Li *et al*. *Complex Eng Syst* 2024;4:26
DOI: 10.20517/ces.2024.64

Complex Engineering Systems

Review

Open Access

A meta-learning approach for predicting asphalt pavement deflection basin area

Zhuoxuan Li[1,2], Xin Jin[3], Xinli Shi[2,3], Jinde Cao[1,2]

[1]School of Mathematics, Southeast University, Nanjing 210096, Jiangsu, China.
[2]Nanjing Modern Multimodal Transportation Laboratory, Ministry of Transport, Nanjing 211135, Jiangsu, China.
[3]School of Cyber Science and Engineering, Southeast University, Nanjing 210096, Jiangsu, China.

Correspondence to: Prof. Xinli Shi, School of Cyber Science and Engineering, Southeast University, Southeast University Road, No. 2, Nanjing 211189, Jiangsu, China. E-mail: xinli_shi@seu.edu.cn; ORCID: orcid.org/0000-0002-4443-608X

How to cite this article: Li Z, Jin X, Shi X, Cao J. A meta-learning approach for predicting asphalt pavement deflection basin area. *Complex Eng Syst* 2024;4:26. http://dx.doi.org/10.20517/ces.2024.64

Received: 25 Sep 2024 **First Decision:** 12 Nov 2024 **Revised:** 7 Dec 2024 **Accepted:** 24 Dec 2024 **Published:** 31 Dec 2024

Academic Editor: Ding Wang **Copy Editor:** Fangling Lan **Production Editor:** Fangling Lan

Abstract

To address the urgent need for accurate pavement performance modeling in pavement design, this study proposes a meta-learning-based few-shot learning method for predicting the Deflection Basin Area (DBA) of asphalt pavements. The method utilizes features such as pavement temperature and load pressure, and applies cyclic DBA data from various pavement types subjected to different pressures. The objective is to predict the trend of DBA changes over cycles at a specific pressure. By leveraging pre-training on diverse pavement datasets, the proposed meta-learning model reduces the training data required for target pavement DBA prediction, enabling better generalization to the target pavement. This approach enhances DBA prediction accuracy even with a small sample size. Compared to traditional machine learning and pre-training methods using data from a single pavement type, the proposed method achieves a Mean Square Error of 13.26 and a Mean Absolute Error of 2.85, demonstrating superior performance. Furthermore, it achieves high prediction accuracy with fewer iterations. Overall, the proposed method effectively predicts DBA across various pavement structures with a few data.

Keywords: Meta-learning, deflection basin area, pre-training, few-shot learning, time series prediction

© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, sharing, adaptation, distribution and reproduction in any medium or format, for any purpose, even commercially, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

1. INTRODUCTION

Highway infrastructure in China has undergone rapid development in recent years, with asphalt pavement (AP) structures being widely adopted across roads of all grades. The majority of these structures use semi-rigid base pavement designs[1]. However, the long-term service of operating highways, subjected to traffic loads and environmental factors, often leads to various degrees of structural damage, including cracking and subsidence. In this context, pavement deflection has emerged as a critical parameter for assessing the bearing capacity of subgrade pavements and plays a pivotal role in evaluating overall pavement conditions[2].

The Falling Weight Deflectometer (FWD) has become the standard tool for non-destructive pavement testing. While FWD detection results contain a wealth of pavement characteristic information, current engineering practices typically rely solely on the maximum deflection at the load center to characterize the surface deflection of the structural layer. This approach, however, underutilizes the comprehensive data provided by FWD testing[3]. Recognizing this limitation, researchers have increasingly focused on establishing correlations between the pavement Deflection Basin Area (DBA) and the structure's bearing capacity. This approach aims to predict pavement usage indicators and evaluate overall structural integrity more comprehensively[4–7]. The DBA, which considers the entire deflection profile rather than a single point, offers a more holistic representation of pavement response to loading.

However, the collection of FWD testing data presents significant challenges. The process is resource-intensive, time-consuming, and costly, making it difficult to conduct repeated measurements across large road networks. These limitations underscore the need for predictive models that can accurately estimate the DBA for various pavement structures without extensive field testing.

Recent advancements in artificial intelligence (AI) have opened new avenues for addressing this challenge. AI methods have been successfully applied across various domains of traffic engineering[8–10], demonstrating their potential for complex prediction tasks. In the context of pavement engineering, AI offers the promise of high-accuracy DBA prediction, potentially revolutionizing how we assess and manage road infrastructure. The motivation behind this research lies in the critical need for more efficient and cost-effective methods of pavement assessment. By developing accurate predictive models for DBA, we aim to enhance the ability of road authorities to monitor pavement conditions, plan maintenance activities, and optimize the allocation of resources. This approach promises to reduce the reliance on extensive field testing and provide a more nuanced understanding of pavement behavior under various conditions.

Despite the potential benefits, achieving high-accuracy DBA prediction remains a significant challenge. This research seeks to address this gap by exploring advanced AI techniques and their application to pavement engineering. By doing so, we aim to contribute to developing more robust and reliable methods for assessing pavement structural integrity, ultimately leading to improved road infrastructure management and longevity. This research aims to develop and design a meta-learning-based prediction model for DBA of APs, which uses measured cyclic data of pavement DBA under different pressures to predict the trend of rebound deflection basin area changes throughout all cycles at a specific pressure. The contributions of this research are as follows.

- This study applies meta-learning algorithms to predict the AP DBA over the entire cycle at a specific pressure for the first time, achieving better performance than traditional machine learning algorithms as measured by Mean Square Error (MSE) and Mean Absolute Error (MAE). These results demonstrate the effectiveness of meta-learning algorithms in small sample regression tasks.
- The proposed meta-learning prediction approach can reduce the need on the amount of target pavement data by reusing relevant data acquired from other pavement structures for training, which is crucial given the challenge of obtaining long-term cyclic data for target pavements in the early stages.
- Compared with the pre-training-based transfer learning method, the proposed meta-learning prediction

uses a limited dataset specific to the target pavement to fine-tune the parameters from pre-training. This strategy not only ensures stability, but also produces better performance in predicting the DBA.

2. RELATED WORK

Machine learning methods can be widely used for data regression prediction tasks [11], and in recent years, they have been increasingly applied to pavement performance prediction tasks. The primary hurdle in pavement performance prediction arises from extracting crucial spatiotemporal features from multidimensional coupled data. Early work, including Artificial Neural Network (ANN) [12], stacked autoencoders [13], and Long Short-term Memory (LSTM) [14], predominantly focused on extracting temporal features but encountered challenges in unveiling latent spatial features within traffic data, leading to suboptimal performance. Yang *et al.* [15] use K-nearest neighbor to characterize crack type and crack width information derived from FWD test data for applied statistical techniques, including ANN and multiple nonlinear regression, to proficiently rut distress in AP using an AP analyzer [16].

Recently, sophisticated learning algorithms have been integrated into engineering modeling applications. Tang *et al*. use genetic algorithms to develop a reverse analysis program that combines finite element analysis and population-based optimization techniques to infer the modulus of the pavement layer [17]. Mabrouk *et al*. use ANNs to calculate the pavement layer modulus as a function of traffic speed deviation, thus predicting the pavement modulus [18]. Zhang *et al.* propose a brand-new framework for predicting time series of ruts, based on multi-level discrete wavelet decomposition and multivariate transfer entropy for feature selection, to achieve higher prediction accuracy [19]. Li *et al.* compare random forest and gradient boosting regression methods for predicting the pavement Asphalt Concrete (AC) layer and use grid search and cross-validation for optimization [20].

