Integrated Real-time Data-driven Model Framework for Optimization of Slurry Control Parameters in SPB-TBM Tunnelling

Sicheng Zhao[1], Xiaojun Li[1], Haijiang Li[2], Qingfeng Du[3], Shuangli Zhang[3], Yi Shen[1],Xuhui Li[4], Guangwen Xue[4]
[1] College of Civil Engineering, Tongji University, Shanghai, China
[2] School of Engineering, Cardiff University, Wales, U.K.
[3] School of Soft Engineering, Tongji University, Shanghai, China
[4] Guangdong Yuehai Pearl River Delta Water Supply Co. Ltd., Guangzhou, China
sichengzhao@tongji.edu.cn

Abstract. Maintaining the stability of excavation face is one of the most concerned problems in Slurry pressure balanced tunnel boring machine (SPB-TBM) construction. To overcome the uncertainty and errors caused by manual adjustment, this paper proposes a novel data-driven model framework to achieve the balance between slurry and soil pressure, which consists of two significant components: (1) an ensemble-learning-based model for predicting slurry pressure; (2) an optimization model based on greedy search strategy for slurry control parameter. The proposed framework was implemented and verified in Pearl River Delta Water Resources Allocation Project and the results demonstrated that the presented framework can make highly-accurate predictions for the slurry pressure and effectively adjust the control parameter values to achieve the balance between slurry and soil pressure.

1. Introduction

The construction of underground engineering plays a critical role in the sustainable development of European cities, by mitigating land use pressure and reducing environmental pollution. Tunnel boring machine (TBM), which is characterized by its high excavation efficiency, minimal environmental impact, and strong adaptability to the stratum, has become a popular option for various types of tunnel projects in Europe (Li et al., 2023), e.g. the London JLE metro (UK), Brescia metro (Italy) and Lyon metro (France) (Rallu et al., 2023). Particularly, slurry pressure balanced (SPB) TBM is widely used in the construction of submarine tunnel because of its advantages in maintaining the stability of excavation face (Zhang et al., 2021).

In SPB-TBM tunnelling, the balance between the slurry pressure (P_s) and the expected water-earth pressure (P_e) is crucial for both efficiency and safety. To reduce ground deformation by using air buffer to mitigate pressure fluctuations in the slurry chamber, indirect-type (German style with air chamber) SPB TBMs are currently mainly used. However, slurry control parameters are still manually adjusted based on the engineer's experience in practise, which has the characteristics of delay and unreliability. Recently, because of its advantages in modelling and controlling non-linear systems (Gao et al., 2019), machine learning (ML) methods have been proverbially employed in TBM tunnelling (Gao et al., 2021). Specifically, several researchers have explored the development of an ML-based approach for dynamically balancing P_s and P_e from various perspectives. Zhou et al. (2013) proposed a predictive control system for air chamber pressure using an Elman recurrent network (ENN). Liu et al. (2010) proposed a method combining mechanism analysis and least squares support vector machine (LS-SVM) technology for balancing chamber pressure. Liu and Zhang (2019) also used LS-SVM to predict earth pressure in the chamber during the tunnelling process. However, these studies do not fully consider the influence of the combined action of tunnelling operating and slurry control parameters, and pay more attention to the theoretical analysis and prediction tasks, without considering how to optimize the operating parameter strategy for practical application.

Ensuring high prediction accuracy is also a crucial challenge for ML tasks. Often, a single ML model performs not so well due to its high bias. However, combining multiple base learners to form an integration model may not always yield robust results due to the variance between the base models. To address this issue, ensemble learning algorithms have emerged as the state-of-the-art solution for improving the predictive performance of a single model by training multiple models and combining their predictions (Zhou and Zhou, 2021). Ensemble methods aims at combining several of these base learners to create a strong learner (or an ensemble model) to achieve better predictive performance. The use of ensemble learning-based methods has also been explored in the field of tunnelling construction for rock mass classification (Liu et al., 2020), estimation of advance rate (Zhou et al., 2021), etc. However, there is a dearth of research on optimizing operating parameters by predicting results using ensemble learning methods.

