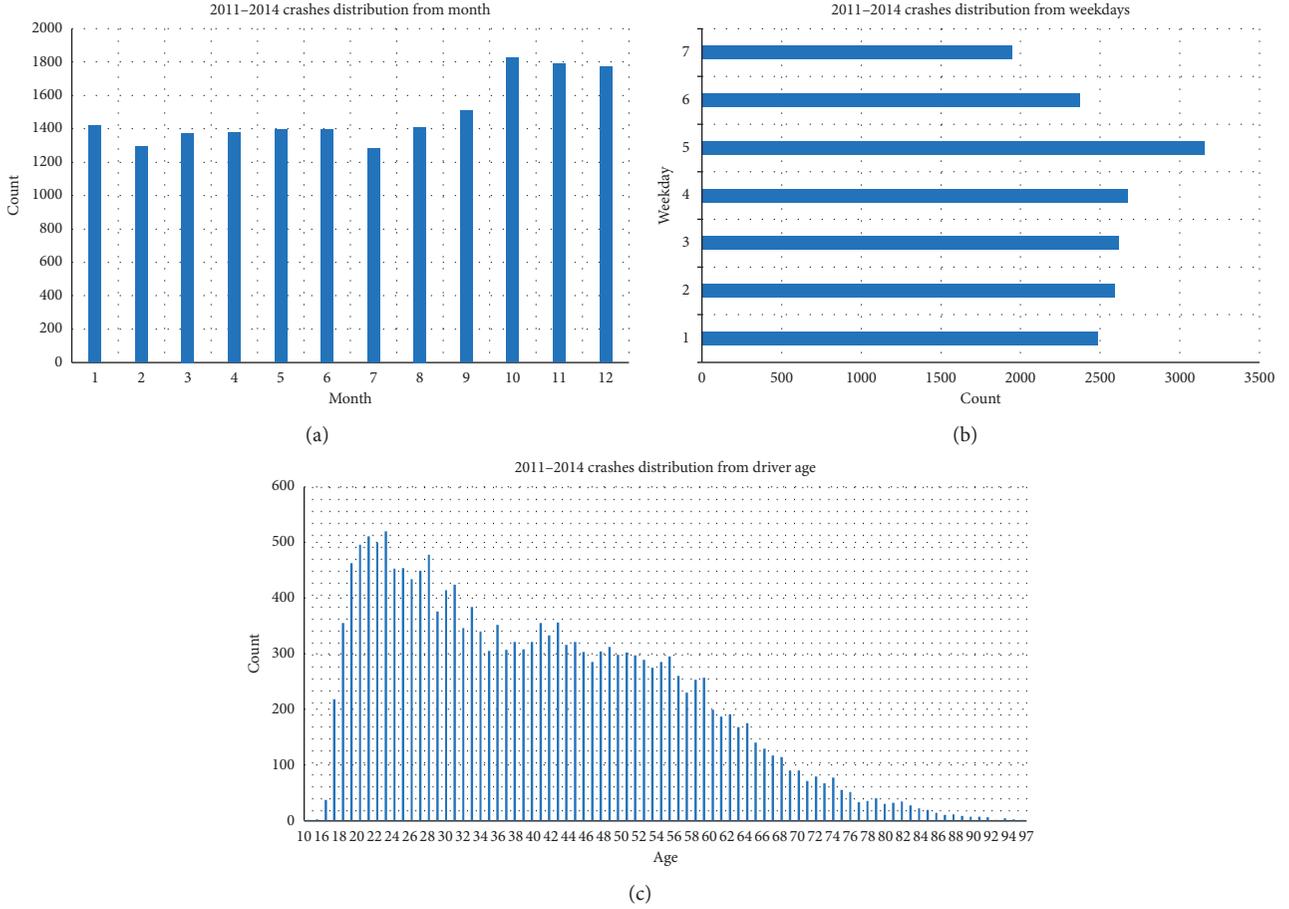


FIGURE 2: Crash distribution based on a (a) month, (b) weekday, and (c) driver age for 2011–2014.

TABLE 1: Summary of the selected variables.

Category	Variables	Value definition
Nonroad	(1) Accident type, (2) month, and (3) weekday (4) Location type, (5) light, and (6) Driver sex (7) Driver age, (8) driver restrain, and (9) vehicle year (10) Vehicle type and (11) weather	Categorical, categorical, and categorical Categorical, categorical, and categorical Continuous, categorical, and categorical Categorical and categorical
	(12) Road characteristics and (13) road surface (14) Road functional class and (15) curve angle (16) Curve direction and (17) gradient direction (18) Gradient percentage and (19) curve radius (20) Curve degree	Categorical and categorical Categorical and continuous Categorical and categorical Continuous and continuous Continuous

Based on the BP method [17], the local gradients $\delta_k^{(2)}(m)$ and $\delta_j^{(1)}(m)$ of the output and hidden layer neurons and the correction values $\Delta w_{k,j}^{(2)}(m)$ and $\Delta w_{j,i}^{(1)}(m)$ of their connection weights are as follows:

$$\begin{aligned}
 \delta_k^{(2)}(m) &= e_k(m) \cdot g_k'(v_k^2(m)), \\
 \Delta w_{k,j}^{(2)}(m) &= a(m) \cdot \Delta w_{k,j}^{(2)}(m-1) + \eta(m) \cdot \delta_k^{(2)}(m) \cdot \psi_k(m), \\
 \delta_j^{(1)}(m) &= g_j'(v_j^{(1)}(m)) \sum_K \delta_k^{(2)}(m) \Delta w_{k,j}^{(2)}(m),
 \end{aligned}
 \tag{11}$$

where $a(m)$ and $\eta(m)$ are the momentum and step size, respectively.

