

Research Article

A Bayesian Neural Network-Based Method to Calibrate Microscopic Traffic Simulators

Qinqin Chen ¹, Anning Ni ¹, Chunqin Zhang,² Jinghui Wang,¹ Guangnian Xiao,³ and Cenxin Yu ¹

¹Department of Transportation Engineering, School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

²School of Civil Engineering and Architecture, Zhejiang Sci-Tech University, Hangzhou 310018, Zhejiang, China

³School of Economics & Management, Shanghai Maritime University, Shanghai 201306, China

Correspondence should be addressed to Anning Ni; nianning@sjtu.edu.cn

Received 15 April 2021; Revised 31 August 2021; Accepted 23 October 2021; Published 26 November 2021

Academic Editor: Chi-Hua Chen

Copyright © 2021 Qinqin Chen 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.

Calibrating the microsimulation model is essential to enhance its ability to capture reality. The paper proposes a Bayesian neural network (BNN)-based method to calibrate parameters of microscopic traffic simulators, which reduces repeated running of simulations in the calibration and thus significantly improves the calibration efficiency. We use BNN with probability distributions on the weights to quickly predict the simulation results according to the inputs of the parameters to be calibrated. Based on the BNN model with the best performance, heuristic algorithms (HAs) are performed to seek the optimal values of the parameters to be calibrated with the minimum difference between the predicted results of BNN and the field-measured values. Three HAs are considered, including genetic algorithm (GA), evolutionary strategy (ES), and bat algorithm (BA). A TransModeler case of highway links in Shanghai, China, indicates the validity of the proposed calibration method in terms of error and efficiency. The results demonstrate that the BNN model is able to accurately predict the simulation and adequately capture the uncertainty of the simulation. We also find that the BNN-BA model produces the best calibration efficiency, while the BNN-ES model offers the best performance in calibration accuracy.

1. Introduction

In the process of efficiently assessing the influence of traffic management schemes and even emerging technologies on traffic performance, simulations are powerful tools to simulate the proposed scenes without the need for field experiments [1]. Microscopic traffic simulator is one of the tools, and it outperforms mesoscopic and macroscopic simulators in the simulation of specific driving behavior and the visual display of traffic scenes [2]. However, considering that the traffic conditions vary from place to place, the default values of parameters from simulator developers are inconsistent with the specific characteristics of the places of simulation. Using the default parameters can cause the simulation results to be out of the range of the places of simulation and cause large deviations in the results. In other

words, because of the location-dependent parameter values, it is essential to determine a set of parameter values before effectively using microscopic traffic simulators. Thus, it is important to calibrate the parameter values, which enables the simulator to be used for traffic analysis under the preferred background.

The calibration needs to search for the set of parameter values, which ensures the minimum difference between the field-measured values and simulation output values. The calibration is still challenging in microscopic traffic simulation containing high uncertainty of the simulation system and a large number of parameters [3]. In addition, the traditional parameter calibration process requires several runs of simulations. Considering that simulation is based on a certain scale of the road network and complex calculations, an inestimable and long period of time is spent on one

simulation. The number of simulations is affected by calibration convergence. Therefore, finding specific parameter values during the calibration process will be time-consuming and computationally expensive, and it is necessary to improve calibration efficiency.

The research on the parameter calibration method can be divided into two categories according to the research methods, which are nonheuristic methods and heuristic optimization methods. Firstly, previous research mainly focused on how to calibrate based on a variety of nonheuristic methods, such as comparison [4], orthogonal experiments [5], indirect calibration method [6], and Bayesian framework-based technique [7]. The calibration problem inherently seeks and optimizes specific parameter values with the minimum difference between field-measured values and simulation output values. It is efficient to treat the calibration as a simulation-based nonlinear optimization problem [8]. Numerous studies have tried to calibrate the parameters from this aspect. A metamodel-based simulation-optimization framework is newly proposed for calibration [9]. With the help of several heuristics, solving the optimization problem becomes easier. The use of HA in calibration has also attracted much attention from researchers. In HA, GA occupies an important position and cannot be ignored. GA has been widely used in the calibration of microscopic traffic simulators (i.e., FRESIM [10], VISSIM [11–13], AIMSUN [14, 15], and PARAMICS [16, 17]) and traffic microsimulation models [18]. For heterogeneous traffic, GA was suitable for obtaining optimized parameter sets [19]. When exploring the impact of different measures of performance on the calibration results, some scholars still preferred GA [20]. The above research shows that the GA-based method performs well in calibration. Similar to GA, ES is based on the same principle of biological evolution, which has also been used in traffic calibration problems. ES was applied to search for two coefficients in the automatic calibration of traffic surveillance cameras [21]. Some scholars used the covariance matrix adaptation-evolution strategy (CMA-ES) to calibrate dynamic traffic assignment models. The parallel feature of CMA-ES made it suitable for the new large-scale field of applications, not limited to real-time deployment [22]. Additionally, simultaneous perturbation stochastic approximation (SPSA) was also implemented in the calibration framework, which turned out to be fast under numerous inputs [23]. In order to calibrate the discrete demand of multimodal microscopic traffic simulation, the weighted SPSA was investigated [24]. In recent years, BA, as an emerging heuristic algorithm, has mainly been used for continuous optimization [25], combinatorial optimization and scheduling [26], engineering problems [27, 28], and parameter estimation [29]. The above literature proves that BA has significant potential to outperform other algorithms in efficiency and robustness. Since there are various types of HA, it is necessary to compare them. For the calibration of microsimulation models, several metaheuristic algorithms were compared with the manual method. The comparison has shown that HA is superior in saving time and getting better-modeled results [30, 31]. Although BA surpasses GA

and particle swarm optimization algorithms in terms of efficiency and accuracy [32], few scholars apply it to the parameter calibration of microscopic traffic simulators or include it in comparative research.

The difficulty of the existing ways is that using heuristics in optimization and calibration needs repeated running of the simulator to obtain outputs, when a different combination of parameter values is being tested. Therefore, calibration still inevitably requires long calculation and simulation time. For the calibration of the microsimulation model, the running time of HA was about 100 hours [30]. Moreover, the GA-based methods took 8.4 hours and about 10 hours to calibrate in two different simulation cases [31, 33]. Although some scholars attempt to save time by using parallel computing technology [34], the research on improving calibration speed is still limited. Reducing the number of simulations in calibration is a possible prerequisite for improving calibration efficiency. Considering that the neural network has been used for prediction in the field of transportation [35], the proposed method substitutes running time-consuming simulations with running the neural network model to gain the predicted simulation outputs in the calibration and thereby greatly improves the efficiency of calibration. It is different from the existing ways. Furthermore, the simulator is a complex nonlinear system. The neural network can fit continuous functions of arbitrary complexity with arbitrary accuracy [36], which can perfectly replace the simulator in a data-driven way. Some studies have made some progress in this area, which also demonstrated the feasibility and benefits of the calibration framework combining neural networks and HA. The applicability of neural networks in calibrating traffic simulators has been studied [37]. An artificial neural network (ANN) set mean target headway and mean reaction time as inputs and outputted the queue length to replace PARAMICS simulation processes. According to the relationship between inputs and outputs given by ANN, GA obtained the calibrated parameter values without simulation. And the consistency between the simulation outputs after calibration and the outputs of ANN verified the validity of the method [38]. Based on VISSIM, a parameter calibration method combining ANN and GA was also proposed. For each parameter set, ANN outputted predictive vehicle speed without simulations. GA took about 1 minute to find parameter values using the trained ANN, but the traditional GA-based method took more than 10 hours. The calibration results reduced the average absolute relative error of the speed in two tested highway sections [33]. Similarly, some scholars have also conducted comparative studies on parameter calibration methods that combine a variety of machine learning models and particle swarm optimization algorithms [39].

The above studies are based on deterministic machine learning models. Nevertheless, probabilistic programming outperforms them in portraying the uncertainty of simulation and driving behavior. BNN has weights that obey the probability distribution to obtain rich prediction results with uncertainty. There are relatively limited studies on the use of BNN in calibration. Only some studies have calibrated the

car-following model in simulation logic based on related ideas. According to the Markov chain Monte Carlo simulation and the Bayesian estimation theory, a stochastic calibration method could estimate the parameter distribution of two car-following models. It was better than the deterministic optimization algorithm from the evaluation of the cost function [40]. Based on real data, probabilistic programming and Bayesian machine learning were applied to the calibration of car-following models. The method showed informative validity and true-to-data uncertainty [41].