To obtain real-time and reliable parameter adjustments to balance the P_s and expected water-earth pressure in SPB-TBM tunnelling, we proposed a novel data-driven model framework. The framework includes an ensemble learning-based model for predicting the P_s, and a greedy search-based optimization model for slurry control parameters. The continuation of this paper is organized as follows: Section 2 gives a detailed explanation of the methodology employed in this paper. Section 3 presents the application case and the framework execution results. Section 4 concludes the researches in this paper.

2. Methodology

The objective of the data-driven model framework proposed in this study is to predict the P_s under different excavation and slurry parameters, and subsequently determine the optimal combination of slurry control parameters that minimizes the difference between the predicted P_s and the expected P_e. According to the pressure balance control mechanism and the field engineering practise, the main parameters affecting the P_s can be divided into two categories: (1) excavation parameters: the excavation speed (v), cutterhead rotation speed (n), cutterhead torque (T), penetration (p) and the total propulsion (F); (2) slurry control parameters: the air chamber pressure (P_a), slurry inlet flow (Q_i) and outlet flow (Q_o).The framework consists of two significant components, as depicted in Figure 1.

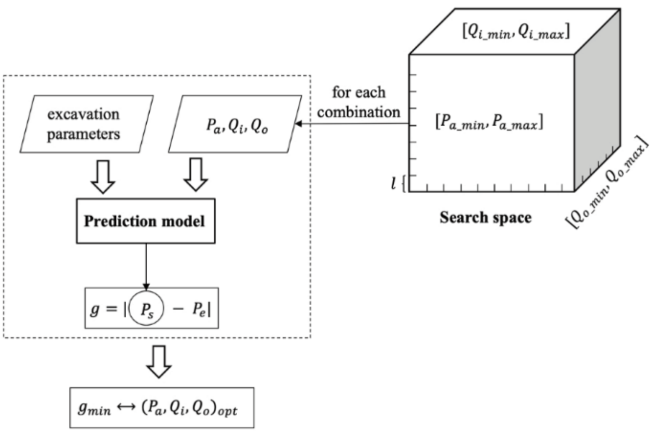


Figure 1: Data-driven model framework

The prediction model is designed to predict P_s by establishing the complicated relationship between P_s and the excavation and control parameters. The predictive performance of this model is critical and should be optimized to the best possible extent. The optimization model, on the other hand, employs a greedy search technique to explore the parameter space and identify various value combinations of the control parameters. This technique is particularly suitable for scenarios where the variation range is small and the search step is limited. The optimization model then determines the optimal values of Q_i, Q_o and P_a by selecting the combination that results in the minimum difference between the predicted P_s and the expected P_e.

2.1 Development of the prediction model

The construction process of the prediction model can be observed in Figure 2. Initially, the extracted samples are divided into training and test sets in a 9:1 ratio. Subsequently, the K-fold cross-validation technique is employed to determine the optimal combination of hyper-parameters. Finally, the models are fitted on the training set, their performance is evaluated on the test set, and the optimal model is selected. To test and compare the performance of different ML-based prediction models, three types of models are employed: (1) Base learners, including multi-linear regression (MLR), decision tree (DT) and back-propagation neural network (BPNN); (2) Bagging. We selected the random forest (RF) algorithm as the representative. (3) Boosting. AdaBoost, GBDT, XGBoost and LightGBM are utilized based on the optimal base learner.