The weights in the network can be updated as

$$W_{j,i}^{(1)} = W_{j,i}^{(1)} + \Delta w_{j,i}^{(1)}(m), \tag{12}$$

$$W_{k,j}^{(2)} = W_{k,j}^{(2)} + \Delta w_{k,j}^{(2)}(m). \tag{13}$$

Other than the general BP method, there are some other modified training functions, including resilient backpropagation (RPROP), conjugate gradient backpropagation,

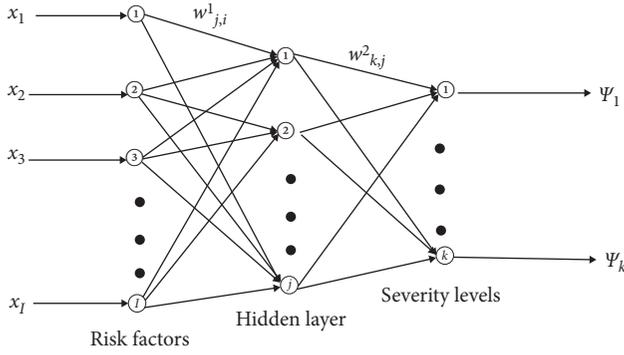


FIGURE 3: General structure of a typical NN used in this study.

gradient descent/momentum, and adaptive back-propagation. These functions have different levels of accuracy and training speeds, and thus, an attempt should be made to find a better solution.

In the traditional BP method, the initialized weights in formulas (6) and (7) are randomly initialized following two different uniform distributions. However, this implementation will highly possibly cause the whole model solution stopping at a local optimum. In order to address this problem, the authors offered a particle swarm optimization method with refined adaptive inertial weights to enhance both local and global searching for the optimal initial weights for BPNN. The details are provided in the following section.

3.2. PSO with Adaptive Inertial Weights. The particle swarm optimization method was a well-known metaheuristic computation method provided in 1995 [21]. Also, it was easy to use in different optimization applications [22]. The standard and original formulas for this method are

$$V_s^{r+1} = V_s^r + c_1 \lambda_1 (pbest_s^r - U_s^r) + c_2 \lambda_2 (gbest_s^r - U_s^r), \quad (14)$$

$$U_s^{r+1} = U_s^r + V_s^{r+1}, \quad (15)$$

where U_s^r is the s th particle at the r th generation and V_s^{r+1} denotes this particle's velocity to the $r+1$ th generation. c_1 and c_2 are two constants usually taken around the value of 2. λ_1 and λ_2 are the two uniform random numbers in the range of $[0, 1]$, $pbest$ is the best position experienced by the particle itself, and $gbest$ is the best position experienced by the particle swarm.

In this paper, the initial weights from BPNN are treated as the particles in the PSO algorithm, and the optimization problem can be described as mapping a decision space X to Y , and for a typical 3-layer neural network, they are encoded in the following set:

$$\begin{cases} \min_{x \in X} Y = f(x), \\ X = [W_1, B_1, W_2, B_2], \end{cases} \quad (16)$$

where W_1 and B_1 refer to the weights and bias connecting the input and hidden layer, while W_2 and B_2 refer to the

weights and bias connecting the hidden and output layer. Also, the transfer objective function Y is the neural network mean squared error (MSE).

For the standard or original PSO, it could solve nonlinear or nondifferentiable problems easily, but the searching space for a particular particle is almost fixed during each phase of generation, which means the model could then be easy and fast to find a solution, a possible solution near the local optimal. Thus, bringing the tradeoff between the local search ability and global search ability ahead [23], an inertial weight is introduced into formula (14), which could be written as follows:

$$V_s^{r+1} = w_s V_s^r + c_1 \lambda_1 (pbest_s^r - U_s^r) + c_2 \lambda_2 (gbest_s^r - U_s^r), \quad (17)$$

where w_s is the inertial weights controlling the global and local optimal searching speed, and it iterated during each generation in a linear or nonlinear form.

Thus, in this paper, different inertial weight setting methods [23–25] including the linear and nonlinear form are compared as follows:

$$w1(s) = \frac{w_{end} + ((w_{start} - w_{end})(S_{max} - s))}{S_{max}}, \quad (18)$$

$$w2(s) = w_{start} - (w_{start} - w_{end}) \left(\frac{s}{S_{max}} \right)^2, \quad (19)$$

$$w3(s) = w_{start} - (w_{start} - w_{end}) \left[\frac{2s}{S_{max}} - \left(\frac{s}{S_{max}} \right)^2 \right], \quad (20)$$

where w_{start} and w_{end} refer to the weights at the start and the end of the generation, and they are usually set to 0.9 and 0.4, respectively, s is the current iteration, and S_{max} is the max generation.

