Hindawi
Advances in Civil Engineering
Volume 2020, Article ID 8824135, 21 pages
https://doi.org/10.1155/2020/8824135

Research Article

Prediction of Highway Tunnel Pavement Performance Based on Digital Twin and Multiple Time Series Stacking

Gang Yu,[1,2] **Shuang Zhang**,[1,2] **Min Hu**,[1,2] **and Y. Ken Wang**[3]

[1]*SILC Business School, Shanghai University, Shanghai 201800, China*
[2]*SHU-SUCG Research Centre for Building Industrialization, Shanghai University, Shanghai 200072, China*
[3]*Computer Information Systems and Technology, University of Pittsburgh, Pittsburgh 15260, PA, USA*

Correspondence should be addressed to Shuang Zhang; zs_melodie@shu.edu.cn

Received 6 August 2020; Revised 3 November 2020; Accepted 10 November 2020; Published 4 December 2020

Academic Editor: Jia-Rui Lin

The existing pavement performance prediction methods are limited to single-factor predictions, which often face the challenges of high cost, low efficiency, and poor accuracy. It is difficult to simultaneously solve the temporal, spatial, and exogenous dependencies between pavement performance data and maintenance, the service life of highways, the environment, and other factors. Digital twin technology based on the building information modeling (BIM) model, combined with machine learning, puts forward a new perspective and method for the accurate and timely prediction of pavement performance. In this paper, we propose a highway tunnel pavement performance prediction approach based on a digital twin and multiple time series stacking (MTSS). This paper (1) establishes an MTSS prediction model with heterogeneous stacking of eXtreme gradient boosting (XGBoost), the artificial neural network (ANN), random forest (RF), ridge regression, and support vector regression (SVR) component learners after exploratory data analysis (EDA); (2) proposes a method based on multiple time series feature extraction to accurately predict the pavement performance change trend, using the highway segment as the minimum computing unit and considering multiple factors; (3) uses grid search with the *k*-fold cross validation method to optimize hyperparameters to ensure the robustness, stability, and generalization ability of the prediction model; and (4) constructs a digital twin for pavement performance prediction to realize the real-time dynamic evolution of prediction. The method proposed in this study is applied in the life cycle management of the Dalian highway-crossing tunnel in Shanghai, China. A dataset covering 2010–2019 is collected for real-time prediction of the pavement performance. The prediction accuracy evaluation shows that the mean absolute error (MAE) is 0.1314, the root mean squared error (RMSE) is 0.0386, the mean absolute percentage error (MAPE) is 5.10%, and the accuracy is 94.90%. Its overall performance is better than a single model. The results verify that the prediction method based on digital twin and MTSS is feasible and effective in the highway tunnel pavement performance prediction.

1. Introduction

In early 2020, China proposed a plan to accelerate the modernization of urban governance with advanced application of technologies, such as the Internet, big data, and artificial intelligence to support the transformation and upgrading of traditional infrastructures, and merge into innovatively infrastructures, such as intelligent transportation infrastructures. According to government statistics, as of the end of 2019, the summed length of highway tunnels of China reached 18.9666 million meters. To comprehensively improve the management level of urban highway tunnels, academia and industry are paying great attention to the life cycle management of urban highway tunnels based on digital twins. A digital twin, proposed by Michael Grieves in 2002, is a mapping relationship between the physical space and virtual space and is intelligent. It maps, records, simulates, predicts, and manages the running track of the life cycle of objects in the physical world and digital virtual space. The life cycle management of urban highway tunnels emphasizes the transformation from traditional corrective maintenance to data-driven preventive maintenance, in which the accurate and timely prediction of highway pavement performance plays a crucial role.

There are two types of research methods in the field of pavement performance prediction. One type is based on traditional statistical models, and the other type is derived from modern machine learning models. Methods based on traditional statistical models are relatively extensive, including deterministic methods such as mechanical prediction models and mechanical-empirical prediction models [1–6], as well as probabilistic methods such as Markov probability models and Bayesian probability models [7–10]. The deterministic model is based on empirical classification, mechanical analysis, or mechanism analysis to predict the pavement performance. Eghbalpoor [1] et al. proposed a three-dimensional finite element model of asphalt pavement to study the damage to the pavement caused by high cyclic loads and their healing effect. Zhang [2] et al. proposed a research method based on morphological dynamics to predict the temperature stress change trend of long-term aging asphalt pavement materials. Shirzad [4] et al. explored the influence of the asphalt mixture ratio on pavement performance and established a mechanical experience model to predict rutting. The probabilistic model is an empirical model dominated by a large amount of measured data. Bianchini [8] et al. proposed an IF-THEN fuzzy neural inference model to predict pavement performance. Li [11] et al. used the method of *K*-means and type-2 fuzzy set to predict the long-term traffic volume. Chen [7] et al. used a Markov chain Monte Carlo (MCMC) simulation to analyze the failure probability of pavement prevention and treatment. Abed [10] et al. used the Monte Carlo simulation method to predict the change probability of pavement performance. The existing traditional statistical models are limited to single-factor predictions, in which single independent variables can hardly reflect the essence of pavement performance. They usually focus on the statistical inference and interpretability of a small sample under certain hypothesis testing conditions. They usually use a small sample size, mainly to represent the relationship between the data and the result variables and the importance of these relationships, and the calculation accuracy is limited. With the increase in the service life of urban highway tunnels, pavement performance is subjected to long-term coupling effects of loads, maintenance situations, environmental conditions, material properties, and other factors, and the resulting pavement diseases aggravate the uncertainty of the driving comfort and safety of the pavement. The predictive effect of traditional statistical methods is sometimes suboptimal in practical engineering.

Studies have shown that machine learning models are superior to traditional statistical methods, such as curve fitting, exponential smoothing, and the autoregressive moving average methods. The accuracy of prediction can be optimized through feature selection, hyperparameter tuning, and integrated learning technology. Gong [12] et al. developed a random forests regression (RFR) model to predict the international roughness index (IRI) performance of flexible pavement based on long-term historical time series data information of disasters, traffic, gas, and structure. The comparison found that it was significantly better than the regularized linear regression model. Xiao and Nie [13] used the regression model and time series GM (1, 1) to predict pavement performance. The maximum difference between the time series predicted value and the regression model predicted value can be reduced by 1%. Bi [14] et al. proposed a time series algorithm suitable for single-factor distributed version to predict the workload of the next stage. Lintonen and Raty [15] proposed a new self-learning stopping criterion for multivariate time series which is a positive-unlabeled (PU) learning case. Liu [16] et al. improved the accuracy of pavement temperature prediction by using a gradient boosting extreme learning machine (GBELM). Gong [17, 18] considered parameters such as materials, structure, traffic, and environment and successfully developed the Mechanistic-Empirical Pavement Design Guide (MEPDG) deep neural network model and GBM method for pavement rutting prediction, and the accuracy was improved. Hu [19] obtained risk indicators from a neural network prediction model and found that the application of data analysis improved the similarity of the predictive performance of the overall pavement safety level. Wang [20] et al. proposed a training method to automatically determine the parameters of network structure for the deep belief network (DBN) with transfer learning (TL-GDBN). Gao [21] et al. proposed 6 non-BP swarm intelligence search algorithms (BBO, PSO, GA, ACO, ES, and PBOL) for model training and developed a new dendritic neuron model (DNM) that considers nonlinearity. Wang and Kumbasar [22] used two swarm intelligence algorithms (PSO and BBBC) to optimize parameters. The machine learning model provides a new method for effectively dealing with multiple factors, ambiguity, randomness, and other issues. It can consider multiple factors and processes high-dimensional data at the same time. It focuses more on the balance between accuracy and complexity. The training set learns the predictive model, and the testset evaluates and verifies its accuracy. No hypothesis conditions are required. The larger the sample size is, the faster the model converges. The generalization integration framework [23], proposed by Wolpert in 1992, is a serial multilevel learning structure with flexibility and uncertainty that can achieve both bagging integration and boosting integration. Lin [24] et al. proposed a stacking model (SMVP) based on real datasets for predicting changes in public bicycle traffic flow. Compared with the traditional single model, the certainty level is increased by 25.58%. Zhai [25, 26] et al. predicted and analyzed the average daily polyvinyl acetate (PM2.5) concentration in Beijing, China, and developed a stacked ensemble model, including the stacking of LASSO, AdaBoost, XGBoost, the GA-MLP-optimized multilayer perceptron, and the support vector regression model (SVR). The results showed that the stacked combination model had high interpretability. Jiang [27] et al. also proposed a stacked ensemble model, which used RF, ERT, XGBoost, LightGBM, RNN, bidirectional RNN, LSTM, and GRU as basic learners, and used logistic regression as the second-level metalearner. In different application scenarios, the stacked models constructed by researchers such as Li [28], Yang [29], and Feng [30] et al. had good results, suggesting an idea for

establishing stacked models to predict pavement performance in the field of urban highway tunnel pavement performance prediction.

In the field of architecture, engineering, construction (AEC), BIM has proved to be an intelligent and parametric digital modeling method that can support the full life cycle of buildings. Since 2019, researchers have begun to conduct applied research on digital twins in the AEC field based on the BIM model [31, 32]. A digital twin refers to a virtual model that is completely consistent with physical entities in the real world and can simulate an entity's behavior and performance in a real environment in real-time. Digital twin technology has the characteristics of traceability, high integration, real-time, and high precision [33, 34]. In the design phase, digital twins can be used for model design, collision detection, pipeline synthesis, and hardcover design. Lu [35] et al. proposed a semiautomatic digital twin system based on images and CAD drawings, aiming at the time-consuming problem of BIM construction. In the construction phase, digital twins can be used for cost budgeting [36], quality management [37], construction collaboration [38], and schedule management [39]. In the operation and maintenance phase, digital twins can be used for equipment asset management [40], safety and prevention management, building space management, and building environment management. Shim [40, 41] et al. proposed a bridge maintenance program based on the concept of digital twins. The program combines a 3D information model-based maintenance information management system with a digital detection system that uses image processing to enhance the decision-making process in terms of the bridge maintenance process reliability.

Therefore, this paper proposes a tunnel pavement performance prediction method based on a digital twin + MTSS for the life cycle operation and maintenance management of urban highway tunnels. Considering many factors, such as the maintenance condition, the service life of highways, and the environment, along with the actual pavement performance data of urban highway tunnels, this method adopts machine learning methods to improve the prediction accuracy, which is conducive to extending the service life of urban highway tunnel pavement, and provides support for operational management. The remainder of this paper is organized as follows: the second section introduces the digital twin for pavement performance prediction and the MTSS prediction model. The third section applies the proposed method in practical engineering projects and compares it with the single model to evaluate the performance of the MTSS model and verifies the applicability of the method. Finally, the fourth section summarizes the whole paper.

2. Theoretical Basis of Regression Models

2.1. EXtreme Gradient Boosting (XGBoost). XGBoost is a highly scalable gradient boosted decision tree system. It has the advantages of good robustness and strong scalability. In recent years, it has shown excellent performance in many fields, such as information technology and software engineering [42], environmental science [30], and economics and finance [36]. XGBoost performs a second-order Taylor expansion on the loss function and uses a quadratic function to approximate the loss function. To prevent overfitting, regularization is introduced into the loss function to constrain the number and weight of leaf nodes, balance the complexity of the model, and optimize each iteration until the following equation is minimized [43].

$$L^{(t)} \simeq \sum_{i=1}^{n} \Big[l\big(y_i, \widehat{y}^{(t-1)}\big) + \partial_{\widehat{y}^{(t-1)}} l\big(y_i, \widehat{y}^{(t-1)}\big) f_t(x_i) + \frac{1}{2} \partial^2_{\widehat{y}^{(t-1)}} l\big(y_i, \widehat{y}^{(t-1)}\big) f_t^2(x_i) \Big] + \Omega(f_t), \tag{1}$$

where $\Omega(f_t) = \gamma T + (1/2)\lambda \|\omega\|^2$, $\sum_{i=1}^{n} l(y_i, \widehat{y}_i)$ is the loss function, $\Omega(f_t)$ is the regularization term, T is the number of leaf nodes, and γ and λ are the regularization parameters.