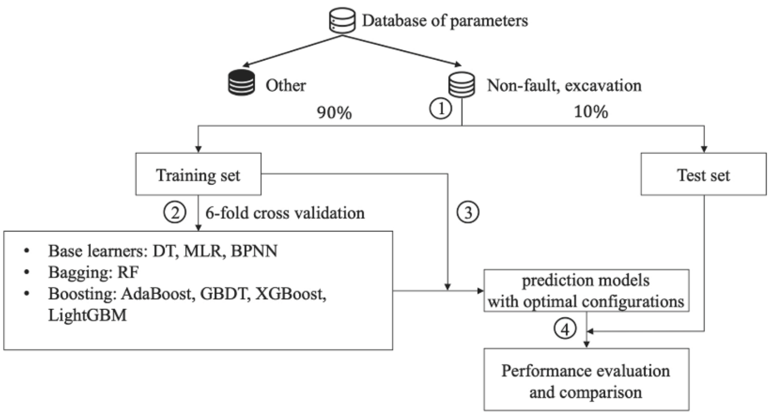


Figure 2: Construction process of prediction model

The prediction performance measurements include mean squared error (*MSE*), mean absolute error (*MAE*), mean absolute percentage error (*MAPE*) and determination coefficient R^2, as Eq. (1) - Eq. (4).

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

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}| \tag{2}$$

$$MAPE = 100\% \times \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y_i}}{y_i}\right| \tag{3}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y_i} - y_i)^2}{\sum_{i=1}^{n}(\bar{y} - y_i)^2} \tag{4}$$

Where n is the number of samples, y_i is the actual value for the ith sample, $\hat{y}_i$ is the corresponding predicted value and $\bar{y}$ is the mean value of the actual values.

2.2 Ensemble learning for regression

There are two frameworks to ensemble the base learners: the independent framework and the dependent framework (Sagi and Rokach, 2018). In the independent framework, each base learner is constructed independently of the others by manipulating the inputs, the outputs, the features, or by injecting randomness (Dietterich, 2002) (e.g., bagging.). In the dependent framework, each additional base learner will affect the weight distribution of all base learners (e.g., boosting).

Bagging. Introduced by Breiman in 1990s (Breiman, 1996), bootstrap aggregating (bagging for short) is commonly used to reduce the variance within a noisy dataset. In bagging algorithms, basic learners are trained independently, and the average of their prediction results is taken as the final result in the regression task, thus generating stability, as shown in Figure 3(a) (Sagi and Rokach, 2018).

One of the typical algorithms is random forest (RF) (Breiman, 2001), which is versatile enough to be applied to large-scale problems, and it performs excellently especially when the number of variables is much larger than the number of observations (Biau and Scornet, 2016). Based on a given training sample set $D_i = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$, an estimate M_i of function M is constructed. Assuming that there are m bootstrap samples, then m almost independent base learners $M_i(X)$ will be obtained, and the ensemble model for regression task is denoted by $M_{reg}(X) = \frac{1}{m}\sum_{i=1}^{m} M_i(X)$.

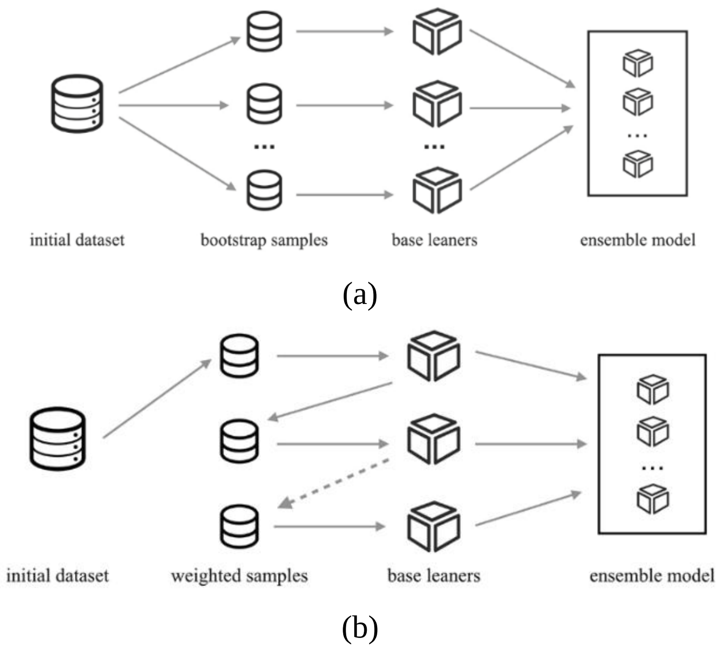


Figure 3: The process of (a) bagging and (b) boosting.