In addition, the frequent collection of pavement structure data is very time-consuming, disrupts traffic flow, and is costly, making it hard to duplicate. These factors might contribute to the oversight of pavement structure aspects in maintenance or repair decisions. Consequently, researchers have invested considerable effort in identifying relatively expedient alternative analysis methods to overcome these limitations. Li *et al*. propose a Chaotic Particle Swarm Optimization (CPSO) to optimize the Extreme Gradient Boosting (XGBoost) model, decreasing the frequency of deflection tests while maintaining estimation accuracy [21]. Zhang *et al*. propose a model for predicting ruts based on multi-source transfer entropy and graph neural networks, which can adapt to sufficient predictive performance and the generalization ability of various complex pavement design data [22]. Shen *et al*. employ LSTM to construct a predictive model for estimating the technical condition score of bridge components [23].

Nevertheless, traditional machine learning methods still face a practical challenge: the accuracy of deep learning regression prediction algorithms heavily depends on the availability of a substantial amount of training data [24]. For tasks such as pavement performance prediction, a large volume of pavement structure data is required, which presents a significant drawback. Collecting such extensive data is time-consuming and resource-intensive, but it remains essential for training deep learning models [25,26]. Furthermore, if the pavement structure used for training does not align with the target pavement structure, retraining the model becomes necessary, which again requires considerable time and effort. These issues highlight a major limitation of traditional machine learning methods: developing accurate performance prediction models for various pavement structures demands both significant time and financial investment.

Given these challenges, alternative approaches such as Model-Agnostic Meta-Learning (MAML) [27] have gained attention in the field. MAML is a powerful meta-learning algorithm designed for few-shot learning, and it has demonstrated superior performance in addressing problems with limited data [28–32]. As a member of optimization-based meta-learning algorithms, MAML differs from traditional methods by enabling models to learn in a way that generalizes quickly to new tasks with minimal data. Other meta-learning algorithms have introduced various modifications for learning weights in task-specific classifiers[33,34]. For example, methods in[35–38] first learn a function to embed the support set and target examples of a few-shot task, and then use the test support set to fine-tune task-specific weights for embedding the target examples. On the basis of meta-learning, Aguiar *et al.* propose selecting meta-features for extracting optimal dataset descriptions, which enhances multi-target regression with high predictive accuracy[39]. Additionally, Jeong *et al.* introduce a meta-learning approach for State-Of-Charge (SOC) estimation in batteries, aiming to reduce the amount of target data needed for training by leveraging deep learning[40]. By overcoming the data limitations inherent in traditional machine learning methods, meta-learning provides a more efficient framework for addressing pavement performance prediction tasks, particularly in situations with scarce or inconsistent data.

Building upon optimization-based meta-learning algorithms, MAML, in particular, has exerted significant influence, inspiring numerous direct extensions in[41–44]. Many of these extensions heavily depend on the foundational structure of the MAML algorithm, encompassing the outer loop (for meta-training) and the inner loop (for task-specific adaptation). However, scant prior research has thoroughly examined the reasons for the success of this core aspect of MAML. To investigate how and why MAML achieves effective few-shot learning[45–47], employs analytical tools such as centered kernel alignment to scrutinize the neural network representations learned by the MAML. This analysis also highlights the proficiency of this algorithm in acquiring practical few-shot learning features.

Meta-learning is sensitive to the choice of initial model parameters. In certain situations, different initial parameters may result in significant performance variations of the model across various tasks. Additionally, the performance of meta-learning may be affected by noise and distribution changes in the input data. When there is considerable noise or shifts in the data distribution within the meta-learning tasks, it can decline the model's generalization performance.

In addition, some robust meta-learning algorithms have been developed for noise rejection. The most effective robust training methods include adversarial training and robustness which establishes a theoretical foundation between accuracy and robustness, and many of their variants such as fast adversarial training methods[48–50], semi-supervised robust training[51,52], adversarial transfer learning. Furthermore, recent research[53–55] has explored the transferability of robustness within the realms of transfer learning and representation learning. Nevertheless, the conventional robust training methods mentioned earlier need to be better suited for MAML in few-shot learning, given the dual optimization characteristics inherent in MAML.

In summary, meta-learning exhibits significant advantages over traditional machine-learning methods when handling small-sample classification and regression tasks. Firstly, meta-learning achieves rapid model adaptation by learning across multiple relevant tasks, enabling it to adjust its parameters to accommodate new tasks quickly. Secondly, meta-learning enhances sample efficiency by acquiring general task-related knowledge, allowing the model to utilize limited samples for learning more effectively. Thirdly, emphasizing the transfer learning concept in meta-learning enables models to share knowledge between tasks, facilitating robust generalization performance. These capabilities represent a notable advantage that traditional machine-learning algorithms lack.

3. FUNDAMENTAL ALGORITHMS

3.1. Gradient boosting decision trees and random forests

Breiman[56] proposed the Random Forests (RF) algorithm, which builds upon the Bagging algorithm by incorporating Bootstrap sampling to construct multiple decision trees. The final prediction of the random forest is made through a majority voting mechanism. The use of Bootstrap sampling enhances the algorithm's accuracy

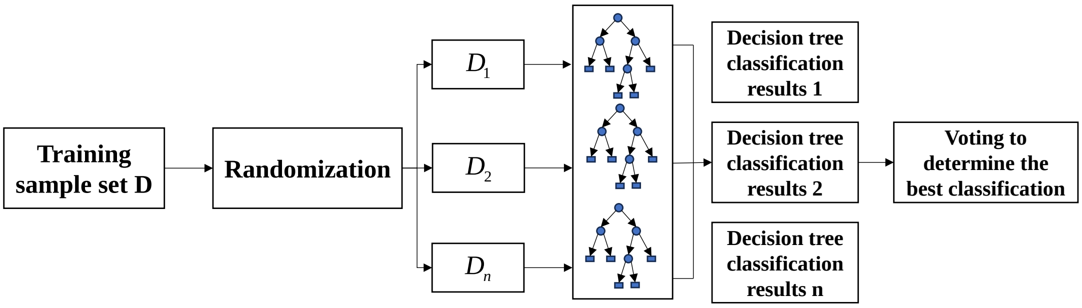

Figure 1. The structure of RF.

and helps mitigate the overfitting issues commonly associated with single decision trees. The structure of RF is shown in Figure 1.

The Gradient Boosting Decision Tree (GBDT)[57] is an ensemble machine learning algorithm based on the Boosting strategy proposed by Friedman. The core idea is to sequentially train multiple weak learners to improve performance iteratively. In each iteration, except for the first decision tree, the objective is to minimize the loss function of the current learner by updating it in the direction of the gradient, ensuring that the loss function decreases with each step. Through continuous iterations, the residuals approach zero, and the results of all trees are combined to generate the final prediction. The specific implementation process of GBDT is as follows:

(1) Initialize weak learner

$$f_0(x) = \text{argmin} \sum_{i=1}^{N} L(y_i, c),$$

where e is the value of the leaf node after the node is divided by the least squares method.

(2) Calculate negative gradient

For each tree $m = 1, 2, \cdots, M$, and for each sample $i = 1, 2, \cdots, N$, calculate the negative gradient, which is the residual:

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right],$$

where $f(x_i)$ is the predicted value of the weak learner, and y_i is the true value of the weak learner.

The obtained residual is used as the true value of the new sample, and the data (x_i, r_{im}), $i = 1, 2, \cdots, N$, is used as the training data of the next tree to obtain a new regression tree $f_m(x)$, whose corresponding leaf node area is R_{jm}, $j = 1, 2, \cdots, J$ is the number of leaf nodes of the regression tree t, and the best fitting value is calculated for the leaf area $j = 1, 2, \cdots, J$, and we have

$$\gamma_{jm} = \text{argmin} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma).$$

Update the strong learner, then we have

$$f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J} \gamma_{jm} I(x \in R_{jm}).$$

Get the final learner

$$f(x) = f_0(x) + \sum_{m=1}^{M} \sum_{j=1}^{J} \gamma_{jm} I(x \in R_{jm}).$$

3.2. Convolutional neural networks for solving regression problems

The network structure of Convolutional Neural Networks (CNNs) consists of an input layer, convolutional layers, pooling layers, and output layers, among others. CNNs are widely used for image classification tasks due to their ability to effectively capture spatial hierarchies in image data. However, when adapted for regression tasks involving numerical data, significant differences arise in various aspects, such as input data handling, the design of convolution and pooling layers, output layer configuration, and the choice of loss function.

