IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received February 25, 2020, accepted March 3, 2020, date of publication March 11, 2020, date of current version March 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2979833

MLLBC: A Machine Learning Toolbox for Modeling the Loss Rate of the Lining Bearing Capacity

SEN ZHANG[1,2], WANYIN WU[3], ZHAO YANG[4], XU LIN[3], ZHIHUA REN[1], AND ZHIXIN YAN[2]

[1]Yunnan Research Institute of Highway Science and Technology, Kunming 650051, China
[2]College of Civil Engineering and Mechanics, Lanzhou University, Lanzhou 730000, China
[3]Union Vision Innovation, Shenzhen 518055, China
[4]School of Mechanical and Electric Engineering, Guangzhou University, Guangzhou 510006, China

Corresponding authors: Wanyin Wu (wanyinwu@qq.com) and Zhao Yang (yangdxng100@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 51768028 and Grant 61501177, and in part by the Yunnan Transportation Science and Technology Funds under Grant 2016(A)01 and Grant 2017(A)04.

ABSTRACT Testing the health of tunnels, as a branch of highway operation, has an extremely important application in public property and even life safety. Among them, there are many factors that cause the tunnel to deform or collapse. The conventional methods use the finite element method (FEM) which are to simulate the bearing capacity loss rate of the lining by using the mechanical method. However, it takes a long time to calculate the stress-strain-situation of the lining model under each condition. This paper explores the machine learning to calculate the loss rate of the lining bearing capacity under more conditions based on FEM simulation data. Here, we establish a machine learning toolbox for modeling the loss rate of the lining bearing capacity named "MLLBC", which contains three main components: 1) data loading; 2) machine learning model deployment; 3) performance evaluation. To ensure the fairness of model evaluation, ten machine learning models use a unified code library. We also conduct experiments on our new dataset which is the loss rate of the lining bearing capacity with different data amounts, as well as experiments on the goodness of model fitting under different ranges of various variables.

INDEX TERMS Toolbox, the loss rate of the lining bearing capacity, machine learning, tunnel health.

I. INTRODUCTION

Lining is a supporting structure with concrete or reinforced concrete built in a tunnel to prevent deformation or collapse of surrounding rocks [22]. With the start of the tunnel construction, the initial stress balance of the stratum has been destroyed, resulting in stress release of the surrounding rocks and the generation of voids, which will cause the lining to deform or even collapse [23]. This not only affects the normal use of the tunnel, but also endangers the safety of the public. It is important to judge whether the tunnel can continue to be used by detecting and analyzing the tunnel supporting structure [5]. Therefore, detecting scientifically the health status of the lining in the tunnel so as to take corrective measures in time can save huge economic losses.

To detect the health status of the tunnel, researchers used ultrasonic and radar sensors to calculate the thickness of the lining, the voids and cracks behind the lining [50]. Then, the actual test results were calculated using the finite element method (FEM) to analyze the bearing capacity of the lining [11]. In addition, researchers have obtained more test results by building similar lining models [17]. In the case of different sizes of cavity sizes, different directions of pressure are further applied to the lining model to analyze the mechanical behavior of the lining [54]. However, due to the health status of the tunnel is related to many conditions, it takes a lot of time to calculate each factor using the finite element analysis software, which causes a lot of inconvenience for practical application.

Recently, researchers have found that exploiting machine learning algorithms can learn some complex statistical patterns effectively [15], [35], such as judging the probability of the disease in medicine [25], analyzing user preferences in the market and fault diagnosis in mechanical parts [52]. Wu *et al.* [56] proposed a model with big lung cancer data to reduce the incidence of malignant diseases Clairand *et al.* [8] introduced neural networks in analyzing user preferences of charging to save charging time. Wang *et al.* [51] and

The associate editor coordinating the review of this manuscript and approving it for publication was Mehul S. Raval.

Wang *et al.* [53] have made significant progress in classifying the fault type of the piston pumps and bearings by utilizing the neural networks.

Over the past decades, several advanced machine learning methods have been proposed for high-quality pattern recognition [45] In general, the research work of regression methods in machine learning can be roughly divided into five categories, including linear [18], kernel [39], tree and forest [60], nearest neighbors [47] and neural network [13]. Firstly, linear based method is the most basic model in machine learning. A major advantage of linear models is that they provide a simple description to predicting a quantitative value. For example, to tackle drug design problems, Lo *et al.* [27] proposed a linear regression method to mine the chemical information and presented the basic principles in drug analysis Experiments validated that the proposed machine learning descriptor can be applied in drug discovery. Kumaret *et al.* [18] introduced a more flexible approach to evaluate the health state of cutting tools. Specifically, this approach used a polynomial regression model based sequential clustering on time series sensor signals, which performed well for monitoring drill-bits. Besides, Wiliński *et al.* [55] proposed a polynomial regression simulation model to provide an investment prediction strategy for finance markets.

The second category of machine learning is kernel-based method, which aims to address the linearly indivisible dataset in low-dimensional space, and kernel-based method have been widely used in sciences and industry fields for solving ranking and regression problems. For instance, Philip *et al.* [39] introduced a support vector regression model for travel time prediction Thus, road users can easily understand the traffic condition and make a decision. Chang *et al.* [6] designed a semi-supervised learning model, which applied kernel ridge regression to unlabeled data for error decomposition and this learning theory provided a promising way analysis to tackle data analysis task in practical applications, such as medicine and business.

Tree and forest method is the third category of machine learning method. Different from kernel-based method, tree and forest method are a summary of expert experience, which are widely used in practical application [44]. For example, Reix *et al.* [40] built a new canonical decision tree model to analysis concordant and non-concordant CT prescriptions and find the decisive factor for breast cancer care. To provide a low-cost sensing strategy for air quality monitoring, Zimmerman *et al.* [60] used random forests to design a comprehensive machine learning calibration model, and this model performed well on real-time air monitoring. Subasi *et al.* [42] utilized random forest method to provide an automatic model for diagnosis of chronic kidney disease.

The fourth type of machine learning method is nearest neighbors-based algorithm, which utilizing the distance between different eigenvalues for cluster analysis, predictive analysis. Specifically, nearest neighbors-based algorithms are considered to be one of the effective regression techniques for data mining. For instance, on the issue of health monitoring application, Vitola *et al.* [49] designed a k-nearest neighbor model to fusion sensor data. Llerena *et al.* [26] introduced radius neighbor regression techniques to approximately evaluate the microphysical parameters of pollution.

The last typical machine learning method is neural network. Neural networks can be used for supervised tasks, such as visual recognition [46], it can also tackle unsupervised tasks. In recent years, due to the complex non-linear problems in practical application, multiple advanced neural networks have been proposed [13]. For example, to tackle prediction tasks, Berahas *et al.* [3] proposed a multi-batch L-BFGS method, which utilized different gradients to update the hessian approximations.

While many machine learning methods have been proposed [29] in various fields, there is no unified platform that can be used to calculate the health of the lining. Therefore, based on previous research, this paper proposes a toolbox for modeling the loss rate of the lining bearing capacity named "MLLBC". It considers several foundational statistical algorithms based on machine learning [24], [43]. Figure 1 shows the framework of calculating the loss rate of linings bearing capacity scheme. MLLBC summarizes four steps by referring to logic of the usage on the existing toolbox [9], [57]: 1) building a physical model of tunnel lining to obtain the loss rate of the lining under different conditions (such as void ratio, stratum stiffness and the angle of load); 2) augmenting simulation data with finite element analysis tool; 3) training the machine learning models in the toolbox with the simulation data of different conditions and the corresponding bearing capacity loss rate; 4) calculating the bearing capacity loss rate under various conditions by one of the pre-trained machine learning models.

It takes a lot of time to calculate various factors by using FEM model, since the health states of the lining is related to many conditions. It is difficult to quickly calculate the lining bearing loss rate under various complicated conditions in the real-word, which causes a lot of inconvenience for practical use. In view of the advantages of machine learning in data processing, a machine learning toolbox is designed and implemented to calculate the loss rate of tunnel lining bearing capacity in this paper. In sum, the main contribution of this paper is the toolbox can replace the finite element analysis tool in real-word conditions by data-driven method to efficiently complete the loss rate calculation of the lining bearing capacity.