The function graph for formulas (18)–(20) is depicted in Figure 4. From Figure 4, can we see that the weight is of a relatively high value at first in order to expand the search space for global optimal; however, at the end of the iteration, it is converging slowly to enhance the local optimal search.

In conclusion, the pseudocode for the whole procedure in Sections 3.1 and 3.2 is formulated as follows:

```

FOR I=1: MAX GENERATION
  FOR J=1: POPULATION SIZE
    PARTICLE INITIALIZE;
    CALCULATE FITNESS (MSE of BP NN);
    UPDATE PARTICLE VELOCITY (WITH ADAPTIVE INERTIAL WEIGHTS);
    UPDATE PARTICLE POSITION;
  END
END
ASSIGN SOLUTION TO BP NN;
BP NN TRAINING, TESTING, VALIDATION;
NN PREDICTION;

```

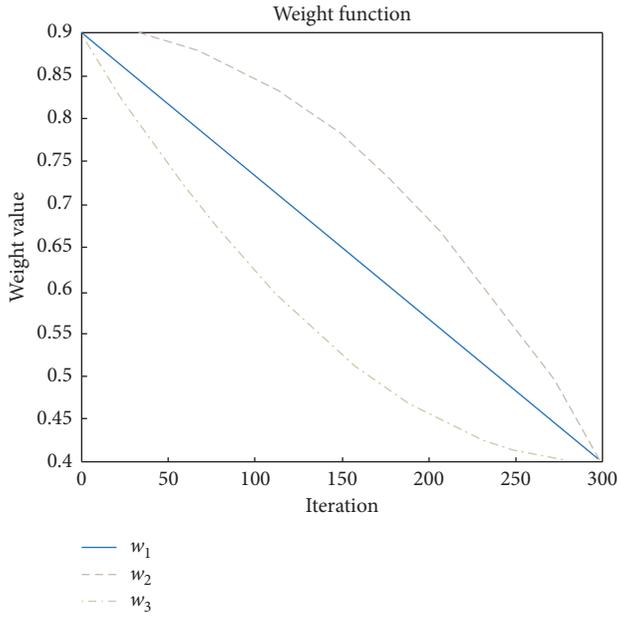


FIGURE 4: Weight function (18)–(20).

The calculation process is given in Figure 5.

3.3. Factor Analysis. To carry out a factor analysis (FA), the factor importance index (FII) is introduced in this paper. According to the nonlinear and classification function in practice, the n -dimensional input vector constructs the whole input space. Thus, from the perspective of engineering math, the first order partial derivative of the outcome Y with respect to the i th variable x_i could explain the connecting weight of that input variable vector, according to the chain rule in calculus, and the math description is as follows:

$$\frac{\partial Y}{\partial X_i} = \frac{\partial Y}{\partial \alpha} \frac{\partial \alpha}{\partial X_i}, \quad (21)$$

where α is the linear output value from the hidden layer to the output layer in BPNN, which could be described as formula (9), and thus, equation (21) is transformed to

$$\frac{\partial Y}{\partial X_i} = \sum_{j=1}^L W_{k,j}^{(2)} \frac{\partial H_j}{\partial X_i} g(\alpha)', \quad (22)$$

where j is the j th nodes in the hidden layer, $W_{k,j}^{(2)}$ refers to the weights through the hidden layer to the output layer, $g(\alpha)'$ denotes the derivative with respect to the activation function between the output and hidden layers.

Considering formula (8) combined with chain rule in calculus, formula (22) could be written as

$$\frac{\partial Y}{\partial X_i} = \sum_{j=1}^L W_{k,j}^{(2)} \frac{\partial H_j}{\partial \beta_j} \frac{\partial \beta_j}{\partial X_i} g(\alpha)', \quad (23)$$

$$\frac{\partial Y}{\partial X_i} = \sum_{j=1}^L W_{j,i}^{(1)} W_{k,j}^{(2)} f(\beta)' g(\alpha)', \quad (24)$$

where β is a short note for formula (8) and $W_{j,i}^{(1)}$ refers to the weights through the input layer to the hidden layer. $f(\beta)'$ denotes the derivative with respect to the activation function between the input layer and the hidden layer.

Through formula (24), we can see that while given a fixed input vector in the i th dimension, the value of $f(\beta)'$ and $g(\alpha)'$ is fixed with respect to all X_i ; thus, considering the remaining part, the FII for the i th variable can be written as

$$R_i = \frac{\sum_{j=1}^L W_{j,i}^{(1)} W_{k,j}^{(2)}}{\sum_i^N \sum_{j=1}^L W_{j,i}^{(1)} W_{k,j}^{(2)}}, \quad (25)$$

where the value of $W_{j,i}^{(1)}$ and $W_{k,j}^{(2)}$ can be calculated through formulas (12) and (13). These values are stored in a dictionary in a program code during the calculation process.

Finally, in order to ease the simulation variance of the model training process, the FII expectation is introduced by running the model for a certain k times:

$$E(R_i) = \frac{1}{K} \sum_{k=1}^K R_i^k. \quad (26)$$

4. Results and Discussion

4.1. NN Model Structure Test. Based on the theory discussed in the previous section, the most primary step in building an NN is to define the number of good hidden layer nodes and a better training function. Usually, the model performance (mean square error, MSE) combined with the total iteration number of convergence is used to test the structure. For the number of hidden layer nodes, based on formula (10), consecutive numbers of 5 through 14 were selected; for the training function, however, one of the following (Table 2) is applied.