In regression tasks, the input data typically takes the form of a numerical feature vector or matrix, denoted as $\mathbf{X}$, where $\mathbf{X} \in \mathbb{R}^{m \times n}$. The convolution layer extracts features from the input data through convolution operations. Mathematically, the output of the convolution layer can be given as:

$$\mathbf{C} = \mathbf{X} * \mathbf{W},$$

where $*$ represents the convolution operation. $\mathbf{X}$ is the input feature matrix, $\mathbf{W}$ represents the convolution kernel (also known as the filter), s denotes the stride of the convolution operation, and p refers to the padding size. The convolution operation involves sliding the kernel $\mathbf{W}$ over the input matrix $\mathbf{X}$, performing element-wise multiplications, and summing the results to produce the output feature matrix $\mathbf{C}$. The choice of s and p directly affects the dimensions of $\mathbf{C}$ and influences how much of the input matrix's boundary information is retained in the output. By adapting these components, CNNs can be effectively utilized for regression tasks, making them versatile tools for a wide range of applications beyond image classification.

Activation functions introduce non-linear transformations, increasing the expressive ability of the model. Commonly used activation functions include the rectified linear unit (ReLU) function. The output of the activation function is given by: $\mathbf{A} = \mathrm{ReLU}(\mathbf{C})$, where $\mathrm{ReLU}(x) = \max(0, x)$. Pooling layers reduce the size of the feature maps through down-sampling while preserving the main features. Common pooling operations include max pooling and average pooling. Assuming the size of the pooling operation is $k \times k$ and the stride is s, the output of the pooling layer can be expressed as:

$$\mathrm{MaxPooling}\left(x_{ij}\right) = \max\left(x_{(i-1)k+1:ik,(j-1)k+1:jk}\right), \tag{1}$$

where x_{ij} represents the value of the pooled output at position (i, j) in the feature map, while k denotes the size of the pooling window, such as $k \times k$. The expression $x_{(i-1)k+1:ik,(j-1)k+1:jk}$ specifies the submatrix of the input feature map covered by the pooling window for the output position (i, j). The max pooling operation extracts the maximum value within this submatrix, effectively reducing the spatial dimensions of the feature map while retaining the most salient features. This dimensionality reduction enhances computational efficiency and helps mitigate overfitting by summarizing the key information from each region of the feature map.

Fully connected layers flatten the output of the pooling layer into a vector and perform linear transformations through matrix multiplication and bias terms. The output of the fully connected layer is given as

$$\mathbf{F} = \mathrm{ReLU}\left(\mathbf{P} \cdot W_f + b_f\right), \tag{2}$$

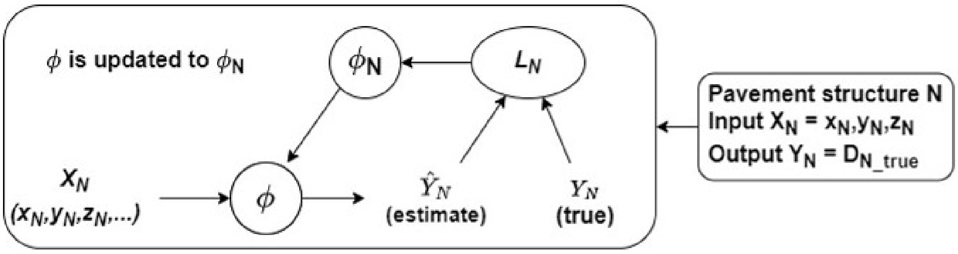

Figure 2. General deep-learning training methods.

where $\mathbf{P} = \text{MaxPooling}(A)$, the weight matrix of the fully connected layer is denoted as W_f, and the bias term is b_f. Activation functions introduce non-linear transformations, increasing the expressive ability of the model and allowing it to learn complex patterns. Commonly used activation functions include the sigmoid, tanh, and ReLU functions. The ReLU function is often chosen for its simplicity, computational efficiency, and ability to mitigate the vanishing gradient problem, making it a popular choice for deep networks.

Finally, the output layer maps the output of the fully connected layer to the range of the predicted values through linear transformations and activation functions, resulting in the output

$$\mathbf{Y} = \mathbf{F} \cdot W_o + b_o, \tag{3}$$

where W_o represents the weight matrix, b_o is the bias term, and Y indicates the predicted data.

In regression tasks, the MSE is a commonly employed loss function. The network parameters are then updated through the backpropagation algorithm based on this loss function to minimize the overall loss. This iterative process allows the CNN algorithm to learn the relationship between input features and regression targets, enabling it to make numerical regression predictions.

Traditional machine learning methods rely on a substantial amount of data for training. When faced with small sample data, it is often unable to train a model with excellent performance. The model parameter update process for the service performance prediction of different pavement structures is shown in Figure 2, where each type of pavement structure must be trained separately as a model to perform regression tasks.

3.3. Pre-training for the prediction of pavement deflection basin area

Pre-training method is an effective approach for few-shot learning, which is an unsupervised learning method. Before predicting the DBA of the target pavement structure data, we use other types of pavement surface data for the usual optimization pre-training. This approach is applicable to deep learning models trained through gradient descent. The comprehensive pre-training process is given in

$$\phi_i \leftarrow \phi_i - \alpha \nabla_{\phi_i} L_i \left(\phi_i, D_i^{\text{pre-train}} \right), \tag{4}$$

where $D_i^{\text{pre-train}}$ is the dataset used for training the DBA, and D_i^{test} is the dataset used for testing. The pre-training is performed on $D_i^{\text{pre-train}}$, updating the parameters ϕ_i using the deep learning model. α denotes a weighting factor, representing the learning rate at which ϕ_i will traverse in the gradient direction. The parameter ϕ_i is set with a random number initially. $D_i^{\text{pre-train}}$ is used to pre-train ϕ_i to minimize the loss function L_i. By repeating Equation (4), the parameter ϕ_i obtained through pre-training serves as the initial parameter for training the model with target pavement data. Then, fine-tuning is performed using target pavement training data to calculate the loss function L_T. The final parameter ϕ_i of the model is initialized as ϕ_T, resulting in a fine-tuned model. Finally, the target data to be trained is input into the final model for prediction. The overall

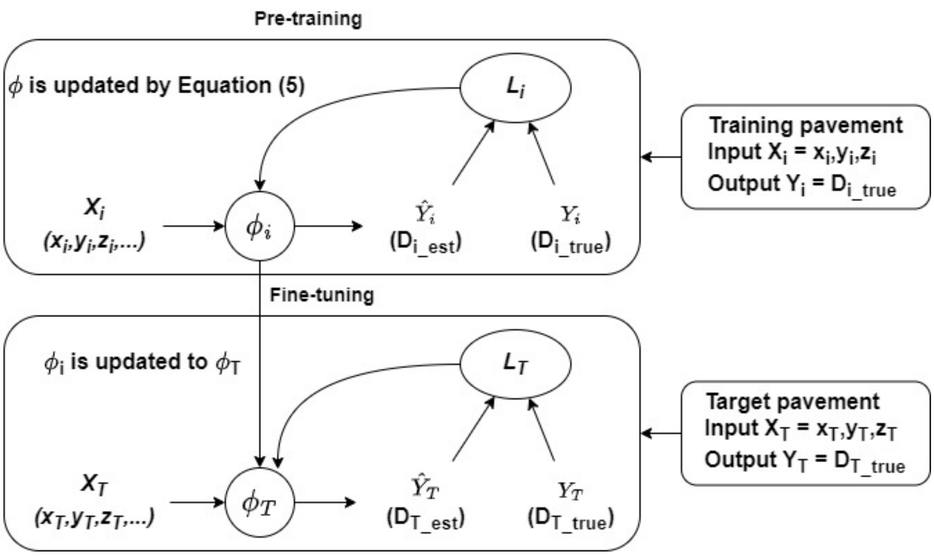

Figure 3. Pre-training methods.

parameter update process is illustrated in Figure 3, where $D_{\text{T\_est}}$ represents the estimated value obtained by the pre-training method, and $D_{\text{T\_true}}$ stands for the true value.

