

Full Length Article

Metaheuristic-optimized neural networks for travel time index prediction: A comparative study of wild horse and coot optimization algorithms

Navid Khorshidi^a, Soheil Rezashoar^b, Pegah Amini^c, Shahriar Afandizadeh Zargari^a,
Hamid Mirzahassein^{d,*}

^a School of Civil Engineering, Iran University of Science and Technology, Tehran, Iran

^b Department of Transportation Planning, Faculty of Engineering, Imam Khomeini International University, Qazvin, Iran

^c Transportation Engineering and Planning, Department of Civil Engineering, Sharif University of Technology, Tehran, Iran

^d Department of Civil – Transportation Planning, Imam Khomeini International University, Qazvin, Iran

ARTICLE INFO

Keywords:

Travel time index

Travel time reliability

Multi-layer perceptron (MLP)

Wild horse optimization (WHO)

Coot optimization algorithm (COA)

ABSTRACT

Accurate travel time prediction is essential for effective Intelligent Transportation Systems (ITSs). Unlike traditional gradient-based methods, which may get trapped in local optima, metaheuristic algorithms efficiently navigate complex optimization landscapes without needing derivative computations. This study compares two metaheuristic algorithms—Wild Horse Optimization (WHO) and Coot Optimization Algorithm (COA)—for optimizing the weights and biases of a Multilayer Perceptron (MLP) neural network to predict the Travel Time Index (TTI). The MLP uses four input variables (traffic volume, weather, crash rate, and section length), one hidden layer with 12 neurons, and one output neuron. The model was trained and tested on a dataset of 472 observations from Virginia's transportation network (2014–2017), with input variables showing mean values of 1.19 (traffic volume), 6.92 mm (precipitation), 0.17 (crash rate), and 2.76 miles (section length), and standard deviations of 0.20, 5.21, 0.12, and 1.64, respectively. Performance was assessed using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Correlation Coefficient (R). The WHO-optimized model outperformed COA, achieving an MSE of 0.0026, RMSE of 0.0511, and R of 0.7516, compared to COA's MSE of 0.0028, RMSE of 0.0533, and R of 0.7262. WHO also showed better computational efficiency (147 s vs. 160 s for COA). These results highlight the WHO-MLP model's superior accuracy and speed, making it ideal for real-time navigation systems.

1. Introduction

Travel time plays a serious role in transportation network analysis and is considered a pivotal variable in shaping user behavior, helping to make informed decisions, and guiding traffic flow planning and management. With the rapid development of transportation infrastructure and the proliferation of Intelligent Transportation Systems (ITSs), accurate travel time prediction has emerged as a fundamental requirement for enhancing the efficiency of these systems. However, traffic congestion—a primary source of travel time variability and uncertainty—poses significant challenges to prediction accuracy [65].

The Federal Highway Administration (FHWA) identifies seven primary sources of congestion that are known to impact transportation network performance: accidents, work zones, weather conditions,

demand fluctuations, special events, physical bottlenecks, and traffic control systems [46]. To evaluate the stability of a network in response to travel time fluctuations, the concept of Travel Time Reliability (TTR) has been introduced. Although a single origin could not be assigned to the advent of TTR, travelers implicitly understood it and would buffer their travel times to allow for unexpected delays. Travel times were not always consistent, as early transportation research started to recognize. An example is the study of Gaver in 1968, where he suggested a departure time choice model in which passengers factored in a “head start time” and took into account the variance of their travel time [21]. A major step in quantifying and integrating reliability into transportation analysis was taken by Jackson and Jucker in 1982. They introduced the “Mean-Variance” framework, where a decision-maker tries to minimize a sum of expected travel time and travel time variability [27].

* Corresponding author.

E-mail addresses: navid.amoei.khorshidi@gmail.com (N. Khorshidi), soheilshoar@edu.ikiu.ac.ir (S. Rezashoar), p.amini75@yahoo.com (P. Amini), zargari@iust.ac.ir (S.A. Zargari), mirzahassein@eng.ikiu.ac.ir (H. Mirzahassein).

<https://doi.org/10.1016/j.treng.2025.100395>

Received 4 July 2025; Received in revised form 28 September 2025; Accepted 7 October 2025

Available online 8 October 2025

2666-691X/© 2025 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Table 1
Summary of Reviewed Studies Focusing Specifically on TTI in TTR Predictive Models.

Reference	Category	Methodology	Scope	Performance Measure/ Description
[80]	Corridor level	Statistical and analytical models	Forecasting hourly TTI	Robustness of Interacting Multiple Model (IMM) predictor could be increased and could perform better than the other predictors.
[81]	Corridor level	Statistical, analytical, and machine learning models	Traffic state forecasting	Lower MAPE compared to other methods (0.94 for their model)
[82]	Corridor level	Statistical and analytical models	Combined linear forecasting techniques to predict TTI during peak and non-peak periods	$0.64 < \text{MAPE} < 1.86$
[79]	Corridor level	Statistical and analytical models	IMM predictor for hourly TTI forecasting, superior stability, and robustness of IMM	$0.84 < \text{MAPE} < 1.45$
[43]	Network level	Statistical and analytical models	Analyzing incident-induced congestion on TTI variations by Spatio-temporal analysis and visualization.	The results emphasize the role of incidents in increasing congestion.
[4]	Corridor level	Statistical and analytical models	Calculating TTI to assess level of service (LOS)	An average TTI of 9.331, indicating poor service due to checkpoints.
[13]	Corridor level	Statistical and analytical models	Developing a model based on field data on traffic volume, travel time, and road geometry	nonlinear regression models to predict TTI
[85]	Network level	Statistical and analytical models	Using spatiotemporal data mining with GPS data from floating vehicles to predict TTI	Forecasting models: Curve Fit, Exponential Smoothing, and Forest-based $R^2 = 0.99$ $\text{RMSE} = 4.20$ $0.61 < \text{Validation RMSE} < 0.87$ $0.57 < \text{Forecast RMSE} < 0.70$
[73]	Network level	Machine learning methods	Using Convolutional Neural Network (CNN) to predict TTI	$0.52 < \text{RMSE} < 1.10$ $0.08 < \text{MAPE} < 0.14$
[8]	Network level	Statistical and analytical models	This study conducts a contrastive analysis of TTI variations under weather conditions using multivariate regression models at different levels	
[72]	Network level	Statistical and analytical models	Comparing traffic flow under non-precipitation and rainy conditions	Maximum of the average TTI increases on the city scale under various rainfall intensities was between 3.3 %, and 10.8 %.
[3]	Network level	Multiple machine learning methods	The proposed approach was also able to identify the effects of weather and seasonality	Developing a framework to forecast (TTI) They used machine learning models (regression): Ridge, Linear, Lasso, DT and SVR, and Ridge was the best.
[6]	Network level	Statistical and analytical models	Quantifying the impact of rainfall on traffic mobility and TTR	Quantile regression and statistical tests to quantify rainfall's effect on travel time distribution in India. The results found 457 % difference between maximum TTI values during rainfall vs. normal conditions.
[61]	Origin-Destination pairs	Statistical and analytical models+ Unsupervised machine learning	Study of ridesharing data in three major US metropolitan areas	The TTI in the three cities typically ranged from 1.1 to 1.6. During peak traffic hours, the TTI reached its highest value (e.g., around 1.5–1.6 in Chicago and Los Angeles), while it remained close to 1.1 during off-peak hours.
[16]	Corridor level	Statistical and analytical models	Evaluate TTR and identifying the factors affecting it.	TTR is significantly affected by several factors, including congestion, time of day, public holidays and characteristics of the traffic flow.
[64]	Corridor level	Statistical and analytical models	Measurement and analysis of travel time and delays on specific routes during morning peak periods	The highest TTI value was observed in Army Chandel expressway link at 8.9 %.
[67]	Corridor level	Statistical models	The effect of the COVID-19 pandemic on TTR	No differences in TTI values when TTDs are formulated using a 5 min or 15 min aggregation interval

The significance of TTR was increasingly acknowledged in the 1990s and 2000s, especially for freight and commuter transportation. The study went on to investigate various theoretical frameworks, such as “Scheduling Delays” and “Mean-Lateness,” for calculating the value of TTR. Improved measurement and analysis of travel time variability was also made possible by the development of increasingly complex data collection techniques. Some of the studies conducted during this period are the study of Small, Noland, and Koskenoja, where they evaluated how a person’s socioeconomic characteristics affect their travel reliability using a stated preference approach [59]. Another outstanding study was conducted by Abdel-Aty, Kitamura, and Jovanis. They used a stated preference survey to examine the impact of travel time variability on route selection [1].

The promotion of TTR as a performance metric was greatly supported by the FHWA. In a report entitled “*Traffic congestion and reliability: Trends and advanced strategies for congestion mitigation*”, the growing problem of unreliable travel time was highlighted and strategies to address it were suggested. According to the FHWA, TTR is defined as “the consistency or predictability of travel times across different time periods and days” [63]. Also, the National Cooperative Highway Research Program (NCHRP) Report 311 further outlines three common definitions of TTR:

- The discrepancy between expected and actual travel times caused by non-recurring congestion;
- The variability in travel times observed over a large sample of daily trips;
- The systemic impact of non-recurring congestion is quantified by the intensity, duration, and spatial extent of congestion events [58].

Although defining definitions for TTR could pave the way toward incorporating TTR into transportation and planning models, some significant flaws in conventional transportation performance metrics such as better justification for operations and management strategies and economic impacts of unreliability paved the way toward defining numerical measures for TTR [22,53]. The numerical metrics included statistical measures such as standard deviation, coefficient of variation, skewness, percentage change, and percentiles, as well as performance-based indices like the Buffer Index (BI), Travel Time Index (TTI), Planning Time Index (PTI), and congestion frequency index [11, 35]. Among them, the TTI is widely recognized as a critical metric for assessing congestion severity. TTI is defined as the ratio of the average actual travel time to the free-flow travel time and has been extensively adopted in transportation research [49].

TTI was first introduced by Texas Transportation Institute, in their “Urban Mobility Report”. This measure was applied to 101 American

cities on a yearly basis since 1982 [53]. Since its introduction, it has been applied to studies trying to address various aspects of transportation systems. Applications in planning and management are one of these focuses and examples of the studies including but not limited to: congestion monitoring [28], examining TTI to improve transportation planning [38], and analyzing TTI to evaluate service quality of road network sections [7]. Also, another category of studies, suggested new approaches, measurements, or confirmed current approaches pertaining to TTR and TTI. Examples of these studies are as follows: the study of Pu who analytically explored relationships between TTI and other reliability measures [49], or proposing a new travel time index (CMTT) that incorporates TTI in equilibrium modeling [77]. Finally, some studies focused on cutting-edge approaches, data sources, or collection techniques for TTI analysis or application [18,32,66].

Another field of study, compatible with this research, is the application of TTI in modeling TTR. Since the focus of this study is developing a novel approach to model TTI, we only focus on studies that consider TTI in their modeling process. These studies either develop or apply models to predict the TTI or investigate how various factors affect TTI. Further details on both of these categories are available in Table 1. In this table, MAPE stands for Mean Absolute Percentage Error, RMSE stands for Root Mean Squared Error, and SVR is Support Vector Regression and DT as Decision Tree.

Upon deeply reviewing the studies, it was revealed that researchers use state-of-the-art predictive methods, mostly machine learning, such as CNNs, Least Squares Support Vector Machines (LS-SVMs), and IMM to predict TTI with high accuracy. Statistical techniques like nonlinear regression and spatiotemporal data mining, which use massive datasets to reveal complex traffic dynamics, improve these strategies.

Classical statistical models predominantly rely on historical data and assumptions about the stability of traffic patterns [47]. While methods like historical averaging demonstrate reasonable performance under stable, low-density traffic conditions [24], they struggle to adapt to dynamic environments.

Linear regression models improve upon simpler approaches in unstable conditions by incorporating variables such as distance, stops, and weather conditions; however, their accuracy is constrained by collinearity and interdependencies among predictors [2,19,33,78]. Time series models (e.g., ARIMA), which assume linear and stationary patterns, achieve limited success in static scenarios but exhibit significant accuracy degradation when applied to real-time data or rapidly changing conditions [9]. The Kalman filter represents a dynamic approach for travel time estimation through state-space modeling and real-time data integration. Extended variants like the Extended Kalman Filter (EKF) have been developed to address the linearity constraints of standard implementations, though challenges persist in complex traffic systems [74].

Despite their widespread adoption, statistical models exhibit limited effectiveness in real-time travel time forecasting due to their reliance on linear assumptions, sensitivity to data noise, and inability to handle complex, real-time data analysis. To address these limitations, machine learning models have emerged as a viable and flexible alternative, leveraging their capacity to discern nonlinear patterns and process vast amounts of data effectively.

Among various methods, Artificial Neural Networks (ANNs) have earned significant attention due to their ability to accurately model dynamic traffic conditions and their key role in enhancing the performance of intelligent transportation systems. Recent studies demonstrate that ANNs, owing to their global approximation capabilities and high computational power, outperform traditional methods and can effectively address challenges such as multicollinearity. These advantages have established ANNs as reliable tools for predicting traffic parameters, earning them a prominent position in transportation engineering literature [17]. Also, ANNs could prove themselves in traffic noise modelling [12].