The last two methods in Table 2 usually provide a fast calculation speed, but tend to be challenging and inefficient when dealing with the big data issues, especially, for the GPU hardware with low configuration. For the present study, considering the sample size, GPU support is not a problem, and thus, this minor difference is not a significant concern.

Theoretically, the number of nodes in the hidden layers should be within the range of (5, 14) based on formula (10); usually, however, the number of hidden layer nodes does not fall below 10, and thus, a combined test using the popular training functions and 10–14 hidden layer nodes number was carried out based on a loop test.

We randomly separated the sample data from 2011 to 2014 to form the dataset of training, testing, and validation with respect to 70%, 15%, and 15%. In detail, a total data of 17839 were divided into 12487, 2676, and 2676, in accordance with training, testing, and validation. The outcome is shown in Table 3.

It can be seen from Table 3 that 12 and 14 hidden layer nodes achieve the best performance (in other words, the lowest MSE) and the BR training function has the lowest MSE. However, when adopting these methods, the gradient value of the GDA, GDX (with an adaptive learning rate), and LM decreases quickly during the very early validation stage

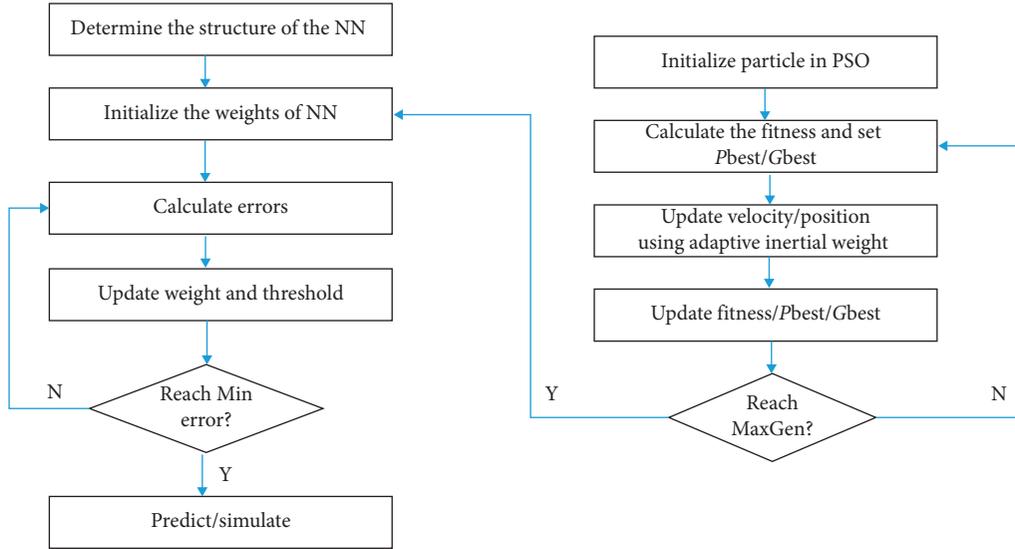


FIGURE 5: Calculation process for the method used in this study.

TABLE 2: Summary of training functions.

No.	Function name	Abbreviation
1	BFGS quasi-Newton backpropagation	BFGS
2	Conjugate gradient backpropagation with Powell-Beale restarts	CGB
3	Conjugate gradient backpropagation with Fletcher-Reeves updates	CGF
4	Conjugate gradient backpropagation with Polak-Ribiere updates	CGP
5	Gradient descent backpropagation	GD
6	Gradient descent with adaptive learning rate backpropagation	GDA
7	Gradient descent with momentum	GDM
8	Gradient descent w/momentum and adaptive learning rate backpropagation	GDX
9	One-step secant backpropagation	OSS
10	RPROP backpropagation	RP
11	Scaled conjugate gradient backpropagation	SCG
12	Levenberg-Marquardt backpropagation	LM
13	Bayesian regulation backpropagation	BR

TABLE 3: Test on the number of neural network hidden layer nodes (best validation performance in terms of MSE).

	10	11	12	13	14	Average
BFGS	0.260	0.252	0.250	0.250	0.208	0.242
CGB	0.202	0.206	0.254	0.244	0.246	0.230
CGF	0.208	0.240	0.204	0.242	0.232	0.224
CGP	0.220	0.222	0.202	0.240	0.202	0.218
GD	0.230	0.232	0.236	0.224	0.232	0.230
GDA	0.204	0.206	0.208	0.204	0.206	0.204
GDM	0.466	0.472	0.228	0.216	0.240	0.324
GDX	0.206	0.202	0.204	0.198	0.204	0.202
OSS	0.200	0.202	0.204	0.244	0.206	0.212
RP	0.206	0.202	0.198	0.206	0.210	0.204
SCG	0.204	0.206	0.198	0.206	0.204	0.204
LM	0.204	0.204	0.198	0.208	0.202	0.202
BR	0.199	0.199	0.199	0.198	0.198	0.198
Average	0.230	0.234	0.214	0.220	0.214	

and, thus, quickly converges to the preset goal, and methods such as BR, GDM, and GD converge slowly, requiring more validation than usual. Thus, in conclusion, the choice of

GDA, GDM, or LM should be better, and considering a lack of GPU support for the simulation environment, we selected the Levenberg-Marquardt backpropagation (LM) as the training function. During the number test, LM with 12 nodes of the hidden layer showed the best performance.