The purpose of pre-training models is to determine the optimal initial parameters ϕ_i for all tasks. However, pre-training does not ensure that employing this ϕ_i for fine-tuning in other tasks will yield a favorable ϕ_T. In essence, the pre-training algorithm concentrates on evaluating the performance of ϕ_i within the current model.

3.4. MAML for the prediction of pavement deflection basin area

The meta-learning algorithm stands out as one of the most effective approaches for addressing few-shot learning challenges. Figure 4 shows the overall process of using meta-learning methods to predict the DBA. The meta-model leverages the related datasets of 18 pavement structures, excluding the target pavement structure, for the training task, while the target pavement data serves as the testing task. Furthermore, each task is subdivided into a query set and a support set, utilized for training and testing in the inner loop, respectively. Algorithm 1 showcases the entire pre-training procedure of meta-learning.

First, meta-learning initializes a model with random parameters ϕ. The meta-model uses pavement data from tasks other than the target pavement, referred to as $D_i^{\text{meta-train}}$. The parameter update is expressed as follows:

$$\phi_i = \phi - \alpha \nabla_\phi L_i \left(\phi, D_i^{\text{support}} \right), \tag{5}$$

where ϕ represents the initial model parameters, and ϕ_i denotes the updated parameters after performing gradient descent on the support data D_i^{support}. The learning rate α controls the step size for each update, and $\nabla_\phi L_i \left(\phi, D_i^{\text{support}} \right)$ refers to the gradient of the loss function L_i with respect to ϕ, computed using the support dataset.

For each class of AP data, the data is further divided into D_i^{support} and D_i^{query}. In the inner loop, the model's parameters are updated using a small number of training samples, enabling the model to quickly adapt to the task. D_i^{support} is used for training, and one or more gradient descent steps are performed to update the model's parameters from ϕ to ϕ_i. Multiple D_i^{query} sets are used during the inner loop, with each gradient update applied

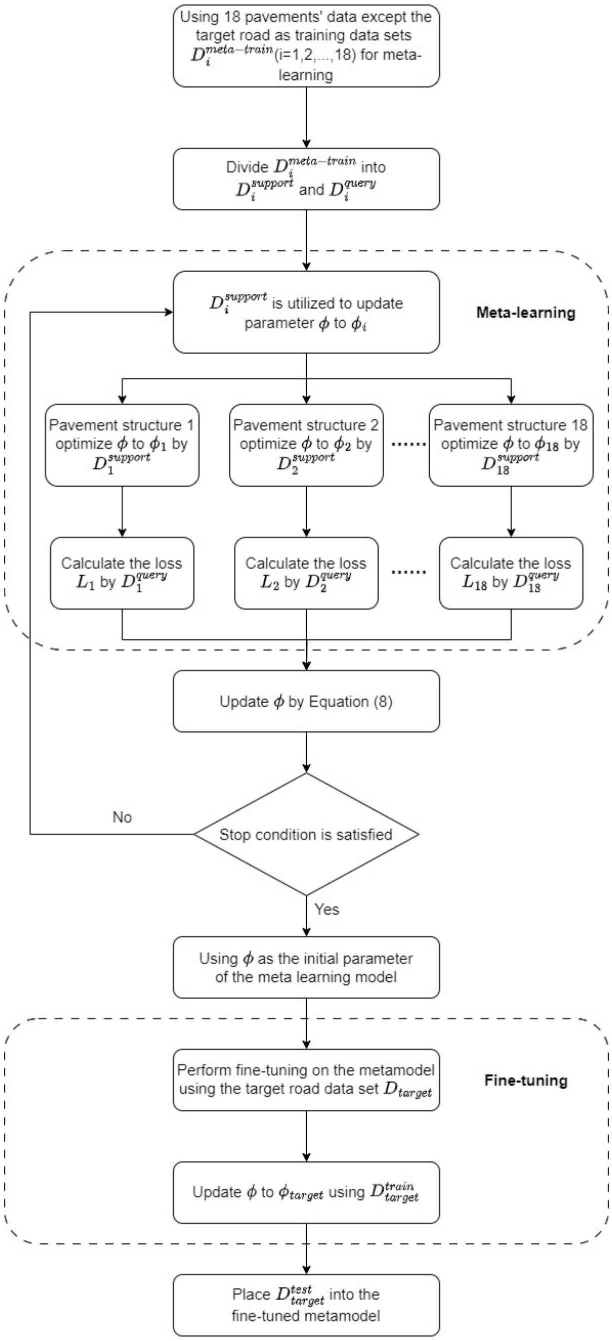

Figure 4. Meta learning algorithm flowchart

to the model with the original parameters ϕ. After the inner loop ends, n updated ϕ_i parameters are obtained.

In the outer loop, the MAML algorithm improves the initial parameters of the model by learning through inner loop iterations on multiple small-sample tasks, enabling it to quickly adapt to new tasks. For each update in the inner loop, the obtained parameters are used as the initial parameters of the new model. The model is then tested using the corresponding test set D_i^{query}, resulting in n loss values. These losses are weighted and averaged to obtain the overall loss function of the meta-model, as given in

$$L_{sum} = \sum_i L_i \left(\phi_i, D_i^{\text{query}} \right). \tag{6}$$

Using the sum of the loss L_{sum}, we perform gradient descent on the initial parameters of the target meta-model. Throughout the entire training process, we continuously perform inner and outer loops on multiple tasks to iteratively optimize the model's parameter ϕ. In this way, when encountering a new task, we only need to use the updated model from the outer loop to rapidly adapt to this task using a small quantity of training data. Ultimately, the MAML algorithm achieves fast learning and adaptation on few-slot tasks by updating the meta-model parameters ϕ, resulting in a well-trained final model, as given in

$$\phi \leftarrow \phi - \beta \nabla_{\phi} L_{sum}. \tag{7}$$

During the prediction phase, we use a portion of the target pavement data as training data ($D_{\text{target}}^{\text{train}}$), which is input into the newly learned meta-model for fine-tuning. Fine-tuning is performed using target pavement training data to calculate the loss function L_T. The initial parameters of the model are updated to ϕ_T to initialize ϕ. Finally, we use the test data ($D_{\text{target}}^{\text{test}}$) for prediction and obtain the predicted results. A flowchart is shown in Figure 5 to illustrate the parameter updates of meta-learning.

Algorithm 1 Meta-learning

```
Input: D_i^{meta-train} : meta-training data
α, β : step size hyperparameters
Initialize φ with a random number
Divide D_i^{meta-train} into D_i^{support} and D_i^{query}
while not done do
    for all D_i^{meta-train} do
        Compute adapted parameters : φ_i = φ − α∇_φ L_i(φ, D_i^{support})
    end for
    Calculate the loss value : L_sum = Σ_i L_i(φ_i, D_i^{query})
    Update φ ← φ − β∇_φ L_sum
end while
```

As observed, in the training process of the MAML model, compared to the impact of ϕ on training task performance, the impact of the parameter ϕ_T trained with ϕ on task performance should be of greater concern. In other words, the MAML algorithm is used to find an initial model parameter ϕ that is more adaptable to other tasks.