Boosting. In boosting, all the base learners are trained sequentially in a highly adaptive way as shown in Figure 3(b), in which each basic model depends on the previous models and tries to correct the predecessor (Schapire et al., 2012). Adaptive Boosting (AdaBoost) and Gradient boosting decision tree (GBDT) are two major algorithms in boosting.

AdaBoost is a machine learning meta-algorithm formulated in 1997 (Freund and Schapire, 1997). In each iteration, the weights of samples predicted incorrectly by the kth learner H_k will be increased. All the base learners will be assigned a certain weight according to the strength of the prediction ability, and usually the base learners with small errors will be assigned a larger weight. The ensemble model is denoted by $H_{reg}(X) = \sum_{k=1}^{m} e_k H_k(X)$, where e_k represents the weight of each weak estimation.

Instead of adjusting weights of samples, Gradient boosting decision tree (GBDT) focuses on the difference between the prediction and the ground truth, which tries to fit the new predictor to the residual errors L_k made by the previous predictor rather than changing the weights for each incorrect and erroneous observation at each iteration (Yang et al., 2020). First, learn a model F^1 based on the training samples. Then in the following k iterations, the base learner F_k is trained according to $L_k = \sum_{i=1}^{N}\left(Y_i - F_k(X_i)\right)^2$. The final prediction is calculated by $F_m(X) = F^1(X) + \rho \times \sum_{k=1}^{m} F_k(m)$, where ρ controls the step size to combine all the wear regression estimations.

Under the framework of GBDT, extreme Gradient Boosting (XGBoost) (Chen et al., 2015) and Light Gradient Boosting Machine (LightGBM) (Ke et al., 2017) are two of the improved algorithms. XGBoost efficiently saves the hardware resources through system optimization and algorithmic enhancements. LightGBM utilizes Gradient-based One side Sampling (GOSS) and Exclusive Feature Building (EFB) techniques to make the model more capable to deal with large amounts of data.

2.3 Optimization of slurry pressure by greedy search strategy

To search for the optimal value combination of the slurry control parameters, the search space needs to be established in advance. The three-dimensional space is composed of P_a, Q_i, and Q_o. In search of [P_{a_min}, P_{a_max}], [Q_{i_min}, Q_{i_max}], [Q_{o_min}, Q_{o_max}], $\{(P_a, Q_i, Q_o)\}$ (the search collection) is obtained. The objective function and the constraint conditions of slurry pressure optimization are shown in Eq. (5).

$$argmin\ g = argmin\ |P_s - P_e|$$

$$s.t. \begin{cases} P_a \in [P_{a\_min}, P_{a\_max}] \\ Q_i \in [Q_{i\_min}, Q_{i\_max}] \\ Q_o \in [Q_{o\_min}, Q_{o\_max}] \\ Q_i < Q_o \\ P_s = f(F, T, n, v, P_a, Q_i, Q_o) \end{cases} \tag{5}$$

Where f is the prediction model.

The greedy search algorithm is efficient to optimize the value combination of control parameters within their specified scopes. Basically, greedy search always makes the optimal choice by exploring in the whole search space. It will look at each step to ensure that the objective function is optimized globally. The process of the greedy search algorithm applied in the optimization of P_s is as shown in Figure 4. The search ranges of the control parameters P_a (Bar), Q_i (m^3/h), and Q_o (m^3/h) are [4, 4.5], [1980, 2000], [1980, 2000], respectively. The search step length l_a, l_i, and l_o are 0.01Bar, 0.1 m^3/h, 0.1m^3/h, respectively.