2.2. Gradient Boosting Decision Tree (GBDT). GBDT is an ensemble method of boosting. It uses the CART decision tree as the basic learner. The GBDT is different from RF that RF can be generated in parallel by the voting method, while GBDT is an ensemble of weight-based weak classifiers, which is the accumulation of the branch results of each tree, and can be generated only serially. Therefore, the GBDT is more concerned with reducing the bias in the training process, while the RF is more concerned with reducing the variance in the training process. At the same time, GBDT has good robustness and high scalability [44].

2.3. Light Gradient Boosting Machine (LightGBM). LightGBM is an open-source distributed gradient boosting framework based on the decision tree algorithm developed by the Microsoft Research Asia Distributed Machine Learning Toolkit (DMTK) team [45]. It is an improved decision tree integration model based on GBDT and XGBoost [46]. Two aspects are improved. (1) Gradient-based one-sided sampling (GOSS): GOSS algorithm sorts the gradients. Samples with a large gradient are selected by percentage, and those with a small gradient are randomly selected to optimize the sampling of the training sample set and reduce the sample points when calculating the gradient. (2) Exclusive feature bundling (EFB): mutually exclusive features are bound to reduce the feature dimension when selecting split points. In a data environment with large training samples and high-dimensional features, it exhibits a faster training speed and efficiency, requires lower memory usage, offers better accuracy, can handle larger-scale data, and supports parallel learning.

2.4. Random Forest (RF). RF is an ensemble bagging model proposed by Breiman [47] that combines the CART tree and random space method without assuming a prior distribution. Each tree is built by the independent random sampling method. The original data subset is sampled and generated by the bootstrap method, and a random subset of the feature attributes is randomly choosed according to a specific

criterion. The selection of the training set is random, and each training set is independent of the others. Random forests generally perform better than single decision trees because the results of random forests are determined by voting on the results of multiple decision trees. Each decision tree in a random forest has its own result, and the random forest selects the result with the largest number of votes as its final result by counting the results of each decision tree.

2.5. Linear Regression. The linear regression method assumes that the relationship between independent variables and target variables is linear. This simplicity often makes linear regression the best choice for small sample analysis and also makes these models relatively easy to interpret and understand, not suitable for too many predictive variables [48].

2.6. Least Absolute Shrinkage and Selection Operator (LASSO). The least absolute shrinkage and selection operator (LASSO) is a regression analysis method with strong robustness and a strong generalization ability for simultaneous feature selection and regularization. The $L1$ norm is introduced to generate sparse solutions at the feature level, which will shrink the unimportant coefficients in the parameters to be estimated, so as to reduce redundancy and make the model more stable when processing different data, thus improving the accuracy and robustness of regression prediction [25].

2.7. Ridge Regression. Ridge differs from LASSO in terms of regularization items, and Ridge introduces the $L2$ norm. Ridge regularization uses the sum of squares of regression coefficients as a penalty function and compresses regression coefficients, making it difficult to compress the coefficients of redundant predictive variables to 0 and easier to over-compress more important regression coefficients [49].

2.8. Artificial Neural Network (ANN). The ANN structure usually consists of three main parts: the input layer, the hidden layer, and the output layer. Each layer is composed of many neurons. The input layer is used to load known parameters and directly determine the output results [25], such as performance index and maintenance type in this study. The hidden layer contains all known information and establishes a nonlinear relationship between the input parameters and the output results. The output layer is used to obtain the unknown prediction result Y. As shown in Figure 1, this study has set up 2 hidden layers with 128 neurons in each layer. The selection range of the hidden layers was set as [32, 64, 128], and the nonlinear hyperbolic tangent (tanh) activation function was adopted. The detailed neural network structure diagram is as follows:

2.9. Support Vector Regression (SVR). The core principle of the support vector machine regression algorithm SVR is to find a regression plane so as to minimize the overall observation error and maximize deviation [50]. The traditional regression methods consider the prediction correct if and only when the regression fitting function is completely equal to y. For example, $(f(x)-y)^2$ is commonly used to calculate the loss in linear regression [51]. However, SVR believes that as long as $f(x)$ deviates from y not too much, the prediction can be considered correct. Instead of calculating losses, the threshold a is set to only calculate the loss of the data point $|f(x)-y|>a$. The SVR model is irrelevant of the dimension of the input space and depends on the number of SVs. The goal of SVR is to solve the following basic problems [25]:

$$\min_{\omega,b,\zeta,\zeta^*} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{n}(\zeta+\zeta^*)$$
$$\text{s.t.}\begin{cases} y_i - \omega^T\phi(x_i) - b \le \varepsilon + \zeta_i \\ \omega^T\phi(x_i) + b - y_i \le \varepsilon + \zeta_i^* \\ \zeta_i \ge 0, \zeta_i^* \ge 0, \quad i = 1,2,\ldots,n, \end{cases} \tag{2}$$

where ω and b are used to determine the fitting function f, C is the regularization constant, ζ_i and ζ_i^* are the relaxation variables to define the up and down deviation.

3. Methods

3.1. Digital Twin for Pavement Performance Prediction. This paper proposes a digital twin for the pavement performance prediction of urban highway tunnels. It is driven by the continuous and circular flow of information and data from the physical world to the digital world and then to the physical world. It adopts a new method by the deep integration of global perception technology and machine learning to establish the digital mapping between the virtual pavement BIM model and physical highway entity. It realizes data sharing, visual analysis, and the intelligent decision support functions of pavement performance trend changes. The digital twin consists of three key parts (Figure 2). (1) Data collection module: the data include pavement performance, maintenance records, tunnel highway structure, traffic flow, and tunnel environment. The pavement performance data come from pavement performance indicators, such as vehicle-mounted laser sensors and high-precision accelerometers. The maintenance records are stored in the operation and maintenance management system. (2) Prediction module: within the stacking integrated learning framework of the MTSS, a pavement performance prediction model based on multiple differentiated models is established to predict the future performance trend change of each segment of the pavement. (3) Parametric analysis module: in the Dynamo environment, which supports visual parameter design, the multisource spatial-temporal data of the BIM model and physical entities are integrated, and the inheritance relationship between the node components is constructed. A Python application is implemented with RevitAPI to realize the visual evolution of the pavement performance.

Figure 3 shows the learning process of digital twins. First, the pavement performance data of the physical environment

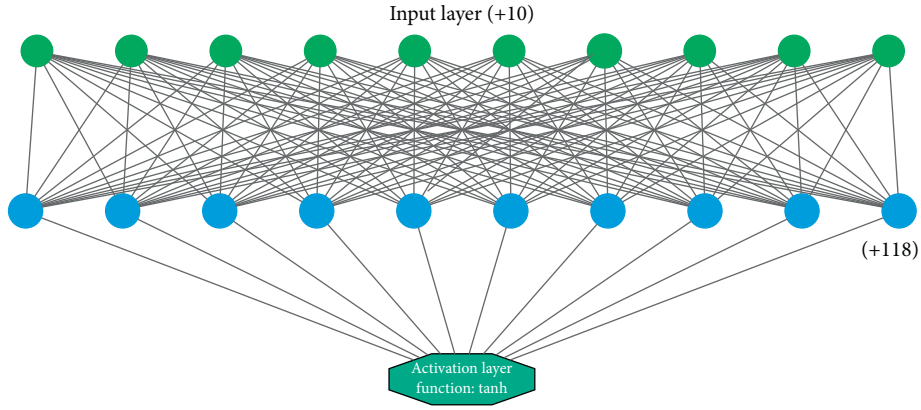


Figure 1: ANN structure diagram.

are collected, and the datasets are divided to build the MTSS prediction model. Then, in the Dynamo environment, the physical pavement entity and the virtual digital BIM model are mapped, the predicted data are synchronized, and the BIM model is assigned. Both the virtual environment and the physical environment contain the bidirectional synchronization process of data acquisition and realization of pavement performance prediction state change [52]. Among them, in the visualization state of pavement performance prediction, the physical pavement entity changes through the digital mapping by using the virtual pavement BIM model. This state change is transferred to the parameter analysis module through the MTSS prediction model. Because the pavement segments may be degraded in the future, the digital twin is used to visualize the decay trend of pavement performance and provide the decision analysis service of the highway tunnel pavement performance prediction and maintenance decision-making by predicting the maintenance timing, analyzing the maintenance cost, and adopting preventive maintenance strategy. The ultimate purpose of all these measures is to extend the life of the highways.

3.2. Multiple Time Series Stacking Prediction Model

3.2.1. Time Series Feature Extraction Based on Multiple Time Series. Feature extraction is very important for the performance and generalization ability of the model. It is used to trim, transform, and refine the input factors, which can improve the interpretability and prediction accuracy of the prediction model [53]. The feature engineering used in this paper includes temporal feature extraction, feature transformation, feature filtering, and feature standardization and one-hot for feature transformation and model-based method for feature filtering [54]. In the preliminary exploratory data analysis (EDA) of the dataset, it is found that the pavement performance indicators, maintenance information, and environmental information dataset have the basic characteristics of the time series. However, it is difficult to process multiple time series in parallel.

The general feature filtering method filters only by the correlation between a single factor and the dependent variable, while the model-based feature filtering method filters by the comprehensive effect of each factor, which can make full use of the combinative characteristics among factors and retain more useful factors. In view of the rapid increase in the number of features as well as feature extraction of time series information, in order to avoid feature collinearity and decreasing of the prediction accuracy by adding useless information, a model-based feature filtering method was adopted to filter the redundant features according to the 95% cumulative contribution threshold (proportion) of the feature factors. Therefore, this paper proposes a method based on multiple time series feature extraction to construct and enhance features and uses a tree-based model to select potentially important features to improve the accuracy of the model.

The time series feature refers to the information contained in the historical sequence. The traditional regression prediction method uses the feature data of the time section at time t to predict the target value at time $t+1$. This method uses only the data of a single time section. However, although the time series method can make use of the characteristics of historical sequences over a period of time, it applies only to the characteristics of a single series and has a strong restriction on the richness of the features. The multiple time series feature extraction method proposed in this paper combines historical sequence information with multiple factors and simultaneously uses the temporal features of multiple factors as the factor input of the regression model, that is, the factor feature data from time $t-m$ to time t are used to predict the target value at time $t+1$. This feature extraction method can effectively combine the features of multiple heterogeneous factors at the same time. Due to the use of temporal features, the change pattern along the time dimension (such as the curvature) of factors can be fully utilized, which can provide richer and more valuable information for prediction and improve the prediction accuracy.

The temporal feature extraction process is shown in Figure 4, where T_j is the target time series, $T_1 - T_{j-1}$ is the characteristic time series, Y_t is the predicted target scalar value of the corresponding sample at time t, and X_t is the factor input of the corresponding sample at time t. X_t is a vector composed of $t-k$ to $t-1$ characteristic data extracted from each characteristic time series by $T_1 - T_{j-1}$