The remainder of the paper is organized as follows. In Sections II, we present related work for toolbox on different industries. In Section III, we detail the toolbox proposed by us. Section IV describes the dataset and the setting, and the experimental results are discussed in Section V. We conclude in Section VI.

II. RELATED WORK

Machine learning toolbox is designed by researchers to help users solve many statistics tasks [34], which can

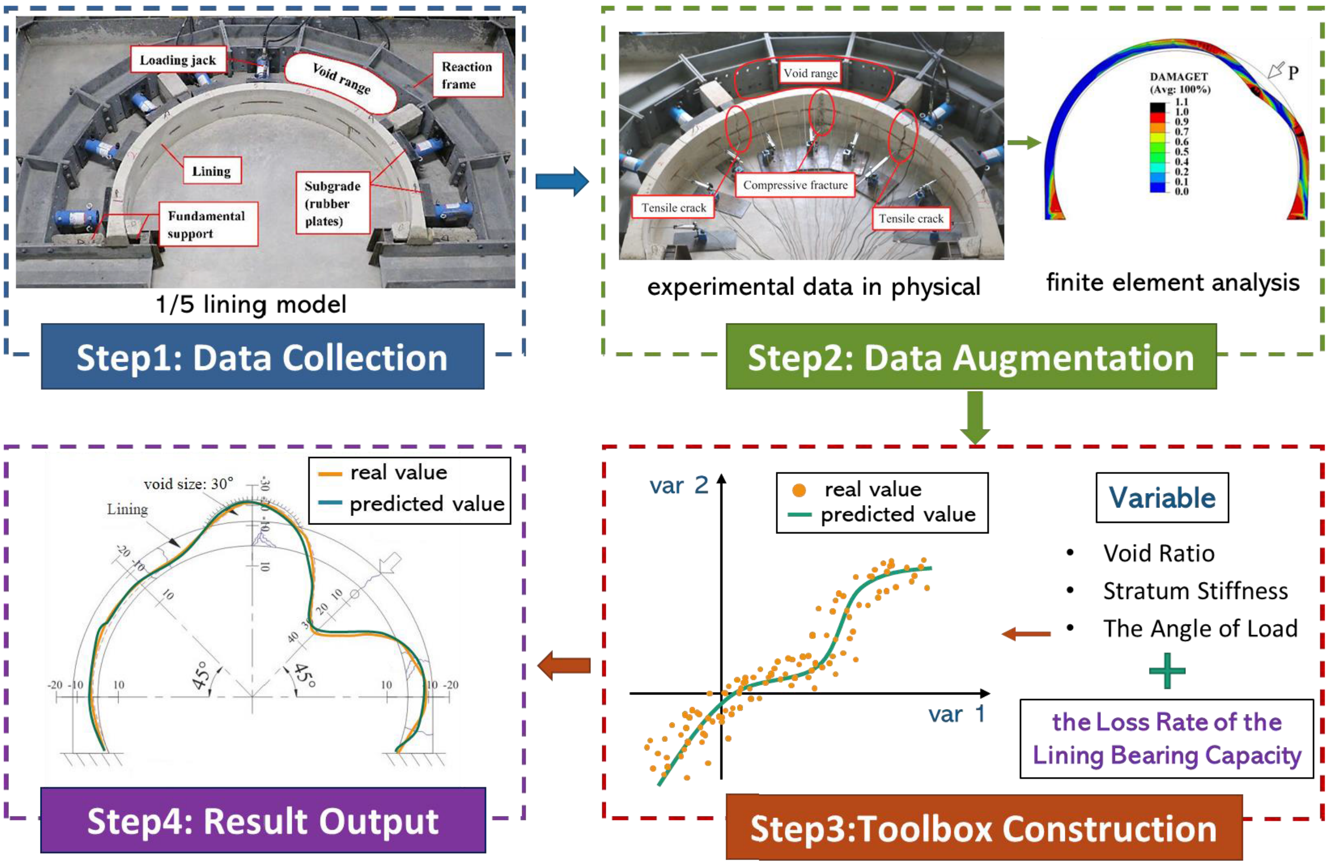


FIGURE 1. The framework of calculating the loss rate of linings bearing capacity.

reduce development time. Nowadays, there exists many open source tools, such as LIBSVM [7], Tensorflow [1] and PyTorch [38]. Specifically, according to the application field, machine learning toolboxes can be generally divided into four categories: 1) Open source machine learning tools for computer vision, natural language processing and audio, such as the software system Detectron [12], which is developed by Facebook's artificial intelligence research company. Moreover, Detectron is written in Python and supported by the Caffe2 deep learning framework. StanfordNLP [28] is a natural language analysis package for Python. And it supports more than 70 human languages. 2) Big data open source machine learning tools For example, Hadoop [41] and Spark [31] are the representative tools that aim to help users deal with big data. 3) Open source tools for reinforcement learning [14] Since reinforcement learning is a hot topic in machine learning, it is important to design effective training environments for reinforcement learning. For example, Google Research Football [19] is a new reinforcement learning environment that allows smart brokers to master the most popular football sport in the world. 4) Open source machine learning tools for model deployment, which aims to help users apply their projects to real-world devices more easily. For example, Apple's CoreML [33] is one popular tool that can build machine learning models into various applications of Apple device. The highlight of CoreML is that users do not need to have extensive knowledge of neural networks or machine learning.

While many machine learning tools have been proposed, there still exists high barriers for the application to specific industries. As for the needs of specific industries, it requires senior engineer for secondary development. The four major industries, which consist of finance, medical care, communications, and building, play an important role in the growth of national economy. Therefore, there are some dedicated toolboxes for the above industries. A review of different tools in this section is summarized as follows.

1) **Financial Toolbox** is developed to build financial knowledge by mathematical modeling and statistical analysis. In recent year, many financial toolboxes are widely used for computational efficiency. For example, to solve the problem of on-line portfolio selection, Li *et al.* [21] proposed a comprehensive toolbox, which can evaluate the performance of different on-line portfolio selection algorithms and develop new algorithms. On the issue of finance shared services project, Neukirchen and Vollmer [34] proposed a change management controlling toolbox, which is a representative research contribution for business management. According to empirical research, Kim [16] proposed a statistical toolbox that can provide a range of statistical instruments for financial researchers. Specially, the various alternatives can eliminate large sample biases

2) **Medical care Toolbox** aims to provide statistical support for mining the inner meaning of medical data. Recently, various medical care tools play an increasingly important role in health care and research. Thangarajh *et al.* [48] introduced a NIH toolbox to deal with duchenne muscular dystrophy, which could obtain cognitive assessment about different causative factors. To analysis complex patient-treatment process Metsker *et al.* [32] utilized graphminer toolbox for data modeling and mining. Besides, this method provides a visualization understanding of the process of treatment. Orava *et al.* [36] proposed a chronic pain assessment toolbox that describes an evaluation of children with disabilities. It has been turns out that the toolbox is a useful resource for the assessment practices, especially for children with cerebral palsy.

3) **Communication Toolbox** provides tools for solving various communication problems through modeling and analysis of signals. For example, Ghimire *et al.* [10] designed a toolbox, which provided various information related to signal oscillations, and eventually evaluated the stability of small signal. Taormina *et al.* [47] proposed an open-source MATLAB toolbox that can be applied in water distribution systems. This toolbox allows users to design a smart water networks through various simulation practices of attack scenarios. Recently, to meet the demand of simple and configurable Cyber-Physical Systems (CPS), Melzer *et al.* [30] proposed a Broker-based SysML Toolbox. The open source toolbox has been demonstrated to fulfil the requirements of baggage tracking system.

4) **Building Toolbox** aims to provide builder-oriented chart of services for architects & constructors. During the past years, to find out specific solutions for users, various building toolboxes are designed by researchers. For instance, Boonstra *et al.* [4] proposed a toolbox that can optimize the building spatial design. The designers utilized building information modeling to arrive at a satisfactory spatial and structural design. On the issue of contaminant event monitoring, Kyriacou *et al.* [20] introduced a MATLAB toolbox namely COMOB, which could correctly monitor the air quality, especially in multi-zone buildings. Moreover, COMOB provided a platform for online detection and isolation of contaminants. For the purpose of finding high-performance homes, Antonopoulos *et al.* [2] described the working mechanism of building America solution center, which was a free toolbox to bring satisfactory practices for members of the building industry, helping their businesses to gain a new edge.