The objective of this paper is to develop an efficient calibration method for microscopic traffic simulators. Although the combination of ANN and GA is efficient, the combination of BNN and other HAs (i.e., ES and BA) is expected to continue to improve efficiency and accuracy. In this paper, the combination of BNN and HA is investigated. The purpose of BNN is to achieve that there is no need to run simulations during the calibration process after training the BNN. Before training the BNN, the parameters that have a significant impact on the simulation results are selected as the parameters to be calibrated. Afterward, the data set of simulation results is derived from running enough simulations based on random combinations of values for selected parameters, which is used to train, verify, and test the BNN. Then, the trained BNN can predict simulation outputs by inputting the selected parameter values. Additionally, the trained BNN with the best results becomes the background of optimization problems. Several HAs, including GA, ES, and BA, are investigated to seek optimal parameter values with the minimum difference between the predicted output of BNN and the field-measured data. Specifically, the proposed parameter calibration method includes the combination of BNN and GA (BNN-GA), the combination of BNN and ES (BNN-ES), and the combination of BNN and BA (BNN-BA). They are applied and verified through a real case in Shanghai, China. Because the proposed parameter calibration method is independent of the simulation platform and the scale of the studied road network, it is universal for microscopic traffic simulators and is also expected to be used in a variety of traffic situations.

The innovations of this paper are summarized as follows:

- (i) The calibration efficiency of the microscopic traffic simulator is further improved under reliable calibration accuracy by using the proposed method. Calibration is challenging and time-consuming due to a large number of parameters and complicated uncertainties. Compared with the traditional method, the proposed calibration method does not need to run simulation during calibration, which greatly saves time. Moreover, derived from the similar efficient calibration framework using neural networks and HA [33, 37–39], the method combining BNN and HA also performs better in efficiency and error.
- (ii) BNN is used in the proposed calibration method. Superior to widely used deterministic machine learning models, BNN is rarely used to predict simulation results, but it can capture uncertainty during calibration.
- (iii) GA is commonly used for calibration, but efficient ES and BA are rarely used. The applicability of ES and BA in calibration is studied in terms of efficiency and accuracy.

The remainder of this paper is organized as follows. Section 2 demonstrates the components of the microscopic traffic simulator parameter calibration method, including the identification of parameters, brief introduction of BNN, and application of HA. Section 3 presents the application and effectiveness of the calibration method combining BNN and HA through a real case study in the TransModeler simulator. Section 4 discusses the results of the case study. Section 5 is the conclusion.

2. Calibration Method of Microscopic Traffic Simulator Parameters

The proposed parameter calibration method consists of three parts: identifying needed parameters to be calibrated, training the BNN model, and applying HA to find the optimal calibrated parameter values. The specific process of the method is shown in Figure 1.

2.1. Identifying Parameters for Calibration. The first step of calibrating parameter values in a traffic simulator is identifying needed parameters before demonstrating the calibration methodology, so a two-step procedure to identify parameters is proposed. Sensitivity analysis is used to verify whether the selected parameters significantly affect the simulation results, and a range test is conducted to know whether the field measurements are in the numerical range of simulator outputs.

2.1.1. Sensitivity Analysis. Generally speaking, based on experience, the initial selection of the parameters to be calibrated can be completed. Then, it is necessary to verify whether the selected parameters significantly affect the simulation results. This process is called sensitivity analysis. The maximum and minimum values are taken for a selected parameter, and the other parameters are kept at their default settings (including the random seed) to obtain the simulation results. Next, the default values of all parameters are kept, and the random seed is randomly used to obtain its effect.

Another role of sensitivity analysis is to explore whether the effect caused by the changes of parameters to be calibrated is significant compared with the effect caused by the random seed in the simulation outputs. If the effect caused by the former is significant, it is essential to do calibration to find the optimum parameter values.

When it comes to specific processes, an example road network is built in the microscopic traffic simulator. First, for a parameter, or called the i th parameter, two simulations are conducted, where the i th parameter uses the maximum and minimum values, respectively, according to the recommendation document, and other parameters take the simulator's

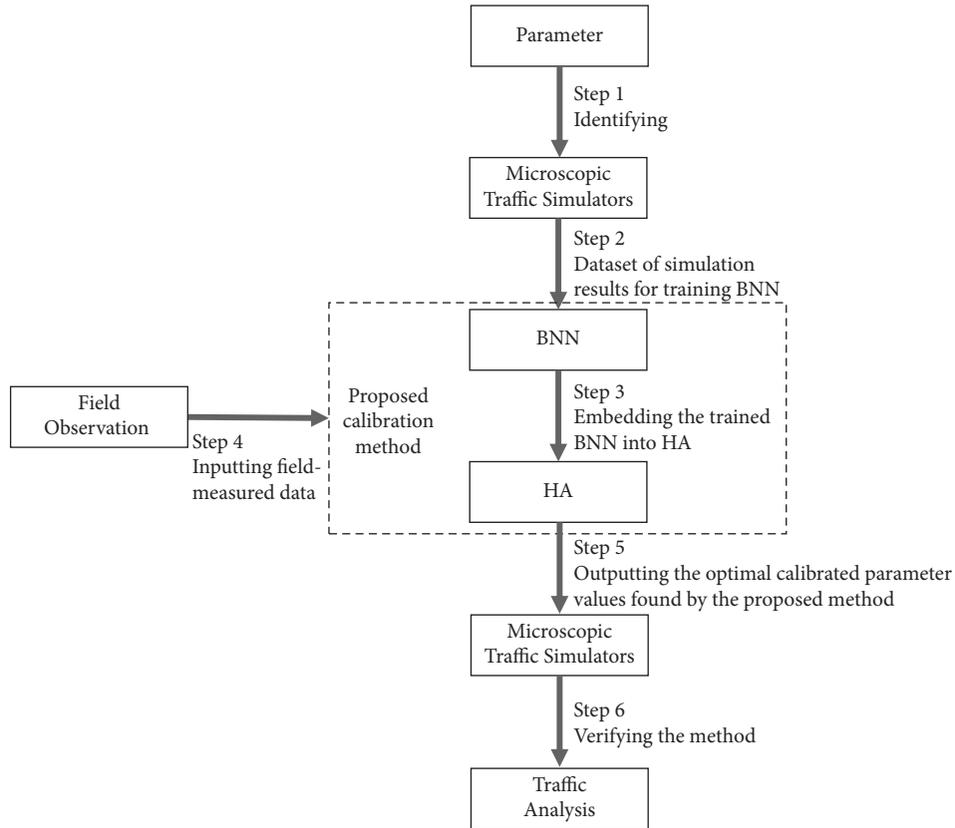


FIGURE 1: Process of the parameter calibration method.

default settings. The difference between the results of the two simulations is denoted as ΔR_i . Afterward, all parameters are set as default. Multiple simulations with randomly generated seeds are performed to obtain the effect caused by the random seed. And ΔR denotes the difference between the minimum and the maximum simulation results. If $\Delta R_i > \Delta R$, it can be concluded that the influence of the i th parameter on results is greater than the influence of the random seeds, so the i th parameter needs calibration.

2.1.2. Range Test. The purpose of the range test is to ensure that the range of the simulation results covers the field-measured data. According to the recommendation document, multiple sets of parameter values within the recommended range are randomly generated. For a set of parameter values, they are inputted into the traffic simulator and the simulation is run several times to calculate the average of the simulation results to reduce randomness. And the same process is repeated for all sets to acquire the range of the simulation results. Eventually, we examine whether the field-measured data are within the range of the simulation results. If not, extra revision to the parameters' range would be required [42].

2.2. Bayesian Neural Network. BNN is a kind of neural network. Different from other neural networks, the weight in BNN replaces a certain value with a probability distribution, as shown in Figure 2. Uncertainty in weights caused by probability distributions can bring benefits of regularization

and richer prediction results with uncertainty. There are two kinds of uncertainty in BNN: epistemic uncertainty and aleatoric uncertainty [43]. Epistemic uncertainty derived from uncertainty in weights can bring prediction results with uncertainty. It can be reduced by enough training data. And aleatoric uncertainty is derived from the inherent noise in the data. The following section only demonstrates the main structure of BNN. For more technical details about BNN, the readers can refer to Reference [44].