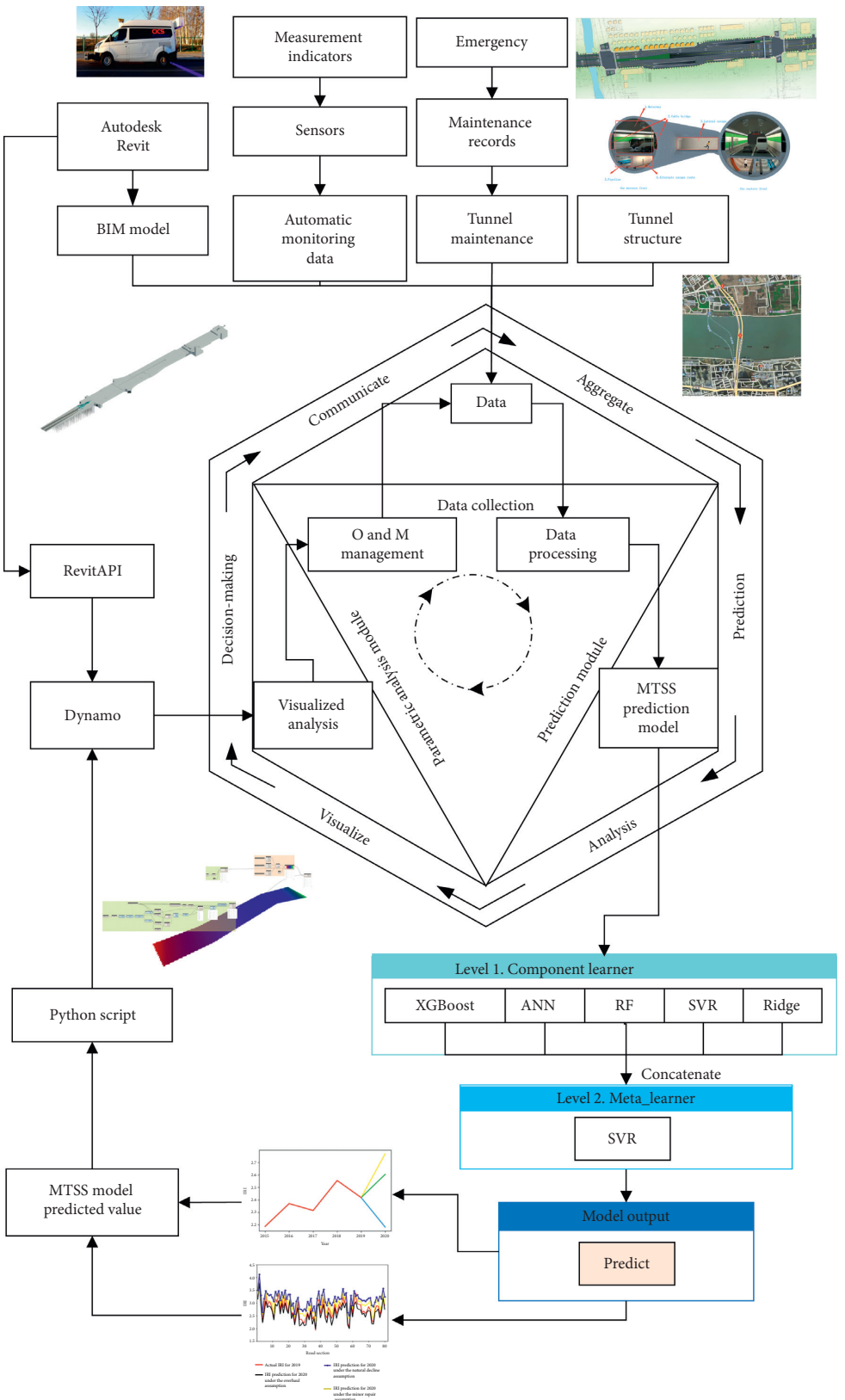


Figure 2: A digital twin for pavement performance prediction.

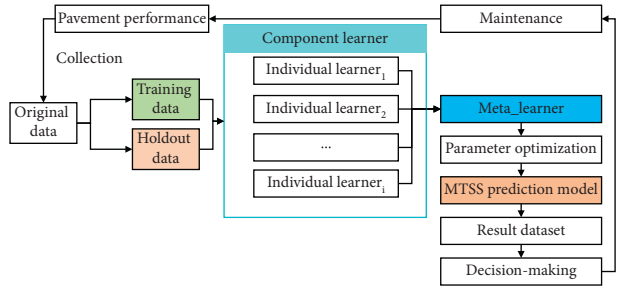
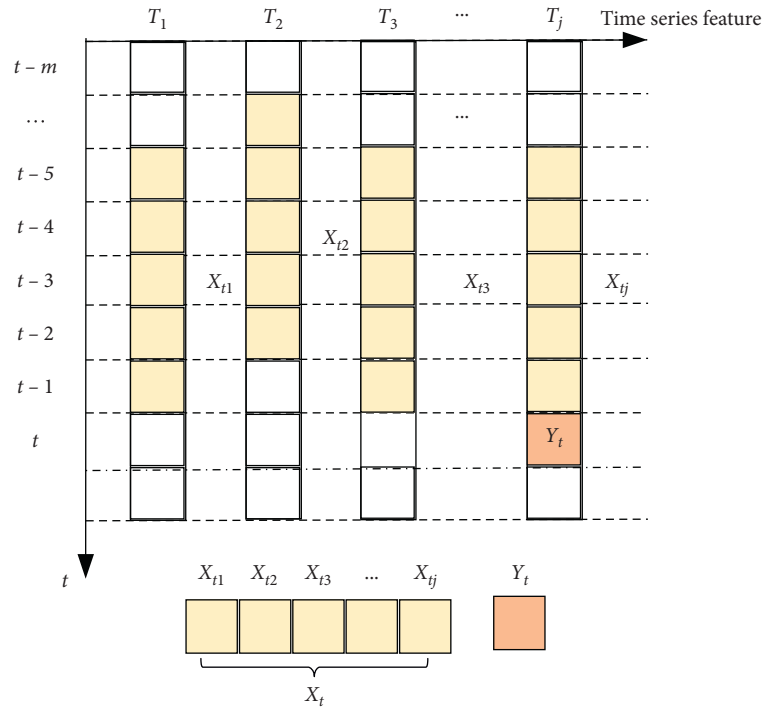


Figure 3: The digital twin learning process.



t_time, $t = 1, 2, ..., n$

m _the service life of a highway, $m = 1, 2, ..., n$

T_j_the j temporal feature, $j = 1, 2, ..., n$

X_{tj}_the feature factor of the j temporal feature at t

Y_t_the predicted value of the pavement performance at t

Figure 4: Time series feature extraction.

characteristic time series. The sample (X_t, Y_t) formed by feature extraction constructs the samples of the regression model.

3.2.2. Filter for Component Learners. In order to make the stacking learning model achieve the optimal prediction effect and improve the robustness of the model, not only the predictive ability of a single individual learner must be considered but also necessary to consider the combined effects of individual learners. Individual learners with stronger learning ability will help improve the overall predictive effect of the model. In the first level, heterogeneous models with a large degree of difference are selected as the component learners, which can maximize the advantages of various algorithms, and at the same time, each differentiated model can learn from each other's strengths [55]. Therefore, this paper conducted a stacking experiment to optimize the stacking method of the component learners when building the model. A total of 9 individual learners, XGBoost, GBDT, LightGBM, RF, Linear, LASSO, Ridge, ANN, and SVR, are initially selected. The essence of the stacking of different models is to observe data from different data space perspectives and data structure perspectives. Based on this principle, base learners should be different from each other. Therefore, this paper uses the Pearson correlation coefficient

to calculate the differences of each model. By selecting algorithms with large differences, we maximize the advantages of different algorithms and obtain the best prediction effect. Pearson correlation formula is shown as equation (3), where X and Y are two single models' outputs.

$$\rho_{X,Y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E\left(X^2\right) - (E(X))^2}\sqrt{E\left(Y^2\right) - (E(Y))^2}}. \quad (3)$$

The stronger the learning ability of each individual learner in stacking and the lower the degree of correlation, the better the prediction effect of the stacked model. The threshold value of correlation between component learners is set to be 0.95. Above the threshold, the correlation is stronger, and less than the threshold, the correlation is weak. As shown in Figure 5, XGBoost, ANN, RF, Ridge, and SVR are selected as the first level of component learners, and SVR as the second level of metalearner. XGBoost has a higher correlation with GBDT and LightGBM, which are both trees boosting algorithms. The correlation between XGBoost and RF is relatively low. This is because although RF is also a tree-based integrated class model, it is a bagging type. The correlations between Ridge and LASSO and linear is strong because Ridge and LASSO are both improved versions of linear regression. ANN is quite different from other models, so the correlation is low.

After the filtering of correlation analysis and combination optimization experiment, as shown in Figure 6, this paper selects XGBoost, ANN, RF, Ridge, and SVR as component learners of level 1 and SVR as metalearner of level 2.

3.2.3. Fusion of the MTSS Prediction Model. This paper first uses the k-fold ($k = 5$) method to train each component learner with the training set and then uses the predicted values from the testset of component learners to train the metalearner. Stacking ensemble learning divides the original dataset $S = \{(X_t, Y_t), t = 1, \ldots, n\}$ into several data subsets, where X_{tj} is the input vector of the sample t with j_{th} temporal features, Y_t is the corresponding predicted value of the sample t, and N is the number of temporal features, that is, each X_t feature vector is $(X_{t1}, X_{t2}, X_{t3}, \ldots, X_{\text{tn}})$. In this paper, the five-fold cross validation method is used. This paper randomly divides the data into five equal size and disjoint subsets $S_1, S_2, S_3, \ldots, S_K$ of the original dataset, where $S_K = S - S_K$ defines S_K and S_{-K} as the k-fold training set and testset, respectively. One of them is taken as the testset, and the other four are used as the training set to train the model each time. MSE_t is calculated based on the testset. Then, the average value is calculated to obtain CV_k:

$$\text{CV}_{(k)} = \frac{1}{k}\sum_{t=1}^{k} \text{MSE}_t. \quad (4)$$

The process of stacking learning is as follows:

Step 1 Divide the original sample data into training data and holdout data and divide the training data into K subsets of basically equal size

Step 2 Uses the k-fold cross validation method to train each individual learner on the training data and validate it on the holdout data

Step 3 Repeat step 2 until all the component learners are trained

Step 4 Use the first level output as the input of the second-level metalearner and train the second-level metalearner, namely, the meta-model (SVR)

As shown in Figure 7, the first level prediction algorithm contains K base learners. The training set S_{-K} is trained with the k_{th} algorithm to obtain the base learner L_k. In k-fold cross validation, for each sample X_{tj} in the k-fold testset S_K, the prediction of the base learner L_k is denoted as Z_{ktj}. After cross validation, the output data of K base learners form a new data sample $S_{\text{new}} = \{((Z_{1tj}, \ldots, Z_{ktj}), Y_t), t = 1, \ldots, k, j = 1, \ldots, N, k = 1, \ldots, K\}$, which is used as the input data of the second level. Train the metalearner L_{new} of the second-level prediction model to detect and correct the prediction errors in the first layer learning algorithm. The SVR model in the second level outputs the final prediction results. The stacking learning framework improves the robustness and generalizability of the prediction model by combining the output predictions of multiple models to obtain an improvement in the overall prediction accuracy. In order to prevent overfitting, the training set of the component learner cannot be directly used in the training of the metalearner. It is necessary to divide the original data reasonably.

3.2.4. Hyperparameter Optimization. The multilevel stacking learning system involves the setting of many parameters. Nevertheless, the combination of parameters of different values has a greater impact on the prediction accuracy of the model, and unreasonable parameter settings can easily cause underfitting or overfitting of the model. Hyperparameter tuning is an optimization method that can be carried out to preprocess or postprocess any learning algorithm. Grid search can improve the generality of the model. To make the MTSS model perform optimally and prevent underfitting or overfitting, it is necessary to optimize the parameters of each base learner and metalearner. First, the single base learner is predicted and analyzed on the original dataset. The regularization technique is used to avoid overfitting. The k-fold cross validation is adopted to divide the dataset into multiple training sets and testsets. The training set is used to construct the model, and the testset is used to evaluate or verify the performance of the model. The structure of the base learner and metalearner algorithm has certain complexity. It is necessary to define many parameters, such as learning_rate, max_depth, and n_estimators. Among the main parameters, learning_rate can control the model weights' updating speed. Max_depth is the maximum depth of a tree. The larger the max_depth is, the more complex the model will be. N_estimators is the number of decision trees. Gamma refers to the complexity penalty term at the level of tree structure. If the value is larger, the number of leaf nodes in the tree will be smaller. Lambda represents the penalty for the output weight of leaf nodes to avoid excessive weight of leaf nodes and

	ANN	GBDT	LASSO	LightGBM	Linear	RF	Ridge	SVR	XGBoost
ANN	1.0000								
GBDT	0.8703	1.0000							
LASSO	0.9004	0.9237	1.0000						
LightGBM	0.8804	0.9635	0.9481	1.0000					
Linear	0.9011	0.9197	0.9975	0.9443	1.0000				
RF	0.8647	0.9439	0.9011	0.9450	0.8972	1.0000			
Ridge	0.9070	0.9245	0.9661	0.9353	0.9616	0.9163	1.0000		
SVR	0.8623	0.8749	0.9553	0.8908	0.9509	0.8595	0.9337	1.0000	
XGBoost	0.8583	0.9515	0.9161	0.9562	0.9126	0.9197	0.8991	0.8576	1.0000

Figure 5: Correlation analysis.

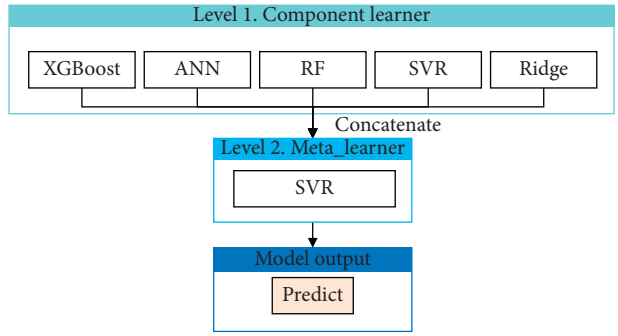


Figure 6: Construction of the integration model.