In view of the toolboxes above, toolboxes are significant for specific industries. Especially with the development of tunnel construction industry, designing an efficient toolbox is vital for guiding tunnel construction. However, at present, there is no dedicated toolbox to quickly analysis the bearing capacity loss rate of the lining. Thus, we introduced the machine learning algorithm in our toolbox to quickly simulate lining bearing capacity

III. TOOLBOX

A. FRAMEWORK

The proposed toolbox provided a comprehensive architecture, which contains data preprocessing, model building and results display. Specifically, to calculate the loss rate of lining bearing capacity more quickly, we designed the dedicated toolbox that contains the following six modules: sensor, FEM simulation, trainer, evaluator, model, predictor, the crosslinking relationship for each module as shown in Figure 2. Each module is divided into four parts: module name, description, variables and interface functions. The detailed descriptions of the variables in each module are provided in Table 1.

Besides, we divide the dataset into training set and testing set, and the training set is defined by 5 different size of subsets. In this way, users can make a comparison about the model performance under different subsets. As for the deployment of machine learning, we selected 5 types of representative machine learning models, and each category contains 2 methods for training, as shown in Table 2. In the next section, we will detailed introduce the 10 machine learning models. Subsequently, we input the test set into the trained model and output the estimated loss rate of the lining bearing capacity. Finally, during the performance evaluation phase, to clearly understand the error between the estimation and real value, we adopted the root-mean square error and goodness of fit to evaluate our models.

B. MACHINE LEARNING

1) LINEAR

Linear model [18] was developed in the noncomputer age, and it is still a useful tool to provide an interpretable description to predicting a quantitative value. Linear methods can be applied in train set with small numbers or sparse data. Consequently, they can sometimes surpass more complex non-linear models.

a: LINEAR REGRESSION

(LR) has been existed for a long time, which is adequate explanation of how the input affects the output in machine learning approaches [27]. Many classical machine learning can be considered as extensions of LR. Assuming a given input vectors $X = [x_1, x_2, \ldots, x_k]$, $x_i \in R^D$, it attempts to learn a model $f(x_i)$ to make the output closer to the real-values $Y = [y_1, y_2, \ldots, y_k]$, $y_i \in R^D$ as accurately as possible. This linear relationship can be written as

$$f(x_i) = wx_i + b \cong y_i, \tag{1}$$

where w and b are unknown coefficients or parameters. The task of the linear regression model is to determine w and b to minimize the error between $f(x_i)$ and y_i, so the function can be reformulated as

$$\underset{(w,b)}{\arg\min} \sum_{i=1}^{k} (f(x_i) - y_i)^2. \tag{2}$$

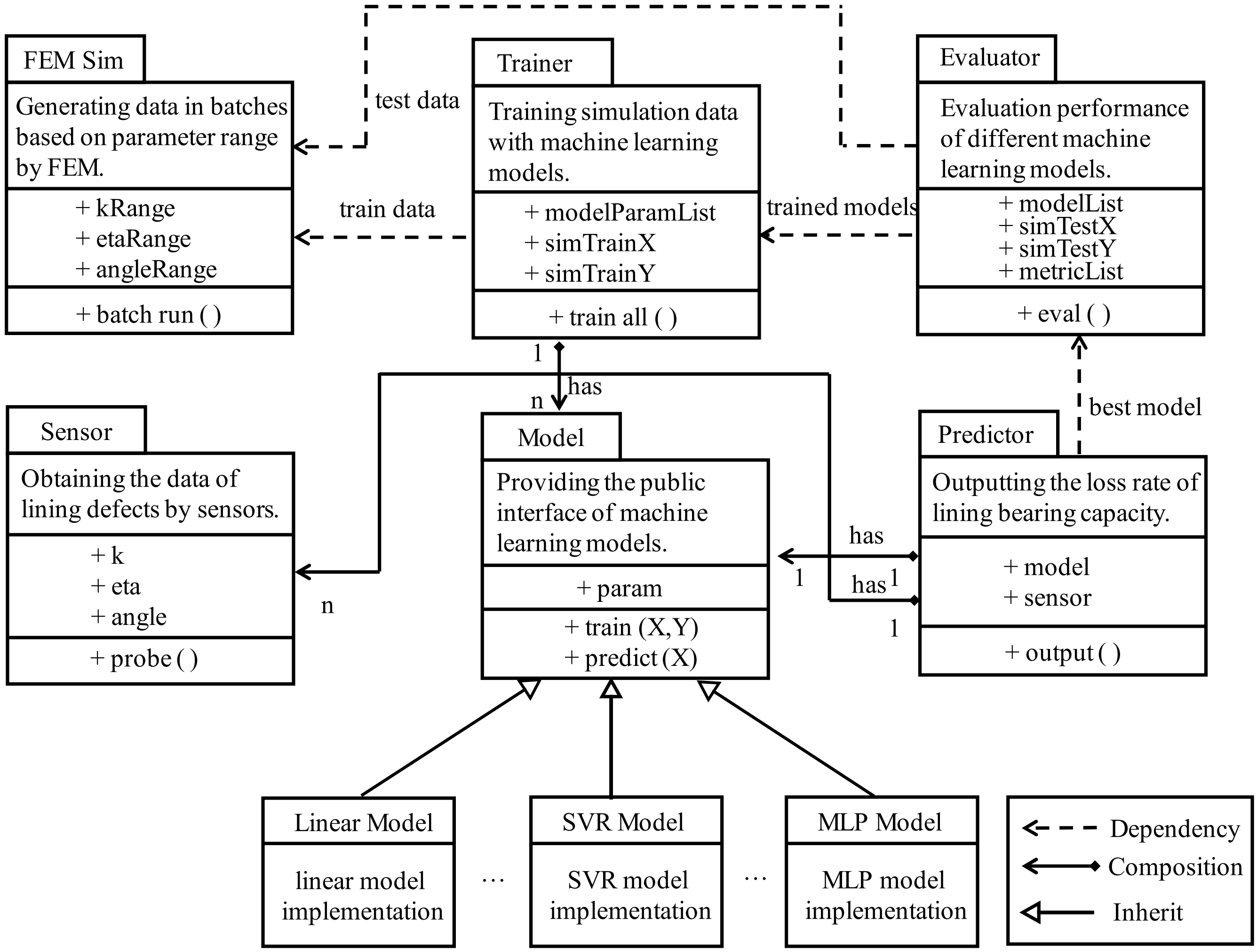


FIGURE 2. The object diagram of MLLBC toolbox.

We can further obtain the values of w and b by the method of least squares. Since less complicated calculations are required, it can deal with large amounts of data But this model is sensitive to high-dimensional data and outliers.

b: POLYNOMIAL REGRESSION

(PR) is a linear model used to accommodate non-linear data. In real world, most of the real-values are far from the prediction straight-line, and the accuracy of the results of the linear regression fitting can be reduced. Any non-linear data can be approximated by polynomials, so polynomial regression can be used to train linear models on non-linear functions of data. Suppose in the case of linear regression, there is a two-dimensional data model

$$f(x_i) = w_0 + w_1x_1 + w_2x_2 + b. \tag{3}$$

Equation (3) can be replaced by this polynomial function

$$f(x_i) = w_0 + w_1x_1 + w_2x_2^2 + w_3x_3^3 + \cdots + w_dx_i^d + b_i. \tag{4}$$

Transform Equation (4) to

$$f(x_i) = w_0 + w_1z_1 + w_2z_2 + w_3z_3 + \cdots + w_dz_d + b_i. \tag{5}$$

Through observation we find that the obtained polynomial regression can be solved with the same technique as the linear regression model. Polynomial regression considers the use of basis functions to build with high-dimensional linear fits, which can be adapted to a larger range of data.

2) KERNEL

The primary objective of the kernel method [39] is to address the linearly indivisible dataset in low-dimensional space. A set of points that cannot be linearly segmented in low-dimensional space is likely to become linearly separable by transforming into high-dimensional space, and kernel method is to find the suitable transformation function.

1) Support Vector Regression (SVR) is an important application branch of Support Vector Machine (SVM) [39]. The purpose of SVM is to find a classification plane that makes the data of different classes furthest away from

TABLE 1. Descriptions of the variables in each module.