The ANNs have gradually proven themselves to be a reliable tool for

modeling complex tasks and datasets, however, relatively fewer number of studies applying them to TTR prediction, compared to travel time prediction, shows room for further application. The study of Van Lint et al. was among the first studies applying ANNs in TTR prediction. They proposed a neural network model to forecast three characteristic percentiles, 10th, 50th, and 90th, claiming neural networks could be applied as an effective tool for predicting these percentiles and were able to make smooth estimates of characteristic percentiles [69]. It is widely acknowledged that TTR should be included in the planning and operational models for metropolitan areas. The report of Liu and Haghani defined this framework in Utah by using ANN to study how adverse weather and crash events can alter TTR [37].

Among other studies focusing on TTR prediction by ANN, many others focused on predicting TRR by various architectures of ANN instead of defining conceptual frameworks. Amrutamanvar et al. focused on Indian urban roads and mentioned their developed model could capture various values of TTR [5]. Similar to the previous study, Afandizadeh et al. investigated the efficiency of multiple machine learning methods (including ANN) on Virginia's interstate network, however in their studies, ANN could not outperform other machine learning models [2,76]. Although they did not explicitly focus on the failure result of ANN, we believe it could be due to small dataset (ANNs usually require large datasets). Also, short-term TTR prediction by reliability interval was first addressed in the study of Wu et al. on corridor level in Texas [71].

Several novel architectures based on ANN have also been used as the main modeling approach in TTR prediction studies. Constrained Long Short-Term Memory (C-LSTM) quantile regression (LSTM is a type of Recurrent Neural Network (RNN), developed to address the vanishing gradient problem encountered by traditional RNNs) [36], a Generative Adversarial Network (GAN) with LSTM for studying TTR at network level by Shao et al. [56] and in peak periods [57]. Also, the Fitnet (a feedforward, fully connected neural network for regression) model [30] was among the studies that utilized ANNs and related architectures for predicting TRR.

Upon reviewing studies, it was revealed that although neural networks exhibit a strong capability to recognize complex patterns and nonlinear relationships, there is no study on TTR prediction applying Multilayer Perception (MLP).

MLP, a feedforward neural network with nonlinear activation functions with fully connected neurons, has been widely used in many aspects of transportation, including travel time prediction (as already stated, it is different from TTR prediction) [24], passenger flow prediction [68], accident prediction [55], lateral movement prediction [75] and many other topics.

To the best of authors' knowledge, in the field of TTR prediction, the Wild Horse Optimization (WHO) and Coot Optimization Algorithm (COA) have not been used yet. Furthermore, it should be also noted their rare application in the transportation context, such that the authors could not find any research utilizing COA and could find only one research on the capacitated vehicle routing problem [20] utilizing WHO.

The objective of this research is to design and evaluate a hybrid framework that integrates metaheuristic algorithms, specifically WHO and COA, to optimize the weights and biases of a MLP neural network for precise travel time prediction within Virginia's transportation network.

The specific objectives are as follows:

1. **Enhancing Prediction Accuracy:** Improving the MLP's performance in travel time prediction by minimizing errors through optimal tuning of weights and biases using metaheuristic algorithms.
2. **Avoiding Local Optima:** Leveraging the exploration and exploitation capabilities of WHO and COA to prevent entrapment in local optima, thereby enhancing the model's generalizability in Virginia.
3. **Comparative Algorithm Evaluation:** Conducting a comparative assessment of WHO and COA's performance in optimizing the MLP based on statistical metrics (R, MSE, and RMSE), regression and

Table 2
Sample data.

	Link Length (Mile)	TCr (Total Crash rate)	AADT Lane.mile	TTI (Unitless)
0.8389	5.33	0.166771	0.422139	1.00
1.1905	5.33	0.199593	0.440901	1.03
1.0971	5.33	0.382892	0.459662	1.00
0.9004	5.33	0.375235	0.469043	1.01
0.8389	5.06	0.087835	0.444664	1.04

overlay plots, and convergence curves to identify the most effective algorithm.

This research introduces several contributions by integrating the novel and previously unapplied metaheuristic algorithms WHO and COA with MLP neural networks for travel time prediction in Virginia, offering the following key innovations:

- 1. Pioneering Application of WHO and COA in Transportation:** Utilizing the innovative WHO and COA algorithms for optimizing MLP weights and biases in the context of travel time prediction, a novel approach that has not been previously explored in this domain.
- 2. Hybrid Framework for Real-World Data:** Developing a framework that combines real-world data with MLP and metaheuristic algorithms to deliver more accurate and practical predictions.

The remainder of this study is structured as follows:

Section 2 introduces the case study and details the dataset used in the research. **Section 3** provides an in-depth exploration of the fundamental principles governing the MLP, WHO, and COA, covering their mechanisms, applications, and the formulation of the optimization problem.

Section 4 offers a comprehensive analysis of the experimental results, evaluating performance through various metrics and plots, while highlighting key findings. **Section 5** concludes the study, summarizing the outcomes and proposing directions for future research.

2. Case study and data

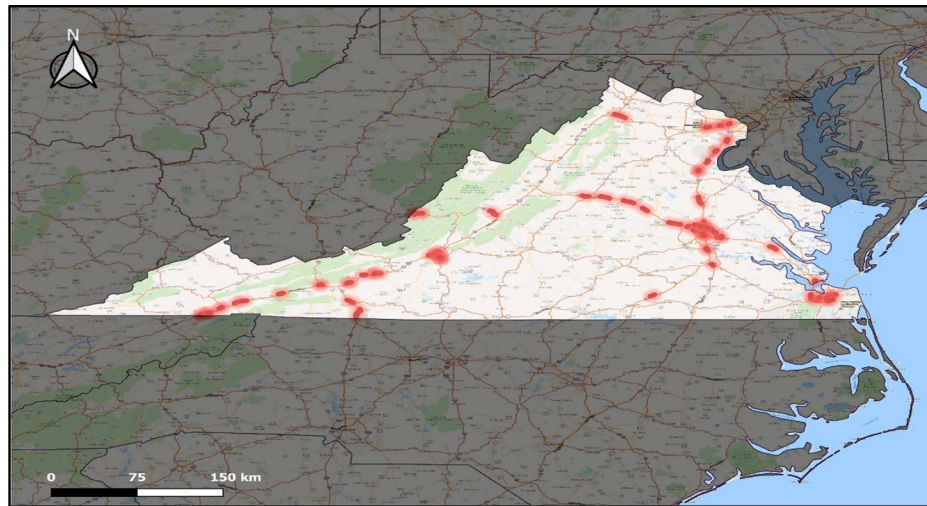
In this study, TTI is used to show the variability of travel time. Also, the study investigates the effect of four independent variables including traffic volume ($\frac{AADT}{Lane.mile}$), weather ($\frac{PRCP}{1000}$), total crash rate, TCr (by considering total number of crashes, AADT and the length of each link in the network) and section length (Link Length) on the existing reliability index. In these variables, AADT is the Annual Average Daily Traffic, Lane shows the number of lanes in the studied segment and PRCP represents total annual precipitation in mm for the specific segment.

All data were collected for 4 consecutive years, 2014 to 2017. The aim of this study is to present a model for predicting TTR using MLP and metaheuristic algorithms. TTR data, specifically TTI, was extracted from INRIX. For this study, 472 observations, calculated as 118 sections in 4 years, indicating one annual observation per section per year was collected.

Table 2 shows an example of data prepared for modeling.

Fig. 1 shows the spatial distribution of these segments across Virginia.

Table 3 presents the statistical details of the dataset, including the total number of observations, mean, standard deviation, minimum, maximum, and the 25th, 50th (median), and 75th percentiles for each feature. The dataset consists of 472 records and encompasses five traffic performance-related features: PRCP/1000 (normalized precipitation rate), Link Length, Total Rate, AADT/(Lane.mile) (average annual daily traffic per lane-mile), and TTI. The normalized precipitation rate has a

**Fig. 1.** Spatial Location of Segments in Virginia Interstate.**Table 3**
Statistical details of the data.

	PRCP 1000	Link Length	Total Rate	AADT Lane.mile	TTI
Count	472.000000	472.000000	472.000000	472.000000	472.000000
Mean	1.192785	2.763220	0.172488	6.922751	1.013962
Std	0.204408	1.639356	0.119722	5.206517	0.077206
Min	0.769700	0.660000	0.022046	0.371058	0.890000
25 %	1.042100	1.640000	0.091598	2.721088	0.980000
50 %	1.161950	2.325000	0.144107	5.431771	1.000000
75 %	1.325425	3.530000	0.214471	10.082861	1.010000
Max	2.044700	7.920000	0.866237	28.048780	1.670000

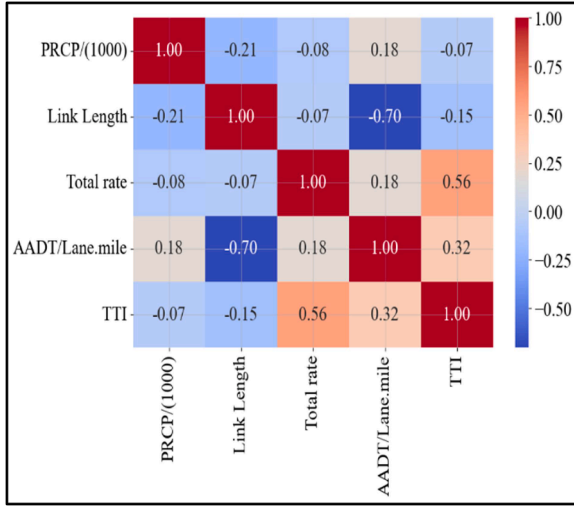


Fig. 2. Correlation Matrix.

mean of 1.1928 and a standard deviation of 0.2044, indicating relatively low variability, with values ranging from 0.7697 (minimum) to 2.0447 (maximum). Link Length averages 2.7632 miles, with a standard deviation of 1.6394, reflecting considerable variation in path lengths, ranging from 0.66 to 7.92 miles. The Total Rate, with a mean of 0.1725 and a standard deviation of 0.1197, exhibits a wide distribution, varying from 0.0220 to 0.8662. AADT/(Lane.mile) has a mean of 6.9228, and its high standard deviation of 5.2065 indicates significant variation in traffic density, with values ranging from 0.3711 to 28.0488. Finally, the TTI, with a mean of 1.0140 and a low standard deviation of 0.0772, suggests relative stability, with values ranging from 0.89 to 1.67. The 25th, 50th, and 75th percentiles for each feature indicate a balanced data distribution, offering valuable insights for analyzing traffic behavior.

2.1. Correlation matrix analysis

The correlation matrix (Fig. 2) examines the relationship between the dependent variable, TTI, and four independent variables: PRCP/1000 (precipitation), Link Length, Total Rate, and AADT/Lane. Mile. TTI shows a very weak negative correlation with PRCP/1000 (-0.07), suggesting that precipitation has negligible impact on travel time. Similarly, a weak negative correlation with Link Length (-0.15) indicates that longer routes may slightly reduce TTI, possibly due to differences in road design or traffic conditions. In contrast, TTI has a moderately strong positive correlation with Total Rate (0.56), highlighting that higher rates significantly increase travel time, making it a key predictor. TTI also exhibits a moderate positive correlation with AADT/Lane. Mile (0.32), implying that higher traffic volumes contribute to longer travel times, though less strongly than Total Rate. Notably, a strong negative correlation (-0.70) between Link Length and AADT/Lane. Mile suggests that longer routes tend to have lower traffic density.

3. Research methods

This section begins by describing the MLP architecture and tackling the optimization challenges of weights and biases in gradient-based methods, followed by an introduction to the structure of metaheuristic algorithms. It elaborates on the core mechanisms of metaheuristic algorithms, such as WHO and COA, within the MLP framework. Additionally, it outlines the evaluation metrics used to measure the model's performance, providing a solid foundation for analyzing the proposed approach.

3.1. Multilayer perceptron (MLP)

The MLP, a form of artificial neural network, falls under the feed-forward neural network class. It consists of processing units, known as neurons, which convert input data to produce the intended output [40].

In an MLP, neurons are organized into three main layers: The input layer, the first layer, consists of neurons that receive input data and transmit it to the subsequent layer, with their number typically matching the count of data features. The hidden layer, the next in sequence, contains neurons that perform mathematical transformations on the data. A MLP may include one or multiple hidden layers, depending on the complexity of the task. The output layer, the final layer, processes data from the hidden layer to produce the end result, with its neuron count determined by the encoding of the expected output.

The MLP functions in a hierarchical, fully connected structure, with each neuron in one layer linked to all neurons in the subsequent layer through weighted connections. For instance, every neuron in the input layer is connected to all neurons in the hidden layer.

Each connection between neurons has a specific weight, reflecting how strongly one neuron influences another. Additionally, hidden and output layer neurons incorporate a bias term (β)—a constant that fine-tunes predictions by shifting the neuron's activation threshold. This bias can either amplify (if positive) or suppress (if negative) the neuron's output, shaping its overall response [34,48]. The core objective of training an MLP is to determine the most effective combination of these weights and biases.