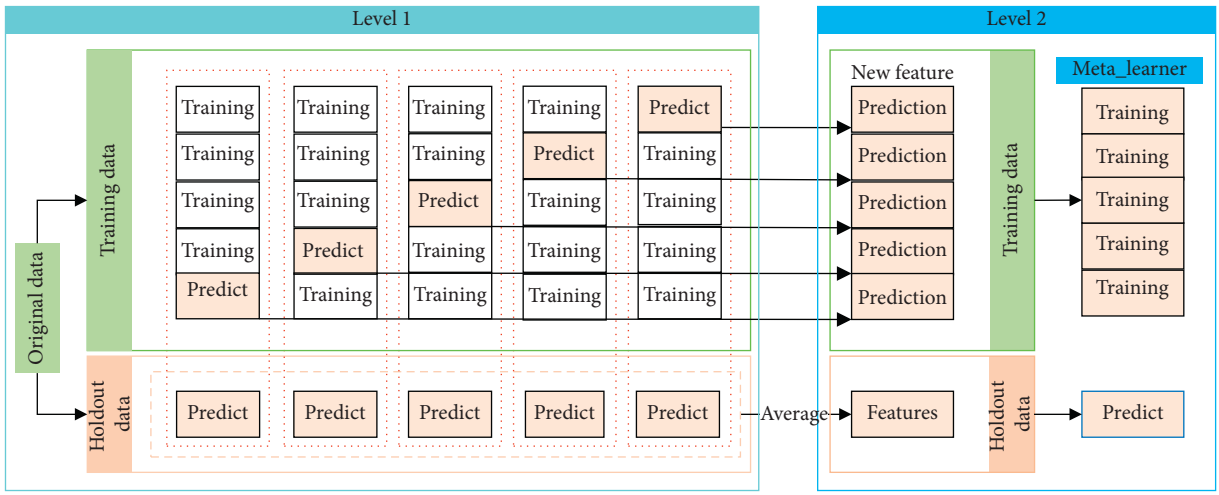


Figure 7: Description of the stacked generalization process.

improve the robustness of the model. The stacking model tuning is different from single model tuning. First, the original sample data are divided into training set and holdout dataset. Second, using the training set, the k-fold cross validation is combined with the grid search method to optimize the parameters of each component learner. In this paper, k is set as five, and the training set is divided into five parts, that is, there are five different subtraining sets and subtestsets. For each component learner, five models are generated based on the training of subtraining sets, respectively, and these five models are, respectively, used to predict their corresponding subtestsets. The optimal parameters are selected with the best five models' testset average accuracy. The parameter range of grid search is carried out according to the hyperparameter set of each model in Table 1. Then, based on the training set generated by the first level model, the parameters of the second level model are also optimized by the combination of k-fold cross validation and the grid search method. The total stacking model is validated on the validation set generated by the first layer model. Finally, the experimental results show that the parameter tuning method can effectively optimize the parameters of the stacking model, and the prediction effect of the stacking model has been improved. The hyperparameter range of each model algorithm grid search method is shown in Table 1.

3.3. Parametric Analysis Module. The parametric analysis module is based on Dynamo's visual programming environment, which associates the physical tunnel entity with the BIM model by constructing the mapping relationship between the components. It realizes the integration of real-time dynamic evolution visualization of pavement performance prediction and data science and provides support for managers' decision-making. The workflow is shown in Figure 8, including six steps: importing the Revit highway BIM model, creating highway segment elements, importing a Python script, importing MTSS highway segment performance prediction results, creating a chromatogram, and visualizing the changing trend of highway segments. First, this paper collects the data information of the 3D BIM model created in Autodesk Revit. Then, we link the MTSS prediction model as a custom node to the Dynamo environment. Finally, the predicted pavement performance trend is displayed in the form of graph visualization to support the decision-making reasoning process.

Step 1 Import the Revit highway BIM model

Dynamo is a lightweight model engine with a series of built-in graphics and bidirectional conversion nodes [56]. In this paper, the four nodes categories, all elements of a category, element geometry, PolyCurve, and ByjoinedCurves, which are wrapped by RevitAPI, are used to realize the component function of importing the highway BIM model into the dynamo engine from Revit.

Step 2 Create highway segment elements

As shown in Figure 8, according to the basic principle of bounding box regression in computer graphics, the bounding box of the highway BIM model in Figure 9(a) is created, and the computational element with the center point of the cube as the minimum (Figure 9(c)) is generated to realize the cut split and create the component function of highway segment elements

Step 3 Import Python script to highway segment elements

The bounding box-generated border range is much larger than the site boundary of the BIM model of the highway surface (Figure 9(b)), and graph screening is more complicated than a numerical screening calculation. Therefore, to improve the efficiency of graphic calculation, this article customizes the Python script and links it to Dynamo by calling the DesignScript library, judging the position of the unit cube according to the projective method and filtering out the unit cubes belonging to the pavement boundary as a highway segment element.

Step 4 Import MTSS highway segment performance prediction results

Through the file. Path node (Figure 8(d)), the dataset of the MTSS model prediction results is imported into Dynamo in the form of an Excel file to provide data support for the visualization of the pavement performance prediction digital twin

Step 5 Create a chromatogram

Create a gradient color chromatogram based on the three primary colors of red, yellow, and blue, map the predicted value to the chromatogram in Figure 10, and display the changes in the performance of the physical image elements of the highway segment through colors to enhance the visualization of the pavement performance

Step 6 Visualize the changing trend of highway segments

Taking each highway segment as a unit, finite element analysis is used to show the future change trend of the current highway segment. The red part indicates that the pavement performance will be reduced, and corresponding maintenance measures need to be taken. The pavement performance trend should be improved. Finally, according to the numerical results of the pavement performance prediction, the future trend of the highway segment is visually displayed in different colors to provide support for maintenance decision-making.

4. Case Study

4.1. Materials. With a total length of 2500 meters and four lanes in both directions, the Dalian highway tunnel in Shanghai is the first cross-river project in China, which

Table 1: Hyperparameter range of each algorithm.

Algorithm name	Hyperparameters selected
XGBoost	learning_rate = [0.01, 0.1, 0.2] n_estimators = [100, 50] max_depth = [30, 50, 100] Gamma = [0.1, 0.2, 0.5] Subsample = [0.9, 0.5] colsample_btree = [0.9, 0.5]
RF	Criterion = [“mse,” “mae”], n_estimators = [10, 100] max_depth = [10, 50, 100] max_features = [0.5, 0.9] min_samples_split = [10] min_samples_leaf = [2, 10]
Ridge	Solver = {“auto,” “svd,” “cholesky,” “lsqr,” “sparse_cg,” “sag”}
SVR	alpha = [0.1, 1, 10] C = [0.01, 0.1, 1], kernel = [“linear,” “poly,” “rbf,” “sigmoid”]
ANN	hindden_cell_num = [32, 64, 128] Activation = [“tanh,” “relu”] Scoring = [“neg_mean_squared_error”] Epochs = [300, 100] batch_size = [20, 50]

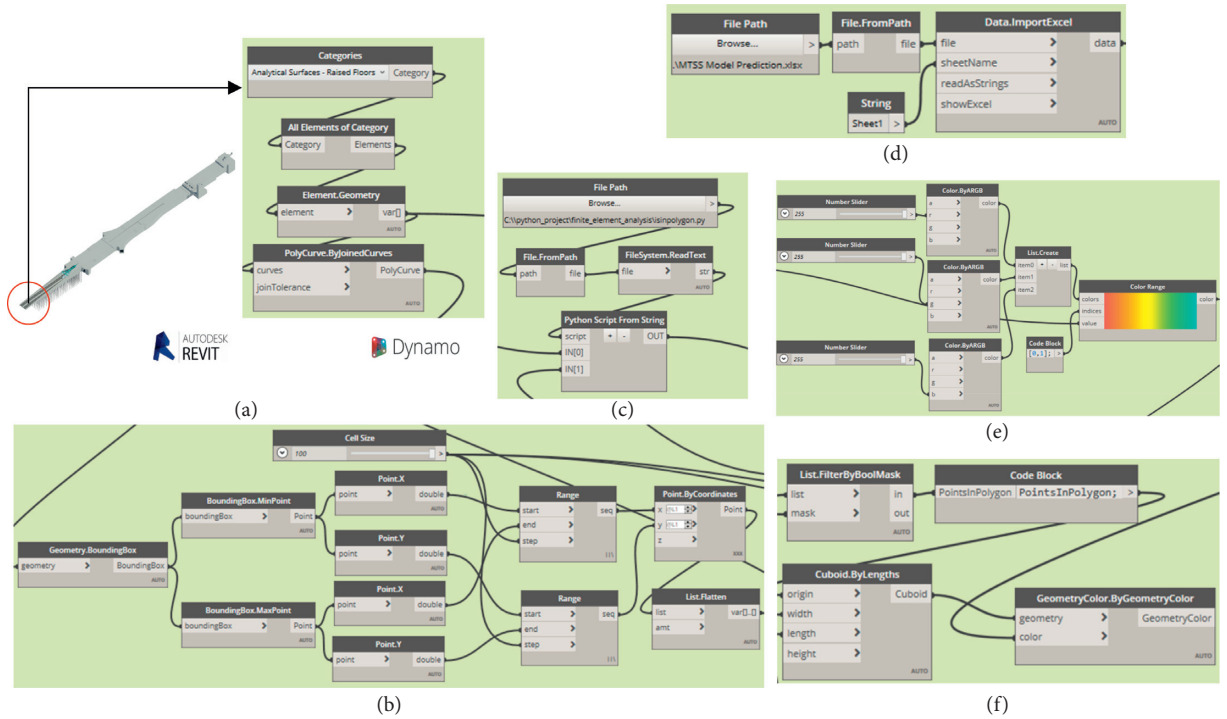


Figure 8: Dynamo’s digital twin workflow. (a) Import the Revit highway BIM model. (b) Create highway segment element. (c) Import Python script. (d) Import MTSS highway segment performance prediction results. (e) Create a chromatogram. (f) Visualize the changing trend of highway segments.

simultaneously adopts two large-scale shields to promote the new technology. This paper uses the historical dataset of the Dalian highway tunnel, including maintenance information and measured IRI data from 2010 to 2019. After data preprocessing, 80% of the dataset is randomly selected as the training set to construct the model, and the other 20% is used as the testset to evaluate and verify the performance of the model. Then, this paper uses the trained model to predict the pavement performance in 2020.

4.2. Data Processing. Whereas the maintenance data of the Dalian highway come from the records of special projects and subprojects over the years, the dataset only records

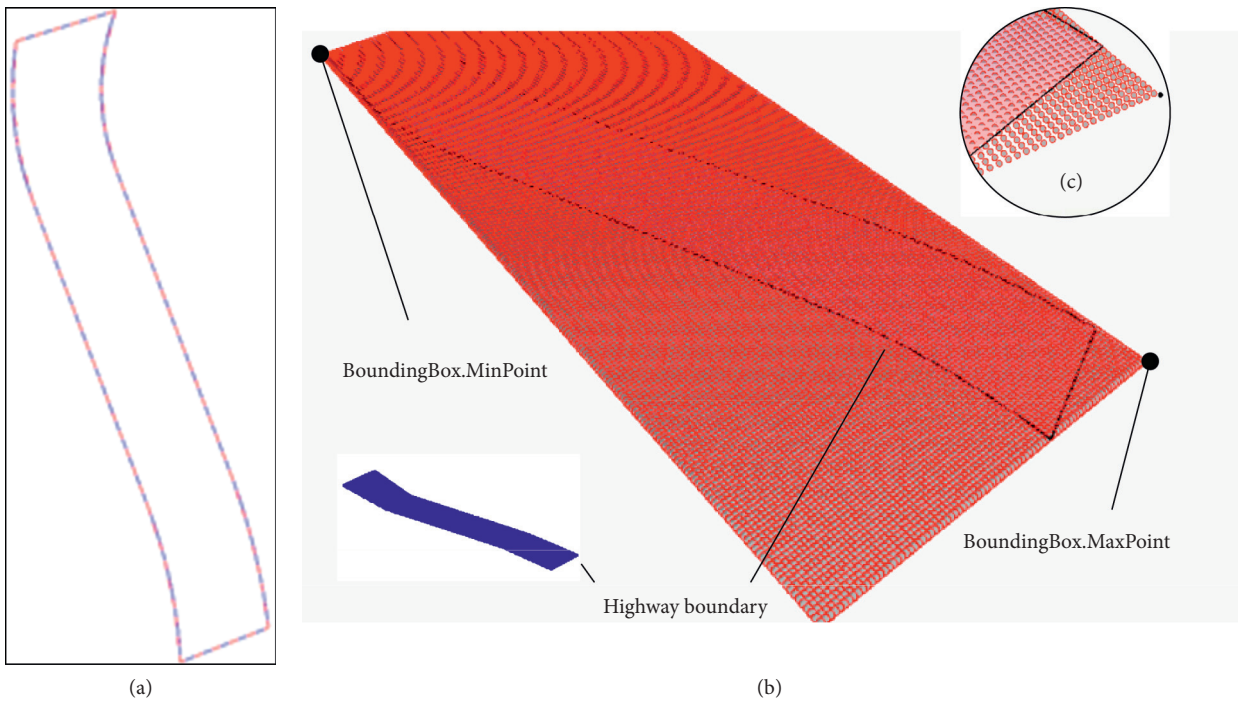


Figure 9: (a) The bounding box of the highway segment element. (b) Instantiate bounding box. (c) Limited cells of highway segments.