Module	Variable	Description
FEM Sim	kRange	The stiffness of surrounding rock (from 20 MPa/m to 850MPa/m).
	etaRange	The ratio of the size of the cavity behind the lining to the complete lining (from 0 to 0.32).
	angleRange	The load applied to the lining model vertically (from 30° to 90°).
Trainer	modelParamList	Parameter lists for machine learning models.
	simTrainX	Train set input of simulation data.
	simTrainY	Train set output of simulation data.
Model	param	The relevant parameters of a single machine learning model.
Evaluator	modelList	List of machine learning models.
	simTestX	Test set input of simulation data.
	simTestY	Test set output of simulation data.
	metricList	List of methods that evaluate the performance of the models.
Sensor	k	The stratum stiffness.
	eta	The void ratio.
	angle	The angle of load.
Predictor	model	Pre-trained machine learning models.
	sensor	The data obtained by the lining defect detection sensors.

TABLE 2. Different categories of machine learning.

Category	Machine Learning
Linear	Linear Regression
	Polynomial Regression
Kernel	Support Vector Regression
	Kernel Ridge Regression
Tree and forest	Decision Tree
	Random Forest
Nearest Neighbors	k-Nearest Neighbor
	Radias Neighbors Regression
Neural Network	Multi-Layer Perceptron
	Multi-Layer Perceptron of the Limited Memory Broyden-Fletcher-Goldfarb-Shanno

that plane, while the purpose of SVR is to find a regression plane that makes the data of the same class nearest to that plane. Assuming a given training data $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, $y_i \in R^D$, the function of SVR can be written as

$$\min_{w,b} \frac{1}{2} \|w\|^2 + c \sum_{i=1}^{m} l_\varepsilon(f(x_i) - y_i), \tag{6}$$

where C is the regularization constant, and l_ε is the loss function. And w and b are unknown coefficients or parameters which need to be learned. The SVR shows that if the deviation between $f(x_i)$ and y_i is not too large, which can be considered that the prediction is correct.

2) Kernel Ridge Regression (KRR) is proposed by combining ridge regression and classification with the kernel trick [6], which has the same learning form as SVR but with different loss function. KRR uses the squared error loss while SVR uses ε-insensitive loss function.

The function of ridge regression can be written as

$$J(w) = (y - Xw)^T(y - Xw) + \lambda \|w\|^2, \tag{7}$$

where λ is the regression coefficient, the optimal solution can be obtained as

$$w = (X^T X + \lambda I_D)^{-1} X^T y = (\sum\nolimits_i x_i x_i^T + \lambda I_D)^{-1} X^T y. \tag{8}$$

By constraining the ridge regression to increase the kernel function, KRR model is obtained. The KRR has an approximate form of solution and is very efficient for medium sized datasets. In addition, KRR does not have the performance of parameter sparsity, so it is slower than the SVR.

3) TREE AND FOREST

Tree and forest-based methods can be applied to both regression and classification problems. The decision tree algorithm represents the result of data classification by tree structure, and each decision point implements a test function with discrete output. By the idea of integrated learning, the random forest-based algorithm can be obtained by integrating multiple trees.

1) Decision tree algorithm is to construct a suitable decision tree by learning the source data [40]. The decision tree generation is mainly divided into the flowing two steps, which are usually achieved by learning the labeled samples: (a) Node splitting: in general, when the attribute represented by a node cannot be judged, the node is divided into two. (b) determination of thresholds: select the appropriate threshold to minimize the classification error rate.

Information entropy (IE) represents the uncertainly of information, it is hoped that the IE of node feature is small, i.e. minimizing the following function

$$H(x) = -p_i(x) \log^{P_i(x)} = -\frac{n_j}{S} \log \frac{n_j}{S}, \tag{9}$$

where n_j denote the number of samples labeled j, and S is the total number of samples.

The commonly used decision trees are ID3, C4.5 and CART, and the classification effect of CART is generally better than that of other decision trees. ID3 depends on the entropy principle to determine the parent node, and for a set of data, the smaller the entropy, the better the classification result. However, ID3 often has the problem of over-learning, so C4.5 improved ID3 by adding optimization terms to constrain over-learning. CART tree is suitable for predicting discrete data results, mainly by calculating the Gini coefficient gain of each set of features to determine the priority rule of decision tree partitioning.

2) Random Forest (RF) is the integration of decision tree. To solve the problem existing in the decision tree model, the training set is resampled to form multiple training subsets.

Each subset generates a decision tree, and all decision trees make decisions by voting to form a random forest [42]. Random forest has many advantages: high accuracy, not easy to over-fitting, excellent noise resistance, high-dimensional data processing capacity and easily parallelized computing with high speed.

Because of the good characteristics of RF in practical application, many improved algorithms based on RF have been proposed, such as extra trees, and totally random trees embedding. Their application fields also have been extended from classification and regression problems to feature conversion and outlier detection.

4) NEAREST NEIGHBORS

Nearest neighbors [47] is an important part of pattern recognition method. As a statistical-based data mining method, neighbors-based algorithm aims to predict the unknown feature value of the current record through a set of historical data records. During the past years, neighbors-based method has been widely used in classification and regression problems and has achieved excellent results.

1) k-nearst Neighbors (kNN) is a well-known statistical method of pattern recognition, which is very important in machine learning classification algorithms [49]. The main idea of kNN algorithm is as follows: in order to judge the categories of unknown samples, the distance between the unknown samples and all known samples is calculated, and the k known samples which are closest to the unknown samples are selected. Then according to the majority-voting rule, the unknown samples were classified as one of the most adjacent samples. The Euclidean distance is often used to calculate the measure of similarity between samples

$$d_{euc}(x, y) = [\sum_{j=1}^{d} (x_j - y_j)^2]^{\frac{1}{2}} = [(x - y)(x - y)^T]^{\frac{1}{2}}. \quad (10)$$

kNN algorithm has many advantages: simple and effective, lower costs of retraining, linear relationship between complexity and training dataset, and suitable for automatic classification of large sample dataset. The effect of kNN algorithm mainly depends on the training set, distance or similar measure and size of k, so it is important to set the parameters according to the source data when dealing with classification or regression problems.

2) Radias Neighbors Regression is one of the nearest neighbors-based methods, and the principle of the radias neighbors regression method is to predict the categories of new sample based on the labelled samples closest to the new sample [26]. The advantages of radias neighbors regression is that the continuous data can be predicted, and it has been successful on numerous classification and regression problems, including handwritten numbers and satellite image scenes. Compared with kNN, better regression results can be obtained by limiting the adjacent radius.

5) NEURAL NETWORK

Neural network is a distributed parallel information processing model that imitates the behavioral characteristics of animal neural networks. To achieve the purpose of information processing, neural network adjusts the interconnection relationship between a large number of internal nodes and it is a hot topic of the new generation of intelligent systems.

1) Multi-Layer Perceptron (MLP), also known as Artificial neural network (ANN) [13], can be modeled as

$$a_h = \sum_{i=1}^{I} w_{ih} x_i, \quad (11)$$

$$b_h = f(a_h), \quad (12)$$

where $f(\cdot)$ is the nonlinear activation function, in which the sigmoid and Tanh are the commonly used activation functions in MLP. The main purpose of $f(\cdot)$ is to improve the fitting ability of neural network by adding nonlinear terms to the computation process between different network layers.

The first layer of MLP is the input layer, the last layer is the output layer, and the middle layers are the hidden layer. Therefore, suitable numbers of hidden layer can be set according to different task and source dataset, thus MLP is widely used in classification and prediction problems.

2) Multi-Layer Perceptron of the Limited Memory Broyden-Fletcher-Goldfarb-Shanno (MLP_LBFGS) is a typical gradient based optimization algorithm [58]. During each iteration, the approximation to the inverse Hessian need to be updated. The updating formula can be determined as

$$\tilde{H}_{i+1}^{-1} = \gamma_i V_i^T \tilde{H}_i^{-1} V_i + \rho_i s_i s_i^T, \quad (13)$$

where the search direction vector $\rho_i = 1/y_i^T s_i$, $s_i = x_{i+1} - x_i$, $V_i = I - \rho_i s_i s_i^T$ and γ_i is the i-th scaling factor. Here I is the $N_m \times N_m$ identity matrix.

Compared with Stochastic Gradient Descent (SGD) algorithm, LBFGS utilize gradient information to approximate the inverse of the Hessian matrix. Specifically, LBFGS exploits the second-order approximation between parameters to accelerate optimization, thus it converges faster and performs better.