4. RIOHTRACK STRUCTURE AND DATA SET

4.1. RIOHTrack structure

The data used by the meta-learning model in this experiment is the AP DBA. The measured data about the DBA is derived from the Ministry of Transport's full-scale pavement test loop project. This project is situated in Beijing and encompasses 25 AP structures. It has a full-scale on-site acceleration road test track called RIOHTrack, which is 2.038 km long. The arrangement of these pavement structures is illustrated in Figure 6.

The data used in this article was measured from different pavement structures. A total of 19 primary experimental pavement structures were established on the test circuit to investigate and compare the long-term performance and evolution of AC structures with varying combinations of structural stiffness. The asphalt concrete layer thickness for these pavement structures ranges from 12 to 48 cm (or 52 cm), covering the spectrum of asphalt concrete layer thicknesses found in highways across China, including the thickness of flexible base layers for thick AP.

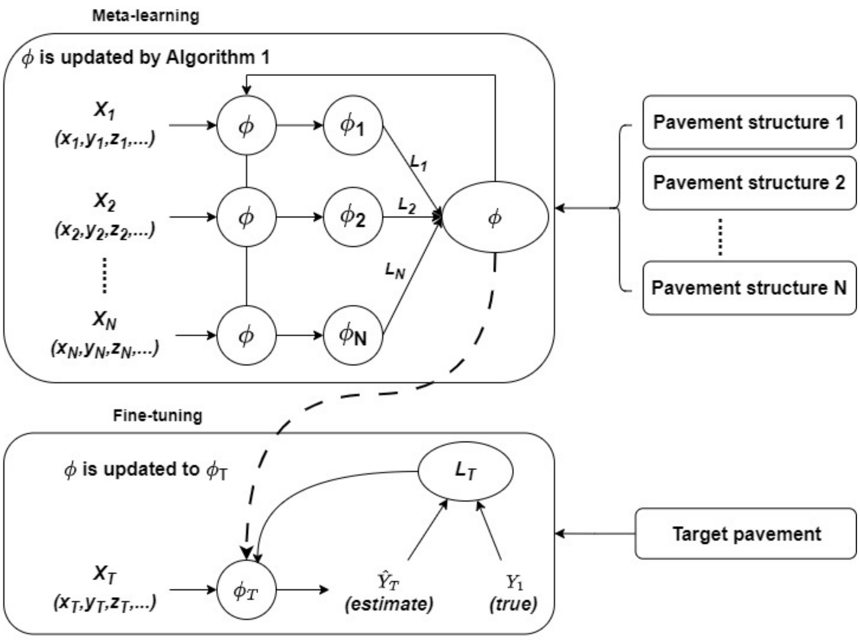

Figure 5. Meta learning method for the performance prediction of the pavement.

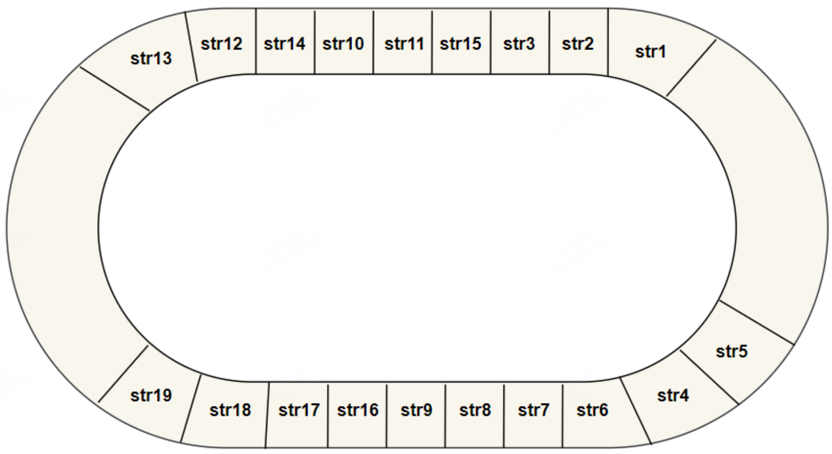

Figure 6. RIOHTRACK test loop pavement.

4.2. Acquisition of datasets

The DBA is measured using a FWD, which mainly consists of a heavyweight (falling weight) and a measuring instrument (usually a level or optical instrument). The FWD is dropped freely from a certain height, causing it to impact the pavement. After that, the pavement undergoes a slight deformation, forming a depression area, the settlement basin. Based on the shape and size of the observed depression area, mathematical formulas or calculation methods can be used to determine the area of the settlement basin.

The measured settlement basin area data has the following characteristics: axle load times, load level, pavement temperature, atmospheric temperature, etc. The data used in the article comes from the DBA measured at pressures of 5, 7, 9, and 11 tons in multiple cycles, respectively. Taking the data of STR1 pavement structure under a pressure of 5 tons as an example, the partial data of the settlement basin area measured by the FWD are shown in Table 1. These variables were selected based on their direct relevance to pavement performance and

Table 1. The pavement feature data(part)

Cycle number	Axle load times	Load level	Pavement temperature (⊠)	Atmospheric temperature (⊠)
N1	4.4533	56.47	12.77	9.17
N2	5.0184	56.47	8.05	11.34
N3	5.4729	56.47	11.66	7.81
N4	5.6846	56.47	26.92	21.85
N5	5.8265	56.47	30.2	25.32
...	...	...	...	...
N111	7.7491	48.09	1.75	3.34

degradation. Axle load times and load level are crucial as traffic load frequency and intensity affect pavement durability. Pavement temperature influences the stiffness and cracking susceptibility of asphalt, while atmospheric temperature affects thermal expansion and environmental wear, all contributing to the pavement's long-term performance.

This text leverages measurement data from the pavement of 19 types of AP as the primary data source. The objective is to regressively predict the area of a particular pavement type's DBA. Due to the brief construction time of the pavement, the data is limited. The deflection basin measurement data is categorized under four distinct pressure conditions, amounting to only 1,792 data cycles. To address the scarcity, we utilize the data under three pressure conditions as the training set to predict pavement data under the remaining pressure conditions. Traditional machine learning methods are less effective for this regression task due to the limited dataset. To overcome the challenge, this paper employs a meta-learning method, using an additional 18 pavement datasets for meta-training, effectively expanding the dataset. The trained meta-learning model is subsequently used to predict the target pavement, achieving regression prediction of the DBA with only a small quantity of training data. Model evaluation is performed using MSE, Root Mean Squared Error (RMSE), and MAE. These metrics gauge the algorithm's performance by reflecting the differences between actual and predicted values, serving as commonly used performance indicators in regression tasks. Smaller values of these indicators indicate more accurate predictions. The metrics are defined in:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{8}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{9}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{10}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}. \tag{11}$$

5. EXPERIMENTAL RESULTS

This section uses meta-learning to experimentally estimate the DBA of pavement to verify its estimation performance compared to pre-training. It also compares the performance of pre-training with other deep learning models that do not use pre-training. It verifies the impact of pre-training on predicting the DBA of the pavements. The purpose of the experiments in this paper is to utilize measured cyclic data of pavement DBA under

Table 2. Comparison of model performance after 10 to 100 iterations

Iterations	10	20	30	40	50	60	70	80	90	100
MSE	172.38	107.33	106.61	55.66	68.56	36.04	63.76	46.35	49.52	40.10
MAE	9.12	9.41	8.17	5.65	6.62	4.74	6.17	5.09	5.41	4.73

different pressures to predict the trend of rebound DBA changes and its variation trend throughout all cycles at a specific pressure. Specifically, the training involves using DBA cyclic data under pressures of 5, 9, and 11 tons, and the goal is to predict the DBA and its variation trend over the entire cycle at a pressure of 7 tons.