A neural network can be considered as a probabilistic model $P(y|x, w)$ (x is the input vector, y is the possible output vector, and w is denoted as the weight). For a training data set $D = (x, y)$, $P(w|D) \propto P(D|w)P(w)$ can be derived based on the Bayesian theory. In BNN, regularization and avoidance of overfitting are achieved by setting prior distributions on w . And maximizing $P(w|D)$ gives the maximum a posteriori (MAP) weights w^{MAP} :

$$\begin{aligned} w^{MAP} &= \arg \max_w \log P(w|D), \\ &= \arg \max_w \log P(D|w) + \log P(w). \end{aligned} \quad (1)$$

Because Bayesian posterior weight distribution $P(w|D)$ is difficult to calculate in a neural network, using a variational distribution to approximate $P(w|D)$ has become the solution. The variational inference tries to minimize the Kullback–Leibler (KL) divergence between a variational distribution on the weights $q(w|\theta)$ and the true $P(w|D)$ by finding the best parameter θ' :

the parameter values to be calibrated are inputted into the traffic simulator. And we simulate several times to obtain an averaged output. A set of parameter values and the corresponding average output forms a piece of training data. For different parameter value sets, the same process is repeated. Finally, numerous training data are acquired for training the BNN, validating the BNN, and testing the BNN.

2.3. Heuristic Algorithms for Calibration. The calibration problem inherently seeks and optimizes specific parameter values with the minimum difference between field-measured values and simulation output values. By using HA, we search for the values of the parameters in the microscopic traffic simulator to minimize the difference between field-measured values and the predicted simulation results from BNN. In the process of the HA, each individual in HA represents a set of parameter values to be calibrated. In multiple iterations, each individual constantly changes according to the rules of the algorithm. BNN is integrated into HA to output the simulation results of each individual. The mean squared error (MSE), also called the fitness value in the HA, is selected as an indicator to measure the difference between field-measured values and the BNN outputs. By minimizing the MSE, HA can find the set of parameter values (the optimal individual) to be calibrated, which is closest to the field situation.

After performing three HAs (GA, ES, and BA), the calibrated parameter values can be further used. Based on the calibrated parameters of microscopic traffic simulation simulators, the consistency between simulation and reality can be verified, and subsequent traffic analysis can be performed.

Before introducing the HA, the common notations are shown in Table 1.

2.3.1. Genetic Algorithm. The main point of GA is that each set of random parameter value combinations (initial population) has its own unique chromosome coding method. Before each iteration, some individuals in the population are randomly eliminated according to certain rules, and the performance of most of the eliminated individuals is poor (i.e., the fitness value is high in the case), and then the remaining individuals continue to reproduce. Under the joint action of elimination, reproduction, and mutation, after many iterations, the whole population will eventually gather at the optimal solution, which means they all have similar chromosome coding methods. The inspiration of GA comes from the law of survival of the fittest in the animal population. The closer the gene is to the optimal solution, the more likely the individual will survive. After several generations, the whole population will continue to adapt to the environment. Therefore, the algorithm has developed certain evolutionary advantages.

In GA, the input parameter set corresponding to each individual in the population corresponds to a chromosome-like code through some coding method with a fitness value. At the start of GA, each individual in the population will get a randomly generated chromosome code. In each iteration,

some chromosomes should be selected randomly according to some rules, and others should be eliminated. The smaller the fitness value of each individual (this study aims to find the minimum fitness value), the greater the probability of being selected. The new individuals in the population are supplemented by the following ways: (1) pairing the selected chromosomes to generate new individuals between the two, and (2) random variation of some existing individuals, which changes a random part of the chromosome code into a new individual. The process repeats until the maximum iterations. Finally, the optimal chromosome code searched by the population can be restored according to the coding method to get the value of each parameter to be calibrated.

The following describes the detailed steps of GA:

Step 0. Initialize: Let $k = 1$. Assign random values to P'_{1i} within the value range ($i = 1, \dots, \beta'_1$). Set $F' = \infty$ and $G' = 0$.

Step 1. Introduce the trained BNN model: For $i = 1, \dots, \beta'_k$, input P'_{ki} into the BNN model to obtain predicted outputs X_{ki} , respectively.

Step 2. Update the best fitness value and the optimal parameter values: For each individual i , calculate $F'_{ki} = \sum_{j=1}^{\gamma'} (X_{kij} - X_{0j})^2 / \gamma'$. If $\min\{F'_{ki}\} < F'$, then let $F' = \min\{F'_{ki}\}$ and $G' = \arg \min\{F'_{ki}\}$.

Step 3. Determine the probability of selecting the particle to enter the next iteration according to the fitness value of each individual, and generate the selected population.

Step 4. From the selected population, the individuals with the cross-ratio q are randomly selected, and their position coordinates are coded with the appropriate coding method, and the new offspring individuals are generated by pairing.

Step 5. Among the new individuals, the individuals with mutation rate r are selected randomly, and the chromosome code is changed. The selected individuals, the newly generated offspring, and the mutated offspring form a new species group.

Step 6. Examine stopping criteria: If $k < N'$, let $k = k + 1$ and continue from Step 1. Otherwise, stop after running Step 2.

2.3.2. Evolutionary Strategy. The idea of ES is similar to that of GA, both of which are derived from biological evolution. In ES, each individual is also coded with a different chromosome. However, unlike GA which uses binary to code chromosomes, ES directly uses real values to code chromosomes. Through multiple iterations of reproduction, mutation, and selection, the entire population will converge on the optimal solution.

In ES, the input parameter set corresponding to each individual in the population is set to the chromosome encoding value. Each individual is given a randomly generated set of parameter values in population initialization. In each iteration, the parameter value of each individual will produce small, random, and unbiased mutations based on

TABLE 1: Common notations.

Notation	Meaning
α'	The number of parameters for calibration
γ'	The number of observed links
β'_k	The population size of the whole population in the k th iteration
F'_{ki}	The current fitness of the i th individual in the k th iteration
F'	The best fitness value (MSE) for all individuals in the population
N'	The maximum number of iterations
$P'_{ki} = (P'_{ki1}, P'_{ki2}, \dots, P'_{ki\alpha'})$	The parameter group of the i th individual in the k th iteration including α' parameters
$X'_0 = (X'_{01}, X'_{02}, \dots, X'_{0\gamma'})$	The set of field-measured data
$G' = (G'_1, G'_2, \dots, G'_\alpha')$	The parameter group associated with F'
$X_{ki} = (X_{ki1}, X_{ki2}, \dots, X_{ki\gamma'})$	The BNN outputs of the i th individual in the k th iteration

an adaptive Gaussian distribution to get offspring [45]. For each individual, the fitness value is calculated separately for the parent and offspring, and then the individual with the smaller fitness value is inherited. The process repeats until the maximum iterations. Eventually, the parameter values of the individual with the smallest fitness are the optimal parameter values to be calibrated.

The following describes the detailed steps of ES:

Step 0. Initialize: Let $k = 1$. Assign random values to P'_{1i} within the value range ($i = 1, \dots, \beta'_1$). Set $F' = \infty$ and $G' = 0$.

Step 1. Introduce the trained BNN model: For $i = 1, \dots, \beta'_k$, input P'_{ki} into the BNN model to obtain predicted outputs X_{ki} , respectively.

Step 2. Update the best fitness value and the optimal parameter values: For each individual i , calculate $F'_{ki} = \sum_{j=1}^{\gamma'} (X_{kij} - X'_{0j})^2 / \gamma'$. If $\min\{F'_{ki}\} < F'$, then let $F' = \min\{F'_{ki}\}$ and $G' = \arg \min\{F'_{ki}\}$.

Step 3. Mutation: Add a random number that follows an adaptive Gaussian distribution with zero mean affected by parameter variance δ_k to the parameter value of each parent individual in the population to obtain a new offspring.

Step 4. Selection: For $i = 1, \dots, \beta'_k$, calculate and compare the fitness values of the i th individual's parent and offspring. Individuals with smaller fitness values are chosen to keep.

Step 5. Examine stopping criteria: If $k < N'$, let $k = k + 1$ and continue from Step 1. Otherwise, stop after running Step 2.

2.3.3. Bat Algorithm. BA simulates the natural process of bat predation using ultrasound. BA has the characteristics of rapid convergence in the early search stage, which reflects the efficiency of the algorithm. The principle of BA is to map each individual in a group of bats to a randomly generated feasible solution and use the process of searching for prey and movement of bats to conduct simulated search and optimization, respectively. The evaluation of the position of the bat is based on the fitness value.

Bats find food by continuously adapting their sound waves during predation, and BA is no exception. First, the

bat performs a global search with louder sound waves and lower pulse frequencies. As the distance to the prey gets closer, the loudness of the sound waves decreases and the pulse frequency increases. During predation, the frequency of bats' emitted pulses and the pulse emission rate are automatically adjusted according to the distance to the prey [46, 47]. The process repeats until the maximum iterations. Finally, the feasible solutions corresponding to the bat individual with the smallest fitness value are the optimal values of parameters to be calibrated.