The MLP architecture includes three primary components: an input layer, a single hidden layer containing m neurons, and an output layer with one neuron. The connection weights between the input and hidden layers are defined as ω_{ij}^H , where $i = \{1, \dots, n\}$ and $j = \{1, \dots, m\}$ correspond to the respective neurons in each layer. Similarly, the weights linking the hidden layer to the output neuron are labeled $\omega_{j,1}^O$. Additionally, each neuron in the hidden and output layers has an associated bias term: β_k for the hidden layer and $k = \{1, \dots, m+1\}$ for the output neuron.

Neurons in the hidden layer perform computations through two sequential operations: weighted summation followed by activation. First, each neuron calculates its net input by summing the weighted outputs from all connected neurons in the preceding layer and adding its own bias term. This combined input for a given hidden neuron j is mathematically represented by (1).

$$Sum_j^H = \sum_{i=0}^n \omega_{ij}^H \times X_i + \beta_j \quad (1)$$

The term ω_{ij}^H corresponds to the connection weight between input neuron i and hidden neuron j . The output value from neuron i , denoted as X_i , serves as input for neuron j . Additionally, each hidden neuron incorporates a bias term β_j that adjusts its activation threshold.

The second processing stage applies a nonlinear transformation to the summed input through an activation function. Typically implemented as a sigmoid function, this crucial step introduces nonlinearity to the network's computations. For any given neuron, the activation output is mathematically defined by (2).

$$y_j^H = f(Sum_j^H) = \frac{1}{1 + e^{-Sum_j^H}} \quad (2)$$

The value y_j^H represents the computed output of hidden neuron j . This output can then serve as input to either another hidden layer or the final output layer. In the case of a single output neuron, the weighted summation is calculated using (3), where:

$$Sum_1^O = \sum_{j=0}^m \omega_{j,1}^O \times y_j^H + \beta_{m+1} \quad (3)$$

- y_j^H is the output from hidden neuron j ,
- $\omega_{j,1}^O$ denotes the connection weight between hidden neuron j and the output neuron, and
- β_{m+1} is the bias term applied to the output neuron.

The ultimate output of the MLP for a given dataset instance is derived from the activation operation of the output neuron, as outlined in (4).

$$y_1^O = f(\text{Sum}_1^O) = \frac{1}{1 + e^{-\text{Sum}_1^O}} \quad (4)$$

The primary objective in training an MLP is to determine the ideal set of connection weights and bias values for both hidden and output layers that collectively minimize the network's MSE. In ANNs, learning represents the adaptive process through which the system acquires predictive capabilities. For MLPs, this knowledge acquisition occurs during training - a cyclical procedure where the network progressively fine-tunes its weights and biases. These adjustments systematically reduce discrepancies between the network's predictions and target outputs, enhancing performance in fundamental tasks like classification and regression analysis.

The backpropagation algorithm, a widely adopted method for training MLPs [70], begins by initializing weights and biases with random values. The MLP processes a collection of labeled data (training set) to produce an output, and the discrepancy between this output and the target value is computed. This error is then propagated backward to refine the weights and biases. The process repeats until the error reaches an acceptable threshold [34].

While these traditional approaches are generally effective, they can encounter issues, such as extended periods of unchanging error levels or getting trapped in local optima. Their performance also depends heavily on the initial weight settings and the tuning of momentum and learning rate, where suboptimal choices may lead to divergence [41].

The performance of MLPs depends greatly on the precise tuning of weights and hyperparameters. Conventional gradient-based techniques, such as backpropagation, are widely used but often face challenges like becoming stuck in local optima or converging prematurely. To address these limitations, metaheuristic algorithms have emerged as powerful tools for optimizing MLP weights and hyperparameters. These algorithms provide global search abilities, operate non-deterministically, and function without gradients, offering greater flexibility than traditional approaches. They are particularly effective in solving complex, high-dimensional, nonlinear problems [15].

Recently, metaheuristic techniques have emerged as effective alternatives to backpropagation. These iterative algorithms efficiently produce high-quality solutions and are highly adaptable, requiring no specialized knowledge to tackle a wide range of problems [26,31,62]. Research demonstrates that metaheuristics excel in optimizing MLP models, effectively handling large sets of weights and biases [14,60].

3.2. Wild horse optimization (WHO)

The WHO algorithm is a swarm-based metaheuristic algorithm introduced in 2021 by Naruei and Keynia. It addresses optimization problems by mimicking the social structure and fourfold behaviors of wild horses. Each individual in the population is considered a horse, and its position in the search space represents a candidate. The modeling process includes 4 main steps:

1. Herding and Social Structure: The population is divided into several groups (herds), each led by a dominant male horse (Stallion). The remaining mares and foals are evenly distributed among the groups. The selection of the stallion (leader) is initially random and then optimized based on fitness.
2. Grazing Behavior: In this phase, members of each group (foals and mares) search for food around their group's stallion. This behavior

serves as a local exploitation mechanism, allowing the group to explore near the current best solution (represented by the stallion's position).

3. Mating Behavior: A unique behavior in wild horses involves foals separating from the main herd before sexual maturity to avoid mating with parents or siblings. In the algorithm, this mechanism is modeled as a crossover operator between horses from different groups, enabling the algorithm to escape local optima traps and explore new regions of the search space.
4. Group Leadership and Competition: The leaders of each group (stallions) guide their members toward the best available region, modeled as a waterhole. Stallions compete with each other to access this waterhole, which enhances the optimization process [45].

The mathematical formulation of the aforesaid steps can be followed in the next subsection.

3.2.1. Mathematical formulation of WHO

The WHO algorithm uses a set of mathematical equations to simulate the aforementioned behaviors. The next steps are from the following resources: [23,39,83,84]

The first step is to model grazing behavior. The position of group members (non-stallion horses) is updated based on the position of their group's stallion. This movement toward the stallion simulates an exploitative search.

$$X_{G,j}^i = 2Z \times \cos(2\pi R) \times (Stallion_{G,j} - X_{G,j}^i) + Stallion_{G,j} \quad (5)$$

In which, $X_{G,j}^i$ is the position of the i th horse in the j -th group at the current iteration, $Stallion_{G,j}$ is the position of the stallion in the j -th group, R is a random number between -2 and 2 and Z is an adaptive parameter.

The second stage is mating behavior. This phase, aimed at preventing premature convergence and increasing population diversity, is simulated using a mean crossover operator and performed with a fixed probability called PC (Crossover Percentage). The mathematical formulation is as follows:

$$X_{G,j}^P = \text{Crossover} \left(X_{G,i}^q, X_{G,j}^z \right), \text{Crossover} = \text{Mean where} \quad (6)$$

The third stage, group leadership and competition, is a stage where stallions lead their group members towards the best overall solution (WH, Waterhole).

$$\overline{Stallion}_{G,j} = \begin{cases} 2Z \times \cos(2\pi RZ) \times (WH - Stallion_{G,j}) + WH & \text{if } rand > 0.5 \\ 2Z \times \cos(2\pi RZ) \times (WH - Stallion_{G,j}) - WH & \text{if } rand \leq 0.5 \end{cases} \quad (7)$$

It should be mentioned that $\overline{Stallion}_{G,j}$ shows new candidate position for the stallion of the j th group, $Stallion_{G,j}$ is the current position of stallion and WH is the location of the water fountain, which is the global best solution.

The last stage, exchange and selection of leaders, ensures that leadership always rests with the best individual in the herd. If the fitness of a non-stallion horse is better than the current stallion, that horse will replace the stallion.

$$Stallion_{G,j} = \begin{cases} X_{G,j}^i & \text{if } f(X_{G,j}^i) < f(Stallion_{G,j}) \\ Stallion_{G,j} & \text{if } f(X_{G,j}^i) \geq f(Stallion_{G,j}) \end{cases} \quad (8)$$

3.3. Coot optimization algorithm (COA)

The Coot Optimization Algorithm (COA) is a nature-inspired metaheuristic algorithm based on the collective behavior of coot birds on water surfaces. It models key behaviors of these birds to address optimization problems. These behaviors include four main mechanisms that

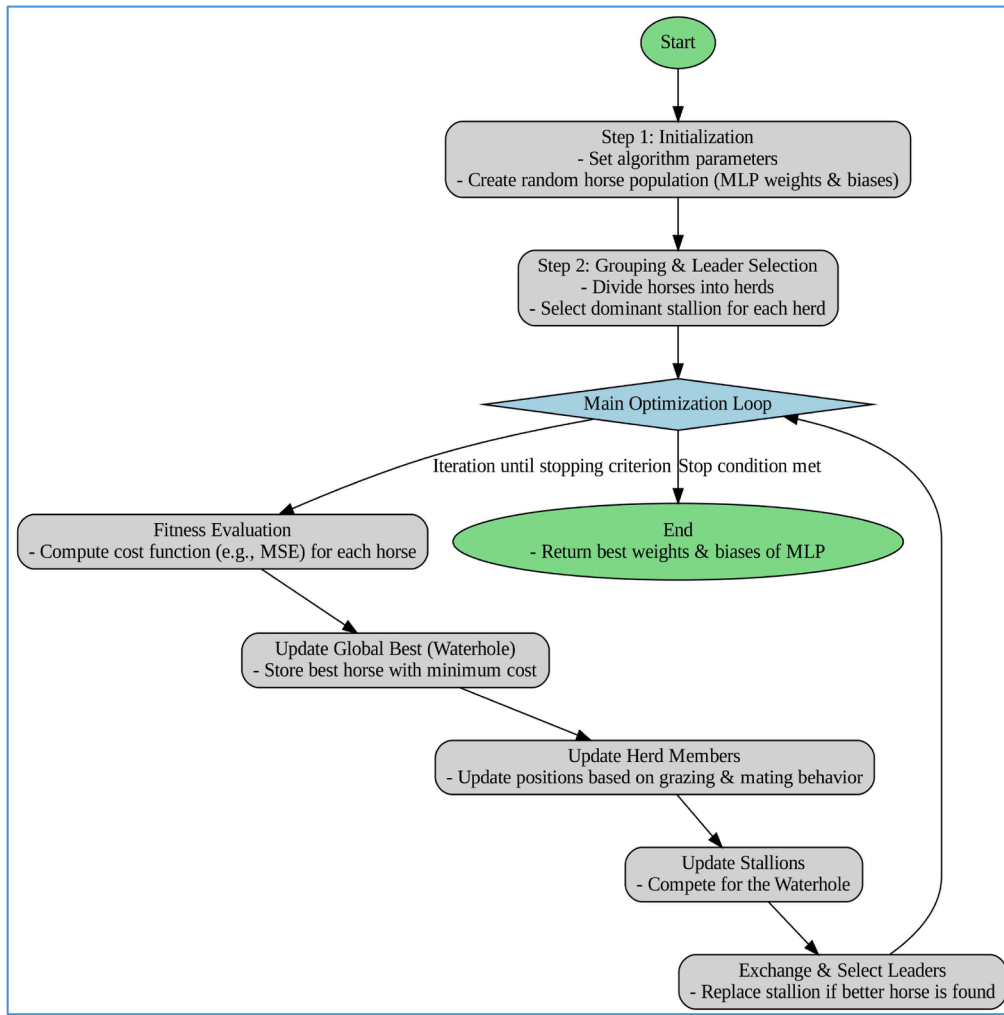


Fig. 3. Flowchart of WHO in Optimizing MLP.

enable the algorithm to perform both exploration (broad search) and exploitation (local search).

The first stage is random movement. Coots move randomly on the water surface to search for and discover new food resources. This behavior enhances the algorithm's exploration process, allowing it to escape local optima traps. After this step, chain movements happen. Coots often move in a chain-like manner, following one another. This behavior is modeled using the average position of two coots, helping the algorithm move toward higher-quality regions. Based on this, adjustment of position happens. This stage refers to fact that each coot adjusts its position relative to its group leader and moves toward them. This behavior facilitates exploitative search around the best local solutions (represented by the leaders). Finally, group leaders, representing the best local solutions, move toward the global best solution. This mechanism ensures the convergence of the entire population toward the global optimum [44]. The mathematical formulation of the aforesaid steps can be followed in the next subsection.

3.3.1. Mathematical formulation of COA

The following equations will be used for simulating the aforesaid behaviors. The following description is taken from these references: [25, 44,52].

First, the population of coots is randomly initialized within the upper and lower bounds of the search space.

$$Cootpos(i) = rand(1, d) \times (ub - lb) + lb \quad (9)$$

In the equation, $cootpos(i)$ is the position of i th coot, d is the dimension of problem and ub and lb represent the upper and lower bounds of search space. Following previous equation, the position of each coot is updated using random movement.

$$Cootpos(i)_{new} = Cootpos(i)_{current} + A \times R2 \times (Q - Cootpos(i)_{current}) \quad (10)$$

$Cootpos(i)_{new}$ is the updated position of i th coot, Q is a randomly generated position within the search range and A is a decreasing parameter that reduces with iterations, balancing exploration and exploitation.