Input: F; T; n; p; v; P_e; l_a: search length of P_a; l_i: search length of Q_i; l_o: search length of Q_o.

Output: $P_a^{opt}; Q_i^{opt}; Q_o^{opt}; P_s^{opt}; g_{min}$: minimal difference between the predicted P_s and the expected P_e.

for P_a **in** $range(P_{a\_min}, P_{a\_max}, l_a)$ **do**

 for Q_i **in** $range(Q_{i\_min}, Q_{i\_max})$ **do**

  if $Q_i > Q_{o\_min}$ **then**

   $Q'_{o\_min} \leftarrow Q_i$

  else

   $Q'_{o\_min} \leftarrow Q_{o\_min}$

  for Q_o **in** $range(Q'_{o\_min}, Q_{o\_max}, l_o)$ **do**

   $P_s \leftarrow f(F, T, n, v, p, P_a, Q_i, Q_o)$

   $g \leftarrow abs(P_s - P_e)$

   if g_{min} **is** None **or** $g < g_{min}$ **then**

    $g_{min} \leftarrow g$, $P_s^{opt} \leftarrow P_s$, $P_a^{opt} \leftarrow P_a$, $Q_i^{opt} \leftarrow Q_i$, $Q_o^{opt} \leftarrow Q_o$

return $P_a^{opt}, Q_i^{opt}, Q_o^{opt}, P_s^{opt}, g_{min}$

Figure 4: Pseudocode of greedy search algorithm

3. Case study

3.1 Project overview and data prepocessing

The Pearl River Delta Water Resources Allocation Project passes through the core urban agglomeration of the Pearl River Delta. The SPB-TBM tunnelling section is around 84.9km long. The data used in our work were collected from the interval GS05# − GS06# with length about 3.39km. All the equipment parameters are collected automatically during the excavation process by the Programmable Logic Controller (PLC) system at an acquisition frequency of 1Hz. Obtained by the interfaces, the parameters of rings between 42 and 1313 from October 20th, 2020 are collected.

Three steps are executed sequentially in the data engineering process. To obtain more effective training data, the parameter values with the null value and the outliers detected according to the 3σ criterion are removed from the final dataset, and then we averaged the data on a minute-granularity basis. Eventually, total 82469 feature vectors were obtained for training and testing of the models. By random search (Bergstra and Bengio, 2012) with 6-fold cross validation, the optimal configuration of hyperparameters are shown Table 1.

Table 1: Configuration of hyper-parameters.

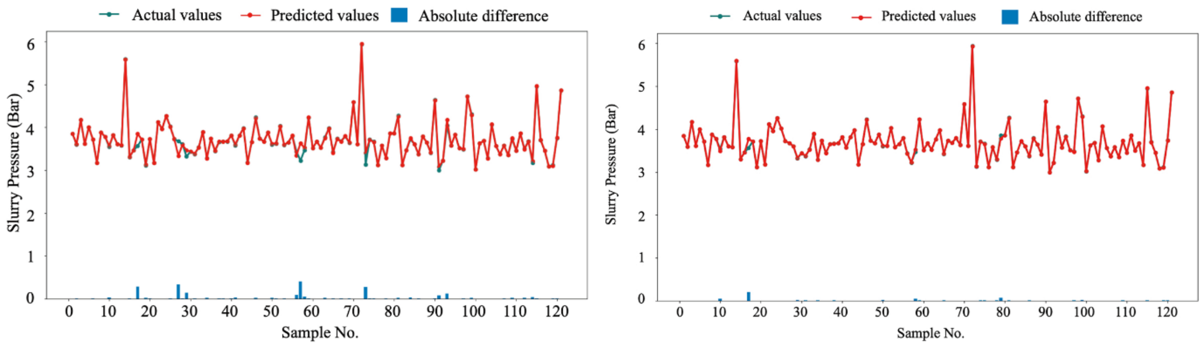
Algorithms	Hyper-parameters	Optimal Values	Algorithms	Hyper-parameters	Optimal Values
DT	Max_depth	30	RF	n_estimators	500
BPNN	Init_mode	"uniform"		max_depth	35