4.2. Results for PSO Optimization. After setting the adaptive inertial weights for the PSO optimizer, we can conclude the following performance graph through each iteration.

To eliminate the random data separation variance, 100-time simulation was conducted, and the average performance for each method is described in Table 4.

From Figure 6 and Table 4, it could be seen that the performance refers to the MSE of the training model, training accuracy refers to the average classification accuracy among the training set (17839), and the prediction accuracy refers to the classification accuracy among the 4984 samples from 2015. Although the fitness (MSE) of PSO with w_1 drops fast in the early stage (almost starts with 22), it converges poorly and has the poor condition of performance and training/testing accuracy, which means it lacks of certain

TABLE 4: Summary of the performance relating to different models.

Category	Performance	Training accuracy (percentage)	Prediction accuracy (percentage)
NN	0.235	78.5	71.6
NN with std. PSO	0.232	78.6	72.4
NN with PSO (w_1)	0.298	73.2	70.5
NN with PSO (w_2)	0.194	80.4	73.1
NN with PSO (w_3)	0.196	79.3	73.6

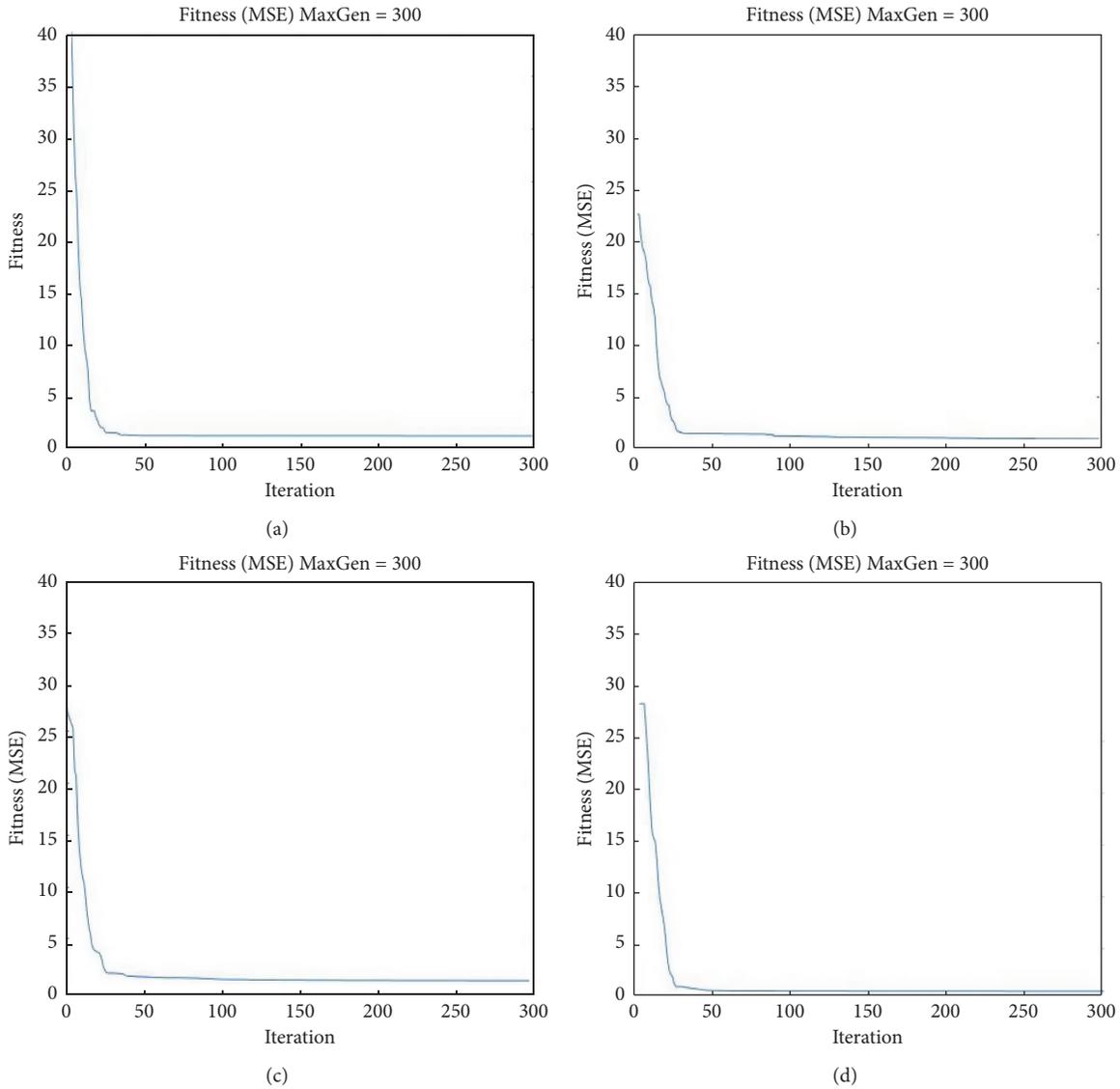


FIGURE 6: NN performance (MSE) adopting different PSO optimizations. (a) PSO without inertial weight; (b) PSO with w_1 ; (c) PSO with w_2 ; and (d) PSO with w_3 .