The second stage of formulation is chain movement modeling. The position of each coot is updated using the average of its position and the previous coot in the chain.

$$Cootpos(i) = 0.5 \times [Cootpos(i-1) + Cootpos(i)] \quad (11)$$

Then, adjustment of position happens. Every coot updates its position based on the position of its group leader.

$$Cootpos(i) = leaderpos(k) + 2 \times R1 \times \cos(2\pi R) \times (leaderpos(k) - Cootpos(i)) \quad (12)$$

In the above equation, the R and $R1$ are random numbers. Finally, group leaders move toward the global best solution (gBest) to accelerate the convergence process. In this equation, $R3$ and $R4$ are random numbers and B is a descending parameter that decreases with repetitions.

$$leaderpos(i) = \begin{cases} B \times R3 \times \cos(2\pi R) \times (gBest - leaderpos(i) + gBest) & \text{if } R4 > 0.5 \\ B \times R3 \times \cos(2\pi R) \times (gBest - leaderpos(i) - gBest) & \text{if } R4 \leq 0.5 \end{cases} \quad (13)$$

3.4. Formulating MLP weights and biases as an optimization problem

Optimizing weights and biases in MLP networks for regression problems faces challenges such as getting trapped in local minima, sensitivity to initial values, and slow convergence. Gradient-based methods, like gradient descent, may exhibit unstable performance due to the non-linear nature of the cost function and data noise, particularly in high-dimensional regression tasks. Metaheuristic algorithms like WHO and COA address these issues by exploring diverse weight and bias configurations to find optimal solutions. This process is executed in nine steps:

1. Initialize the MLP Structure:

- Define the architecture of the MLP, including the number of input neurons (based on data features), hidden layers and neurons (based on problem complexity), and output neurons (based on the expected output encoding).

2. Generate Initial Population:

- Create a randomly initial set of weight and bias configurations for the MLP.
- Each of weight and bias sets represents a candidate solution (e.g., a horse in WHO or a bird in COA).

3. Apply Weights and Biases to MLP:

- Assign each candidate weight and bias sets to the connections between the inputs, hidden, and output layers of the MLP.

4. Evaluate Weight and Bias Sets:

- Train the MLP using the training dataset with each of weight and bias sets.
- Compute the objective function, such as MSE, to measure the prediction accuracy of each weight and bias sets based on the MLP's performance on the training data.

5. Run Metaheuristic Optimization:

- Apply the WHO or COA algorithm to iteratively update the weight and bias sets:
 - Exploration Phase:** Broadly search the weight and bias spaces to identify promising regions for new solutions.
 - Exploitation Phase:** Refine the current weight and bias sets to improve solution quality.

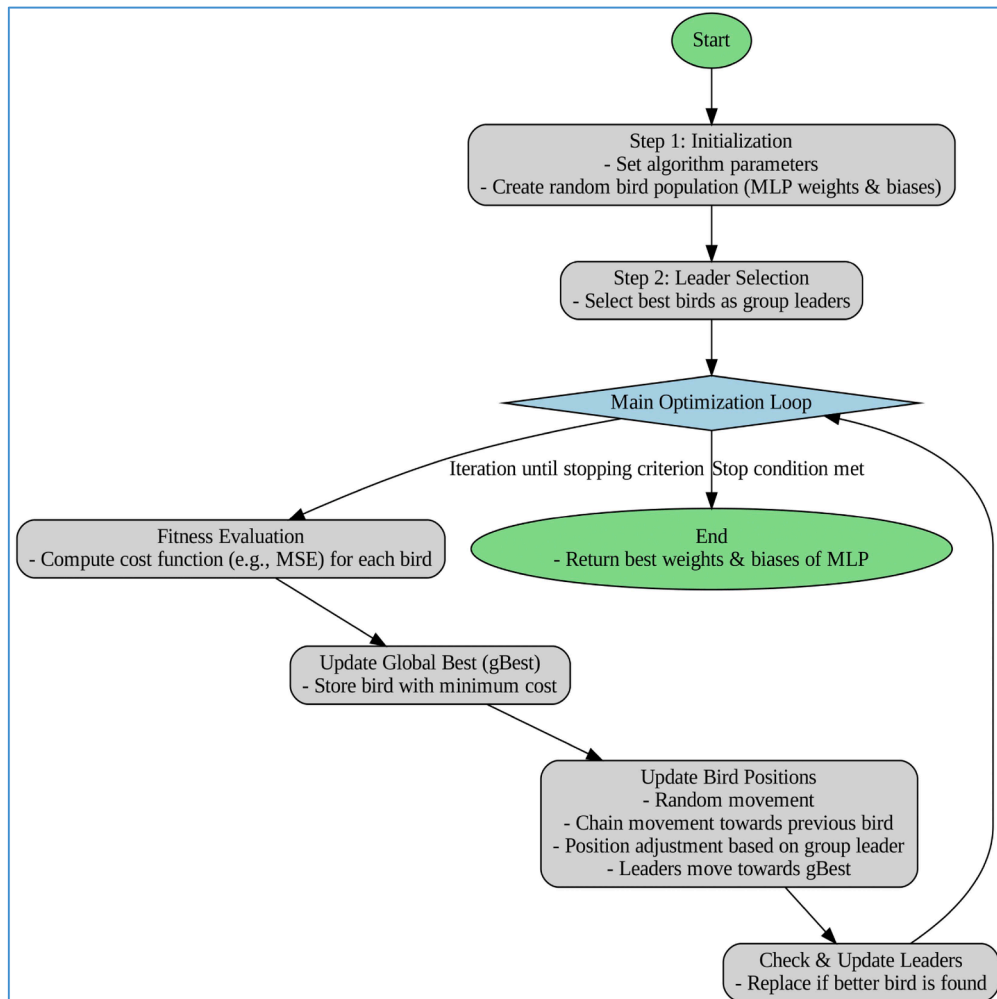


Fig. 4. Flowchart of COA in Optimizing MLP.

Table 4
Selected main parameters and values.

Common Parameter	Description	Selected Value
Max_Iter	Number of Iterations of Algorithm	400
N	Population Size	40
Objective Function	Error Function	MSE

Table 5
Selected Parameters for the MLP Architecture.

Parameter	Value
Number of Data Samples	472
Training Size	70 %
Testing Size	30 %
Number of Hidden Layer Neurons	12
Number of Hidden Layers	1
Number of Input Layer Neurons	4
Number of Output Neurons	1
Hidden Layer Activation Function	Tansig
Output Layer Activation Function	Purelin

Table 6
Software and Hardware Specifications.

CPU	AMD EPYC 7763 64-Core Processor
RAM	32 GB
Operating System	Windows 10 Pro
Modeling Software	MATLAB 2023

- Update the weights and biases based on the specific mechanisms of WHO (e.g., herd movement) or COA (e.g., coot flock behavior).
- 6. Iterate Until Convergence:**
 - Repeat steps 3–5 for a predefined number of iterations or until the objective function (e.g., MSE) converges to a satisfactory minimum, indicating optimized weights and biases.
- 7. Select Optimal Weights and Biases:**
 - Identify the weight and bias sets with the lowest error (best objective function value) as the final solution.
- 8. Test:**
 - Apply the optimized weights and biases to the MLP and evaluate its performance on a separate test dataset to confirm prediction accuracy and generalization.
- 9. Analyze Results:**
 - Compare the performance of WHO and COA in terms of convergence speed, error reduction, and MLP prediction accuracy to assess their effectiveness in optimizing.

To better illustrate how the two aforesaid algorithms, WHO and COA can be utilized in the case of this paper, two figures, Figs. 3 and 4 are presented.

3.5. Models configuration and optimization settings

In this sub section, travel time data modeling was performed based on the methodology outlined in the previous section. Travel time prediction is a critical issue in traffic management, intelligent transportation systems, and route planning, requiring accurate and reliable models.

Neural networks, such as the MLP, are well-suited for this task due to their ability to model nonlinear relationships between variables. However, the performance of these networks heavily depends on optimal weights and biases. Weights and biases in a neural network determine the influence of each input on the output (predicted travel time).

Optimizing weights and biases by minimizing the error function enhances prediction accuracy and prevents issues such as overfitting or poor generalization to new data. Metaheuristic algorithms, such as the

WHO and COA offer significant advantages over traditional methods (e.g., gradient descent) in optimizing neural network for travel time prediction.

First, these algorithms do not require computing the derivative of the objective function, which is highly beneficial in complex problems like travel time prediction, where the error function may be non-differentiable due to multiple nonlinear variables. Second, metaheuristic algorithms excel at escaping local optima, which is crucial in travel time prediction given the diverse patterns and noise. Gradient-based methods may get trapped in local optima, failing to identify optimal weights and biases for accurate travel time prediction. Third, these algorithms can simultaneously explore multiple weight/bias sets through parallel search, which is highly efficient for optimizing in high-dimensional data. Finally, the flexibility of these algorithms in adjusting parameters (e.g., population size or iteration count) enables adaptation to various travel time prediction scenarios.

In the WHO and COA, the number of iterations and the initial population size played a crucial role in balancing accuracy and computational efficiency. The initial population in both algorithms consisted of 40 solutions (horses in WHO and coot birds in COA), randomly generated within the search space, and with 400 iterations, optimal exploration and exploitation were ensured for complex problems.

Table 4 shows the values selected to the main parameters of each algorithm for the study.

To model a system with four independent variables and one dependent variable, the MLP was configured with four neurons in the input layer and one neuron in the output layer. A single hidden layer was adopted, as it typically provides adequate accuracy for most nonlinear problems while maintaining computational efficiency.

The number of neurons in the hidden layer was determined through careful empirical analysis to strike a balance between model complexity and performance. Increasing the number of hidden neurons could lead to overfitting and increased sensitivity to noise in the dataset, resulting in an overly complex network. Conversely, having too few neurons might hinder the network's ability to learn effectively. Therefore, the number of hidden neurons was iteratively adjusted to minimize the difference between actual and predicted outputs while achieving the target error level. Through trial and error and evaluation of results, the number of neurons selected for the neural network was set to 12.

Transfer functions were employed to map input signals to outputs: the hyperbolic tangent (tanh) function was selected for the hidden layer due to its effectiveness in capturing nonlinear relationships, while a linear function (Purelin) was applied in the output layer to ensure direct correspondence with the dependent variable.

Tanh, an S-shaped nonlinear function, maps inputs to values between -1 and 1 , making it suitable for hidden layers as it centers data and enables stronger gradients during backpropagation. In contrast, purelin outputs the input directly without transformation ($f(x) = x$). While tanh introduces nonlinearity for complex pattern learning, purelin is typically used in output layers for regression tasks where unbounded, continuous outputs are needed.

Table 5 provides the architectural details of the neural network.

In the metaheuristic approaches of WHO and COA, the calculation of weights and biases for optimizing a neural network with four input nodes, one hidden layer with 12 neurons, and a single output node is performed iteratively using metaheuristic search: the weights from the input to the hidden layer ($4 \times 12 = 48$), weights from the hidden layer to the output ($12 \times 1 = 12$), biases for the 12 hidden neurons, and one bias for the output neuron, totaling 73 parameters per individual in a population of 40, resulting in 2920 parameters per iteration.

In WHO, these parameters are refined through the dynamic behavior of wild horse groups, where individuals within subgroups gravitate toward the group leader (the individual with the lowest MSE) while inter-group migrations introduce diversity, striking a balance between exploitation (updating based on the best solution) and exploration (random perturbations).

Table 7
Modeling Results for COA-MLP and WHO-MLP.

Algorithm	Metric	Train Set	Test Set	All Data
COA	MSE	0.0028	0.0029	0.0028
	RMSE	0.0530	0.0541	0.0533
	R (%)	0.7429	0.7213	0.7262
	Time (Seconds)	160.7993		
WHO	MSE	0.0026	0.0025	0.0026
	RMSE	0.0515	0.0502	0.0511
	R (%)	0.7592	0.7628	0.7516
	Time (Seconds)	147.4705		

Conversely, in COA, parameter improvement is driven by the coordinated movements of coot birds, encompassing chain-like movements toward the leader and random diving-like searches, which enhance exploration of the search space. In both algorithms, the best individual, comprising 73 parameters, is carried forward to the next generation, and the entire population's 2920 parameters are updated using fitness-driven strategies (based on MSE), ensuring gradual convergence toward optimal weights and biases that minimize error.

In this study, the experiments were conducted using the software and hardware specifications outlined in Table 6.

3.6. Evaluation metrics

To thoroughly assess the forecasting model's accuracy, three key metrics were employed. These metrics provide a detailed assessment of the model's predictive capability from various perspectives, with their respective formulas presented (Eqs. (14) to 16).

In these formulas, y_{act} represents the actual value of the variable, y_{pre} denotes the predicted value, $\bar{y}_{act}$ is the mean of the actual values, $\bar{y}_{pre}$ is the mean of the predicted values, and n indicates the number of collected data points.