5.1. Performance after a small number of iterations

A meta-learning prediction model trained on multiple pavement data (excluding the target pavement) can predict the target pavement DBA by fine-tuning a small quantity of pavement data in several gradient steps. To assess the effectiveness of meta-learning in DBA prediction, this study conducted a performance comparison based on gradient steps. A limited dataset specific to the target pavement was employed to fine-tune pre-trained DBA prediction models constructed using DNNs. In addition, this study also uses calibrated linear unit activation functions and the Adam optimizer. Multiple error measurement indicators, including the coefficient of determination (R^2), MSE, etc., are used to evaluate the prediction performance of the basin area of AP for meta-learning, pre-training, and traditional deep learning methods without pre-training. The basin area of the AP estimation model is trained using an NVIDIA 3060 graphics processing unit and Pytorch deep learning framework. Taking road surface STR19 as an example, the MAML algorithm is tested and the results are shown in Table 2.

Table 2 illustrates a noteworthy reduction in MSE and MAE after ten to 100 iterations, underscoring that DNN models with meta-learning can rapidly adapt to accurately predict the pavement DBA.

5.2. Prediction of the area of pavement deflection basin

The CNN model used in this experiment consists of one convolutional layer and two fully connected layers. The convolutional layer has 1 input channel, 10 output channels, and a kernel size of 2. The first linear layer has an input dimension of 10 and an output dimension of 10, containing ten neurons. The second linear layer has an input dimension of 10 and an output dimension of 1, containing 1 neuron. The learning rate is 0.001. The GBDT model uses a parameter $n\_estimators$ of 500, which indicates the number of trees, and the learning rate as 0.01. The RF regression model used in this experiment has a parameter n_estimators of 200, the loss function is selected as MSE, max_depth is set to None, and min_samples_split is 2. The pre-trained deep learning models studied include meta-learning and pre-training learning. The Deep Neural Network (DNN) structure studied in this research consists of 40 neurons and three hidden layers. The specific parameters are shown in the Table 3.

The performance of meta-learning in predicting the area of deflection basins using data from different pavement structures. To test the exactitude of meta-learning in predicting the area of deflection basins, experiments are conducted on pre-training data involving various pavement conditions without considering the similarity between targets. Table 4 summarizes the DBA estimation errors for each method. As observed, among the studied methods, the deflection basin area estimation using meta-learning achieves the most accurate results (MSE = 13.262, MAE = 2.854) despite using a pre-trained pavement structure different from the target pavement. Among the compared ML methods, the RF model has the least accurate prediction results, with MSE = 372.982 and MAE = 17.161, as it did not receive sufficient training (only using 424 target data points).

Taking the STR19 road as an example, the MAML algorithm uses the data of STR19 as the target road and performs meta-training on other roads. We intend to forecast the DBA of STR19 within one cycle. The MAML model uses parameters of 0.01 for the inner loop learning rate, 0.001 for the outer loop learning rate, and 1 for

Table 3. Model Parameters Summary

Model	Parameter	Value
CNN	Convolutional layer	1 input channel, 10 output channels, kernel size = 2
	First linear layer	Input dimension = 10, Output dimension = 10
	Second linear layer	Input dimension = 10, Output dimension = 1
	Learning rate	0.001
GBDT	Number of estimators	500
	Learning rate	0.01
RF	Number of estimators	200
	Loss function	MSE (Mean Squared Error)
	Max depth	none
	Min samples split	2
DNN	Number of neurons	40 (3 hidden layers)

Table 4. Comparison of model performance

Model	MSE	MAE	R^2
RF	372.982	17.161	-
GBDT	228.281	14.651	-
CNN	151.741	10.144	0.745
Pre-training	16.645	3.253	0.902
MAML	13.262	2.854	0.922

Table 5. Comparison of model performance

Pavement	STR1	STR2	STR3	STR4	STR5	STR6	STR7	STR8	STR9
MSE	20.132	13.830	57.710	63.206	50.844	63.216	60.360	14.453	36.942
MAE	3.394	2.805	5.677	3.680	3.757	4.622	3.895	3.075	4.765
R^2	0.933	0.843	0.791	0.622	0.604	0.630	0.696	0.855	0.776
Pavement	STR10	STR11	STR12	STR13	STR14	STR15	STR16	STR17	STR18
MSE	56.620	11.873	52.740	50.941	69.873	20.132	39.535	23.012	19.111
MAE	5.658	2.788	5.480	4.917	5.045	3.394	4.946	3.863	5.335
R^2	0.840	0.888	0.644	0.781	0.787	0.933	0.895	0.885	0.914

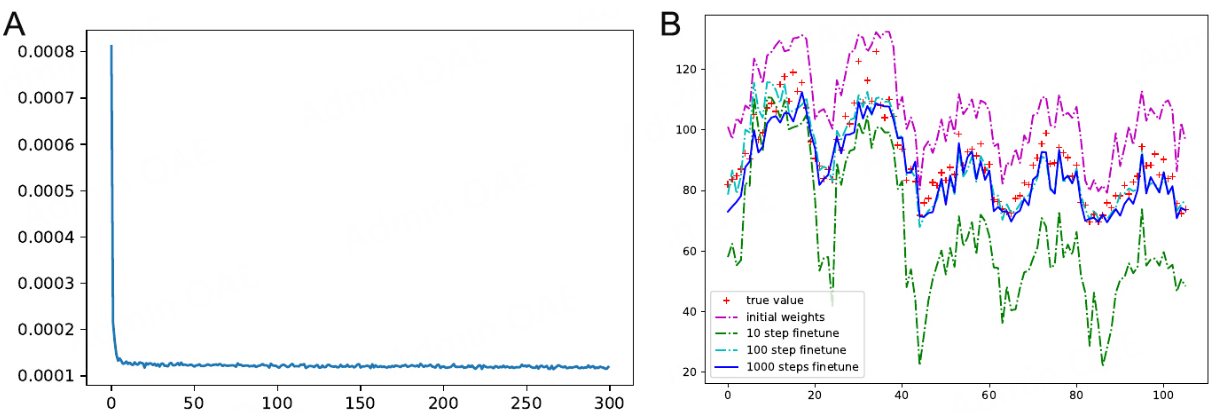

Figure 7. (A) The training loss values of the MAML model, (B) The model prediction performance after different steps.

the inner steps. The experimental results are shown in Figure 7.

5.3. Predictions with various pavement types

This section outlines the prediction performance of meta-learning when utilizing pavement data with diverse structures. To validate the exactitude of meta-learning in predicting the DBA, the subsequent experiments are conducted under the condition that the meta-training data is composed of pavement data with diverse structures without considering the similarity between targets.

This experiment aims to verify that the meta-learning algorithm can have a good estimation performance when facing pavement data with different structures rather than being only applicable to specific data. The specific process of the experiment is similar to Section 5.2. The experiment is conducted on 19 pavements, with one target structure to be predicted and the other 18 pavements as the training data for meta-learning.

Table 5 lists the DBA prediction results for 18 different pavement structures. The MSE and MAE are significantly lower than those of the CNN, GBDT, and other models used in the comparative experiments. The estimation performance of the meta-learning algorithm is superior. Apart from the different data on the target pavement, the same conditions are applied in this experiment. As shown in Figure 8A-R, the fitting and estimation effects for different structures of the pavement are generally sound. Therefore, it is concluded that the meta-learning method could achieve good estimation performance for different types of pavement structures.

6. CONCLUSIONS

This paper proposes a meta-learning-based approach to predict the DBA of AP. By leveraging data from multiple pavement structures, excluding the target pavement, the proposed method significantly reduces the required training data volume, enabling accurate DBA prediction with small sample sizes. Traditional machine learning methods, which rely on training with data from a single pavement type, are less effective in estimating the DBA of the target pavement. In contrast, the proposed meta-learning method effectively reuses data from different pavement structures to pre-train deep learning models, improving the model's generalization capabilities.

Through experiments, we demonstrated that the meta-learning approach outperforms traditional machine learning, neural networks, and pre-training methods in predicting the DBA of APs. The results show that the proposed method is effective and resource-efficient for estimating pavement performance, especially in the early stages of pavement development.