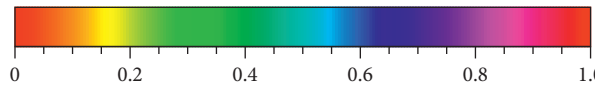


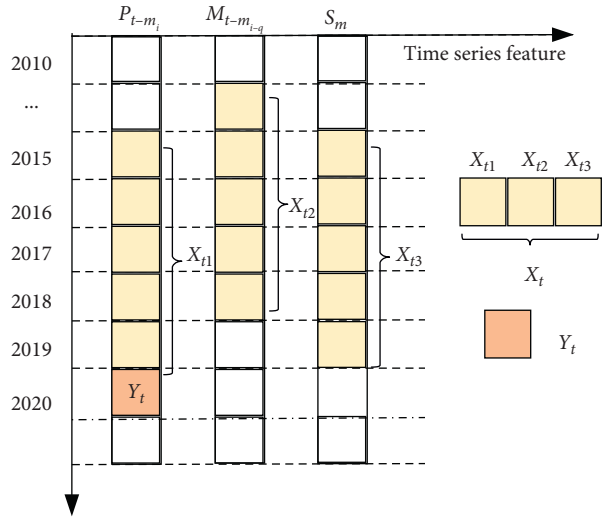
Figure 10: Mapping relationship between pavement performance and chromatography.

information such as east-west lines, fast and slow lanes, and light numbers. There is no direct record of highway segment elements, which does not meet the data requirements of the MTSS prediction model. Therefore, according to the actual tunnel space structure and the light number arrangement rules, this paper maps each maintenance record to the highway segment elements in accordance with the recorded natural decay, minor repair, or overhaul maintenance to solve the problem that the current maintenance record does not satisfy the prediction model to subdivide the highway into segments.

4.3. Result of Feature Selection

4.3.1. Results Based on Temporal Feature Extraction. The measured data of pavement roughness are affected by factors such as the service life of highways, maintenance level, COVI, traffic volume, and other factors. Among them, the service life of highways, maintenance conditions, and history performance data have the most significant influence on pavement performance prediction. Clearly, the pavement performance prediction is affected by multiple time series, and the measured data have multiple temporal features. The study of Kitaha et al. showed that the performance change was more reliable than the value itself [57]. Therefore, this paper selects IRI $P_{t-m_i}(i = 0, 1, 2, 3, 4, 5)$ of the previous five years as the feature factors describing IRI, defines $M_{t-m_i-q}(i = 1, 5; q =$ natural decay, minor repair, andoverhaul) as the maintenance feature factors, and defines $S_m(m = 0, 1, \ldots, 10)$ as the feature factor of the service life of highways. The extraction process based on the time series features is shown in Figure 11. The results of the extraction are shown in Tables 2-3. The sample is composed of each record X_t and the predicted value Y_t of the pavement performance.

4.3.2. The Significance of Feature Factors. To verify the temporal, spatial, and exogenous dependencies between pavement performance data and multiple factors, this paper uses the MTSS model to score P_{t-m_i}, M_{t-m_i-q}, and S_m multiple feature factors in the year t during the training process. The feature importance score is calculated based on the gain of the number of branches in the tree structure, mainly calculating the sum of the number of occurrences of features in all trees. If the number of occurrences is higher, the characteristic factor is more important. In Figure 12, the feature factors for 2020 are automatically ranked. According to the significance of the feature factors, it is found that the importance of maintenance features in the t year accounts for the largest proportion, that is, the maintenance situation in the predicted year has the greatest impact on pavement performance prediction; P_{t-1}, that is, pavement performance changes in the previous year have a greater impact on the



t_time, t = 2020, 2019, ... , 2010

m_the service life of a highway, m = 1, 2, 3, ... , 10

T_j_ the j temporal feature j = 1, 2, ... , n

X_{tj}–the feature factor of the j temporal feature at t

Y_j–the predicted value of the pavement performance at t

Figure 11: Time series feature extraction of the Dalian highway.

Table 2: Extraction results of time series feature.

Sample	IRI (m/km)					Maintenance status					Year
	t−5	t−4	t−3	t−2	t−1	t−1	t−2	t−3	t−4	t−5	
0	2.84	2.23	3.14	3.37	2.83	Natural decay	Overhaul	Minor repair	Natural decay	Overhaul	2015
1	2.23	3.14	3.37	2.83	3.56	Overhaul	Minor repair	Natural decay	Overhaul	Minor repair	2016
2	3.14	3.37	2.83	3.56	3.77	Minor repair	Natural decay	Overhaul	Minor repair	Natural decay	2017
3	3.37	2.83	3.56	3.77	3.22	Natural decay	Overhaul	Minor repair	Natural decay	Minor repair	2018
4	2.83	3.56	3.77	3.22	3.20	Overhaul	Minor repair	Natural decay	Minor repair	Minor repair	2019

Table 3: All X factors after one-hot transformation.

Factor name	Sample number				
	0	1	2	3	4
P_{t-5}	2.84	2.23	3.14	3.37	2.83
P_{t-1}	2.83	3.56	3.77	3.22	3.20
P_{t-3}	3.14	3.37	2.83	3.56	3.77
P_{t-2}	3.37	2.83	3.56	3.77	3.22
P_{t-4}	2.23	3.14	3.37	2.83	3.56
$M_{t-1\_overhaul}$	0	1	0	0	1
$M_{t-1\_minor\ repair}$	0	0	1	0	0
$M_{t-1\_natural\ decay}$	1	0	0	1	0
$M_{t-4\_overhaul}$	0	1	0	0	0
$M_{t-4\_minor\ repair}$	0	0	1	0	1
$M_{t-4\_natural\ decay}$	1	0	0	1	0
$M_{t-3\_overhaul}$	0	0	1	0	0
$M_{t-3\_minor\ repair}$	1	0	0	1	0
$M_{t-3\_natural\ decay}$	0	1	0	0	1
$M_{t\_overhaul}$	1	0	0	0	0
$M_{t\_minor\ repair}$	0	1	0	1	1
$M_{t\_natural\ decay}$	0	0	1	0	0
$M_{t-2\_overhaul}$	1	0	0	1	0
$M_{t-2\_minor\ repair}$	0	1	0	0	1
$M_{t-2\_natural\ decay}$	0	0	1	0	0

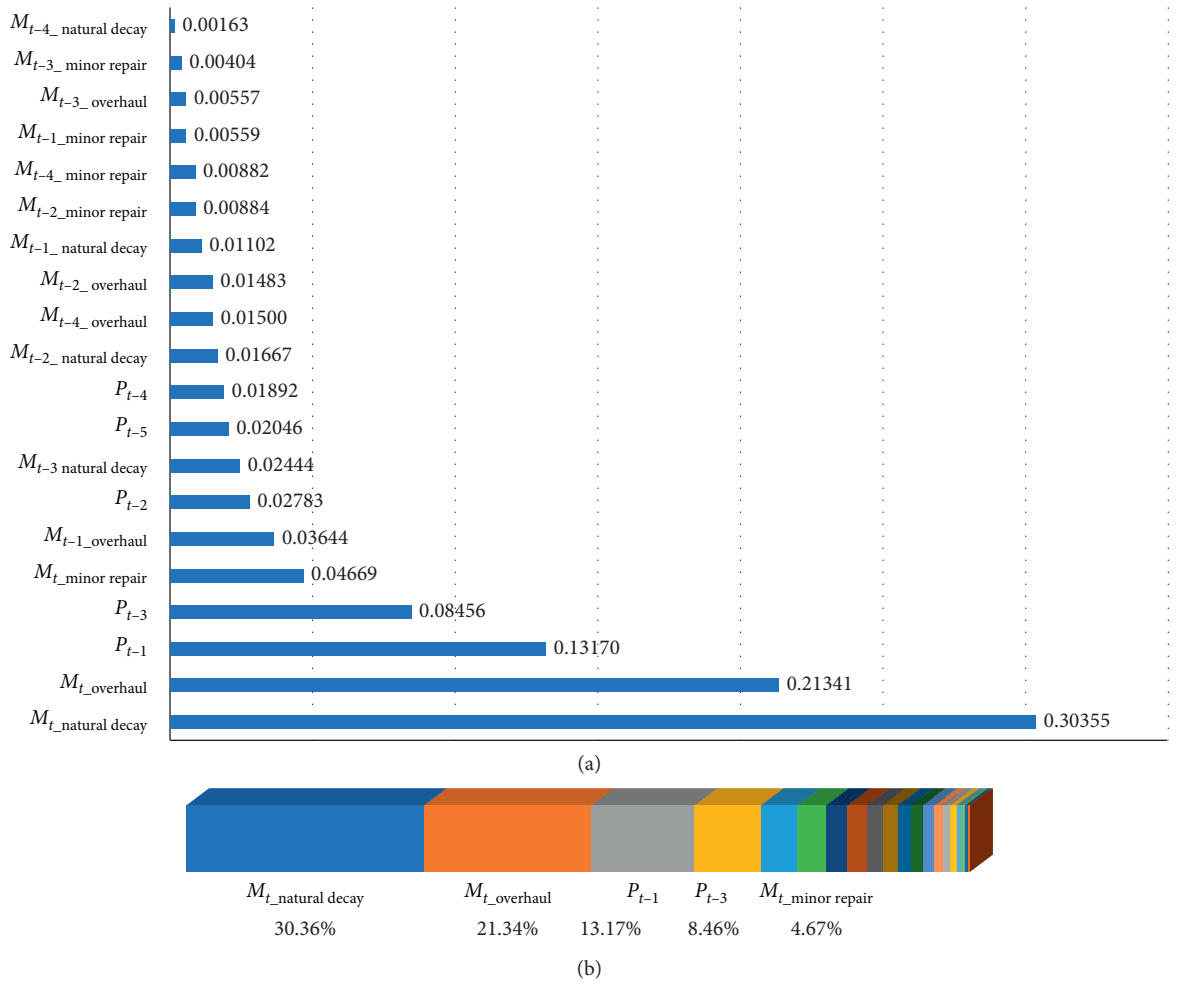


Figure 12: Feature significance.

predicted year; the overhaul has a greater impact on P_{t-m_i} than the minor repair. The closer the m_i value is to zero, the greater the feature importance is of P_{t-m_i}. It can be seen that maintenance types have a great impact on pavement performance.

4.4. Analysis of MTSS Prediction Model Results

4.4.1. MTSS Prediction Model Evaluation. Without the loss of generality, to evaluate the accuracy of the MTSS prediction model, this paper compares the stacking method with a single SVR, Ridge, XGBoost, ANN, and RF machine learning models using the same data to conduct a multicriterion evaluation. MAE and RMSE are standard indicators for model evaluation in machine learning. Their values are often used to directly measure the pros and cons of the prediction model results, and they can also better reflect the degree of deviation of the predicted value from the true value. The lower the value is, the smaller the error and the higher the prediction accuracy. When the outliers have a very large degree of individual deviation, even if the number

Table 4: Prediction error considering time series features.