The unique notations of BA are shown in Table 2.

The following describes the detailed steps of BA:

Step 0. Initialize: Let $k = 1$. Assign random values to P'_{1i} within the value range ($i = 1, \dots, \beta'_1$). Set $F' = \infty$ and $G' = 0$. Initialize other parameters in BA.

Step 1. Introduce the trained BNN model: For $i = 1, \dots, \beta'_k$, input P'_{ki} into the BNN model to obtain predicted outputs X_{ki} , respectively.

Step 2. Update the best fitness value and the optimal parameter values: For each individual i , calculate $F'_{ki} = \sum_{j=1}^{\gamma'} (X_{kij} - X'_{0j})^2 / \gamma'$. If $\min\{F'_{ki}\} < F'$, then let $F' = \min\{F'_{ki}\}$ and $G' = \arg \min\{F'_{ki}\}$.

Step 3. Update according to the following equation: $f_i = \min\{f\} + (\max\{f\} - \min\{f\}) \cdot \beta$, $V_{ki}^{t+1} = V_{ki}^t + (X_{ki}^t - G') \cdot f_i$, $X_{ki}^{t+1} = X_{ki}^t + V_{ki}^{t+1}$, where β is a randomly generated variable and subject to a uniform distribution from 0 to 1.

Step 4. Generate a random number a . If $a > r_i^k$, $G' = G' + \epsilon L^k$, where ϵ is a randomly generated variable and subject to a uniform distribution from -1 to 1 .

Step 5. Generate a random number b . If $b < L_i^k$ and $F'_{ki} < F'$, use $r_i^{k+1} = \max\{r\} \cdot (1 - e^{-mk})$ and $L_i^{k+1} = n \cdot L_i^k$ to generate a new individual, where m and n are constants.

Step 6. Examine stopping criteria: If $k < N'$, let $k = k + 1$ and continue from Step 1. Otherwise, stop after running Step 2.

3. Case Study

3.1. Background. The field-measured data used for the case study are collected from three highway links of the North-South Elevated Road that are easily congested in Shanghai,

TABLE 2: Unique notations of BA.

Notation	Meaning
f_i	The frequency of the i th bat
V_i^k	The velocity of the i th bat in the k th iteration
r_i^k	The pulse emission rate of the i th bat in the k th iteration
L_i^k	The loudness of the i th bat in the k th iteration
L^k	The average loudness of the whole bat population in the k th iteration
$V_{ki} = (V_{ki1}, V_{ki2}, \dots, V_{ki\alpha})$	The velocity set of the i th bat in the k th iteration including α parameters

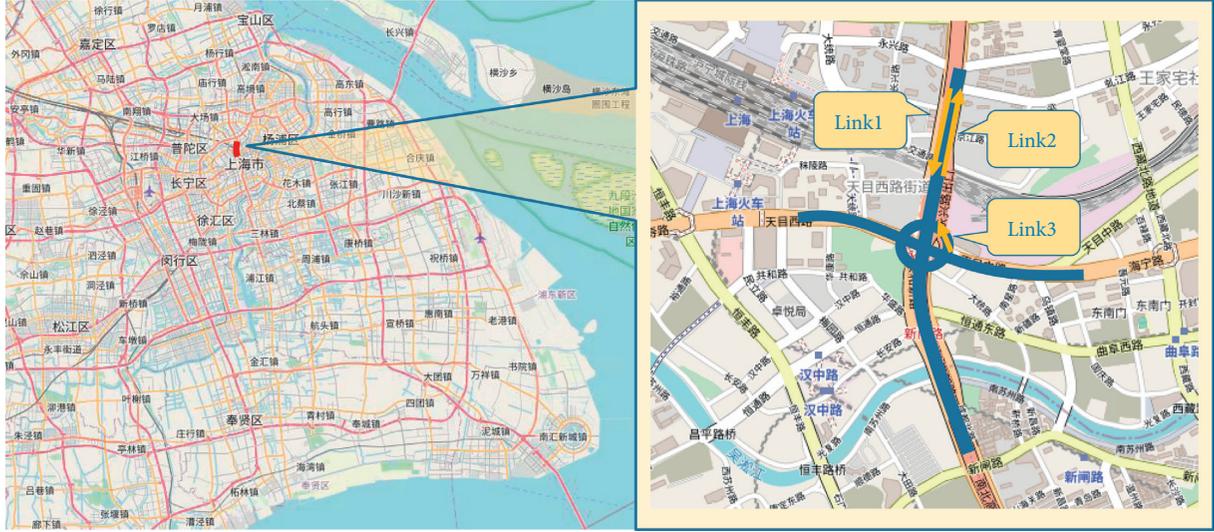


FIGURE 3: The geographical location of the three highway links in Shanghai, China.

China. The total length of the links is 2 km as shown in Figure 3. The data include the time and speed of the vehicle passing through the monitoring point of the link and the license plate number. The data collection time is from 9 am to 10 am on May 28, 2018, and the data source is Shanghai Municipal Traffic Management Bureau. Although the highway links do not have traffic lights, ground-level roads and nearby ramps have. Therefore, information about related traffic control is further gathered. Network geometric features, vehicular characteristics, and traffic volume are also collected. All information is inputted into the foundation of the traffic simulation model. In this case, we need to calibrate the average traffic speed on highway links.

Simulation is carried out in a universal microscopic traffic simulator TransModeler to verify the application of the proposed method in calibrating the parameters. TransModeler has various parameters that affect the simulation results (i.e., driver behaviors, car-following model parameters, and so on). Some parameters are the same in different regions, so they do not need calibrating. Nevertheless, parameters concerning driver behavior are different in different regions, which may have a huge impact on the simulation output and need calibration.

3.2. Parameter Identification. Some parameters that may need to be calibrated are initially selected in TransModeler. First, the car-following models in TransModeler are the basic

logic for realizing simulation, so it is necessary to select the parameters in the models for calibration. The car-following models that are universal for different vehicle types are shown in the following equations:

$$A'_i[t + \Delta t] = a' \frac{(V_i[t])^{b'}}{(D_{i,i-1}[t])^{c'}} (V_{i-1}[t] - V_i[t]) + \varepsilon_i, \quad (7)$$

$$A''_i[t + \Delta t] = a'' \frac{(V_i[t])^{b''}}{(D_{i,i-1}[t])^{c''}} (V_{i-1}[t] - V_i[t]) + \vartheta_i. \quad (8)$$

Among them, equation (7) is about acceleration and equation (8) is about deceleration. $A'_i[t + \Delta t]$ denotes acceleration of the i th vehicle at time $t + \Delta t$. $A''_i[t + \Delta t]$ denotes deceleration of the i th vehicle at time $t + \Delta t$. $V_i[t]$ denotes the speed of the i th vehicle at time t . $V_{i-1}[t]$ denotes the speed of the $i - 1$ th vehicle (the front vehicle of the i th vehicle) at time t . $D_{i,i-1}[t]$ denotes the distance between the i th vehicle and the $i - 1$ th vehicle at time t . ε_i and ϑ_i denote errors. Readers can refer to Reference [48–50] to know more about car-following models. Based on the above models, six parameters (a' , b' , c' , a'' , b'' , and c'') to be calibrated are selected. Subsequently, some other parameters are also selected, including the critical value of car-following speed, car-following time headway, and transverse speed when a vehicle changes

lane, transverse speed when a vehicle is in a transition section, and the maximum look-ahead distance of the driver's visibility. They are denoted as $P_1, P_2, P_3, P_4,$ and $P_5,$ respectively.

In the sensitivity analysis, average traffic speed (u) and traffic volume (q) on each highway link are regarded as simulation results. Thus, ΔR in Section 2.1 consists of Δq and Δu in this case. The number of random seeds in TransModeler is 10. Figures 4 and 5 illustrate the results of sensitivity analysis concerning q and $u,$ respectively. Among them, the error bars represent the effect caused by the random seed from 10 simulations. The results of the sensitivity analysis are obtained by selecting the maximum and minimum values of the corresponding parameters and using the default values for other parameters to run the simulation. Comprehensively, Δq and Δu of $a', b', c', a'', b'',$ and c'' are much larger than those of other parameters. Thus, $a', b', c', a'', b'',$ and c'' are selected for calibration.

In the range test, 460 sets of values of 6 selected parameters are generated at random within the corresponding recommended ranges from the instructions of TransModeler. For each set, we run 10 simulations and take the average value of the simulation speed as the results. The results of the range test verify that the field-measured speeds are within the range of the distribution of simulation speeds.