	Batch_size	128	GBDT	learning_rate	0.1
	Optimizer	"Adam"		n_estimators	400
	Decay_rate	0.5		max_depth	35
	Learning-rate	0.05		loss	"Is"
	Activation	"relu"	XGBoost	learning_rate	0.1
	Drop-rate	0.5		n_estimators	450
LightGBM	learning_rate	0.2		max_depth	30
	n_estimators	500	Adaboost	learning_rate	0.1
	max_depth	35		n_estimators	450
	Boosting_type	"goss"		loss	"linear"

3.2 Results and discussion

In this study, a comparison experiment is conducted on the pre-processed dataset to predict P_s using different prediction models. The statistic results were listed in Table 2. Initially, basic learners, including MLR, DT and BPNN, were constructed and compared. DT was found to perform the best among the base learners, achieving an R^2 value of up to 0.98, as shown in Figure 5(a). Thus, DT was selected for bagging and boosting algorithms.

Subsequently, bagging and boosting algorithms were constructed, and the evaluation results indicated that the AdaBoost model performed better than other models on the test set. The MSE, MAE, MAPE and R^2 were 0.002 Bar^2, 0.005 *Bar*, 0.127 % and 0.996, respectively. As show in Figure 5(b), the predicted values were almost entirely fitted with the actual values. Therefore, in the next Section, the optimal fitted AdaBoost model was selected as the prediction algorithm in the data-driven framework.



(a) (b)

Figure 5: Fitting results on the test set: (a) DT and (b) Adaboost

Table 2: The evaluation results of base learning algorithms on the test set.

Model	**MSE(Bar^2)**	**MAE(*Bar*)**	**MAPE(%)**	**R^2**
MLR	0.279	0.338	8.412	0.341
DT	0.008	0.014	0.355	0.980
BPNN	0.014	0.046	1.158	0.968
RF	0.004	0.016	0.394	0.992

AdaBoost	0.002	0.005	0.127	0.996
GBDT	0.008	0.013	0.312	0.982
XGBoost	0.004	0.021	0.526	0.990
LightGBM	0.009	0.048	1.220	0.978

Based on theoretical P_e calculations and engineering experience, the anticipated value of P_e in the full section sandstone and mudstone fraction zones is 4 *Bar*. To further evaluate the performance of the data-driven model framework, 50 samples belonging to 25 rings were randomly selected from the entire dataset. These samples were compared with the actual values, and the results are illustrated in Figure 6.

As shown in Fig. 6(a), the values of P_s optimized by the framework fluctuated slightly near the expected value (4 *Bar*), with a mean absolute gap between the optimized and the expected values of only 0.074 *Bar*. In contrast, the actual values fluctuated violently with a mean error of approximately 0.318 *Bar*. This is likely due to the fact that, as shown in Figure 6(b) to 6(d), during the actual tunnelling process, operators only set the P_a to balance P_s with P_e. Consequently, since Q_i and Q_o were not controlled, and the floating difference of P_a was 0.912 *Bar*, the actual values of P_s fluctuated violently.

The contrast results indicated the data-driven model framework is capable of ensuring that P_s is more consistent with P_e and can provide operators with more rational and reliable assistance in adjusting control parameters. Furthermore, the results suggest that the proposed framework has the superior ability to track the conditions of P_s when these critical control parameters have changed.