1. **R (Correlation Coefficient):** The correlation coefficient is calculated using a specific formula. It ranges from +1 (indicating perfect positive correlation) to -1 (indicating perfect negative correlation). This measure is determined by assessing the degree of association between two variables relative to their maximum possible impact. Known as the Pearson correlation, this coefficient is a widely used metric in data analysis.

$$R = \frac{\frac{1}{n} \sum_{i=1}^n (y_{act} - \bar{y}_{act})(y_{pre} - \bar{y}_{pre})}{\sqrt{\frac{1}{n} \sum_{i=1}^n (y_{act} - \bar{y}_{act})^2} \sqrt{\frac{1}{n} \sum_{i=1}^n (y_{pre} - \bar{y}_{pre})^2}} \quad (14)$$

2. **MSE (Mean Squared Error):** Calculates the average of squared differences between predicted and actual values. By squaring errors, MSE ensures positive and negative deviations do not cancel out, while heavily penalizing larger errors. Its values range from 0 (perfect fit) to infinity, with units in squared data terms.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_{pre} - y_{act})^2 \quad (15)$$

3. **RMSE (Root Mean Square Error):** Represents the square root of MSE, converting error values back to the original data units. Like MSE, lower RMSE values indicate better accuracy, but its unit consistency makes errors more interpretable for practical applications.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_{pre} - y_{act})^2} \quad (16)$$

4. Numerical results and discussion

This section first introduces the parameters and configurations considered for the MLP, WHO and COA, followed by a detailed analysis of the results for each model individually, comparing their performance using various evaluation metrics.

4.1. Analysis results

This section focuses on analyzing the results of the models, providing a detailed evaluation of their performance.

4.1.1. R, MSE and RMSE metrics

Table 7 shows the results obtained from the hybrid models. The WHO demonstrated outstanding performance in optimizing neural network for travel time prediction in this study. The MSE for the train, test, and all datasets was reported as 0.0026, 0.0025, and 0.0026, respectively. These low values indicate high model accuracy in travel time prediction and the algorithm's ability to find weights and biases that minimize prediction error. The slight reduction in MSE for the test data (0.0025) compared to the train data (0.0026) suggests desirable model generalization, which is critical for predicting travel times. The RMSE for the train, test, and all datasets was calculated as 0.0515, 0.0502, and 0.0511, respectively, confirming minimal deviation of predictions from

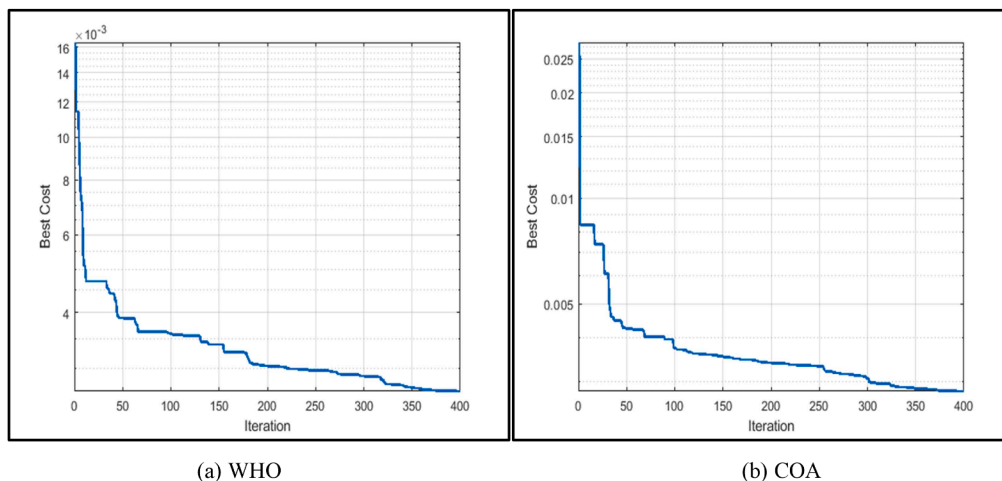


Fig. 5. Convergence Curves.

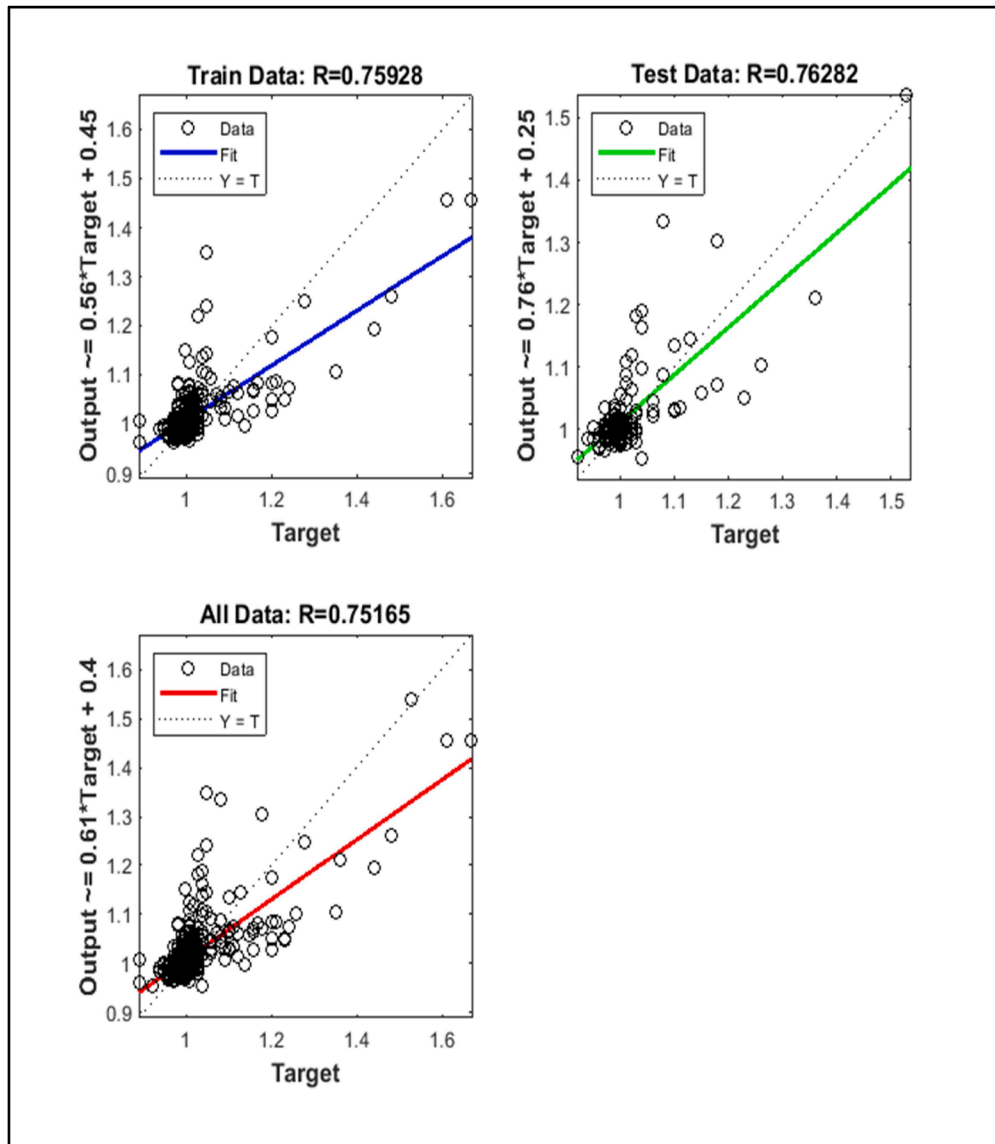


Fig. 6. The WHO Regression Plots.

actual travel times. The R for the train, test, and all datasets was 0.7592, 0.7628, and 0.7516, respectively, indicating that the model explains 75.92 %, 76.28 %, and 75.16 % of the variance in travel times. These values reflect good alignment for travel time prediction, though an R below 0.8 suggests potential for improvement.

The COA also delivered acceptable performance in travel time prediction but performed slightly worse than WHO. The MSE for the train, test, and all datasets was 0.0028, 0.0029, and 0.0028, respectively, indicating higher error compared to WHO. This difference may lead to less accurate travel time predictions, especially for test data (e.g., new routes or varying traffic conditions). The RMSE values for the train, test, and all datasets were 0.0530, 0.0541, and 0.0533, respectively, showing greater deviation than WHO. The R for the train and test datasets in COA was 0.7429 and 0.7213, respectively, demonstrating less ability to explain travel time variance compared to WHO.

In terms of computational efficiency, WHO outperformed COA with a computational time of 147 s compared to 160 s for COA. This difference, despite identical iteration (400) and population sizes (40), highlights WHO's superior efficiency. In travel time prediction applications, particularly real-time systems like dynamic route planning, lower computational time is crucial, as users expect rapid responses for

decision-making.

4.1.2. Convergence curves

The convergence results (Fig. 5) of the WHO and COA methods are compared based on key metrics such as final best cost, convergence speed, stability, and overall efficiency. The WHO method achieves a superior final best cost of 0.0026582 at iteration 401, approximately 5.7 % lower than COA's final cost of 0.0028187, indicating WHO's better precision in reaching an optimal solution. In terms of convergence speed, WHO demonstrates faster initial progress, reducing the cost to 0.0053931 by iteration 10 and 0.0038909 by iteration 50, while COA starts slower, maintaining a cost of 0.008405 until iteration 15 but showing significant improvements in the mid-range (iterations 50–150), reaching 0.0037373 by iteration 100. However, WHO regains the lead in later iterations with steady improvements. Regarding stability, WHO exhibits longer periods of cost stagnation (e.g., iterations 13–29 at 0.0047146), suggesting occasional entrapment in local optima, but maintains smoother transitions overall. COA, conversely, shows larger cost jumps (e.g., from 0.0073811 to 0.0060842 at iteration 28), indicating better escape from local optima but with more fluctuations in later stages. Overall, WHO is more suitable for problems requiring high

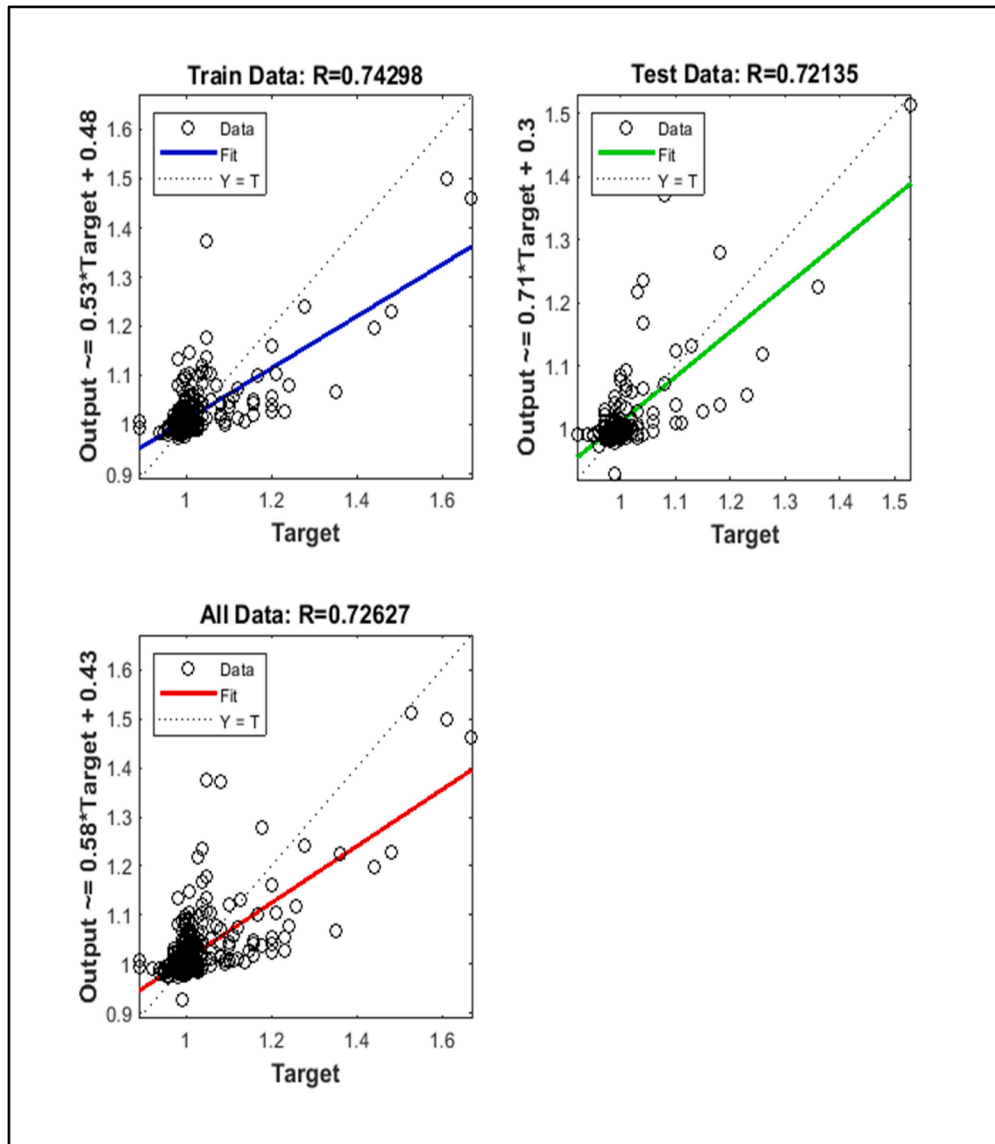


Fig. 7. The COA Regression Plots.

precision due to its lower final cost and stable improvements, while COA excels in mid-range exploration and escaping local optima, making it preferable for problems needing broader search capabilities.