Algorithm name	Prediction error		
	MAE	RMSE	MAPE
SVR	0.1527	0.0435	5.51%
Ridge	0.1526	0.0429	5.66%
XGBoost	0.1603	0.0491	5.75%
ANN	0.1533	0.0439	5.64%
RF	0.1689	0.0514	6.52%
Stacking	0.1314	0.0386	5.10%

Table 5: Prediction error without considering time series features.

Algorithm name	Prediction error		
	MAE	RMSE	MAPE
SVR	0.1815	0.0653	7.31%
Ridge	0.2033	0.0746	8.00%
XGBoost	0.1994	0.0748	7.69%
ANN	0.1937	0.0745	7.71%
RF	0.1909	0.0706	7.57%
Stacking	0.1808	0.0643	7.23%

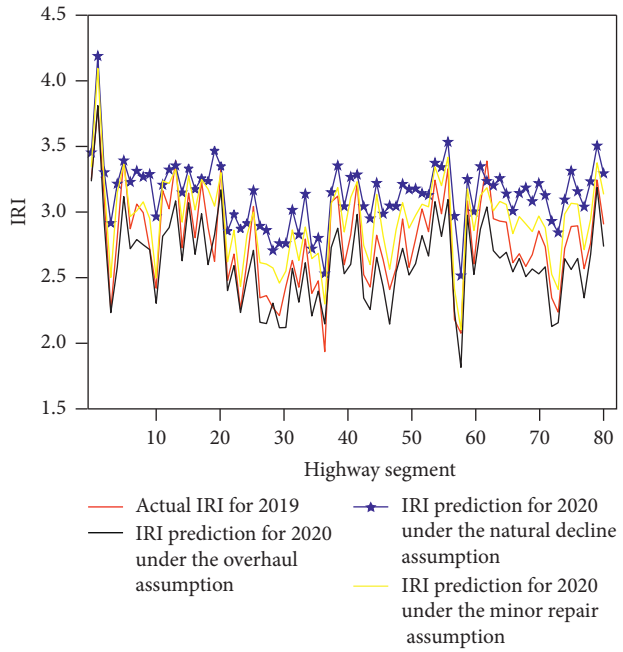


Figure 13: Variation prediction trend of the pavement performance of the Dalian highway in 2020.

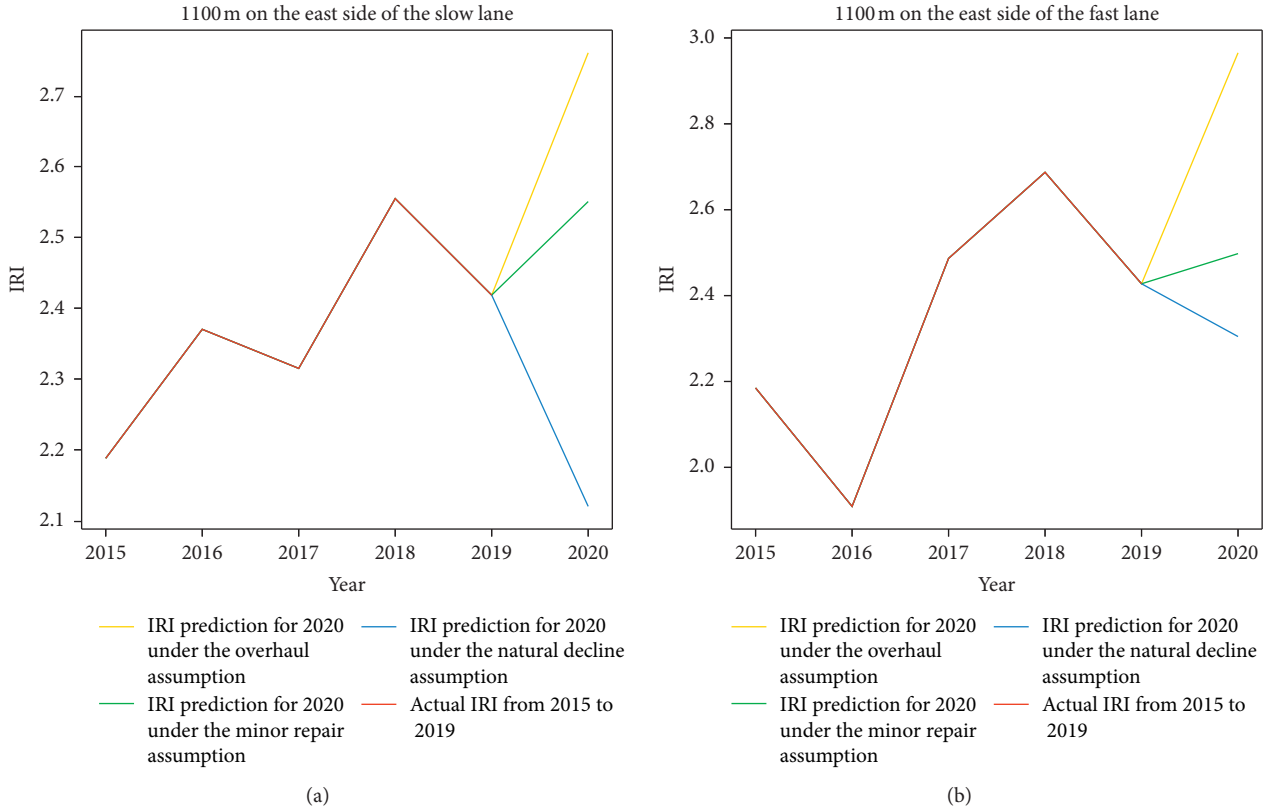


Figure 14: Continued.

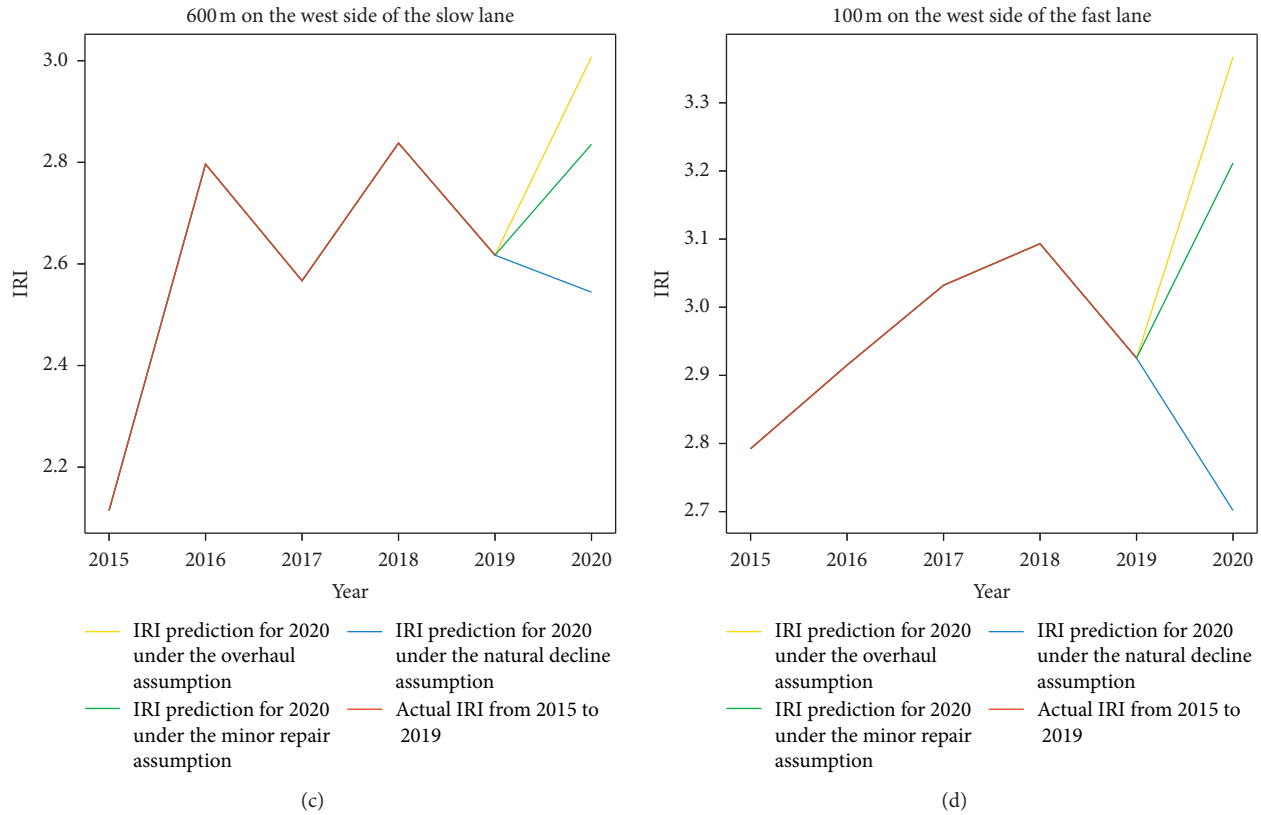


Figure 14: Trend chart of some segments of the pavement performance of the Dalian highway in 2020. (a) 1100 m on the east side of the slow lane. (b) 1100 m on the east side of the fast lane. (c) 600 m on the west side of the slow lane. (d) 600 m on the west side of the fast lane.

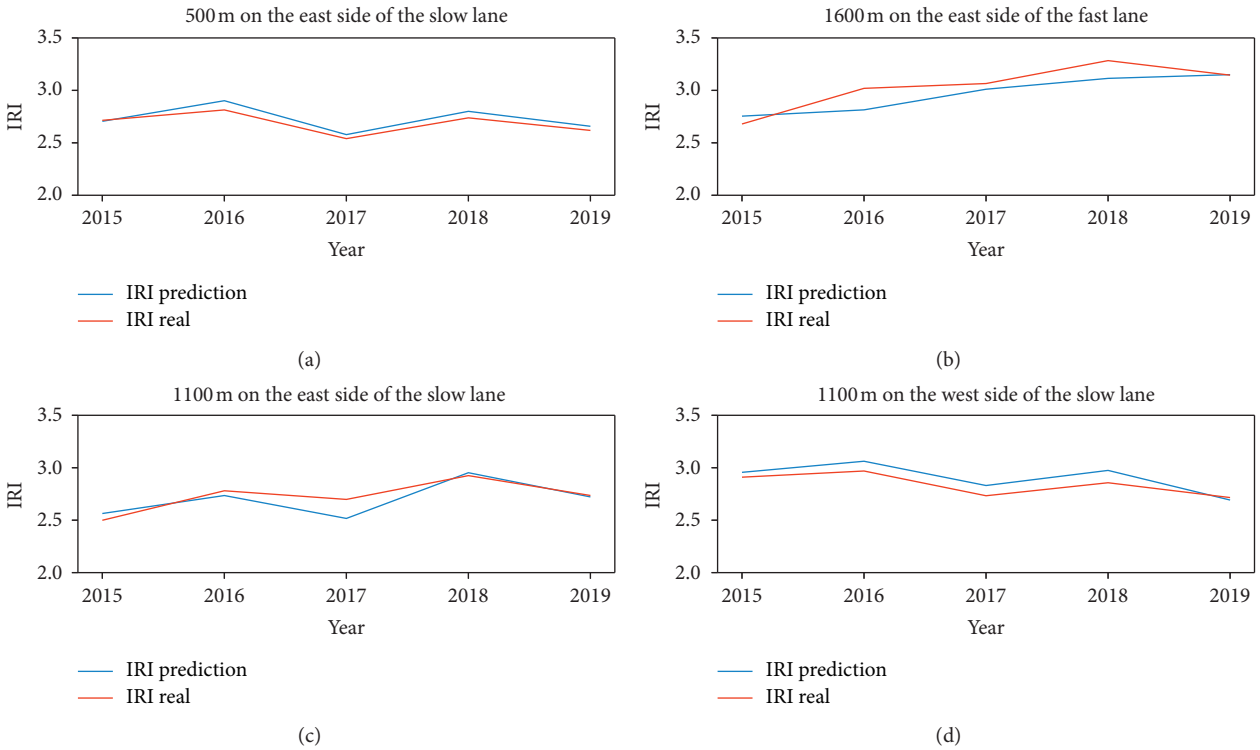


Figure 15: Segment prediction accuracy. (a) 500 m on the east side of the slow lane. (b) 1600 m on the east side of the fast lane. (c) 1100 m on the west side of the slow lane. (d) 1100 m on the west side of the slow lane.