C. PERFORMANCE EVALUATION

In our toolbox, to intuitively evaluate the error between the results predicted by the machine learning model and the real-world mechanical experimental results, and the degree of fitting of the bearing capacity loss rate of the lining with the machine learning model, we use two indicators to evaluate the performance of the model: root-mean square error and goodness of fit.

1) ROOT-MEAN SQUARE ERROR (RMSE)

RMSE is also known as standard error, which is used to measure the deviation between the predicted value and the true value. It is expressed as

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2}, \tag{14}$$

where n is the number of measurements. From Equation (10) we note that RMSE close to 0 indicates that the higher the accuracy of the model. The root-mean square error is quite sensitive to the outlier (such as the error is extraordinarily large or small) reflection in a group of measurements.

2) GOODNESS OF FIT (R^2)

The R^2 is also called the coefficient of determination, which provides an alternative assessing of fit. To calculate R^2, given by

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - f(x_i))^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{15}$$

where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, which is the mean value of the true value. The value range of R^2 is $(-\infty, 1]$, and we would expect R^2 is close to 1 extremely, which demonstrates that the model fits the data better. R^2 near 0 indicates that the model unexplained much of the response.

IV. EXPERIMENTS

In this section, comparative experiments were conducted using different models on the dataset to demonstrate the machine learning algorithm in our toolbox that can quickly model the loss rate of lining bearing capacity. Performance was assessed using the correlation coefficient and confusion matrices for each algorithm. The experimental details of our proposed toolbox are as follows.

A. DATASETS

Since there are no standardized, publicly-available loss rate of the lining bearing capacity datasets, we establish a novel dataset by FEM model. Our dataset was obtained using the Concrete Damage Plasticity (CDP) model in ABAQUS tool [11] based on the bearing capacity experiments of real lining models, which is a set of finite element software for engineering simulation. Here, we first describe the process of obtaining the actual bearing capacity of the lining model. The lining model is made according to a 1/5 scale of the prototype of the tunnel lining in the real scene. The width and height of the lining model are 2.3m × 1.56m, the thickness is 80cm, and the length is 300mm. Figure 3 shows the dimensions of the lining model. In this experiment, two lining models were established based on the lining materials divided into non-reinforced concrete (NC) and reinforced concrete (RC).

To eliminate the difference between FEM simulation and physical measurement, we assume that the two main failure modes of concrete are tensile cracking and compressive crushing. The stress-crushing (or cracking) strain relationship of concrete is related to the grid size. By controlling

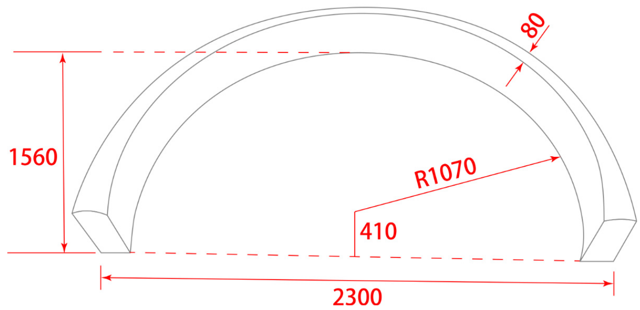


FIGURE 3. The dimensions of the lining model. The thickness of the lining model is 80cm, the inner radius is 1070cm, the outer diameter is 2300cm, and the height from the arch foot to the vault is 1560cm.

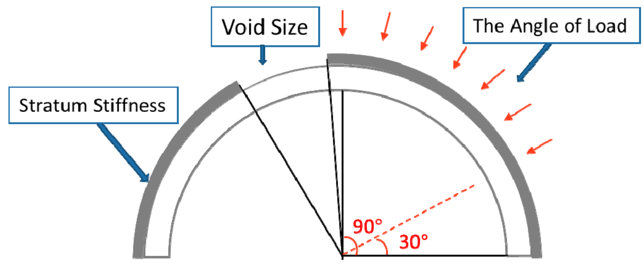


FIGURE 4. The dimensions of the lining model. A model of the surrounding rock is included next to the outer layer of the lining model. This is to simulate the stratum stiffness where there are no surrounding rocks is to simulate voids. In addition, different angles of load are applied to the outermost layer of the surrounding rock.

the parameters of compressive fracture bearing capacity and tensile fracture bearing capacity of a single unit of concrete, the plastic behavior of different units is consistent. Our experiment compares the calculation results of no-void linings at 5mm, 10mm, and 20mm grid sizes. And finds that the load-displacement curve shapes and peak loads at the three grid sizes are very close, indicating that the simulation results of the lining under different grids have converged, that is, the FEM model is close to the result of the physical model.

Three combined experiments were performed on two kinds of lining models to calculate the loss rate of lining bearing capacity under different working conditions. Examples are shown in Figure 4: 1) the angle of load: the load applied to the lining model vertically (from 30° to 90°); 2) void ratio: the ratio of the size of the cavity behind the lining to the complete lining (from 0 to 0.32); 3) stratum stiffness: the stiffness of surrounding rock (from 20 MPa/m to 850MPa/m). There are 11 groups of such combination experiments in lining model. Based on these 11 groups of data, 1500 different working conditions are calculated using ABAQUS.

B. THE SETTING AND ENVIRONMENT

We divided the dataset into five train sets (20/50/200/500/1000 working conditions) and a test set (500 working conditions). Our toolbox was performed in the Python 3 and running on i7-7500 CPU. To ensure fairness in the contrast experiment, the environment was kept consistent throughout. In addition, it takes about 2 hours to model a group of projects,

TABLE 3. The root-mean squard error of the NC lining on machine learning models.

Train Set	LR	PR	SVR	KRR	DT	RF	kNN	RNR	MLP_sgd	MLP_lbfgs
20	0.1500	0.2696	0.1731	0.1220	0.1511	0.1075	0.2419	0.3494	0.2837	0.1852
50	0.116	0.1588	0.1082	0.1029	0.1042	0.0707	0.1661	0.348	0.1265	0.0898
100	0.1152	0.0705	0.1007	0.1030	0.0799	0.0582	0.1207	0.1586	0.2532	0.0592
500	0.1092	0.0526	0.0951	0.1001	0.0575	0.0435	0.0712	0.0725	0.0928	0.0358
1000	0.1084	**0.0507**	0.0937	0.0987	**0.0492**	**0.0432**	**0.0574**	0.0616	0.0845	**0.0347**

TABLE 4. The goodness of fit of the NC lining on machine learning models.

Train Set	LR	PR	SVR	KRR	DT	RF	kNN	RNR	MLP_sgd	MLP_lbfgs
20	0.8542	0.5774	0.3056	0.8113	0.8314	0.8999	-0.2756	0	-14.6128	0.7369
50	0.8974	0.815	0.8592	0.8943	0.9063	0.9578	0.5940	0	0.8868	0.9353
100	0.8988	0.9572	0.9009	0.9069	0.9431	0.9702	0.8252	0.4435	-3.4378	0.9720
500	0.9035	0.9753	0.9207	0.9166	0.9717	0.9833	0.951	0.9449	0.9253	0.9891
1000	0.9048	0.9770	0.9250	0.9197	0.9793	**0.9837**	0.9699	0.9606	0.9421	**0.9896**

TABLE 5. The root-mean squard error of the RC lining on machine learning models.

Train Set	LR	PR	SVR	KRR	DT	RF	kNN	RNR	MLP_sgd	MLP_lbfgs
20	0.2290	0.3163	0.3048	0.2295	0.1993	0.1582	0.3631	0.5091	0.2150	0.2655
50	0.1917	0.1900	0.2009	0.1674	0.1782	0.1535	0.2408	0.5042	0.3272	0.1352
100	0.1929	0.0916	0.1676	0.1644	0.1389	0.1253	0.1828	0.2438	0.3557	0.1015
500	0.1846	0.0721	0.1357	0.1417	0.0913	0.0926	0.1027	0.1068	0.1713	0.0539
1000	0.1831	**0.0698**	0.1273	0.1345	0.1282	0.0830	0.0817	0.0901	0.0836	**0.0492**

TABLE 6. The goodness of fit of the RC lining on machine learning models.