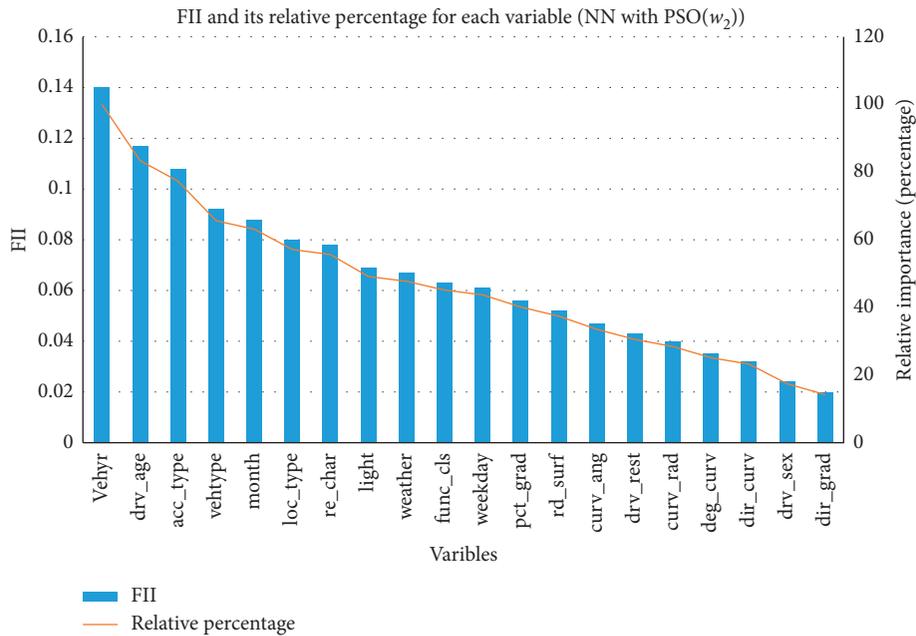
ability for local optimal searching. On the other hand, PSO with w_2 and w_3 (two nonlinear formulas) outperforms the other methods not only in global optimal search (both from 28, while NN with standard PSO drops from nearly 40) but also in the local optimal search (with good performance and accuracy), and the NN and NN with standard PSO are in the middle level with an acceptable result. It can be assumed that

with more data flushing in, the PSO with nonlinear will perform a dominating advantage over the other methods.

As shown in Table 4, the accuracy of training is better than the results of the prediction; the relatively lower sample number may be a major contributor. Notably, the performance from all categories is at the same rate related to the sample size, and it can be concluded that although the

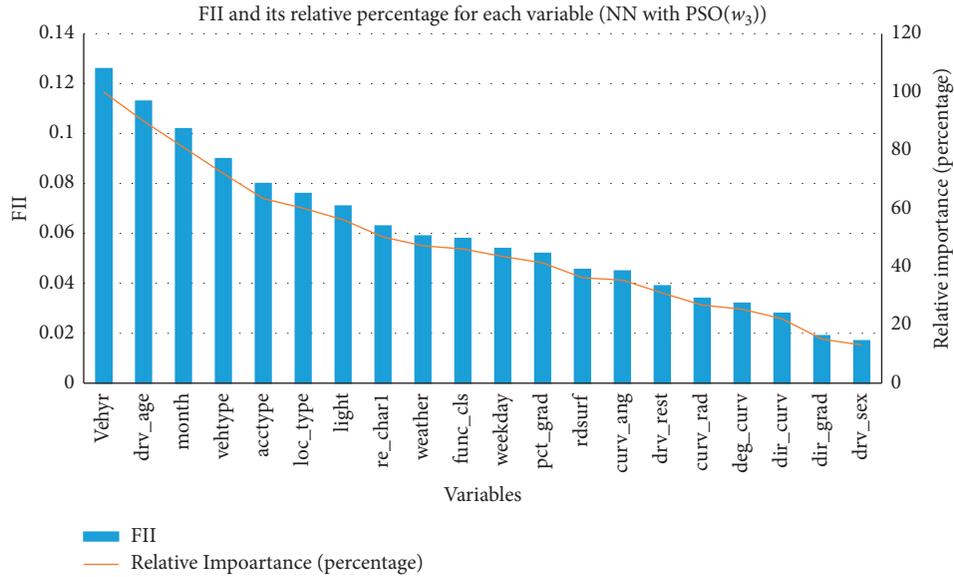
TABLE 5: Summary of FII expectation ($K = 100$) and relative percentage for two different methods.