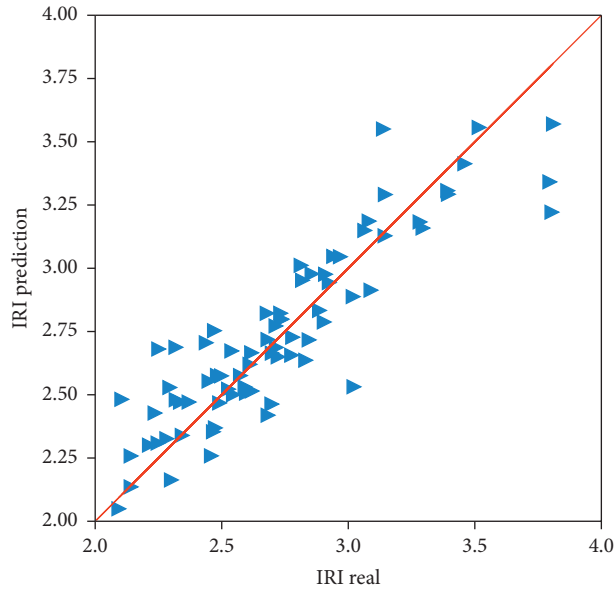


Figure 16: MTSS model prediction and actual error map.

Table 6: Pavement performance evaluation criteria.

The evaluation indicator	Superior (A)	Good (B)	Qualified (C)	Unqualified (D)
Expressway	≤2.6	> 2.6, ≤4.1	> 4.1, ≤7.3	> 7.3
Primary and secondary arterial highways	≤4.1	> 4.1, ≤5.7	> 5.7, ≤7.8	> 7.8
Branch pavement	≤4.6	> 4.6, ≤6.6	> 6.6, ≤8.3	> 8.3

of outliers is small, the performance of the standard indicators will deteriorate. Therefore, this paper also adds an MAPE indicator to solve the problem of the robustness of the evaluation indicator by replacing the average with the quantile of the error. Suppose y_i is the true value of sample i, $\hat{y}_i$ is the predicted value of sample i, and n is the number of samples, the evaluation index is defined as

$$\begin{aligned} \text{MAE} &= \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|, \\ \text{RMSE} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2}, \\ \text{MAPE} &= \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{\hat{y}_i}\right| \times 100\%, \end{aligned} \tag{5}$$

where MAE denotes the average absolute error between the predicted value and the actual value, RMSE denotes the root mean square error, and MAPE denotes the average relative error.

In Table 4, the stacking model has the best performance in comparison with the single model. The MAE of the stacking model is only 0.1314, the RMSE is 0.0386, and the MAPE is only 5.10%. The results show that the accuracy of the predicted value obtained by the MTSS model can reach 94.90%, which is higher than that of other single models. In the single prediction model, the SVR prediction error is small and the accuracy is the highest.

Without considering multiple timing features, the MAE of stacking was 0.1808, the RMSE was 0.0643, and the MAPE is 7.23% (Table 5). Considering multiple timing features, the MAE of stacking was 0.1314, the RMSE was 0.0386, and the MAPE is 5.10% (Table 4). Regarding the benchmark performance without considering multiple timing features, the MTSS prediction model shows better results that MAE is decreased by 0.0494, RMSE is decreased by 0.0257, and MAPE is decreased by 2.13%. Without considering the timing characteristics, the accuracy of model prediction is significantly reduced. It can be seen that multitemporal feature extraction is effective for improving the accuracy of the model, and the supplement of historical information can improve the prediction accuracy of the model.

In summary, the stacking model integrates a variety of heterogeneous prediction algorithms and can use the advantages of multiple algorithms in different data spaces and structures to construct better models. Therefore, the MTSS model is better than a single machine learning in terms of pavement performance prediction. The MTSS model shows better performance.

4.4.2. MTSS Prediction Model Results. From the analysis of the forecast cycle of the entire pavement of the Dalian highway, Figure 13 shows the comparison trend of the actual IRI value in 2019 and the IRI predicted values under three

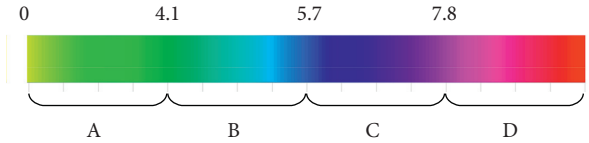
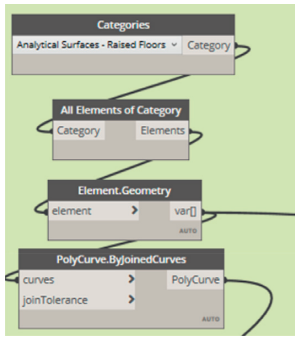
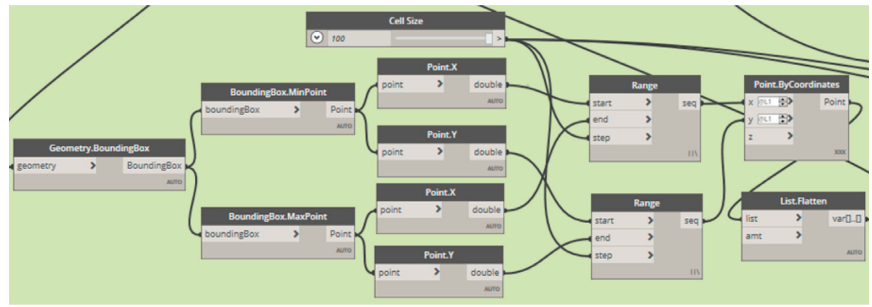


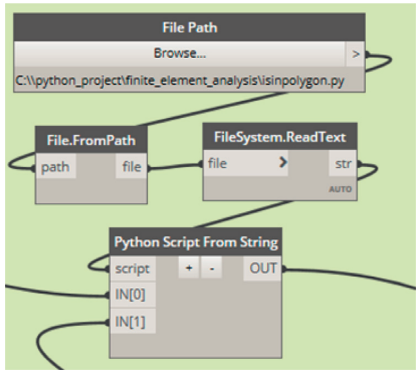
Figure 17: Pavement performance chromatogram of the Dalian highway tunnel.



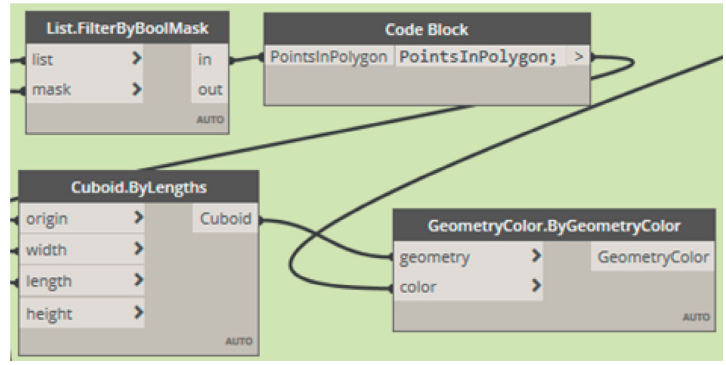
(a)



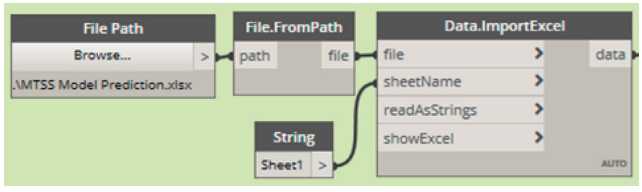
(b)



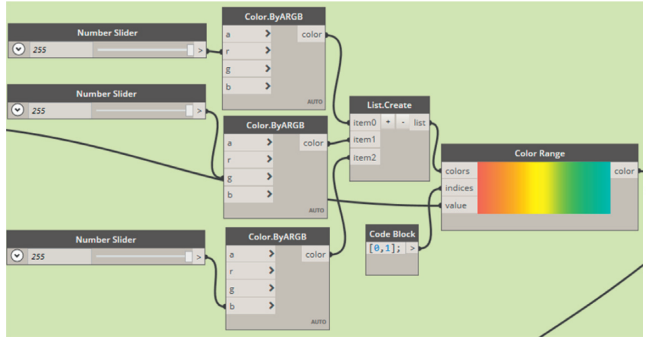
(c)



(f)



(d)



(e)

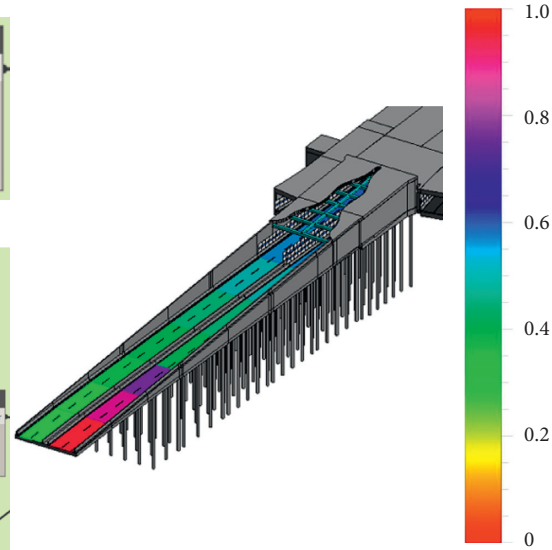


Figure 18: Visual presentation of pavement performance prediction in Dynamo. (a) Import the Revit highway BIM model. (b) Create highway segment element. (c) Import Python script. (d) Import MTSS highway segment performance prediction results. (e) Create a chromatogram. (f) Visualize the changing trend of highway segments.

different maintenance assumptions in 2020. The pavement performance will decline in turn under the assumption of overhaul, minor repair, and natural decay, which is in line with the normal law. In other words, the pavement performance after major repair is obviously higher than that after minor repair, and the pavement performance after minor repair is higher than that after natural decay.

According to the prediction of each segment in Figure 14, the IRI will decrease by 0.2 under the assumption of natural decay for the 1100 km slow lane from Puxi to Pudong (Eastern Route) in 2020. Under the assumption of minor repair, the IRI will be improved by 0.2. Under the assumption of overhaul, the IRI will be improved by 0.4. In 2020, on the 1100 km segment of the Puxi to Pudong (Eastern Route) expressway, the IRI decreases by 0.1 under the assumption of natural decay; under the assumption of minor repair, the IRI is improved by 0.2; under the assumption of overhaul, the IRI is increased by 0.6. The IRI will decrease by 0.05 under the assumption of natural decay for the 600 km slow lane from Pudong to Puxi (West Route) in 2020. Under the assumption of minor repair, the IRI will be improved by 0.3; under the assumption of overhaul, IRI will be improved by 0.6. The IRI will decrease by 0.1 in the 100 km segment of the expressway from Pudong to Puxi (West Route) in 2020 under the assumption of a natural recession. Under the assumption of minor repair, IRI will be improved by 0.2; under the assumption of overhaul, IRI will be improved by 0.6.

Figure 15 shows the comparison between the predicted value and the actual value of some highway segments. In scatter Figure 16, diagonals identify that the line X equals to Y, and the X-axis and Y-axis, respectively, represent the actual value and predicted value of IRI on the verification set. The closer the scatter is to the diagonal line, the better the prediction accuracy is. It is obvious that the model has a good fitting effect, which further indicates that the prediction effect of the MTSS model is feasible.

4.5. Application of Digital Twins. According to Shanghai Engineering Construction Code DGJ08-92-2013 "Technical Regulations for Urban Pavement Maintenance," the main highway network connected at both ends of the Dalian highway tunnel belongs to the primary and secondary arterial highways. It needs to be classified according to the ratings in Table 6 to map the pavement performance values and the chromatogram.

Applying the digital twin for pavement performance prediction proposed in Section 3.1, taking each section of the Dalian highway tunnel as an example, as shown in Figure 17, the performance numerical results of the visualized prediction highway segments are used to enhance the visualization by the color mapped in the chromatogram. According to the pavement performance evaluation standard level, the pavement performance trend is predicted by the MTSS prediction model in the Dynamo visualization parametric analysis module (Figure 18), and the change law of the highway segment performance varies under different maintenance assumptions.