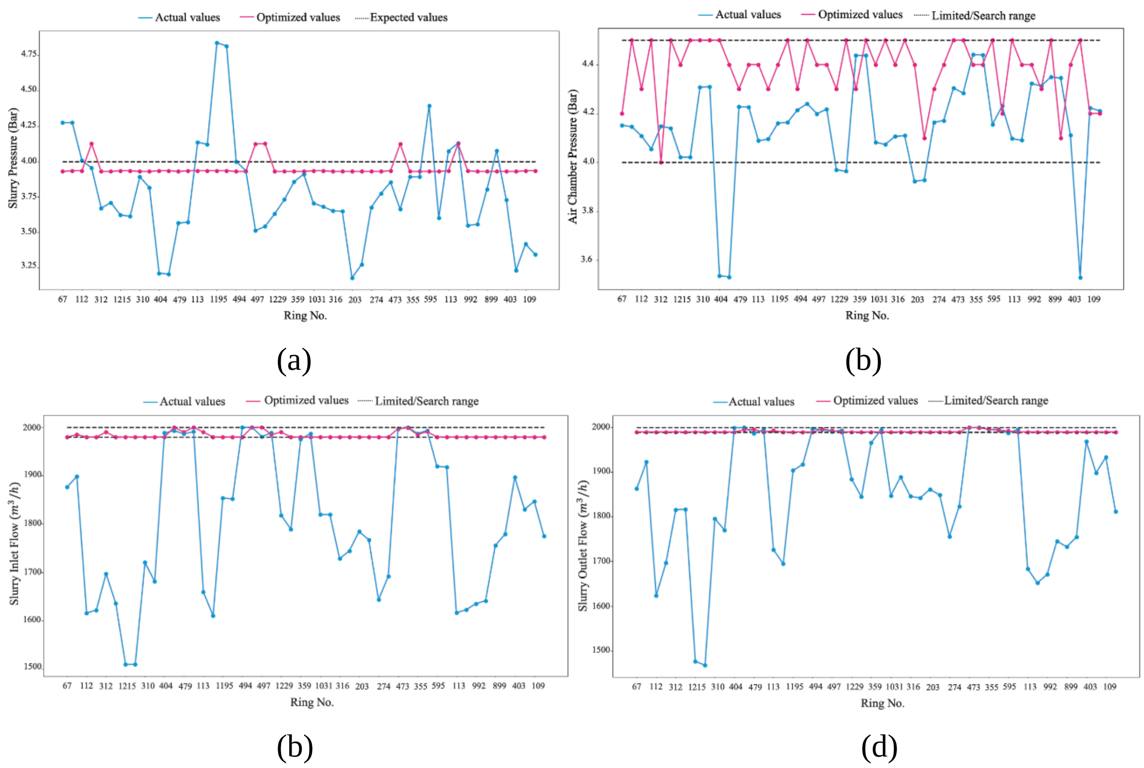


Figure 6: Optimized results: (a) P_s, (b) P_a, (c) Q_i, and (d) Q_o.

4. Conclusions

The stability of the excavation face is crucial for safety and efficiency in SPB-TBM tunnelling. To address the limitations of manual adjustment of slurry control parameters, this paper proposed a data-driven model framework to achieve the slurry and earth pressure balance. Ensemble learning algorithms were employed to forecast P_s more accurately based on the collected excavation and control parameters. A greedy search strategy was utilized to determine the optimal combination of slurry control parameters that minimized the difference between the predicted P_s and the expected P_e.

The proposed framework was empirically tested using data collected from the Pearl River Delta Water Resources Allocation Project in China. The results of the analysis indicated that AdaBoost ensemble model exhibited superior performance compared to other ensemble models. Specifically, the AdaBoost model achieved higher accuracy and fewer errors, as evidenced by the MSE, MAE, MAPE and R^2 of 0.002, 0.005, 0.127 and 0.996, respectively. Thus, the expected P_s can be accurately tracked.

Subsequently, a greedy search algorithm was employed to optimize the slurry control parameters, aiming to minimize the disparity between the predicted P_s and the expected P_e. The results of the optimization process indicated that the proposed data-driven model framework was effective, with a mean absolute gap of only 0.074 *Bar* between the optimized P_s and the expected P_e. This aspect of the analysis underscores the potential value of the proposed framework in offering valuable insights and guidance to operators involved in the tunnelling process.