Variables	Category	NN with PSO (w_2)		NN with PSO (w_3)	
		FII	Relative percentage (%)	FII	Relative percentage (%)
Acctype	Nonroad	0.108	77.3	0.080	63.4
Month	Nonroad	0.088	63.1	0.102	81.0
Weekday	Nonroad	0.061	43.7	0.054	43.5
loc_type	Nonroad	0.080	57.1	0.076	60.1
rd_char1	Road	0.078	55.6	0.063	50.2
Rdsurf	Road	0.052	37.4	0.456	36.2
Light	Nonroad	0.069	49.1	0.071	56.1
Weather	Nonroad	0.067	47.7	0.059	47.2
func_cls	Road	0.063	45.1	0.058	46.1
drv_sex	Nonroad	0.024	17.4	0.017	13.1
drv_rest	Nonroad	0.043	30.5	0.039	31.0
Vehetype	Nonroad	0.092	65.6	0.090	72.0
dir_curv	Road	0.032	23.2	0.028	22.1
dir_grad	Road	0.020	14.2	0.019	15.1
drv_age	Nonroad	0.117	83.4	0.113	90.0
Vehyr	Nonroad	0.140	100.0	0.126	100
curv_ang	Road	0.047	33.5	0.045	35.4
pct_grad	Road	0.056	40.1	0.052	41.3
deg_curv	Road	0.035	25.1	0.032	25.4
curv_rad	Road	0.040	28.4	0.034	26.8



(a)

FIGURE 7: Continued.



(b)

FIGURE 7: Summary of results for factor analysis. (a) FII and its relative percentage for each variable (NN with PSO (w_2)); (b) FII and its relative percentage for each variable (NN with PSO (w_3)).

prediction accuracy applied to other samples could result in a marginally worse accuracy, the model can explain all variance and contributors.

4.3. Results for Factor Analysis. The final results for two PSO with nonlinear adaptive inertial weight are described in Table 5.

From Figure 7 and Table 5 we can see that two methods (PSO with w_2 and PSO with w_3) provide almost the same ranking with respect to each variable, despite some minor difference, for example, MONTH and LIGHT are ranked slightly higher in the second method, while DRIVER SEX are ranked slightly lower. Nearly seven or eight of the 20 variables have a relative importance value exceeding 50%, including the vehicle year, driver age, accident type, month, location type, and road functional class (also LIGHT in the second method), with vehicle year and driver age taking the top two values. Additionally, the related road factors such as curve and gradient variables have the least importance, and the only important factor contributing to the severity prediction is the road functional class, which indicates whether a crash occurred on a certain type of road. Thus, it can be concluded that the relative road variables contribute less than the relative nonroad variables, particularly, compared with the relative vehicle factors (vehicle year and type). Therefore, the policy makers should pay more attention to the vehicle and driver regulation rules, as well as the road design, to reduce the possible severity level in the future.

Driver age and month are two other important factors in predicting a crash severity. From the sample size, we can see that the most severe crashes occur during the winter in the Washington State (December, January, and February), and drivers below the age of 25 and above the age of 60 are more prone to encountering severe injury crashes. The month may

account for the rainy season in the mountainous Seattle area, whereas age may be derived from the fact that younger people and older people are more prone to making severe mistakes.

5. Conclusions

In this study, a thorough artificial neural network (ANN) was developed to address the problems of the crash severity level modeling and factor analysis (FA). Besides the test of different types of training structure and methods, more importantly, a nonlinear adaptive PSO optimization method was proposed in order to solve the tradeoff problem between the global and local search ability among the previous studies. The detail test of different algorithm confirmed our hypothesis. The additional contributing factor analysis also offers a different point of view compared with former statistical analysis. The main conclusions can be concluded as follows:

- (1) The number 12 hidden layer nodes fit the model developed in this paper well; and the BP method (Levenberg–Marquardt) can be better utilized when aided by fast hardware
- (2) The simulation result showed that the PSO optimizer with nonlinear adaptive inertial weight outperforms the standard PSO and PSO with linear adaptive inertial weight
- (3) Through the factor analysis (FA), it can be found that, among all 20 variables, nonroad-related variables can account for most of the severity prediction variance, and the rainy mountainous area in Seattle may be the reason for the importance of the month as a factor and, also, the impact of driver age, where

younger and older people are more prone to encountering a severe crash

The main innovations can be concluded as follows:

- (1) Traditional studies often used statistical methods like the Poisson regression, negative binary regression, and generalized logit or probit model for the identification and mathematical qualification of the inner internal triggers and their impact on crash severity, while this paper utilized FA as the analytical tool, which is unusual for the current research system of crash severity, and we think that our attempt extended the methods of crash severity analyses, and more research could be conducted in the future work.
- (2) FA, as a traditional statistical implement, also can serve as a powerful explanatory tool in the last stage of the model, and our work has proved it. The application of FA in this paper indicated that the basic statistical method is still useful and efficient while the AI methods sometimes did not have an agreeable explanation for the inner mechanism of the data.

The method developed in this study can be applied to a big data analysis of traffic accidents and be used as a fast-useful tool for policy makers and traffic safety researchers. The authors recognize that much can be further investigated. In this paper, only crash severity was discussed. Further research could be conducted from the perspective of the collision type (e.g., head-on collisions and rear-end collisions). Besides, the dataset could be enlarged in the future research to improve the accuracy.