However, the study also has limitations. One potential limitation is the impact of sample size on the model's prediction accuracy. Further research could explore how variations in sample size affect model performance and investigate ways to mitigate the effects of limited data. Additionally, the applicability of the proposed method to different pavement types and conditions remains an area for future exploration. Future studies could also focus on extending the model's capabilities to include real-time data integration and testing in various environmental conditions.

DECLARATIONS

Authors' contributions

Methodology, validation, visualization, and writing-original draft: Li Z, Jin X
Conceptualization, writing-reviewing, supervision, and editing: Shi X, Cao J

Availability of data and materials

The data cannot be shared publicly as the partner (company) does not permit public disclosure. It is available from the corresponding author upon reasonable request.

Financial support and sponsorship

This work was supported by the National Key Research and Development Project of China under Grant (No. 2020YFA0714300) and the Open Project of Nanjing Modern Multimodal Transportation Laboratory (MTF2023004).

Conflicts of interest
Cao J is an Advisory Board Member of the journal *Complex Engineering Systems*. Shi X is an Junior Editorial Board Member of the same journal. Cao J and Shi X were not involved in any steps of editorial processing, notably including the selection of reviewers, manuscript handling and decision-making, while the other authors have declared that they have no conflicts of interest.

Ethical approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Copyright
© The Author(s) 2024.

REFERENCES
1. Wang X. Design of pavement structure and material for full-scale test track. *J Highw Transp Res Dev* 2017;34:30-7. DOI
2. Fu G, Zhao Y, Zhou C, Liu W. Determination of effective frequency range excited by falling weight deflectometer loading history for asphalt pavement. *Constr Build Mater* 2020;235:117792. DOI
3. Habbouche J, Hajj EY, Sebaaly PE, Morian NE. Damage assessment for M-E rehabilitation design of modified asphalt pavements: challenges and findings. *Transp Res Rec* 2018;2672:228-41. DOI
4. Huang W, Liang S, Wei Y. Surface deflection-based reliability analysis of asphalt pavement design. *Sci China Technol Sci* 2020;63:1824-36. DOI
5. Vyas V, Singh AP, Srivastava A. Prediction of asphalt pavement condition using FWD deflection basin parameters and artificial neural networks. *Road Mater Pavement Des* 2021;22:2748-66. DOI
6. Liu H, Cao J, Huang W, Shi X, Zhou X, Li Z. A multidimensional framework for asphalt pavement evaluation based on multilayer network representation learning: a case study in RIOHTrack. *Expert Syst Appl* 2024;237:121370. DOI
7. Wang W, Zhang H, Li T, et al. An interpretable model for short term traffic flow prediction. *Math Comput Simul* 2019;171:264-78. DOI
8. Li Z, Korovin I, Shi X, Gorbachev S, Gorbacheva N, Huang W. A data-driven rutting depth short-time prediction model with metaheuristic optimization for asphalt pavements based on RIOHTrack. *IEEE/CAA J Autom Sin* 2023;10:1918-32. DOI
9. Li Z, Cao J, Shi X, Huang W. QPSO-AHES-RC: a hybrid learning model for short-term traffic flow prediction. *Soft Comput* 2023;97:9347-66. DOI
10. Yang Y, Li Z, Chen J, Liu Z, Cao J. TRELM-DROP: an impavement non-iterative algorithm for traffic flow forecast. *Physica A* 2024;633:129337. DOI
11. Pan JS, Zhang Z, Chu SC, Zhang SQ, Wu JMT. A parallel compact marine predators algorithm applied in time series prediction of Backpropagation neural network (BNN) and engineering optimization. *Math Comput Simulat* 2024;220:65-88. DOI
12. Saltan M, Terzi S. Modeling deflection basin using artificial neural networks with cross-validation technique in backcalculating flexible pavement layer moduli. *Adv Eng Softw* 2008;39:588-92. DOI
13. Lv Y, Duan Y, Kang W, Li Z, Wang FY. Traffic flow prediction with big data: a deep learning approach. *IEEE Trans Intell Transp Syst* 2015;16:865-73. DOI
14. Ma X, Tao Z, Wang Y, Yu H, Wang Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp Res Part C Emer Technol* 2015;54:187-97. DOI
15. Yang Q, Deng Y. Evaluation of cracking in asphalt pavement with stabilized base course based on statistical pattern recognition. *Int J Pavement Eng* 2019;20:417-24. DOI
16. Karballaeezadeh N, Ghasemzadeh Tehrani H, Mohammadzadeh Shadmehri D, Shamshirband S. Estimation of flexible pavement structural capacity using machine learning techniques. *Front Struct Civ Eng* 2020;14:1083-96. DOI
17. Tang X, Yang X. Inverse analysis of pavement structural properties based on dynamic finite element modeling and genetic algorithm. *Int J Transp Sci Technol* 2013;2:15-30. DOI
18. Mabrouk GM, Elbagalati OS, Dessouky S, Fuentes L, Walubita LF. Using ANN modeling for pavement layer moduli backcalculation as a function of traffic speed deflections. *Constr Build Mater* 2022;315:125736. DOI
19. Zhang J, Cao J, Huang W, Shi X, Ji X, Zhou X. A hybrid framework for asphalt pavement rutting prediction modeling and influencing factors analysis based on multilevel wavelet decomposition and transfer entropy. *Appl Math Model* 2023;121:714-30. DOI
20. Li M, Dai Q, Su P, You Z, Ma Y. Surface layer modulus prediction of asphalt pavement based on LTPP database and machine learning for Mechanical-Empirical rehabilitation design applications. *Constr Build Mater* 2022;344:128303. DOI
21. Li Z, Shi X, Cao J, Wang X, Huang W. CPSO-XGBoost segmented regression model for asphalt pavement deflection basin area prediction. *Sci China Technol Sci* 2022;65:1470-81. DOI