3.3. Building BNN. The same data set as the range test is used for building the BNN model. The data set is randomly divided into a training set, a validation set, and a test set according to the ratio of 8 : 1 : 1. Moreover, the basic framework of the BNN model is implemented based on the logic introduced in the previous subsection, and the BNN model is built based on Python and its neural network API, Keras. The trained BNN model is evaluated by the indicators mean absolute error (MAE), MSE, root mean squared error (RMSE), and mean absolute percentage error (MAPE). After the training and the evaluation of multiple BNN models, we comprehensively selected the model with the lowest error on the validation set to find the relationship between the simulation results and the parameters to be calibrated as close as possible. The selected BNN model has two hidden layers, each with 23 neurons. Rectified linear unit (ReLU) is selected as the neuron activation function. The values of hyperparameters $\pi, \sigma_1,$ and σ_2 are 0.5, 1.5, and 0.1, respectively.

The prediction ability of the BNN model is evaluated by the consistency of the simulation speeds in TransModeler and the predicted speeds from the BNN model based on the test set in the scatter plots with uncertainty, as shown in Figures 6–8. The horizontal axis of the scatter plot is the simulation speed, and the vertical axis is the predicted speed. The error bars of the data points in the vertical axis indicate the prediction range of the BNN model. If the predicted speeds are closer to the simulation speeds, in other words, the closer the data points are to the diagonal in the scatter plot, the better the model's predictive ability. And based on the test set, the indicators MAE, MSE, RMSE, and MAPE also demonstrate the predictive accuracy of the BNN model in Table 3. Although the BNN prediction error on Link 3 is

slightly larger, especially on MAPE, which is due to the small denominator, the overall prediction accuracy is high.

3.4. Using HA for Seeking the Optimal Parameter Values. Based on the previously trained BNN model, three HAs are performed to seek the optimal parameter value set. HAs are implemented in Python. To reduce randomness, we run 10 times for each HA.

In this case, the parameters of GA are set as follows: $\beta'_1 = 100, N' = 200, q = 0.8,$ and $r = 0.55$ [33, 38]. The setting of these parameters significantly affects the performance of GA. The population size β'_k and the maximum iterations N' mainly affect the convergence of the final results. If the population size is too large or the maximum iterations are too small, the algorithm is difficult to converge. Additionally, the cross-ratio q and the mutation rate r control the renewal of the population. If the cross-ratio is too high, it is easy to destroy the existing favorable mode. If the crossover ratio is too low, the renewal of the population is inefficient. A low mutation rate will rapidly reduce the diversity of the population, which can lead to an irreparable loss of effective genes. However, if the mutation rate is too high, the probability of the high-order mode being destroyed is also very high. Therefore, in order to ensure the efficiency and accuracy of GA, it is necessary to determine the reasonable population size, generation size, cross-ratio, and mutation rate.

Based on the actual situation of this case study, the parameters of ES take the same values as GA: $\beta'_1 = 100, N' = 200, \delta_{\max} = 12,$ and $\delta_{\min} = 1.$ The effects of the population size β'_k and the maximum number of iterations N' in ES are similar to those in GA. Moreover, δ_k is calculated as in the following equation in each iteration:

$$\delta_k = \delta_{\min} + (\delta_{\max} - \delta_{\min}) \frac{k}{N'}, \quad (9)$$

δ_k controls the variation of the population. When running ES at the beginning, δ_k is relatively large to facilitate global search. The global search aims to find the region where the optimal parameter values are located as much as possible. Otherwise, the results tend to converge to the local optimal solution, especially for large-scale problems. With the increase in iterations, δ_k gradually becomes smaller for local search to get the optimal parameter values.

According to Reference [29, 47] and actual case background, the parameters of BA in this case are set as follows: $\beta'_1 = 100, N' = 50, m = n = 0.9, f_{\max} = 1,$ and $f_{\min} = 0.$ Small population sizes will bring about a poor search effect. Nevertheless, when the size is large, the calculation complexity will increase and the efficiency will be reduced. Because BA converges quickly, the maximum number of iterations can be reduced compared with GA and ES. The determination of the maximum number of iterations also requires a trade-off. If the number of iterations is too small, the randomness will increase when the algorithm stops searching, and if the number of iterations is too large, the efficiency of solving will also be

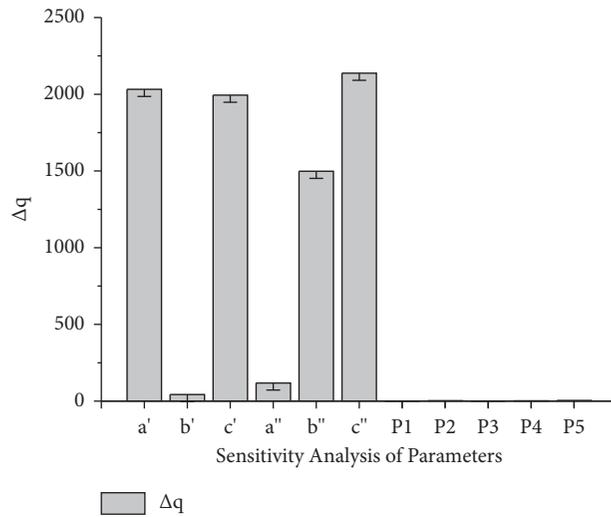


FIGURE 4: Sensitivity analysis results of traffic volume. The error bars represent the difference of traffic volume in 10 simulations with randomly generated seeds.

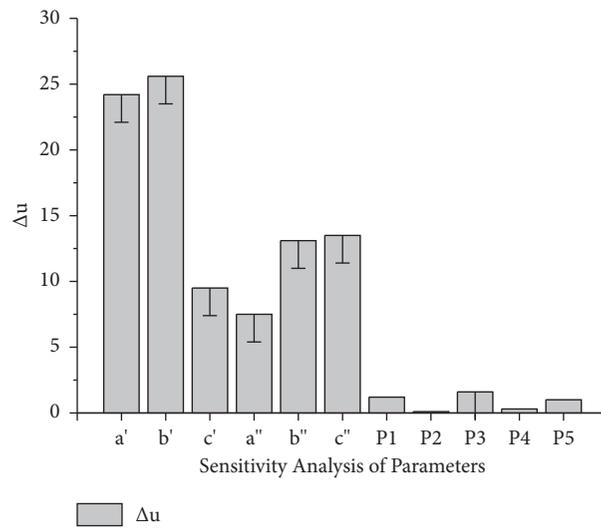


FIGURE 5: Sensitivity analysis results of average traffic speed. The error bars represent the difference of average traffic speed in 10 simulations with randomly generated seeds.

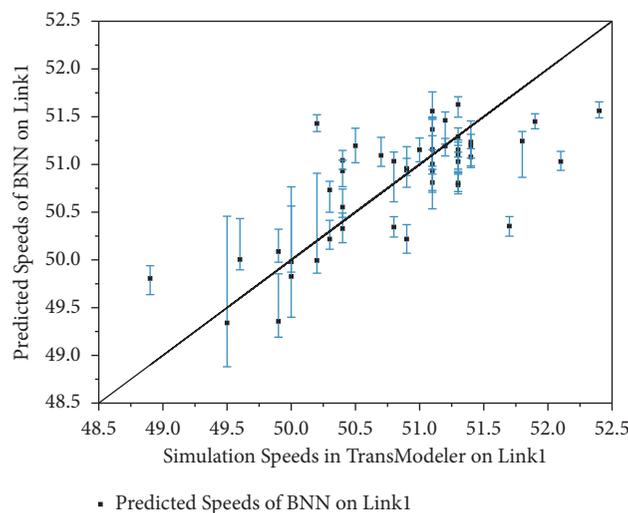


FIGURE 6: Scatter plot of simulation speeds and predicted speeds on Link 1. The error bars represent the 95% confidence interval of predicted speeds in 1000 predictions.

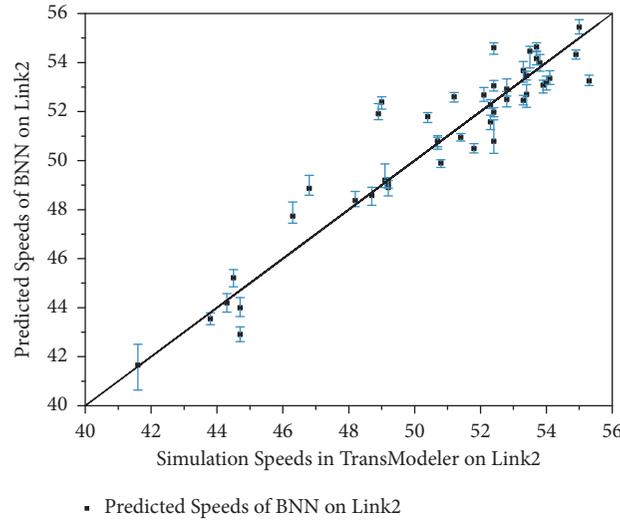


FIGURE 7: Scatter plot of simulation speeds and predicted speeds on Link 2. The error bars represent the 95% confidence interval of predicted speeds in 1000 predictions.