Train Set	LR	PR	SVR	KRR	DT	RF	kNN	RNR	MLP_sgd	MLP_lbfgs
20	0.8016	0.6800	-0.3079	0.6463	0.8405	0.8912	-0.4101	0	0.8047	0.6928
50	0.8621	0.8646	0.7094	0.8631	0.8745	0.9045	0.5627	0	-1.2475	0.9330
100	0.8532	0.9661	0.8560	0.8791	0.9148	0.9313	0.7962	0.3227	-2.4564	0.9607
500	0.8607	0.9787	0.9213	0.9182	0.9657	0.9622	0.9502	0.9416	0.8708	0.9882
1000	0.8641	**0.9801**	0.9338	0.9281	0.9318	**0.9703**	**0.9709**	0.9593	0.9697	**0.9901**

so we used three computers and performed a one-month finite element calculation at the same time.

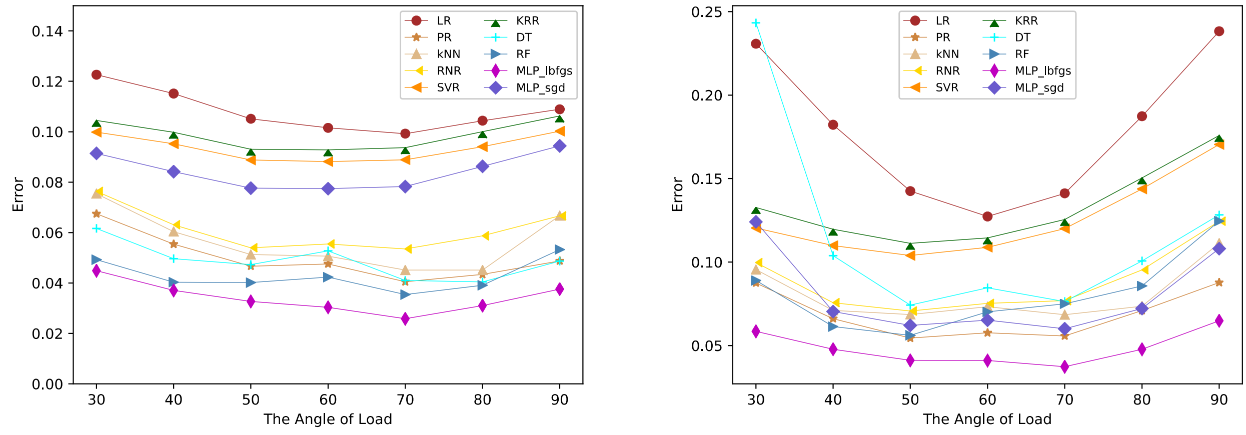
V. RESULTS AND DISCUSSION

Comparative experiments were conducted using different models on the dataset to demonstrate the machine learning algorithm that can quickly simulate reinforced concrete. We report our results using the root-mean square error (RMSE) and goodness of fit (R^2) for each model. The details of the performance comparison between the models in our proposed toolbox are as follows.

A. COMPARISON OF VARYING TRAIN SET

To analyze comprehensively machine learning performance in our toolbox, we performed experiments on the varying number of train set. The root-mean square error and goodness of fit on the non-reinforced concrete are shown in Table 3 and Table 4. The results on reinforced concrete are shown in Table 5 and Table 6. In general, we observe that the model performances go down as the number of train set increases because we can obtain the more information from a much larger sample to improve the stability of the model. However, we note that the linear regression is not obviously affected by the number of train set in either RC or NC. This may be attributed to a single straight line is difficult to fit the distribution of a larger number of data. In addition, these table shows that the multi-layer perceptron increases very significantly as the number of train set. This demonstrates that the multi-layer perceptron requires a lager train data to make the model stable.

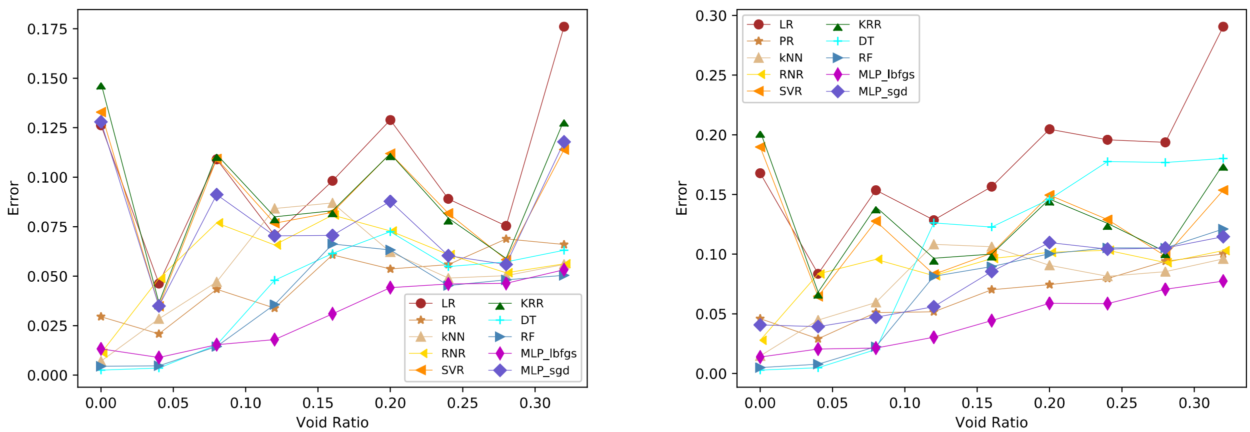
Here, we also note that the number of train set is only 20 or 50, decision tree and random forest show respectable performance compared to other models. This is because the logical structure of models such as tree or forest is simple,



(a) The Angle of Load on NC lining

(b) The Angle of Load on RC lining

FIGURE 5. **The root-mean square error of the angle of load in different value ranges on the NC lining and RC lining.**



(a) Void Ratio on NC lining

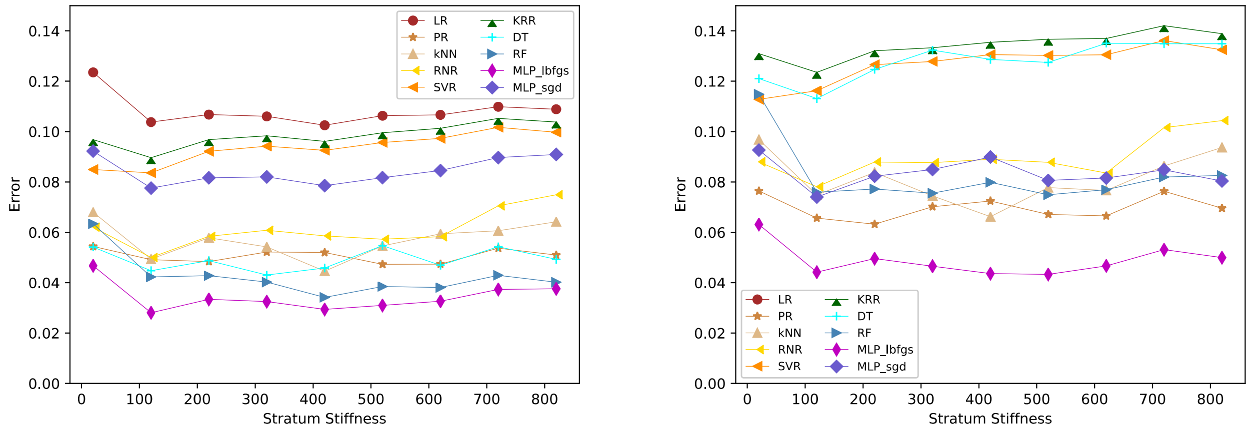
(b) Void Ratio on RC lining

FIGURE 6. **The root-mean square error of the void ratio in different value ranges on the NC lining and RC lining.**

and there are no complex calculation functions, and training on a small number of data is suitable for such models instead. This suggests that the model of tree or forest in terms of information gain is most suitable to the small sample problem. It is worth noting that the performance of polynomial regression is relatively superior to when the number of training sets is 1000. Compared with linear regression, it can increase the higher-order term to approach the measurements until it is satisfied. It is quite interesting that the MLP_sgd is not the best performance in the case of large train set, while the performance of MLP_lbfgs is the best. This is because SGD needs to be adjusted manually, such as the learning rate and convergence criteria. However, LBFGS uses a larger minibatch to estimate the expected excitation value of each node, and the performance has been significantly improved. This is also related to our data dimension. In our experiments, the input data dimension was 3, the MLP_lbfgs shows its superior performance in low-dimensional data processing.