4.1.3. Regression plots

The regression analysis of the Figs. 6 and 7, illustrates the model's performance across different datasets. The top-left plot represents the train data, where a high degree of alignment between the model's predictions and actual values is evident. This is quantified by a regression coefficient (R) of approximately 0.76 for the model optimized using the WHO, indicating robust training accuracy. In contrast, the model optimized with the COA achieves a train accuracy with an R value of approximately 0.74, demonstrating that WHO outperforms COA. The R values for the test and all datasets, shown in the remaining plots, are nearly identical to those of the train data for the WHO-optimized model, suggesting consistent performance across all data subsets. With close approximation, the network's predictions align well with the target outputs, supporting the hypothesis that the neural network, particularly when optimized with the WHO, is an effective model for travel time prediction.

4.1.4. Overlay plots

The overlay between the neural network model's predictions and actual values was analyzed. The neural network model, optimized using the WHO, accurately captures the trends across the test, and all datasets.

In Figs. 8 and 9, the black curve represents the target or actual values and the red curve represents the predicted or output values for the WHO and COA, respectively. The minimal discrepancy between target and output values, reflects low prediction error and highlights the model's high accuracy and robustness. Notably, the WHO demonstrates superior performance and closer alignment with actual values compared to the COA in both the test and all datasets, underscoring the effectiveness of WHO in optimizing the neural network for time travel prediction.

4.1.5. Comparative performance analysis of metaheuristic algorithms

In this section, to evaluate the effectiveness of the models presented in this study, the results are compared with several other algorithms. The selected algorithms are Ant Colony Optimization (ACO) [10,50], Genetic Algorithm (GA) [29], and Particle Swarm Optimization (PSO) [54]. ACO mimics the behavior of a colony of real foraging ants to find the most cost-effective path. The shortest or most optimal path is discovered through the stigmergy process. This is a social network

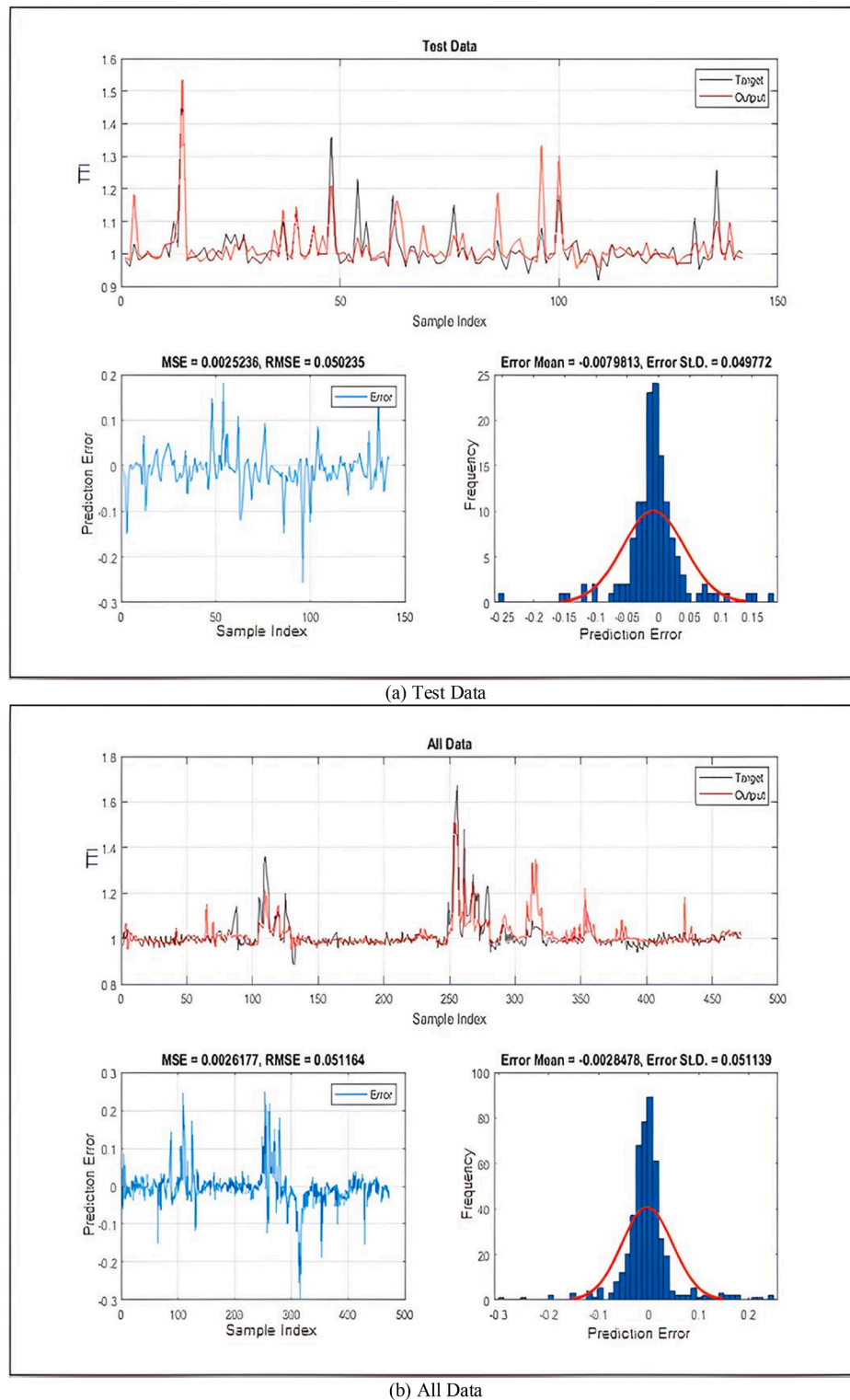


Fig. 8. WHO Overlay Plots.

mechanism in which pheromones guide agents toward promising solutions. ACO is designed to address the most challenging combinatorial optimization problems as well as network applications, such as routing and load balancing. GA is a search algorithm based on the concepts of natural selection and genetics (crossover and mutation). PSO mimics the collective behavior of birds or fish searching for food. In PSO, each particle in a swarm does not exchange materials with other particles. A particle is influenced by its current position, the best position in the

swarm, and its velocity.

The results of Table 8 show a comparative evaluation of the outcomes of these five metaheuristic algorithms. WHO exhibits the best performance across all metrics, achieving the lowest MSE (0.0026 on all data) and RMSE (0.0511 on all data) while maintaining the highest correlation coefficient ($R = 0.7516$ on all data). COA follows closely, with slightly higher errors (MSE = 0.0028, RMSE = 0.0533 on all data) and a marginally lower R (0.7262 on all data). In contrast, ACO, PSO,

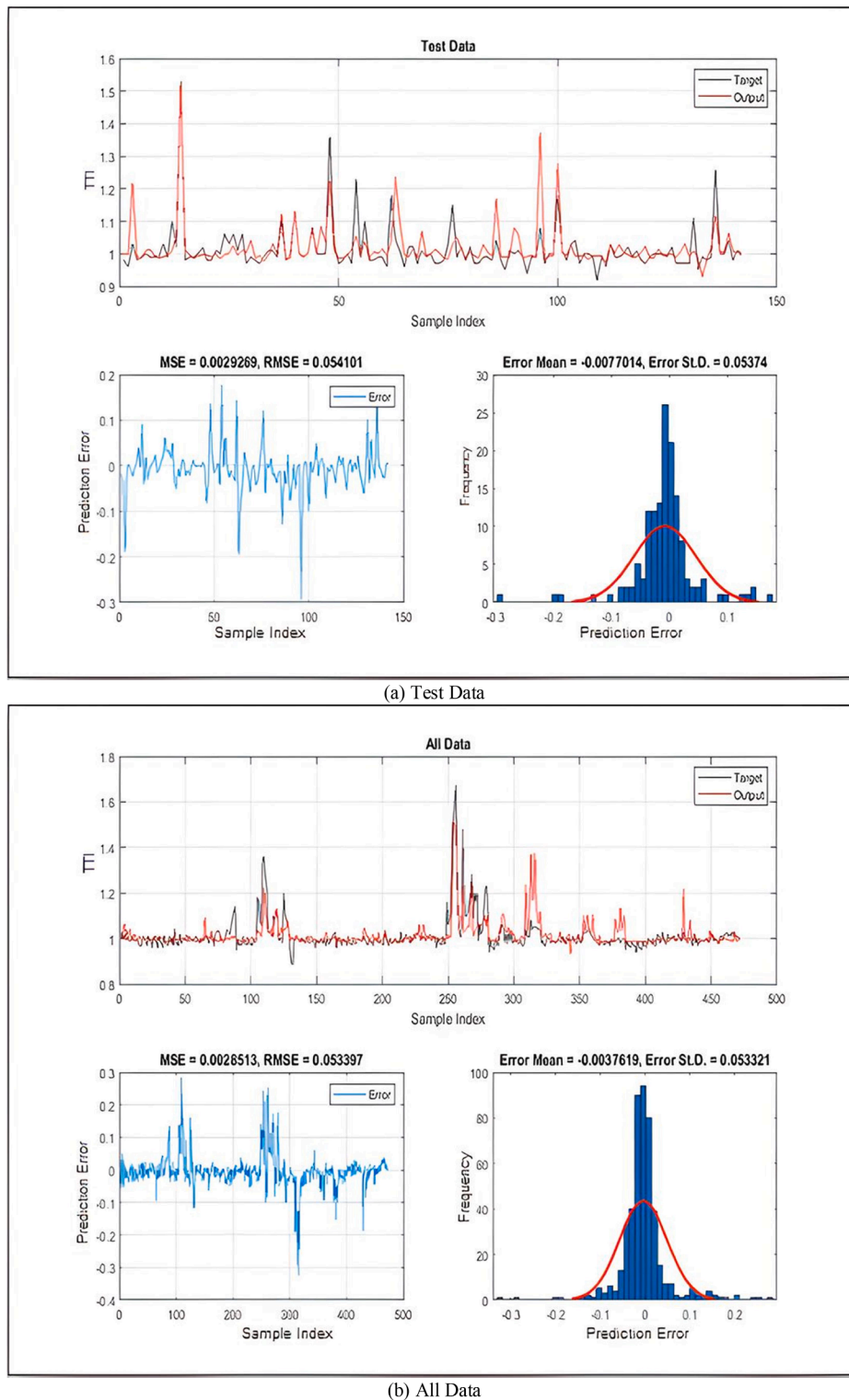


Fig. 9. COA Overlay Plots.

and GA show noticeably weaker performance. PSO performs better than ACO and GA, with an MSE of 0.0035 and RMSE of 0.0593 on all data, and an R of 0.68, but it still lags behind COA and WHO. ACO and GA exhibit the poorest results, with GA having the highest errors (MSE = 0.0039, RMSE = 0.0626 on all data) and the lowest correlation (R = 0.65 on all data), followed closely by ACO (MSE = 0.0037, RMSE = 0.061, R = 0.66 on all data). Across train, test, and full datasets, WHO shows better optimization, likely because it explores the solution space more

efficiently, while GA and ACO converge slower and tune parameters less effectively for this MLP.

The execution times shown in Fig. 10 provide a comparative analysis of five metaheuristic algorithms. WHO demonstrates the fastest performance with an execution time of 147.47 s, indicating its efficiency in navigating the solution space. COA follows closely with a time of 160.79 s, still performing well but slightly slower than WHO, likely due to differences in their search mechanisms. PSO, with an execution time of

Table 8
Comparison of MLP Optimization Performance Using Metaheuristic Algorithms.

Algorithm	Metric	Train Set	Test Set	All Data
COA	MSE	0.0028	0.0029	0.0028
	RMSE	0.053	0.0541	0.0533
	R (%)	0.7429	0.7213	0.7262
WHO	MSE	0.0026	0.0025	0.0026
	RMSE	0.0515	0.0502	0.0511
	R (%)	0.7592	0.7628	0.7516
ACO	MSE	0.0037	0.0038	0.0037
	RMSE	0.0608	0.0616	0.061
	R (%)	0.67	0.65	0.66
PSO	MSE	0.0035	0.0036	0.0035
	RMSE	0.0591	0.06	0.0593
	R (%)	0.69	0.67	0.68
GA	MSE	0.0039	0.004	0.0039
	RMSE	0.0624	0.0632	0.0626
	R (%)	0.66	0.64	0.65

175.8 s, is noticeably slower than both WHO and COA, reflecting its reliance on iterative particle updates, which introduces additional computational overhead. ACO and GA exhibit the longest execution times at 195.7 and 200.4 s, respectively. The slower performance of ACO can be attributed to its complex pheromone-based search process, while GA's extended runtime stems from its computationally intensive operations, such as selection, crossover, and mutation. These results highlight WHO's superior efficiency for this specific MLP optimization task, while ACO and GA are hindered by their higher computational complexity.