22. Zhang J, Cao J, Huang W, Shi X, Zhou X. Rutting prediction and analysis of influence factors based on multivariate transfer entropy and graph neural networks. *Neural Netw* 2023;157:26-38. DOI
23. Shen C, Jiang B, Yue L. LSTM combined with BIM technology in the management of small and medium-sized span highway concrete beam bridges. *Results Eng* 2023;20:101539. DOI
24. Sharma A, Sachdeva SN, Aggarwal P. Predicting IRI using machine learning techniques. *Int J Pavement Res Technol* 2023;16:128–37. DOI
25. Tommasi T, Orabona F, Caputo B. Learning categories from few examples with multi model knowledge transfer. *IEEE Trans Pattern Anal Mach Intell* 2013;36:928-41. DOI
26. Pan SJ, Yang Q. A survey on transfer learning. *IEEE Trans Knowl Data Eng* 2010;22:1345-59. DOI
27. Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv* 2017. pp. 1126-35. DOI
28. Vinyals O, Blundell C, Lillicrap T, Kavukcuoglu K, Wierstra D. Matching networks for one shot learning. *arXiv* 2016. DOI
29. Snell J, Swersky K, Zemel RS. Prototypical networks for few-shot learning. *arXiv* 2017. DOI
30. Santoro A, Bartunov S, Botvinick M, Wierstra D, Lillicrap T. Meta-learning with memory-augmented neural networks; 2016. Available from: https://proceedings.mlr.press/v48/santoro16.pdf [Last accessed on 28 Dec 2024]
31. Ravi S, Larochelle H. Optimization as a model for few-shot learning; 2016. Available from: https://openreview.net/forum?id=rJY0-Kcll [Last accessed on 28 Dec 2024].
32. Nichol A, Schulman J. Reptile: a scalable metalearning algorithm. *arXiv* 2018. Available from: https://yobibyte.github.io/files/paper_notes/Reptile___a_Scalable_Metalearning_Algorithm__Alex_Nichol_and_John_Schulman__2018.pdf [Last accessed on 28 Dec 2024]
33. Chu X, Cai F, Cui C, Hu M, Li L, Qin Q. Adaptive recommendation model using meta-learning for population-based algorithms. *Inf Sci* 2019;476:192-210. DOI
34. Yu H, Zhang Q, Liu T, Lu J, Wen Y, Zhang G. Meta-ADD: a meta-learning based pre-trained model for concept drift active detection. *Inf Sci* 2022;608:996-1009. DOI
35. Lee Y, Choi S. Gradient-based meta-learning with learned layerwise metric and subspace. *arXiv* 2018. pp. 2927-36. DOI
36. Gordon J, Bronskill J, Bauer M, Nowozin S, Turner RE. Meta-learning probabilistic inference for prediction. *arXiv* 2018. DOI
37. Bertinetto L, Henriques JF, Torr PHS, Vedaldi A. Meta-learning with differentiable closed-form solvers. *arXiv* 2018. DOI
38. Lee K, Maji S, Ravichandran A, Soatto S. Meta-learning with differentiable convex optimization. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 15-20 June 2019; Long Beach, CA, USA. pp. 10657-65. DOI
39. Aguiar GJ, Santana EJ, de Carvalho ACPFL, Barbon Junior S. Using meta-learning for multi-target regression. *Inf Sci* 2022;584:665-84. DOI
40. Jeong D, Bae S. Estimating battery state-of-charge with a few target training data by meta-learning. *J Power Sources* 2023;553:232238. DOI
41. Antoniou A, Edwards H, Storkey A. How to train your MAML. *arXiv* 2018. DOI
42. Finn C, Xu K, Levine S. Probabilistic model-agnostic meta-learning. *arXiv* 2018. DOI
43. Grant E, Finn C, Levine S, Darrell T, Griffiths T. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv* 2018. DOI
44. Rusu AA, Rao D, Sygnowski J, et al. Meta-learning with latent embedding optimization. *arXiv* 2018. DOI
45. Morcos AS, Raghu M, Bengio S. Insights on representational similarity in neural networks with canonical correlation. *arXiv* 2018. DOI
46. Kornblith S, Norouzi M, Lee H, Hinton G. Similarity of neural network representations revisited. *arXiv* 2019. pp. 3519-29. DOI
47. Yang J, Wang X, Luo Z. Few-shot remaining useful life prediction based on meta-learning with deep sparse kernel network. *Inf Sci* 2024;653:119795. DOI
48. Shafahi A, Saadatpanah P, Zhu C, et al. Adversarially robust transfer learning. *arXiv* 2019. DOI
49. Wong E, Rice L, Kolter JZ. Fast is better than free: Revisiting adversarial training. *arXiv* 2020. DOI
50. Andriushchenko M, Flammarion N. Understanding and improving fast adversarial training. *arXiv* 2020. DOI
51. Carmon Y, Raghunathan A, Schmidt L, Liang P, Duchi JC. Unlabeled data improves adversarial robustness. *arXiv* 2019. DOI
52. Uesato J, Alayrac JB, Huang PS, Stanforth R, Fawzi A, Kohli P. Are labels required for improving adversarial robustness? *arXiv* 2019. DOI
53. Chan A, Tay Y, Ong YS. What it thinks is important is important: Robustness transfers through input gradients. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 13-19 June 2020; Seattle, WA, USA. pp. 332-41. DOI
54. Utrera F, Kravitz E, Erichson NB, Khanna R, Mahoney MW. Adversarially-trained deep nets transfer better: Illustration on image classification. *arXiv* 2020. DOI
55. Salman H, Ilyas A, Engstrom L, Kapoor A, Madry A. Do adversarially robust imagenet models transfer better? *arXiv* 2020. DOI
56. Breiman L. Random forests. *Mach Learn* 2001;45:5-32. DOI
57. Friedman JH. Greedy function approximation: a gradient boosting machine. *Ann Statist* 2001;29:1189-232. DOI

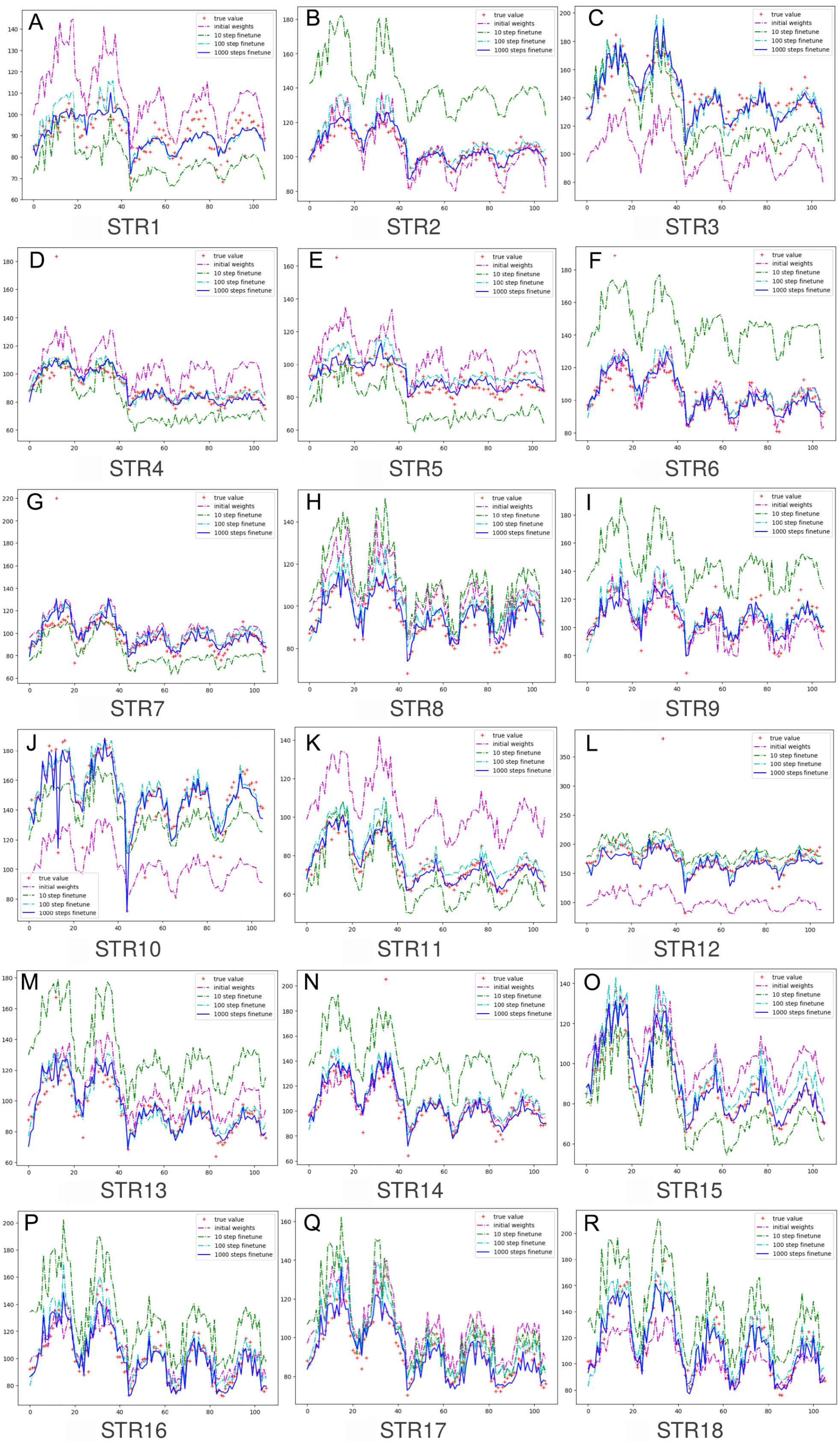

Figure 8. The results of estimating the deflection basin area of the remaining 18 pavements through meta-learning.