B. COMPARISON OF VARYING VARIABLE

In this section, to more intuitively illustrate the fit of the model on different variables, Figure 5 to Figure 7 show the model performance of each variable in different value ranges. We set the number of train set to 1000. In general, the root-mean squard errors of MLP_lbfgs, polynomial regression, k-nearest neighbor, decision tree and random forest are small, and the stronger performance of random forest in each variable case shown in the figure. This suggests that the simpler the structure of the model, the more suitable it is for the task of calculating the loss rate of the bearing capacity of the lining. This task has the characteristics of a small number of variables input and fewer train samples. In addition, it is not surprising that the performance of random forests is better than that of decision trees. This is because random forest is an ensemble learning method essentially, which can better integrate the differences of individual classifiers and make the final generalization performance improve.



(a) Stratum Stiffness on NC lining

(b) Stratum Stiffness on RC lining

FIGURE 7. The root-mean square error of the stratum stiffness in different value ranges on the NC lining and RC lining.

Next, we analyze the performance of different machine learning models on each variable. We found some interesting phenomena. The results in Figure 5(a) show that all models achieve the best performance when angle of load is at 60°. This is because there are many experiments in the case of the angle of load at 60° in the original physics experiment on RC dataset. These results further corroborate what we observe in Table 3 to Table 6, with decision tree and random forest being among the better performing machine learning models. Generally speaking, MLP_lbfgs have the best performance. Given the involved in working with lower dimensional feature spaces, these results suggest that tree model and neural network are widely used to predict the loss rate of lining bearing capacity, which seems to be a reasonable choice and provides better performance to the other choices in most cases.

VI. CONCLUSION

In this work, we propose a toolbox that can quickly calculate the loss rate of lining bearing capacity and make a design about the machine learning models in the toolbox. In doing so, we introduce a new dataset of loss rate of lining bearing capacities with 1500. To verify the effectiveness of our proposed toolbox, we conducted comprehensive experiments with different machine learning models. Experiment results exploited that random forest perform well in terms of performance and computing efficiency when the dataset is small. In addition, when the sample is sufficient, using quasi-Newton algorithm to optimize the multilayer perceptron can achieve the best results. And MLP_lbfgs has the lowest average error rate under different variables. The above conclusions can be used as model selection guide to calculate the loss rate of the realistic lining bearing capacity.

REFERENCES

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, J. Davis, J. Dean, and M. Kudlur, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th Symp. Oper. Syst. Design Implement.*, 2016, pp. 265–283.

[2] C. A. Antonopoulos, M. C. Baechler, and T. L. Gilbride, "A toolbox necessity: Finding best practices in high-performance homes with the building America solution center," Pacific Northwest Nat. Lab. (PNNL), Richland, WA, USA, Tech. Rep. PNNL-SA-134350, 2018.

[3] A. S. Berahas, J. Nocedal, and M. Takác, "A multi-batch L-BFGS method for machine learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1055–1063.

[4] S. Boonstra, K. van der Blom, H. Hofmeyer, J. van den Buijs, and M. T. Emmerich, "Coupling between a building spatial design optimisation toolbox and BouwConnect BIM," in *Proc. Adv. Inform. Comput. Civil Construct. Eng.*, 2019, pp. 95–102.

[5] M. Cai, "Influence of stress path on tunnel excavation response—Numerical tool selection and modeling strategy," *Tunnelling Underground Space Technol.*, vol. 23, no. 6, pp. 618–628, Nov. 2008.

[6] X. Chang, S.-B. Lin, and D.-X. Zhou, "Distributed semi-supervised learning with kernel ridge regression," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 1493–1514, Jan. 2017.

[7] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, Apr. 2011.

[8] J. M. Clairand, J. Rodríguez-García, and C. Álvarez-Bel, "Smart charging for electric vehicle aggregators considering users' preferences," *IEEE Access*, vol. 6, pp. 54624–54635, 2018.

[9] Y. Gao, X. Liu, and J. Xiang, "FEM simulation-based generative adversarial networks to detect bearing faults," *IEEE Trans Ind. Informat.*, to be published.

[10] S. Ghimire, P. K. Dhital, and A. K. Mishra, "Small signal stability analysis toolbox: A MATLAB based GUI," in *Proc. 2nd Int. Conf. Adv. Comput. Commun. Paradigms (ICACCP)*, Feb. 2019, pp. 1–5.

[11] E. Giner, N. Sukumar, J. E. Tarancón, and F. J. Fuenmayor, "An Abaqus implementation of the extended finite element method," *Eng. Fract. Mech.*, vol. 76, no. 3, pp. 347–368, Feb. 2009.

[12] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, "Detectron," Tech. Rep., 2018. [Online]. Available: https://github.com/facebookresearch/Detectron

[13] C. N. Gupta, R. Palaniappan, S. Swaminathan, and S. M. Krishnan, "Neural network classification of homomorphic segmented heart sounds," *Appl. Soft Comput.*, vol. 7, no. 1, pp. 286–297, Jan. 2007.

[14] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, May 1996.

[15] K. He, R. Wang, D. Tao, J. Cheng, and W. Liu, "Color transfer pulse-coupled neural networks for underwater robotic visual systems," *IEEE Access*, vol. 6, pp. 32850–32860, 2018.

[16] J. H. Kim, "Tackling false positives in business research: A statistical toolbox with applications," *J. Econ. Surv.*, vol. 33, no. 3, pp. 862–895, Jul. 2019.

[17] C. Koch, A. Vonthron, and M. König, "A tunnel information modelling framework to support management, simulations and visualisations in mechanised tunnelling projects," *Automat. Construct.*, vol. 83, pp. 78–90, Nov. 2017.