Data Availability

The dataset used in this study was made up of data requested from the Highway Safety Information System (HSIS), and requests for access to these data should be made by filling the form at the following link: <https://www.hsisinfo.org/datarequest.cfm>.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The authors would like to thank the Highway Safety Information System, U.S.A, Smart Transportation Application and Research Laboratory (STAR Lab) at the University of Washington, U.S.A, and Pacific Northwest Transportation Consortium Region 10, U.S.A. They also thank National Natural Science Foundation of China (Grant nos. 51778141 and 71871078), China Scholarship Council, and Jiangsu Creative PhD Student Sponsored Project (KYLX15_0157) for providing essential data and support.

References

- [1] Washington State Department of Transportation, *Fatality Analysis Reporting System (FARS) Auxiliary Datasets Analytical User's Manual 2007–2016, Annual Collision Summary*, Washington State Department of Transportation, Olympia, WA, USA, 2015.
- [2] D. Lord and F. Mannering, "The statistical analysis of crash-frequency data: a review and assessment of methodological alternatives," *Transportation Research Part A: Policy and Practice*, vol. 44, no. 5, pp. 291–305, 2010.
- [3] X. Ma, S. Chen, and F. Chen, "Multivariate space-time modeling of crash frequencies by injury severity levels," *Analytic Methods in Accident Research*, vol. 15, pp. 29–40, 2017.
- [4] R. O. Mujalli, G. López, and L. Garach, "Bayes classifiers for imbalanced traffic accidents datasets," *Accident Analysis & Prevention*, vol. 88, pp. 37–51, 2016.
- [5] Y. Li, D. Ma, M. Zhu, Z. Zeng, and Y. Wang, "Identification of significant factors in fatal-injury highway crashes using genetic algorithm and neural network," *Accident Analysis and Prevention*, vol. 111, pp. 354–363, 2018.
- [6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [7] AASHTO, *Highway Safety Manual*, American Association of State Highway Transportation Officials, Washington, DC, USA, 2010.
- [8] R. Elvik, P. Christensen, and A. Amundsen, "Speed and road accidents: an evaluation of the power model," *TOI Report*, vol. 740, 2004.
- [9] F. Russo, M. Busiello, and G. Dell'Acqua, "Safety performance functions for crash severity on undivided rural roads," *Accident Analysis & Prevention*, vol. 93, pp. 75–91, 2016.
- [10] J. Park and M. Abdel-Aty, "Safety performance of combinations of traffic and roadway cross-sectional design elements at straight and curved segments," *Journal of Transportation Engineering, Part A: Systems*, vol. 143, no. 6, Article ID 4017015, 2017.
- [11] M. Ahmed, H. Huang, M. Abdel-Aty, and B. Guevara, "Exploring a Bayesian hierarchical approach for developing safety performance functions for a mountainous freeway," *Accident Analysis & Prevention*, vol. 43, no. 4, pp. 1581–1589, 2011.
- [12] B. Debrabant, U. Halekoh, W. H. Bonat, D. L. Hansen, J. Hjelmberg, and J. Lauritsen, "Identifying traffic accident black spots with Poisson-Tweedie models," *Accident Analysis & Prevention*, vol. 111, pp. 147–154, 2018.
- [13] X. Pei, S. C. Wong, and N. N. Sze, "A joint-probability approach to crash prediction models," *Accident Analysis & Prevention*, vol. 43, no. 3, pp. 1160–1166, 2011.
- [14] K. El-Basyouny, S. Barua, and M. T. Islam, "Investigation of time and weather effects on crash types using full bayesian multivariate poisson lognormal models," *Accident Analysis and Prevention*, vol. 73, pp. 91–99, 2014.
- [15] M. G. Karlaftis and E. I. Vlahogianni, "Statistical methods versus neural networks in transportation research: differences, similarities and some insights," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 3, pp. 387–399, 2011.
- [16] L.-Y. Chang, "Analysis of freeway accident frequencies: negative binomial regression versus artificial neural network," *Safety Science*, vol. 43, no. 8, pp. 541–557, 2005.
- [17] Q. Zeng and H. Huang, "A stable and optimized neural network model for crash injury severity prediction," *Accident Analysis & Prevention*, vol. 73, pp. 351–358, 2014.

- [18] H. Huang, Q. Zeng, X. Pei, S. C. Wong, and P. Xu, "Predicting crash frequency using an optimised radial basis function neural network model," *Transportmetrica A: Transport Science*, vol. 12, no. 4, pp. 330–345, 2016.
- [19] S. S. Haykin, *Neural Networks and Learning Machines*, Prentice Hall, Piscataway, NJ, USA, 3rd edition, 2009.
- [20] J. S. Judd, *Neural Network Design & the Complexity of Learning*, The MIT Press, Cambridge, MA, USA, 2000.
- [21] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks 1942–1948*, Piscataway, NJ, USA, 1995.
- [22] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.
- [23] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Computational Intelligence*, pp. 69–73, Ankorage, AK, USA, June 1998.
- [24] L. Zhang, H. Yu, and S. Hu, "A new approach to improve particle swarm optimization," *Genetic and Evolutionary Computation—GECCO 2003*, vol. 2327, pp. 134–139, 2003.
- [25] S. Liang and W. Dong, "PSO algorithm with dynamic inertial weight vector and dimension mutation," *Computer Engineering & Application*, vol. 47, no. 5, pp. 29–32, 2011.