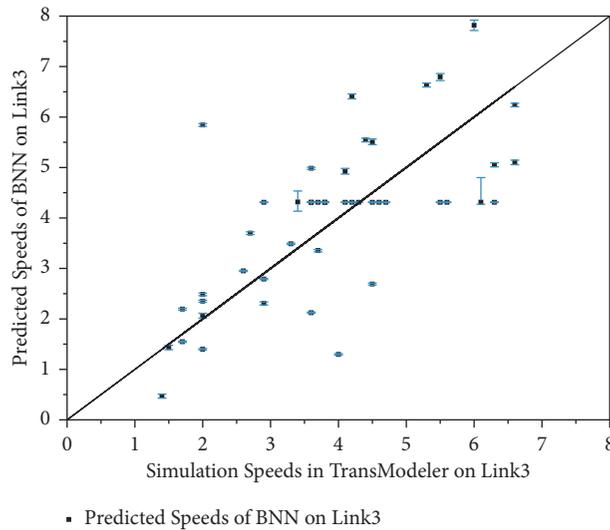


FIGURE 8: Scatter plot of simulation speeds and predicted speeds on Link 3. The error bars represent the 95% confidence interval of predicted speeds in 1000 predictions.

reduced. The values of the constants m and n and frequency also affect the efficiency and accuracy of the search. Therefore, it is necessary to determine the parameter values of BA.

4. Discussion

The previously proposed parameter calibration method combining ANN and GA (ANN-GA) is also introduced into the case study as a comparison [33, 39]. The ANN used has four hidden layers, each with 25 neurons. The dropout index is 0.2. The parameters of GA are the same as those in Section 3.4.

The change of the best fitness value in the iteration is shown in Figure 9. Table 4 shows the values of the calibrated parameter a', b', c', a'', b'' , and c'' , the MSE between the

predicted result of neural network using the optimal parameter values found by HA and the field-measured data (the minimum optimal fitness value in 10 runs), and the average calculation time of HA in 10 runs. No matter from the perspective of MSE or the calculation time, BNN-BA has the highest accuracy and the highest calculation efficiency, followed by BNN-ES, and BNN-GA is the lowest. ANN-GA not only requires the longest average computation time but also has the largest MSE. This suggests that BA is an excellent heuristic optimization algorithm in this case.

The average calculation time of each HA is less than 31 seconds, which is almost negligible. It is worth noting that the average calculation time in table is just the time to run HA for calibration without the simulation. In this case, it takes 416 seconds to run a simulation of TransModeler. If the traditional calibration method is used, the simulation needs

TABLE 3: MAE, MSE, RMSE, and MAPE of the trained BNN model.

No.	MAE	MSE	RMSE	MAPE (%)
Link 1	0.369	0.234	0.484	0.7
Link 2	0.807	1.246	1.116	1.6
Link 3	0.924	1.451	1.205	25.9

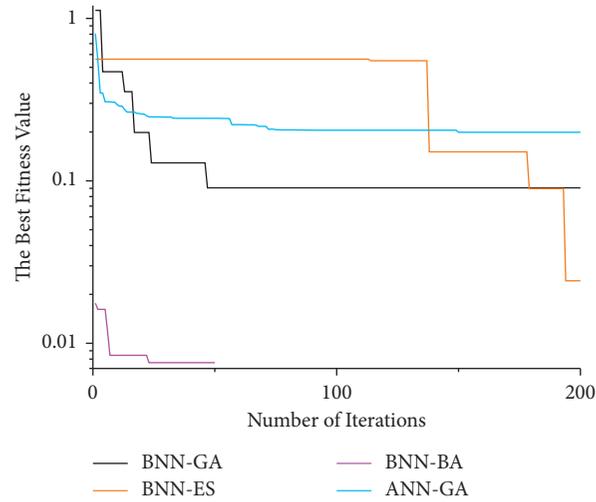


FIGURE 9: The best fitness value of the four methods.

TABLE 4: Results of the calibration methods.

Method	a'	b'	c'	a''	b''	c''	MSE (optimal fitness value)	Average calculation time (seconds)
BNN-GA	3.75	-1.35	-2.70	0.38	2.32	-0.03	0.091	18.898
BNN-ES	5.00	-2.14	-2.21	0.22	1.74	2.51	0.018	16.467
BNN-BA	4.73	-2.25	-1.62	-1.10	-0.62	-0.40	0.001	8.054
ANN-GA	4.99	-0.58	-2.99	-0.27	-2.99	-2.99	0.199	30.298

to be run many times, and the calibration time will be several times longer than 416 s. If the traditional GA-based calibration method is performed, 20000 simulations (the population size multiplied by the number of iterations) need to be run during the calibration based on the same parameters of GA. Even if the traditional calibration method using high-efficiency BA is performed, 5000 simulations still need to be run during the calibration based on the same parameters. Thus, the calibration time of the traditional method is much longer than that of the proposed method. Even if the proposed method considers the number of simulations for obtaining the training dataset, the total number of simulations in this method is still much lower than that of the traditional method. Furthermore, it is only necessary to simulate in order to obtain data for training the BNN model before the calibration process. Therefore, the time saved by not running the simulation during the calibration process is the biggest benefit of the proposed method.

Apart from this, although the number of highway links in the previous case study is only three, considering that the parameters to be calibrated and the road network scale are independent, the impact of the road network scale on the

calculation time for running HA to calibrate is limited. If the influence exists, it would likely to be derived from performing the trained BNN model to predict the simulation results and updating the fitness value in HA. However, predicting and updating only need simple calculations, which take very little time. In a word, the shortened expected calculation time almost without the influence of network scale is also the benefit of the proposed method.

Figure 10 illustrates the average speeds of highway links based on BNN-GA, including the output results of TransModeler using the default parameter values (uncalibrated), the calibrated output results of TransModeler using the optimal parameter values found by BNN-GA (calibrated by BNN-GA), and the BNN predicted results using the optimal parameter values found by the combination of BNN-GA (predicted by BNN). The points in Figure 10 indicate the average speed measured in the field. Figures 11–13 show the calibration results of BNN-ES, BNN-BA, and ANN-GA, respectively, in the same way. It is obvious that the calibrated results and the BNN predicted results have close similarity, which verifies that the trained BNN model can replace TransModeler to get simulation outputs, not only in accuracy but also in

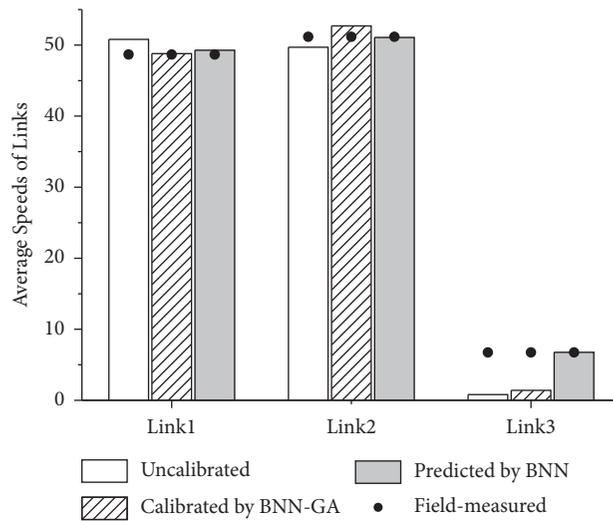


FIGURE 10: Calibration results using BNN-GA.

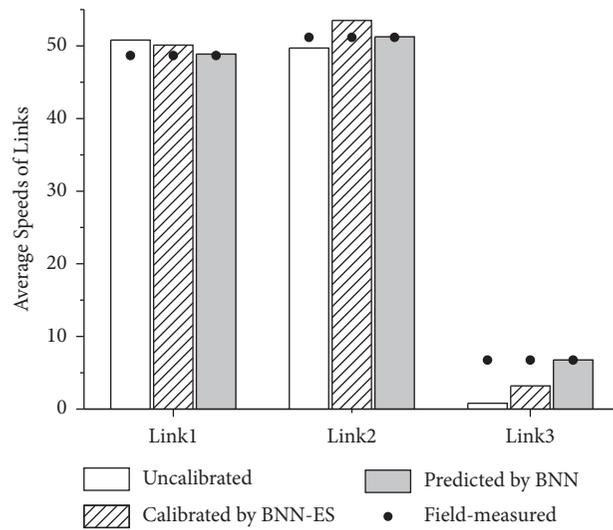


FIGURE 11: Calibration results using BNN-ES.