[18] A. Kumar, R. B. Chinnam, and F. Tseng, "An HMM and polynomial regression based approach for remaining useful life and health state estimation of cutting tools," *Comput. Ind. Eng.*, vol. 128, pp. 1008–1014, Feb. 2019.
[19] K. Kurach, A. Raichuk, P. Stańczyk, M. Zając, O. Bachem, L. Espeholt, C. Riquelme, D. Vincent, M. Michalski, O. Bousquet, and S. Gelly, "Google research football: A novel reinforcement learning environment," 2019, *arXiv:1907.11180*. [Online]. Available: http://arxiv.org/abs/1907.11180
[20] A. Kyriacou, M. P. Michaelides, D. G. Eliades, C. G. Panayiotou, and M. M. Polycarpou, "COMOB: A MATLAB toolbox for sensor placement and contaminant event monitoring in multi-zone buildings," *Building Environ.*, vol. 154, pp. 348–361, May 2019.
[21] B. Li, D. Sahoo, and S. C. H. Hoi, "OLPS: A toolbox for on-line portfolio selection," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1242–1246, 2016.
[22] T. Li, "Damage to mountain tunnels related to the wenchuan earthquake and some suggestions for aseismic tunnel construction," *Bull. Eng. Geol. Environ.*, vol. 71, no. 2, pp. 297–308, May 2012.
[23] D.-X. Liang, Z.-Q. Jiang, S.-Y. Zhu, Q. Sun, and Z.-W. Qian, "Experimental research on water inrush in tunnel construction," *Natural Hazards*, vol. 81, no. 1, pp. 467–480, Mar. 2016.
[24] X. Liu, H. Huang, and J. Xiang, "A personalized diagnosis method to detect faults in a bearing based on acceleration sensors and an FEM simulation driving support vector machine," *Sensors*, vol. 20, no. 2, p. 420, 2020.
[25] Y. Liu, Q. Zhang, G. Zhao, Z. Qu, G. Liu, Z. Liu, and Y. An, "Detecting diseases by human-physiological-parameter-based deep learning," *IEEE Access*, vol. 7, pp. 22002–22010, 2019.
[26] C. Llerena, D. Müller, R. Adams, N. Davey, and Y. Sun, "Estimation of microphysical parameters of atmospheric pollution using machine learning," in *Proc. Int. Conf. Artif. Neural Netw.*, 2018, pp. 579–588.
[27] Y.-C. Lo, S. E. Rensi, W. Torng, and R. B. Altman, "Machine learning in chemoinformatics and drug discovery," *Drug Discovery Today*, vol. 23, no. 8, pp. 1538–1546, Aug. 2018.
[28] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The stanford CoreNLP natural language processing toolkit," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics, Syst. Demonstrations*, 2014, pp. 55–60.
[29] G. Manogaran, P. M. Shakeel, A. S. Hassanein, P. M. Kumar, and G. C. Babu, "Machine learning approach-based gamma distribution for brain tumor detection and data sample imbalance analysis," *IEEE Access*, vol. 7, pp. 12–19, 2019.
[30] S. Melzer, J. P. Speichert, O. C. Eichmann, and R. God, "Simulating cyber-physical systems using a broker-based SysML toolbox," in *Proc. 7th Int. Workshop Aircr. Syst. Technol. (AST)*, 2019, pp. 411–420.
[31] X. Meng, J. Bradley, B. Yavu, E. Sparks, S. Venkataraman, D. Liu, D. Xin, R. Xin, M. J. Franklin, R. B. Zadeh, M. A. Zaharia, A. Talwalkar, D. Liu, and J. Freeman, "MLlib: Machine learning in apache spark," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1235–1241, 2016.
[32] O. Metsker, S. Kesarev, E. Bolgova, K. Golubev, A. Karsakov, A. Yakovlev, and S. Kovalchuk, "Modelling and analysis of complex patient-treatment process using GraphMiner toolbox," in *Proc. Int. Conf. Comput. Sci.*, 2019, pp. 674–680.
[33] A. Morshed, P. P. Jayaraman, T. Sellis, D. Georgakopoulos, M. Villari, and R. Ranjan, "Deep osmosis: Holistic distributed deep learning in osmotic computing," *IEEE Cloud Comput.*, vol. 4, no. 6, pp. 22–32, Nov. 2017.
[34] R. Neukirchen and M. Vollmer, "Controlling toolbox für ein erfolgreiches ChangeManagement im finance shared services projekt," *Controlling*, vol. 19, no. 2, pp. 91–98, 2007.
[35] L. Nkenyereye, L. Nkenyereye, S. M. R. Islam, C. A. Kerrache, M. Abdullah-Al-Wadud, and A. Alamri, "Software defined network-based multi-access edge framework for vehicular networks," *IEEE Access*, vol. 8, pp. 4220–4234, 2020.
[36] T. Orava, C. Provvidenza, A. Townley, and S. Kingsnorth, "Screening and assessment of chronic pain among children with cerebral palsy: A process evaluation of a pain toolbox," *Disab. Rehabil.*, vol. 41, no. 22, pp. 2695–2703, Oct. 2019.
[37] S. Pang, Y. Zhang, M. Ding, X. Wang, and X. Xie, "A deep model for lung cancer type identification by densely connected convolutional networks and adaptive boosting," *IEEE Access*, vol. 8, pp. 4799–4805, 2020.
[38] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, and A. Lerer, "Automatic differentiation in PyTorch," Tech. Rep., 2017.
[39] A. M. Philip, G. Ramadurai, and L. Vanajakshi, "Urban arterial travel time prediction using support vector regression," *Transp. Developing Economies*, vol. 4, no. 1, pp. 7–16, Apr. 2018.
[40] N. Reix, M. Lodi, S. Jankowski, S. Molière, E. Luporsi, S. Leblanc, L. Scheer, I. Ibnouhsein, J.-C. Benabu, V. Gabriele, A. Guggiola, J.-M. Lessinger, M.-P. Chenard, F. Alpy, J.-P. Bellocq, K. Neuberger, C. Tomasetto, and C. Mathelin, "A novel machine learning-derived decision tree including uPA/PAI-1 for breast cancer care," *Clin. Chem. Lab. Med.*, vol. 57, no. 6, pp. 901–910, May 2019.
[41] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The Hadoop distributed file system," in *Proc. MSST*, vol. 10, 2010, pp. 1–10.
[42] A. Subasi, E. Alickovic, and J. Kevric, "Diagnosis of chronic kidney disease by using random forest," in *Proc. CMBEBIH*, 2017, pp. 589–594.
[43] W. Song, J. Xiang, and Y. Zhong, "A simulation model based fault diagnosis method for bearings," *J. Intell. Fuzzy Syst.*, vol. 34, no. 6, pp. 3857–3867, Jun. 2018.
[44] D. Tao, J. Cheng, Z. Yu, K. Yue, and L. Wang, "Domain-weighted majority voting for crowdsourcing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 1, pp. 163–174, Jan. 2019.
[45] D. Tao, Y. Guo, B. Yu, J. Pang, and Z. Yu, "Deep multi-view feature learning for person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2657–2666, Oct. 2018.
[46] D. Tao, Y. Guo, Y. Li, and X. Gao, "Tensor rank preserving discriminant analysis for facial recognition," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 325–334, Jan. 2018.
[47] R. Taormina, S. Galelli, H. C. Douglas, N. O. Tippenhauer, E. Salomons, and A. Ostfeld, "A toolbox for assessing the impacts of cyber-physical attacks on water distribution systems," *Environ. Model. Softw.*, vol. 112, pp. 46–51, Feb. 2019.
[48] M. Thangarajh, A. J. Kaat, G. Bibat, J. Mansour, K. Summerton, A. Gioia, K. K. Hardy, and K. R. Wagner, "The NIH toolbox for cognitive surveillance in Duchenne muscular dystrophy," *Ann. Clin. Transl. Neurol.*, vol. 6, no. 9, pp. 1696–1706, 2019.
[49] J. Vitola, F. Pozo, D. Tibaduiza, and M. Anaya, "A sensor data fusion system based on k-nearest neighbor pattern classification for structural health monitoring applications," *Sensors*, vol. 17, no. 2, pp. 417–426, 2017.
[50] A. S. Voznesenskii and V. V. Nabatov, "Identification of filler type in cavities behind tunnel linings during a subway tunnel surveys using the impulse-response method," *Tunnelling Underground Space Technol.*, vol. 70, pp. 254–261, Nov. 2017.
[51] S. Wang, J. Xiang, Y. Zhong, and Y. Zhou, "Convolutional neural network-based hidden Markov models for rolling element bearing fault identification," *Knowl.-Based Syst.*, vol. 144, pp. 65–76, Mar. 2018.
[52] S. Wang and J. Xiang, "A minimum entropy deconvolution-enhanced convolutional neural networks for fault diagnosis of axial piston pumps," *Soft Comput.*, vol. 24, no. 4, pp. 2983–2997, Feb. 2020.
[53] S. Wang, J. Xiang, Y. Zhong, and H. Tang, "A data indicator-based deep belief networks to detect multiple faults in axial piston pumps," *Mech. Syst. Signal Process.*, vol. 112, pp. 154–170, Nov. 2018.
[54] S. Wang, Y. Jian, X. Lu, L. Ruan, W. Dong, and K. Feng, "Study on load distribution characteristics of secondary lining of shield under different construction time," *Tunnelling Underground Space Technol.*, vol. 89, pp. 25–37, Jul. 2019.
[55] A. Wiliński, A. Smoliński, and W. Nowicki, "Investment funds management strategy based on polynomial regression in machine learning," in *Intelligent Systems for Computer Modelling*. Cham, Switzerland: Springer, 2016, pp. 87–97.
[56] J. Wu, P. Guan, and Y. Tan, "Diagnosis and data probability decision based on non-small cell lung cancer in medical system," *IEEE Access*, vol. 7, pp. 44851–44861, 2019.
[57] J. Xiang and Y. Zhong, "A novel personalized diagnosis methodology using numerical simulation and an intelligent method to detect faults in a shaft," *Appl. Sci.*, vol. 6, no. 12, p. 414, 2016.
[58] F. Zhang and A. C. Reynolds, "E48: Optimization algorithms for automatic history matching of production data," in *Proc. 8th Eur. Conf. Math. Oil Recovery*, 2002, pp. 1–11.
[59] X. Zhao, W. Zang, R. Lv, and W. Cui, "Effective information filtering mining of Internet of brain things based on support vector machine," *IEEE Access*, vol. 7, pp. 191–202, 2019.
[60] N. Zimmerman, A. A. Presto, S. P. N. Kumar, J. Gu, A. Hauryliuk, E. S. Robinson, A. L. Robinson, and R. Subramanian, "A machine learning calibration model using random forests to improve sensor performance for lower-cost air quality monitoring," *Atmos. Meas. Techn.*, vol. 11, no. 1, pp. 291–313, 2018.