5. Conclusions

In this paper, a highway tunnel pavement performance prediction approach based on a digital twin + MTSS is proposed. This method adopts digital twin technology and machine learning technology to predict the pavement performance data through the established MTSS model. The multiple time series feature extraction method and the automatic hyperparameter tuning technology of the MTSS model are the key factors improving the accuracy of pavement performance prediction. The stacking integrated learning method can effectively reduce the risk of poor generalization performance of a single model and falling into a local minimum. The relative error between the predicted value and the actual value of the MTSS model is small, which improves the accuracy of the prediction. The visual information service of the digital twin provides a reliable data-driven management method for preventive maintenance, which helps to improve the management efficiency of prediction and decision-making.

Data Availability

The online access data used to support the findings of this study have not been made available because the data source involves the internal life cycle operation and maintenance management system and is not open to the public.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (71701121), the Chinese Ministry of Education of Humanities and Social Science Project (17YJC630021), and the Key Program of the Shanghai Science and Technology Commission: research on intelligent identification and control strategy optimization of the shield tunneling state.

References

[1] R. Eghbalpoor, M. Baghani, and H. Shahsavari, "An implicit finite element framework considering damage and healing effects with application to cyclic moving load on asphalt pavement," *Applied Mathematical Modelling*, vol. 70, pp. 139–151, 2019.

[2] D. Zhang, B. Birgisson, X. Luo, and I. Onifade, "A new long-term aging model for asphalt pavements using morphology-kinetics based approach," *Construction and Building Materials*, vol. 229, p. 117032, 2019.

[3] A. C. Collop and D. Cebon, "A model of whole-life flexible pavement performance," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 209, no. 6, pp. 389–407, 1995.

[4] S. Shirzad, M. A. Aguirre, L. Bonilla, M. A. Elseifi, S. Cooper, and L. N. Mohammad, "Mechanistic-empirical pavement performance of asphalt mixtures with recycled asphalt

shingles," *Construction and Building Materials*, vol. 160, pp. 687–697, 2018.

[5] X. Ma, Z. Dong, F. Chen, H. Xiang, C. Cao, and J. Sun, "Airport asphalt pavement health monitoring system for mechanical model updating and distress evaluation under realistic random aircraft loads," *Construction and Building Materials*, vol. 226, pp. 227–237, 2019.

[6] J. Chao and Z. Jinxi, "Prediction model for asphalt pavement temperature in high-temperature season in Beijing," *Advances in Civil Engineering*, vol. 2018, Article ID 1837952, 11 pages, 2018.

[7] X. Chen, Q. Dong, X. Gu, and Q. Mao, "Bayesian analysis of pavement maintenance failure probability with Markov chain Monte Carlo simulation," *Journal of Transportation Engineering, Part B: Pavements*, vol. 145, no. 2, pp. 1–9, 2019.

[8] A. Bianchini and P. Bandini, "Prediction of pavement performance through neuro-fuzzy reasoning," *Computer-Aided Civil and Infrastructure Engineering*, vol. 25, no. 1, pp. 39–54, 2010.

[9] N.-F. Pan, C.-H. Ko, M.-D. Yang, and K.-C. Hsu, "Pavement performance prediction through fuzzy regression," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10010–10017, 2011.

[10] A. Abed, N. Thom, and L. Neves, "Probabilistic prediction of asphalt pavement performance," *Road Materials and Pavement Design*, vol. 20, no. sup1, pp. S247–S264, 2019.

[11] R. Li, Y. Huang, and J. Wang, "Long-term traffic volume prediction based on type-2 fuzzy sets with confidence interval method," *International Journal of Fuzzy Systems*, vol. 21, no. 7, pp. 2120–2131, 2019.

[12] H. Gong, Y. Sun, X. Shu, and B. Huang, "Use of random forests regression for predicting IRI of asphalt pavements," *Construction and Building Materials*, vol. 189, pp. 890–897, 2018.

[13] S. Xiao and M. Nie, "The preventive maintenance of highway based on data mining," *MATEC Web of Conferences*, vol. 139, pp. 1–5, 2017.

[14] J. Bi, H. Yuan, and M. Zhou, "Temporal prediction of multiapplication consolidated workloads in distributed clouds," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1763–1773, 2019.

[15] T. Lintonen and T. Raty, "Self-learning of multivariate time series using perceptually important points," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 6, pp. 1318–1331, 2019.

[16] B. Liu, S. Yan, H. You et al., "Road surface temperature prediction based on gradient extreme learning machine boosting," *Computers in Industry*, vol. 99, pp. 294–302, 2018.

[17] H. Gong, Y. Sun, Z. Mei, and B. Huang, "Improving accuracy of rutting prediction for mechanistic-empirical pavement design guide with deep neural networks," *Construction and Building Materials*, vol. 190, pp. 710–718, 2018.

[18] H. Gong, Y. Sun, W. Hu, and B. Huang, "Neural networks for fatigue cracking prediction using outputs from pavement mechanistic-empirical design," *International Journal of Pavement Engineering*, vol. 145, no. 2, pp. 1–11, 2019.

[19] Q. Hu, "A review of data analytic applications in road traffic safety. Part 2: prescriptive modeling," *Sensors (Switzerland)*, vol. 20, no. 4, 2020.

[20] G. Wang, J. Qiao, J. Bi, W. Li, and M. Zhou, "TL-GDBN: growing deep belief network with transfer learning," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 2, pp. 874–885, 2019.

[21] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 2, pp. 601–614, 2019.

[22] J. Wang and T. Kumbasar, "Parameter optimization of interval Type-2 fuzzy neural networks based on PSO and BBBC methods," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 1, pp. 247–257, 2019.

[23] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[24] F. Lin, J. Jiang, J. Fan, and S. Wang, "A stacking model for variation prediction of public bicycle traffic flow," *Intelligent Data Analysis*, vol. 22, no. 4, pp. 911–933, 2018.

[25] B. Zhai and J. Chen, "Development of a stacked ensemble model for forecasting and analyzing daily average PM2.5 concentrations in Beijing, China," *Science of The Total Environment*, vol. 635, pp. 644–658, 2018.

[26] J. Chen, J. Yin, L. Zang, T. Zhang, and M. Zhao, "Stacking machine learning model for estimating hourly PM2.5 in China based on Himawari 8 aerosol optical depth data," *Science of The Total Environment*, vol. 697, p. 134021, 2019.

[27] M. Jiang, J. Liu, L. Zhang, and C. Liu, "An improved Stacking framework for stock index prediction by leveraging tree-based ensemble models and deep learning algorithms," *Physica A: Statistical Mechanics and Its Applications*, vol. 541, no. 258, p. 122272, 2020.

[28] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, "A stacking model using URL and HTML features for phishing webpage detection," *Future Generation Computer Systems*, vol. 94, pp. 27–39, 2019.

[29] X. Yang, B. Dindoruk, and L. Lu, "A comparative analysis of bubble point pressure prediction using advanced machine learning algorithms and classical correlations," *Journal of Petroleum Science and Engineering*, vol. 185, Article ID 106598, 2020.

[30] L. Feng, Y. Li, Y. Wang, and Q. Du, "Estimating hourly and continuous ground-level PM2.5 concentrations using an ensemble learning algorithm: the ST-stacking model," *Atmospheric Environment*, vol. 223, Article ID 117242, 2020.

[31] D. Systèmes, *Meet Virtual Singapore, the City's 3D Digital Twin Available*, GovInsider, Adelaide, Australia, 2019.

[32] C. Boje, A. Guerriero, S. Kubicki, and Y. Rezgui, "Towards a semantic construction digital twin: directions for future research," *Automation in Construction*, vol. 114, p. 103179, 2020.

[33] Y. Lu, C. Liu, K. I.-K. Wang, H. Huang, and X. Xu, "Digital Twin-driven smart manufacturing: connotation, reference model, applications and research issues," *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101837, 2020.

[34] S. Liu, J. Bao, Y. Lu, J. Li, S. Lu, and X. Sun, "Digital twin modeling method based on biomimicry for machining aerospace components," *Journal of Manufacturing Systems*, pp. 1–16. In press, 2020.

[35] Q. Lu, L. Chen, S. Li, and M. Pitt, "Semi-automatic geometric digital twinning for existing buildings based on images and CAD drawings," *Automation in Construction*, vol. 115, p. 103183, 2020.

[36] Z. Ma and Z. Liu, "BIM-based intelligent acquisition of construction information for cost estimation of building projects," *Procedia Engineering*, vol. 85, pp. 358–367, 2014.

[37] Z. Ma, S. Cai, N. Mao, Q. Yang, J. Feng, and P. Wang, "Construction quality management based on a collaborative system using BIM and indoor positioning," *Automation in Construction*, vol. 92, pp. 35–45, 2018.

[38] Z. Wang and E. Rezazadeh Azar, "BIM-based draft schedule generation in reinforced concrete-framed buildings," *Construction Innovation*, vol. 19, no. 2, pp. 280–294, 2019.

[39] M. Sami Ur Rehman, M. J. Thaheem, A. R. Nasir, and K. I. A. Khan, "Project schedule risk management through building information modelling," *International Journal of Construction Management*, pp. 1–11, 2020.
[40] C.-S. Shim, N.-S. Dang, S. Lon, and C.-H. Jeon, "Development of a bridge maintenance system for prestressed concrete bridges using 3D digital twin model," *Structure and Infrastructure Engineering*, vol. 15, no. 10, pp. 1319–1332, 2019.
[41] J. Heaton, A. K. Parlikad, and J. Schooling, "Design and development of BIM models to support operations and maintenance," *Computers in Industry*, vol. 111, pp. 172–186, 2019.
[42] G. Y. N. N. Adi, M. H. Tandio, V. Ong, and D. Suhartono, "Optimization for automatic personality recognition on twitter in Bahasa Indonesia," *Procedia Computer Science*, vol. 135, pp. 473–480, 2018.
[43] T. Chen and T. He, "xgboost: eXtreme Gradient Boosting," pp. 1–4, 2015.
[44] H. Meng, X. Wang, and X. Wang, "Expressway crash prediction based on traffic big data," in *Proceedings of the ACM's International Conference Proceedings Series*, pp. 11–16, New York, NY, USA, July 2018.
[45] G. Ke, "LightGBM: A highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, vol. 30, pp. 3147–3155, 2017.
[46] S. Zhou, L. Sun, W. Xing et al., "Hyperspectral imaging of beet seed germination prediction," *Infrared Physics & Technology*, vol. 108, no. 74, p. 103363, 2020.
[47] Y. L. Pavlov, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2019.
[48] A. Mechelli and S. Vieira, *Machine Learning: Methods and Applications to Brain Disorders*, Academic Press, London, UK, 2019.
[49] R. Kissell, *Lgorithmic Trading Methods: Applications Using Advanced Statistics, Optimization, and Machine Learning Techniques*, Academic Press, Cambridge, MA, USA, 2020.
[50] M. Ali, I. Ullah, W. Noor, A. Sajid, A. Basit, and J. Baber, "Predicting the session of an P2P IPTV user through support vector regression (SVR)," *Engineering, Technology & Applied Science Research*, vol. 10, no. 4, pp. 6021–6026, 2020.
[51] Z. Nie, F. Shen, D. Xu, and Q. Li, "An EMD-SVR model for short-term prediction of ship motion using mirror symmetry and SVR algorithms to eliminate EMD boundary effect," *Ocean Engineering*, vol. 217, Article ID 107927, 2020.
[52] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the Digital Twin: a systematic literature review," *CIRP Journal of Manufacturing Science and Technology*, vol. 29, pp. 36–52, 2020.
[53] E. A. G. Isabelle, *An Introduction to Feature Extraction*, Springer, Berlin, Germany, 2006.
[54] S. O. M. Smith and D. Susarla, *Feature Engineering Made Easy*, pp. 211–264, Packt Publishing, Birmingham, UK, 2018.
[55] P. Ruano, *Introduction to Machine Learning*, MIT press, Cambridge, MA, USA, 4th edition, 2016.
[56] N. C. Brown, V. Jusiega, and C. T. Mueller, "Implementing data-driven parametric building design with a flexible toolbox," *Automation in Construction*, vol. 118, Article ID 103252, 2020.
[57] A. Kitaha and K. P. Biligiri, "Prioritization-optimization process algorithm to manage pavement networks during the non-availability of historical data," *Journal of Testing and Evaluation*, vol. 45, no. 2, Article ID 20150287, 2017.