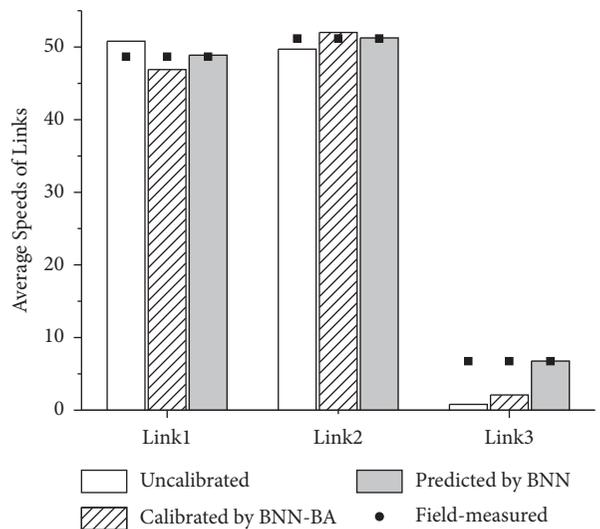


FIGURE 12: Calibration results using BNN-BA.

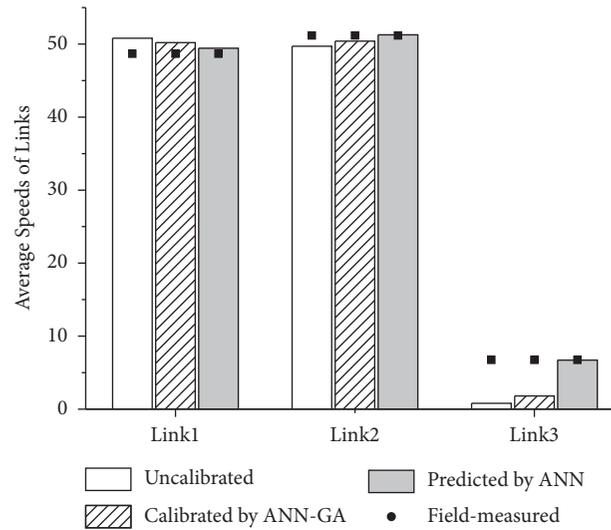


FIGURE 13: Calibration results using ANN-GA.

TABLE 5: Square error of three links.

No.	Uncalibrated	Calibrated by BNN-GA	Calibrated by BNN-ES	Calibrated by BNN-BA	Calibrated by ANN-GA
Link 1	4.558	0.018	2.059	3.115	2.356
Link 2	2.148	2.355	5.450	0.696	0.586
Link 3	35.286	28.518	12.533	21.531	24.406
Mean (MSE)	13.997	10.297	6.681	8.448	9.116

computationally efficiency. And the similarity between the BNN predicted results and the field-measured data suggests that inputting the parameter values found by the combination of BNN and HA into the BNN model can perfectly estimate actual road conditions.

Table 5 presents the square error of the default output results of TransModeler (uncalibrated) and the calibrated output results of TransModeler using the optimal parameter values found by the combination of neural network and HA, respectively.

Although there is some error in the simulation results predicted by BNN, the error is within acceptable limits compared with the improvement in the efficiency of the calibration calculation. No matter which HA is used for calibration, the MSE is lower than that using default and uncalibrated parameter values. This verifies the validity of the parameter calibration method proposed in this paper. Two calibration methods using GA (BNN-GA and ANN-GA) have poor performance in terms of MSE and the average calculation time for calibration. Considering the trade-off between MSE and calculation time, BNN-ES and BNN-BA perform well. BNN-ES has the lowest MSE (6.681) after calibration, while BNN-BA requires a shorter calculation time (8.054s) and has a relatively low MSE (8.448).

5. Conclusions

In this paper, a new parameter calibration method for microscopic traffic simulators combining BNN and HA is proposed. The objective of the BNN is to quickly predict

simulation results with uncertainty based on inputs of traffic simulator parameter values without real simulation. Based on the trained BNN model, the purpose of running HA is to search for the optimal values of the parameters to be calibrated in the traffic simulator to minimize the difference between field-measured values and the predicted simulation results from BNN without simulation. The combination of BNN and HA avoids running the simulator repeatedly during the calibration, which can significantly decrease the computation time of calibration. The research innovation lies not only in the huge improvement of calibration efficiency but also in which the method is universal for microscopic traffic simulators.

Based on a real case, the calibration method combining BNN and HA is applied to calibrate parameter values in TransModeler, which also validates the proposed method in error and efficiency. The result shows that the BNN model captures the uncertainty of the simulation and has high predictive accuracy. And it takes almost no time to seek the parameter value set with a small MSE by using the combination of BNN and HA. When comparing the average calculation time, BNN-BA has the highest calculation efficiency, followed by BNN-ES. Moreover, in terms of the MSE between the calibrated output results and the field-measured data, BNN-ES performs best, BNN-BA performs moderately, and BNN-GA performs worst. Therefore, considering that the total calibration accuracy is high in the balance between calibration efficiency and calibration error, BNN-ES and BNN-BA have their own advantages to calibrate parameters in microscopic traffic simulators compared with BNN-GA and the existing ANN-GA.

The proposed method can quickly obtain the calibrated parameter values after training the BNN, which facilitates the simulation to be closer to the actual situation. The parameter value found is suitable for a certain area where the case is located. The BNN model in the proposed method can also be used by traffic authorities to obtain predictive real-time traffic conditions without running microscopic traffic simulators.

Future research can focus on how to optimize the hyperparameters in the BNN and establish a more reliable variational distribution to better approximate the Bayesian posterior weight distribution and predict the distribution of simulation outputs. Furthermore, based on the inherent characteristics of HA, the results inevitably end at a locally optimal solution. It may be meaningful to further improve HA for calibration to find the global optimal solution rather than the locally optimal solution.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by Project of Shanghai Science and Technology Innovation Action Plan (grant no. 20511101800) and National Natural Science Foundation of China (grant no. 52002243).