5. Conclusions and future studies

Accurate travel time prediction is essential for ITSs. Traditional gradient-based methods, such as backpropagation, often get trapped in local optima, limiting their performance. Metaheuristic algorithms offer a robust alternative by exploring global solution spaces efficiently without derivatives. This study evaluated five nature-inspired algorithms—WHO, COA, ACO, PSO, and GA—for optimizing the weights and biases of an MLP to predict the TTI. The MLP model used four inputs (traffic volume, weather, crash rate, and section length), one hidden layer with 12 neurons, and one output neuron. It was trained on 472 observations from Virginia's transportation network (2014–2017). Weights and biases are encoded into vectors as candidate solutions, with fitness evaluated by Mean Squared Error (MSE). The metaheuristic algorithms iteratively minimize MSE, refining the MLP's parameters and avoiding local optima. Results highlighted WHO's superior performance, achieving an MSE of 0.0026, RMSE of 0.0511, and R of 0.7516, alongside a computational time of 147 s. COA followed with slightly weaker performance, recording an MSE of 0.0028, RMSE of 0.0533, R of 0.7262, and a computational time of 160 s. In contrast, ACO, PSO, and

GA demonstrated notably poorer performance. PSO achieved an MSE of 0.0035, RMSE of 0.0593, and R of 0.68, with a computational time of 175.8 s, indicating moderate performance but slower convergence compared to WHO and COA. ACO performed worse, with an MSE of 0.0037, RMSE of 0.061, R of 0.66, and a longer computational time of 195.7 s, reflecting its higher computational complexity. GA exhibited the weakest results, with an MSE of 0.0039, RMSE of 0.0626, R of 0.65, and the highest computational time of 200.4 s, likely due to its computationally intensive operations like crossover and mutation. WHO's efficiency and higher accuracy make it particularly well-suited for real-time applications, such as intelligent navigation systems and urban traffic management, where precise travel time predictions under varying conditions are critical. COA, while effective, shows slower convergence and occasional plateaus, which may reduce reliability in complex scenarios. ACO, PSO, and GA, with their higher errors and longer computational times, are less suitable for such demanding applications due to their limited optimization efficiency. To enhance the performance of these models, several strategies are proposed. Optimizing algorithmic settings, such as adjusting population size or iteration count, can enhance the balance between accuracy and computational speed for time-sensitive applications. Improving neural network design by increasing the number of neurons or incorporating advanced activation functions can boost predictive accuracy. Additionally, developing hybrid models by combining metaheuristic algorithms like WHO or COA with methods such as ACO, PSO, GA, or the Red Deer Algorithm (RDA) [51] can improve performance. Furthermore, incorporating permutation feature importance [42] can be explored to analyze the relative importance of features, enabling better feature selection and model refinement for improved predictive outcomes.

Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

CRediT authorship contribution statement

Navid Khorshidi: Writing – original draft, Methodology, Formal analysis, Conceptualization. **Soheil Rezashoar:** Writing – original draft, Software, Investigation, Formal analysis. **Pegah Amini:** Writing – original draft, Investigation, Funding acquisition, Formal analysis, Data curation. **Shahriar Afandizadeh Zargari:** Writing – review & editing, Project administration, Conceptualization. **Hamid Mirzahosseini:** Writing – review & editing, Supervision, Project administration, Methodology, Conceptualization.

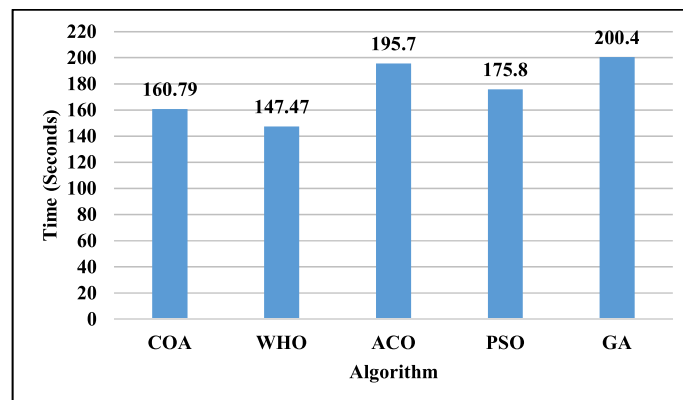


Fig. 10. Execution Time Comparison of Metaheuristic Algorithms for MLP Optimization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors will make available, upon reasonable request, the data and code that support the findings of this study.

References

- [1] M.A. Abdel-Aty, R. Kitamura, P.P. Jovanis, Investigating effect of travel time variability on route choice using repeated-measurement stated preference data, *Transp. Res. Rec.* (1493) (1995) 39–45. <https://stars.library.ucf.edu/scopus/1990/2064>.
- [2] S. Afandizadeh Zargari, N. Amoei Khorshidi, H. Mirzahassein, N. Kalantari, Comparative approach for predicting travel time reliability (a case study of Virginia interstate), *Innov. Infrastruct. Solut.* 6 (4) (2021) 229, <https://doi.org/10.1007/s41062-021-00597-8>.
- [3] Y. Ai, Y. Yu, W. Pu, L. Gao, Y. Ren, Network-level travel time prediction considering the effects of weather and seasonality, in: *International Conference on Transportation and Development* 2023, 2023.
- [4] R. Al-Rubae, N. Al-Etahi, J. Alsaif, S. Al-Fadhili, Effects of checkpoints on urban travel time, in: *IOP Conference Series: Materials Science and Engineering*, 2018.
- [5] R. Amrutamanvar, G. Joshi, S.S. Arkatkar, R.S. Chalumuri, Empirical travel time reliability assessment of Indian urban roads, in: *Recent Advances in Traffic Engineering: Select Proceedings of RATE 2018*, 2020.
- [6] S. Banik, L. Vanajakshi, Impact of rainfall on traffic mobility and reliability under Indian traffic conditions, *Transp. dev. econ.* 10 (2) (2024) 29, <https://doi.org/10.1007/s40890-024-00219-9>.
- [7] A. Bennecke, B. Friedrich, M. Friedrich, J. Lohmiller, Time-dependent service quality of network sections, *Procedia-Soc. Behav. Sci.* 16 (2011) 364–373, <https://doi.org/10.1016/j.sbspro.2011.04.457>.
- [8] H. Bi, Z. Ye, H. Zhu, Data-driven analysis of weather impacts on urban traffic conditions at the city level, *Urban. Clim.* 41 (2022) 101065, <https://doi.org/10.1016/j.uclim.2021.101065>.
- [9] D. Billings, J.-S. Yang, Application of the ARIMA models to urban roadway travel time prediction—a case study, in: *2006 IEEE International Conference on Systems, Man and Cybernetics*, 2006.
- [10] X. Chen, L. Yu, T. Wang, A. Liu, X. Wu, B. Zhang, Z. Lv, Z. Sun, Artificial intelligence-empowered path selection: a survey of ant colony optimization for static and mobile sensor networks, *IEE Access* 8 (2020) 71497–71511, <https://doi.org/10.1109/ACCESS.2020.2984329>.
- [11] Z. Chen, W. Fan, Data analytics approach for travel time reliability pattern analysis and prediction, *J. Mod. Transp.* 27 (4) (2019) 250–265, <https://doi.org/10.1007/s40534-019-00195-6>.
- [12] F. Cirianni, G. Leonardi, Artificial neural network for traffic noise modelling, *ARNP J. Eng. Appl. Sci.* 10 (22) (2015) 10413–10419. http://www.arnpjournals.org/journals/research_papers/rp_2015/jeas_1215_3079.pdf.
- [13] M. CP, K. Karuppanagounder, Performance prediction model for urban dual carriageway using travel time-based indices, *Transp. dev. econ.* 6 (1) (2020) 2, <https://doi.org/10.1007/s40890-019-0090-8>.
- [14] S. Ding, C. Su, J. Yu, An optimizing BP neural network algorithm based on genetic algorithm, *Artif. Intell. Rev.* 36 (2011) 153–162, <https://doi.org/10.1007/s10462-011-9208-z>.
- [15] T. Dokeroğlu, E. Sevinc, T. Kucukyilmaz, A. Cosar, A survey on new generation metaheuristic algorithms, *Comput. Ind. Eng.* 137 (2019) 106040, <https://doi.org/10.1016/j.cie.2019.106040>.
- [16] D. Doley, A.K. Maurya, An assessment of travel time dependability in urban corridors: guwahati City case Study, in: *2024 16th International Conference on COMMUNICATION SYSTEMS & NETWORKS (COMSNETS)*, 2024.
- [17] M. Dougherty, A review of neural networks applied to transport, *Transp. Res. C: Emerg. Technol.* 3 (4) (1995) 247–260, [https://doi.org/10.1016/0968-090X\(95\)00009-8](https://doi.org/10.1016/0968-090X(95)00009-8).
- [18] A.M. El Amrani, M. Fri, O. Benmoussa, N. Rouky, A deep reinforcement learning framework for last-mile delivery with public transport and traffic-aware integration: a case study in Casablanca, *Infrastruct. (Basel)* 10 (5) (2025) 112, <https://doi.org/10.3390/infrastructures10050112>.
- [19] L. Eleftheriadou, X. Cui, A framework for defining and estimating travel time reliability, <https://trid.trb.org/View/801806>, 2007.
- [20] C. Fang, Y. Cai, Y. Wu, A discrete wild horse optimizer for capacitated vehicle routing problem, *Sci. Rep.* 14 (1) (2024) 21277, <https://doi.org/10.1038/s41598-024-72242-0>.
- [21] D.P. Gaver Jr, Headstart strategies for combating congestion, *Transp. Sci.* 2 (2) (1968) 172–181, <https://doi.org/10.1287/trsc.2.2.172>.
- [22] P. Goodwin, The economic costs of road traffic congestion. <https://discovery.ucl.ac.uk/id/eprint/1259/>, 2004.
- [23] Ş. Güler, E. Eker, N. Yumuşak, Improvement of wild horse optimizer algorithm with random walk strategy (IWHO), and appointment as MLP supervisor for solving energy efficiency problem, *Energ. (Basel)* 18 (11) (2025) 2916, <https://doi.org/10.3390/en18112916>.
- [24] Z.K. Gurmu, W.D. Fan, Artificial neural network travel time prediction model for buses using only GPS data, *J. Public Trans.* 17 (2) (2014) 45–65, <https://doi.org/10.5038/2375-0901.17.2.3>.
- [25] A. Hasnaoui, A. Omari, Z.-e. Azzouz, Coot algorithm for optimization and management of residential power demand, *Prz. Elektrotech.* 99 (2023), <https://doi.org/10.15199/48.2023.06.49>.
- [26] C. Huang, Y. Li, X. Yao, A survey of automatic parameter tuning methods for metaheuristics, *IEEE trans. evol. comput.* 24 (2) (2019) 201–216, <https://doi.org/10.1109/TEVC.2019.2921598>.
- [27] W.B. Jackson, J.V. Jucker, An empirical study of travel time variability and travel choice behavior, *Transp. Sci.* 16 (4) (1982) 460–475, <https://doi.org/10.1287/trsc.16.4.460>.
- [28] S. Jung, K. Wunderlich, J. Larkin, A. Toppen, Rapid nationwide congestion monitoring: the urban congestion reporting program, in: *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749)*, 2004.
- [29] S. Katoch, S.S. Chauhan, V. Kumar, A review on genetic algorithm: past, present, and future, *Multimed. Tools. Appl.* 80 (5) (2021) 8091–8126, <https://doi.org/10.1007/s11042-020-10139-6>.
- [30] N. KHORSHIDI, S.A. ZARGARI, H. MIRZAHOSSEIN, H. HEIDARI, T. WALLER, PREDICTING TRAVEL-TIME RELIABILITY IN ROAD NETWORKS: A FITRNET-BASED APPROACH—A CASE STUDY OF ENGLAND, *Sci. J. Sil. Univ. Technol. Ser. Transp.* 126 (2025) 79–95, <https://doi.org/10.20858/sjstut.2025.126.5>.
- [31] N. Khorshidi, S.A. Zargari, S. Rezashoar, H. Mirzahassein, Optimizing travel time reliability with XAI: a Virginia interstate network case using machine learning and meta-heuristics, *Mach. Learn. Appl.* (2025) 100709, <https://doi.org/10.1016/j.mlwa.2025.100709>.
- [32] X. Kong, Y. Zhang, W.L. Eisele, X. Xiao, Using an interpretable machine learning framework to understand the relationship of mobility and reliability indices on truck drivers' Route choices, *IEEE Trans. Intell. Transp. Syst.* 23 (8) (2021) 13419–13428, <https://doi.org/10.1109/TITS.2021.3124221>.
- [33] S. Kranzinger, M. Steinmaßl, Which aggregation fits best? The use of linear regression to show the influence of temporal and spatial aggregation of sparse probe vehicle data on the explanation of travel time reliability, *Transp. Res. Rec.* 2676 (4) (2022) 210–227, <https://doi.org/10.1177/03611981211057881>.
- [34] A. Krogh, What are artificial neural networks? *Nat. Biotechnol.* 26 (2) (2008) 195–197, <https://doi.org/10.1038/nbt1386>.
- [35] B. Kuhn, L. Higgins, A. Nelson, M. Finley, G. Ullman, S. Chrysler, K. Wunderlich, V. Shah, C. Dudek, Lexicon for conveying travel time reliability information, 0309273072, <https://nap.nationalacademies.org/catalog/22604/lexicon-for-conveying-travel-time-reliability-information>, 2014.
- [36] H. Li, Z. Wang, X. Li, H. Wang, Y. Man, J. Shi, Travel time probability prediction based on constrained LSTM quantile regression, *J. Adv. Transp.* 2023 (1) (2023) 9910142, <https://doi.org/10.1155/2023/9910142>.
- [37] X.C. Liu, N. Haghighi, Travel-time reliability in simulation and planning models: utah case study (SHRP2 L04 IAP Round 7). <https://rosap.ntl.bts.gov/view/dot/54020>, 2019.
- [38] K. Lyman, R.L. Bertini, Using travel time reliability measures to improve regional transportation planning and operations, *Transp. Res. Rec.* 2046 (1) (2008) 1–10, <https://doi.org/10.3141/2046-01>.
- [39] V. Mandala, S.N.D. Surabhi, V. Balaji, D.R. Patil, S.F. Waris, G. Shobana, Wild horse optimizer and support vector machine (SVM) classifier predicts the heart disease converging nature-motivated optimization and machine learning, *Integr. Biomed. Res.* 8 (3) (2024) 1–12, <https://doi.org/10.25163/angiotherapy.839535>.
- [40] S. Mirjalili, S. Mirjalili, Evolutionary multi-layer perceptron. *Evolutionary Algorithms and Neural Networks: Theory and Applications*, 2019, pp. 87–104, https://doi.org/10.1007/978-3-319-93025-1_7.
- [41] S. Mirjalili, S.M. Mirjalili, A. Lewis, Let a biogeography-based optimizer train your multi-layer perceptron, *Inf Sci (Nij)* 269 (2014) 188–209, <https://doi.org/10.1016/j.ins.2014.01.038>.
- [42] H. Mirzahassein, S. Rezashoar, Feature importance analysis of optimized machine learning modeling for predicting customers satisfaction at the United States airlines, *Mach. Learn. Appl.* (2025) 100734, <https://doi.org/10.1016/j.mlwa.2025.100734>.
- [43] A. Najnin, J.C. Xia, G. Wright, T.G. Lin, Spatio-temporal analysis and visualisation of incident induced traffic congestion using real time online routing information, in: *The Ninth International Conference on Advanced Geographic Information Systems, Applications, and Services, Nice, France, 2017*.
- [44] I. Naruei, F. Keynia, A new optimization method based on COOT bird natural life model, *Expert. Syst. Appl.* 183 (2021) 115352, <https://doi.org/10.1016/j.eswa.2021.115352>.
- [45] I. Naruei, F. Keynia, Wild horse optimizer: a new meta-heuristic algorithm for solving engineering optimization problems, *Eng. Comput.* 38 (Suppl 4) (2022) 3025–3056, <https://doi.org/10.1007/s00366-021-01438-z>.
- [46] National Academies of Sciences, E., & Medicine, Analytical procedures for determining the impacts of reliability mitigation strategies, in: <https://nap.nationalacademies.org/catalog/22806/analytical-procedures-for-determining-the-impacts-of-reliability-mitigation-strategies>, 2013.
- [47] S. Oh, Y.-J. Byon, K. Jang, H. Yeo, Short-term travel-time prediction on highway: a review of the data-driven approach, *Transp. Rev.* 35 (1) (2015) 4–32, <https://doi.org/10.1080/01441647.2014.992496>.
- [48] J.C. Principe, C. Lefebvre, C.L. Fancourt, Dataflow learning in coupled lattices: an application to artificial neural networks. *Handbook of Global Optimization*:

- Volume 2, Springer, 2002, pp. 363–386, https://doi.org/10.1007/978-1-4757-5362-2_10.
- [49] W. Pu, Analytic relationships between travel time reliability measures, *Transp. Res. Rec* 2254 (1) (2011) 122–130, <https://doi.org/10.3141/2254-13>.
- [50] S. Rezashoar, A.A. Rassafi, Analyzing the impact of ant colony optimization parameters for path searching behavior, *Civ. Proj.* 6 (11) (2025), <https://doi.org/10.22034/cpj.2024.482516.1326>.
- [51] S. Rezashoar, A.A. Rassafi, Analyzing the performance of the Red Deer optimization algorithm in comparison to other metaheuristic algorithms, *J. AI Data Min.* 13 (1) (2025) 53–61, <https://doi.org/10.22044/jadm.2025.14868.2586>.
- [52] H.K. Rushdi, F.M. Al-Naima, Coot optimization algorithm for parameter estimation of photovoltaic model, *MEST J.* 10 (2) (2022) 177–185, <https://doi.org/10.12709/mest.10.10.02.16>.
- [53] D. Schrank, T. Lomax, The 2003 urban mobility report. <https://rosap.ntl.bts.gov/view/dot/15904>, 2003.
- [54] S. Sengupta, S. Basak, R.A. Peters, Particle Swarm Optimization: a survey of historical and recent developments with hybridization perspectives, *Mach. learn. knowl. extr.* 1 (1) (2018) 157–191, <https://doi.org/10.3390/make1010010>.
- [55] R. Shamsashtiany, M. Ameri, Road accidents prediction with multilayer perceptron MLP modelling case study: roads of Qazvin, Zanjan and Hamadan, *J. Civ. Eng. Mater. Appl.* 2 (4) (2018) 181–192, <https://doi.org/10.22034/jcema.2018.91998>.
- [56] F. Shao, H. Shao, D. Wang, W.H. Lam, A multi-task spatio-temporal generative adversarial network for prediction of travel time reliability in peak hour periods, *Phys. A: Stat. Mech. Appl.* 638 (2024) 129632, <https://doi.org/10.1016/j.physa.2024.129632>.
- [57] F. Shao, H. Shao, D. Wang, W.H. Lam, M.L. Tam, A generative adversarial network-based framework for network-wide travel time reliability prediction, *Knowl. Based. Syst.* 283 (2024) 111184, <https://doi.org/10.1016/j.knsys.2023.111184>.
- [58] T. Shaw, Performance Measures of Operational Effectiveness For Highway Segments and Systems, 311, Transportation Research Board, 2003. https://onlinepubs.trb.org/onlinepubs/nchrp/nchrp_syn_311.pdf.
- [59] K.A. Small, R.B. Noland, P. Koskenoja, Socio-economic attributes and impacts of travel reliability: a stated preference approach. <https://escholarship.org/uc/item/8sp2315k>, 1995.
- [60] Such, F.P., Madhavan, V., Conti, E., Lehman, J., Stanley, K.O., & Clune, J. (2017). Deep neuroevolution: genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint arXiv:1712.06567*. <https://doi.org/10.48550/arXiv.1712.06567>.
- [61] Y. Sun, Y. Chen, Travel time variability in urban mobility: exploring transportation system reliability performance using ridesharing data, *Sustainability* 16 (18) (2024) 8103, <https://doi.org/10.3390/su16188103>.
- [62] J. Swan, S. Adriaansen, A.E. Brownlee, K. Hammond, C.G. Johnson, A. Kheiri, F. Krawiec, J.J. Merelo, L.L. Minku, E. Özcan, Metaheuristics “in the large, *Eur J Oper Res* 297 (2) (2022) 393–406, <https://doi.org/10.1016/j.ejor.2021.05.042>.
- [63] C. Systematics, Traffic congestion and reliability: trends and advanced strategies for congestion mitigation. <https://rosap.ntl.bts.gov/view/dot/20656>, 2005.
- [64] S.H. Taher, Z.A. Alkaissi, Analysis the reliability of travel time in urban corridors in Baghdad City, *J. Eng.* 30 (07) (2024) 202–217, <https://doi.org/10.31026/j.eng.2024.07.12>.
- [65] R. Tanwar, P.K. Agarwal, Assessing travel time performance of multimodal transportation systems using fuzzy-analytic hierarchy process: a case study of Bhopal City, *Heliyon* 10 (17) (2024), <https://doi.org/10.1016/j.heliyon.2024.e36844>.
- [66] D. Thapa, S. Mishra, A. Khattak, M. Adeel, Assessing driver behavior in work zones: a discretized duration approach to predict speeding, *Accid. Anal. Prev.* 196 (2024) 107427, <https://doi.org/10.1016/j.aap.2023.107427>.
- [67] E. Tufoor, L. Rilett, High volume freeway travel time reliability and the COVID-19 pandemic, *Transp. Res. Rec* 2678 (12) (2024) 96–113, <https://doi.org/10.1177/03611981221090929>.
- [68] A. Utku, S.K. Kaya, Multi-layer perceptron based transfer passenger flow prediction in Istanbul transportation system, *Decis. Mak.: Appl. Manag. Eng.* 5 (1) (2022) 208–224, <https://doi.org/10.31181/dmame0315052022u>.
- [69] J. Van Lint, H.J. Van Zuylen, Monitoring and predicting freeway travel time reliability: using width and skew of day-to-day travel time distribution, *Transp. Res. Rec* 1917 (1) (2005) 54–62, <https://doi.org/10.1177/0361198105191700107>.
- [70] P.J. Werbos, Generalization of backpropagation with application to a recurrent gas market model, *Neural netw.* 1 (4) (1988) 339–356, [https://doi.org/10.1016/0893-6080\(88\)90007-X](https://doi.org/10.1016/0893-6080(88)90007-X).
- [71] Z. Wu, L.R. Rilett, W. Ren, New methodologies for predicting corridor travel time mean and reliability, *Int. j. urban sci.* 26 (3) (2022) 517–540, <https://doi.org/10.1080/12265934.2021.1899844>.
- [72] M.-M. Xiong, Z.-Y. Han, B. Lu, J. Guo, S. Xu, Y.-J. Yang, H. Huang, Effects of rainfall on the weekday traffic flow in major cities of the Beijing–Tianjin–Hebei region, China, in 2021, *Adv. Clim. Change Res.* 13 (6) (2022) 858–867, <https://doi.org/10.1016/j.accres.2022.11.009>.
- [73] X. Xu, Y. Sun, Y. Bai, K. Zhang, Y. Liu, X. Zhao, Seq2Img-DRNET: a travel time index prediction algorithm for complex road network at regional level, *Expert. Syst. Appl.* 185 (2021) 115554, <https://doi.org/10.1016/j.eswa.2021.115554>.
- [74] J.-S. Yang, Travel time prediction using the GPS test vehicle and Kalman filtering techniques, in: *Proceedings of the 2005, American Control Conference*, 2005, 2005.
- [75] S. Yoon, D. Kum, The multilayer perceptron approach to lateral motion prediction of surrounding vehicles for autonomous vehicles, in: *2016 IEEE Intelligent Vehicles Symposium (IV)*, 2016.
- [76] S.A. Zargari, N. Amoei Khorshidi, N. Kalantari, Planning level estimation of travel time reliability for virginia interstate highways. <https://trid.trb.org/View/1759699>, 2021.
- [77] W.-y. Zhang, W. Guan, L.-y. Song, H.-j. Sun, Alpha-reliable combined mean traffic equilibrium model with stochastic travel times, *J. Cent. South. Univ* 20 (12) (2013) 3770–3778, <https://doi.org/10.1007/s11771-013-1906-z>.
- [78] X. Zhang, J.A. Rice, Short-term travel time prediction, *Transp. Res. C: Emerg. Technol.* 11 (3–4) (2003) 187–210, [https://doi.org/10.1016/S0968-090X\(03\)00026-3](https://doi.org/10.1016/S0968-090X(03)00026-3).
- [79] Y. Zhang, Hourly traffic forecasts using interacting multiple model (IMM) predictor, *IEEe Signal. Process. Lett* 18 (10) (2011) 607–610, <https://doi.org/10.1109/LSP.2011.2165537>.
- [80] Y. Zhang, Y. Liu, Traffic forecasting using least squares support vector machines, *Transportmetrica* 5 (3) (2009) 193–213, <https://doi.org/10.1080/18128600902823216>.
- [81] Y. Zhang, Y. Liu, Traffic forecasts using interacting multiple model algorithm, in: *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 2009.
- [82] Y. Zhang, Y. Liu, Analysis of peak and non-peak traffic forecasts using combined models, *J. Adv. Transp* 45 (1) (2011) 21–37, <https://doi.org/10.1002/atr.128>.
- [83] W. Zhao, Y. Liu, J. Li, T. Zhu, K. Zhao, K. Hu, A Wild Horse Optimization algorithm with chaotic inertia weights and its application in linear antenna array synthesis, *PLoS. One* 19 (7) (2024) e0304971, <https://doi.org/10.1371/journal.pone.0304971>.
- [84] R. Zheng, A.G. Hussien, H.-M. Jia, L. Abualigah, S. Wang, D. Wu, An improved wild horse optimizer for solving optimization problems, *Mathematics* 10 (8) (2022) 1311, <https://doi.org/10.3390/math10081311>.
- [85] R. Zhou, H. Chen, H. Chen, E. Liu, S. Jiang, Research on traffic situation analysis for urban road network through spatiotemporal data mining: a case study of Xi'an, China, *IEEe Access* 9 (2021) 75553–75567, <https://doi.org/10.1109/ACCESS.2021.3082188>.