In future research, the generalization issue of the proposed framework can be further improved to address the challenges posed by diverse tunnelling scenarios, geological conditions, and other relevant factors. This can be achieved by refining and expanding the framework, as well as integrating advanced machine learning techniques and optimization algorithms.

Acknowledgements

This work was supported by the Water Resources Allocation Project in Pearl River Delta (Grant No.CD88-GC022020-0038) and the National Natural Science Foundation of China (Grant No.52078304).

Reference

Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization. Journal of machine learning research 13.

Biau, G., Scornet, E., 2016. A random forest guided tour. Test 25, 197-227.

Breiman, L., 1996. Bagging predictors. Machine Learning 24, 123-140.

Breiman, L., 2001. Random forests. Machine learning 45, 5-32.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., 2015. Xgboost: extreme gradient boosting. R package version 0.4-2 1, 1-4.

Dietterich, T.G., 2002. Ensemble learning. The handbook of brain theory and neural networks 2, 110-125.

Freund, Y., Schapire, R.E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences 55, 119-139.

Gao, X., Shi, M., Song, X., Zhang, C., Zhang, H., 2019. Recurrent neural networks for real-time prediction of TBM operating parameters. Automation in Construction 98, 225-235.

Gao, X., Song, X., Shi, M., Zhang, C., Zhang, H., 2021. Real-time Forecast Models for TBM Load Parameters Based on Machine Learning Methods. arXiv preprint arXiv:2104.06353.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y., 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree, 31st Annual Conference on Neural Information Processing Systems (NIPS), Long Beach, CA.

Li, X., Zhao, S., Shen, Y., Li, G., Zhu, H., 2023. Integrated parameter optimization approach: Just-in-time (JIT) operational control strategy for TBM tunnelling. Tunnelling and Underground Space Technology 135, 105040.

Liu, Q., Wang, X., Huang, X., Yin, X., 2020. Prediction model of rock mass class using classification and regression tree integrated AdaBoost algorithm based on TBM driving data. Tunnelling and Underground Space Technology 106.

Liu, X., Shao, C., Liu, R., 2010. Modeling and simulation on earth pressure control for shield tunneling. Journal of China Coal Society 35, 575-579.

Liu, X., Zhang, K., 2019. Earth pressure balance control of shield tunneling machine based on nonlinear least squares support vector machine model predictive control. Measurement & Control 52, 3-10.

Rallu, A., Berthoz, N., Charlemagne, S., Branque, D., 2023. Vibrations induced by tunnel boring machine in urban areas: In situ measurements and methodology of analysis. Journal of Rock Mechanics and Geotechnical Engineering 15, 130-145.

Sagi, O., Rokach, L., 2018. Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8, e1249.

Schapire, R.E., Freund, Y., Schapire, R.E., Freund, Y., 2012. Boosting Foundations and Algorithms Preface.

Yang, J., Zhao, C., Yu, H., Chen, H., 2020. Use GBDT to predict the stock market. Procedia Computer Science 174, 161-171.

Zhang, W.G., Li, H.R., Wu, C.Z., Li, Y.Q., Liu, Z.Q., Liu, H.L., 2021. Soft computing approach for prediction of surface settlement induced by earth pressure balance shield tunneling. Underground Space 6, 353-363.

Zhou, C., Ding, L.Y., He, R., 2013. PSO-based Elman neural network model for predictive control of air chamber pressure in slurry shield tunneling under Yangtze River. Automation in Construction 36, 208-217.

Zhou, J., Qiu, Y., Zhu, S., Armaghani, D.J., Khandelwal, M., Mohamad, E.T., 2021. Estimation of the TBM advance rate under hard rock conditions using XGBoost and Bayesian optimization. Underground Space 6, 506-515.

Zhou, Z.-H., Zhou, Z.-H., 2021. Ensemble learning. Springer.