References

- [1] W. Wu, R. Sun, A. Ni, Z. Liang, and H. Jia, "Simulation and evaluation of speed and lane-changing advisory of CAVS at work zones in heterogeneous traffic flow," *International Journal of Modern Physics B*, vol. 34, no. 21, Article ID 2050201, 2020.
- [2] M. Yanai, K. Abe, T. Yamada, H. Fujii, and S. Yoshimura, "Cluster analysis for a series of microscopic traffic simulation results," *Journal of Advanced Simulation in Science and Engineering*, vol. 4, no. 1, pp. 78–98, 2018.
- [3] J. Barceló, "Fundamentals of traffic simulation," *Springer-Verlag GmbH*, vol. 145, no. 145, 2010.
- [4] H. Yang, S. Han, and X. Chen, *Parameter Calibration and Application for the Vissim Simulation Model*, Urban Transport of China, Beijing, China, 2006.
- [5] Y. Kang, M. Zhai, H. Zhang, and C. Zhang, "Parameter calibration of VISSIM simulation model based on the orthogonal experiment," in *Proceedings of the Tenth International Conference of Chinese Transportation Professionals*, Beijing, China, 2010.
- [6] M. Farzaneh and H. Rakha, "Procedures for calibrating TRANSYT platoon dispersion model," *Journal of Transportation Engineering*, vol. 132, no. 7, pp. 548–554, 2006.
- [7] A. Agarwal, D. Ziemke, and K. Nagel, "Calibration of choice model parameters in a transport scenario with heterogeneous traffic conditions and income dependency," *Transportation Letters*, vol. 12, no. 7, pp. 441–450, 2020.
- [8] C. Zhang, C. Osorio, and G. Flötteröd, "Efficient calibration techniques for large-scale traffic simulators," *Transportation Research Part B: Methodological*, vol. 97, pp. 214–239, 2017.
- [9] A. U. Z. Patwary, W. Huang, and H. K. Lo, "Metamodel-based calibration of large-scale multimodal microscopic traffic simulation," *Transportation Research Part C: Emerging Technologies*, vol. 124, Article ID 102859, 2021.
- [10] R.-L. Cheu, X. Jin, K.-C. Ng, Y.-L. Ng, and D. Srinivasan, "Calibration of FRESIM for Singapore expressway using genetic algorithm," *Journal of Transportation Engineering*, vol. 124, no. 6, pp. 526–535, 1998.
- [11] S. J. Kim, W. Kim, and L. R. Rilett, "Calibration of Microsimulation Models Using Nonparametric Statistical Techniques," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1935, no. 1, pp. 111–119, 2005.
- [12] B. Park and H. Qi, "Development and evaluation of a procedure for the calibration of simulation models," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1934, no. 1, pp. 208–217, 2005.
- [13] S. Menneni, C. Sun, and P. Vortisch, "Microsimulation calibration using speed-flow relationships," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2088, no. 1, pp. 1–9, 2008.
- [14] O. Giuffrè, A. Granà, M. L. Tumminello, and A. Sferlazza, "Estimation of passenger car equivalents for single-lane roundabouts using a microsimulation-based procedure," *Expert Systems with Applications*, vol. 79, pp. 333–347, 2017.
- [15] S. Chiappone, O. Giuffrè, A. Granà, R. Mauro, and A. Sferlazza, "Traffic simulation models calibration using speed-density relationship: an automated procedure based on genetic algorithm," *Expert Systems with Applications*, vol. 44, pp. 147–155, 2016.
- [16] R. Omrani and L. Kattan, "Concurrent estimation of origin-destination flows and calibration of microscopic traffic simulation parameters in a high-performance computing cluster," *Journal of Transportation Engineering, Part A: Systems*, vol. 144, no. 1, Article ID 04017068, 2018.
- [17] T. Ma and B. Abdulhai, "Genetic algorithm-based optimization approach and generic tool for calibrating traffic microscopic simulation parameters," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1800, pp. 6–15, 2002.
- [18] G. Amirjamshidi and M. J. Roorda, "Multi-objective calibration of traffic microsimulation models," *Transportation Letters*, vol. 11, no. 6, pp. 311–319, 2019.
- [19] K. Bhattacharyya, B. Maitra, and M. Boltze, "Calibration of micro-simulation model parameters for heterogeneous traffic using mode-specific performance measure," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2674, no. 1, pp. 135–147, 2020.
- [20] C. Wang and C. Xu, "On the effects of various measures of performance selections on simulation model calibration performance," *Journal of Advanced Transportation*, vol. 2018, pp. 1–16, 2018.
- [21] M. Dubska, A. Herout, R. Juránek, and J. Sochor, "Fully automatic roadside camera calibration for traffic surveillance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1162–1171, 2015.
- [22] B. Kostic, L. Meschini, and G. Gentile, "Calibration of the demand structure for dynamic traffic assignment using flow and speed data: exploiting the advantage of distributed

- computing in derivative-free optimization algorithms,” *Transportation Research Procedia*, vol. 27, pp. 993–1000, 2017.
- [23] D. K. Hale, C. Antoniou, M. Brackstone, D. Michalaka, A. T. Moreno, and K. Parikh, “Optimization-based assisted calibration of traffic simulation models,” *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 100–115, 2015.
- [24] S. Oh, R. Seshadri, C. L. Azevedo, and M. E. Ben-Akiva, “Demand calibration of multimodal microscopic traffic simulation using weighted discrete SPSSA,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2673, no. 5, pp. 503–514, 2019.
- [25] Q. Liu, L. Wu, W. Xiao, F. Wang, and L. Zhang, “A novel hybrid bat algorithm for solving continuous optimization problems,” *Applied Soft Computing*, vol. 73, pp. 67–82, 2018.
- [26] R. M. Rizk-Allah and A. E. Hassanien, “New binary bat algorithm for solving 0-1 knapsack problem,” *Complex & Intelligent Systems*, vol. 4, no. 1, pp. 31–53, 2018.
- [27] X. S. Yang and A. Hossein Gandomi, “Bat algorithm: a novel approach for global engineering optimization,” *Engineering Computations*, vol. 29, no. 5, pp. 464–483, 2012.
- [28] S. Srivastava and S. K. Sahana, “Application of bat algorithm for transport network design problem,” *Applied Computational Intelligence and Soft Computing*, vol. 2019, Article ID 9864090, 12 pages, 2019.
- [29] X. S. Yang and X. He, “Bat algorithm: literature review and applications,” *International Journal of Bio-Inspired Computation*, vol. 5, no. 3, p. 141, 2013.
- [30] A. D. Lidbe, A. M. Hainen, and S. L. Jones, “Comparative study of simulated annealing, tabu search, and the genetic algorithm for calibration of the microsimulation model,” *Simulation*, vol. 93, no. 1, pp. 21–33, 2017.
- [31] M. Yu and W. Fan, “Calibration of microscopic traffic simulation models using metaheuristic algorithms,” *International Journal of Transportation Science and Technology*, vol. 6, no. 1, pp. 63–77, 2017.
- [32] X. S. Yang, *A New Metaheuristic Bat-Inspired Algorithm*, Springer, Berlin, Germany, 2010.
- [33] H. Shahrokhi Shahraki, C. Alecsandru, R. Maghsoudi, and L. Amador, “An efficient soft computing-based calibration method for microscopic simulation models,” *Journal of Transportation Safety & Security*, vol. 10, no. 4, pp. 367–386, 2018.
- [34] N. Dadashzadeh, M. Ergun, A. S. Kesten, and M. Zura, “Improving the calibration time of traffic simulation models using parallel computing technique,” in *Proceedings of the IEEE MT-ITS2019—6th International Conference on Models and Technologies for Intelligent Transportation Systems*, June 2019.
- [35] G. Xiao, R. Wang, C. Zhang, and A. Ni, “Demand prediction for a public bike sharing program based on spatio-temporal graph convolutional networks,” *Multimedia Tools and Applications*, vol. 80, no. 15, pp. 22907–22925, 2020.
- [36] K. Hornik, “Approximation capabilities of multilayer feed-forward networks,” *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.
- [37] I. Ištoka Otković, T. Tollazzi, and M. Šraml, “Calibration of microsimulation traffic model using neural network approach,” *Expert Systems with Applications*, vol. 40, no. 15, pp. 5965–5974, 2013.
- [38] N. T. Ratrou, S. M. Rahman, and I. Reza, “Calibration of PARAMICS model: application of artificial intelligence-based approach,” *Arabian Journal for Science and Engineering*, vol. 40, no. 12, pp. 3459–3468, 2015.
- [39] Y. Liu, B. Zou, A. Ni, L. Gao, and C. Zhang, “Calibrating microscopic traffic simulators using machine learning and particle swarm optimization,” *Transportation letters*, vol. 13, no. 4, pp. 295–307, 2020.
- [40] M. Rahman, M. Chowdhury, T. Khan, and P. Bhavsar, “Improving the efficacy of car-following models with a new stochastic parameter estimation and calibration method,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2687–2699, 2015.
- [41] F. Abodo, A. Berthoume, S. Zitzow-Childs, and L. Bobadilla, “Strengthening the case for a bayesian approach to car-following model calibration and validation using probabilistic programming,” “, IEEE, in *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference—ITSC Auckland*, New Zealand, , 2019.
- [42] P. Byungkyu and Q. Hongtu, “Microscopic simulation model calibration and validation for freeway work zone network - a case study of VISSIM,” in *Proceedings of the IEEE Intelligent Transportation Systems Conference*, September 2006.
- [43] A. Kendall and Y. Gal, “What uncertainties do we need in bayesian deep learning for computer vision?” in *Proceedings of the 31st Conference on Neural Information Processing System*, Long Beach, CA, USA, 2017.
- [44] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, “Weight uncertainty in neural networks,” *Computer Science*, 2015.
- [45] N. Hansen, D. V. Arnold, and A. Auger, “Evolution strategies,” in *Springer Handbook of Computational Intelligence*, pp. 871–898, Springer, Berlin, Germany, 2015.
- [46] X. S. Yang and S. Xin, “Bat algorithm for multi-objective optimisation,” *International Journal of Bio-Inspired Computation*, vol. 3, no. 5, p. 267, 2011.
- [47] A. H. Gandomi and X.-S. Yang, “Chaotic bat algorithm,” *Journal of Computational Science*, vol. 5, no. 2, pp. 224–232, 2014.
- [48] H. Subramanian, *Estimation of Car-Following Models*, Massachusetts Institute of Technology, Cambridge, MA, USA, 1996.
- [49] K. I. Ahmed, *Modeling Drivers’ Acceleration and Lane Changing Behavior*, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999.
- [50] H. Farah and H. N. Koutsopoulos, “Do cooperative systems make drivers’ car-following behavior safer?” *Transportation Research Part C: Emerging Technologies*, vol. 41, pp. 61–72, 2014.