sustainability

Article

Improved Machine Learning Model for Urban Tunnel Settlement Prediction Using Sparse Data

Gang Yu [1,2], Yucong Jin [1,2,*], Min Hu [1,2], Zhisheng Li [3], Rongbin Cai [3], Ruochen Zeng [1] and Vijiayan Sugumaran [4]

[1] SILC Business School, Shanghai University, Shanghai 201800, China; gyu@shu.edu.cn (G.Y.); minahu@shu.edu.cn (M.H.); rczeng@shu.edu.cn (R.Z.)
[2] SHU-SUCG Research Centre for Building Industrialization, Shanghai University, Shanghai 200072, China
[3] Shanghai Urban Construction City Operation (Group) Co., Ltd., Shanghai 200023, China; lizs@suog.cn (Z.L.); cairb@suog.cn (R.C.)
[4] School of Business Administration, Oakland University, Rochester, MI 48309, USA; sugumara@oakland.edu
* Correspondence: jyc_kx@shu.edu.cn

Abstract: Prediction tunnel settlement in shield tunnels during the operation period has gained increasing significance within the realm of maintenance strategy formulation. The sparse settlement data during this period present a formidable challenge for predictive Artificial Intelligence (AI) models, as they may not handle non-stationary relationships effectively or have the risk of overfitting. In this study, we propose an improved machine learning (ML) model based on sparse settlement data. We enhance training data via time series clustering, use time decomposition to uncover latent features, and employ Extreme Gradient Boosting (XGBoost) v1.5.1 with Bayesian Optimization (BO) v1.2.0 for precise predictions. Comparative experiments conducted on different acquisition points substantiate our model's efficacy, the in-training set yielding a Mean Absolute Error (*MAE*) of 0.649 mm, Root Mean Square Error (*RMSE*) of 0.873 mm, Mean Absolute Percentage Error (*MAPE*) of 3.566, and Coefficient of Determination (R^2) of 0.872, and the in-testing set yielding a *MAE* of 0.717 mm, *RMSE* of 1.048 mm, *MAPE* of 4.080, and R^2 of 0.846. The empirical results show the superiority of the proposed model compared to simple ML models and a complex neural network model, as it has a lower prediction error and higher accuracy across different sparse settlement datasets. Moreover, this paper underlines that accurate settlement predictions contribute to achieving some Sustainable Development Goals (SDGs). Specifically, preventive tunnel maintenance strategies based on predictive results can enhance tunnels' long-term operational reliability, which is in accordance with SDG 9 (Industry, Innovation, and Infrastructure) and SDG 11 (Sustainable Cities and Communities).

Keywords: operation period; settlement prediction; sparse data; machine learning

check for updates

Citation: Yu, G.; Jin, Y.; Hu, M.; Li, Z.; Cai, R.; Zeng, R.; Sugumaran, V. Improved Machine Learning Model for Urban Tunnel Settlement Prediction Using Sparse Data. *Sustainability* **2024**, *16*, 4693. https://doi.org/10.3390/su16114693

Academic Editors: Davide Settembre-Blundo and Minghua Huang

Received: 25 March 2024
Revised: 19 May 2024
Accepted: 29 May 2024
Published: 31 May 2024

1. Introduction

The structural settlement changes of urban tunnels may seriously affect the structure, endangering the safety, stability, and lifespan of the tunnel [1–3]. Accurately predicting settlement is vital for monitoring changes and implementing preventive maintenance, making it a critical concern in academic and industrial domains [4–6]. A preventive maintenance strategy can be made based on estimating future tunnel structural settlement. It identifies areas needing attention and interventions to prevent more severe structural damage. This approach significantly improves the long-term operational performance of the tunnel and ensures its long-term safety and reliability [7]. Thus, it develops a sustainable and reliable transportation infrastructure for regional economic development aligned with SDG 9 (Industry, Innovation, and Infrastructure). Furthermore, this maintenance strategy also aligns with the target outlined in SDG 11 (Sustainable Cities and Communities), which emphasizes the importance of offering safe, affordable, accessible, and sustainable transport systems for all.

Machine Learning (ML) exhibits remarkable capacity in capturing complex non-linear relationships among multiple variables, introducing a new avenue for tunnel settlement prediction [8]. Presently, ML-based research focuses on the construction period, employing prediction models like Support Vector Regression (SVR) [9], Random Forest (RF) [10], General Regression Neural Network (GRNN) [11], and Back Propagation Neural Network (BPNN) [12]. Notably, Shi et al. [13] utilized Support Vector Machine (SVM) to precisely predict the arch crown settlement of a shallow buried tunnel; Zhang et al. [14] employed Extreme Gradient Boosting (XGBoost) to estimate excavation-induced settlement considering parameters such as excavation speed, soil pressure, and water content; Moghaddasi and Noorian-Bidgoli [15] developed a hybrid model of Artificial Neural Network optimized by an Imperialist Competitive Algorithm (ICA-ANN) to forecast maximum surface settlement for minimizing the impact of subway tunnel excavation on the urban area above. To address the challenge of effectively capturing time-dependent characteristics [16], Deep Learning (DL) models such as Long Short-Term (LSTM) [17] and Wavenet [18] have been applied to improve prediction accuracy. Ge et al. [19] proposed a Deep Belief Network optimized by a Whale Optimization Algorithm (WO-DBN) to predict shield-induced settlement, while Zhang et al. [20] combined kinetic correlation analysis with Conv1d to introduce an expanding DL method for real-time ground settlement prediction.

Unlike the construction period, where high-frequency settlement data are directly acquired using real-time sensors, the operation period mainly relies on manually placed acquisition points to collect the tunnel surface heights. Due to the monitoring frequency and methods limitations, the collected data exhibit sparse characteristics of short length and univariate form. Therefore, the input data for training the prediction model only contain sparse univariate time-series data. Direct application of sophisticated AI models can result in inadequate parameter training or overfitting. Moreover, tunnel settlement during the operation period is influenced by multiple factors [21], emphasizing the importance of feature extraction for prediction accuracy. Currently, most prediction models solely utilize the original temporal features of the univariate data without fully exploring the underlying influencing factors, thereby leading to subpar accuracy and generalization. In summary, tunnel settlement prediction during the operation period surpasses conventional time series prediction in complexity.

To overcome these challenges, this paper proposes an improved ML model based on sparse datasets to predict tunnel settlement during the operation period. The model's effectiveness is validated using real datasets from a cross-river tunnel in Shanghai. This study has the following contributions.

1. To address ineffective parameter training caused by sparse settlement data, this paper utilizes a K-Means cluster model based on Dynamic Time Wrapping (DTW) to divide data from different acquisition points into distinct groups. It augments training samples and enhances parameter training efficiency;
2. To address the limitations of exclusively exploring temporal features in settlement data, this paper applies the Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) model to decompose the univariate settlement data into multi-dimensional data. This approach fully explores the hidden influencing factors and improves feature mining capability;
3. To achieve precise tunnel settlement prediction across different locations, this study adopts an effective XGBoost with Bayesian optimization (BO)-informed parameter selection. This predictive model learns decomposed features for each group, dynamically optimizing parameter combinations to improve predictive performance, ensuring precise and stable predictions.

The subsequent sections are structured as follows: Section 2 provides an overview of the existing research in settlement prediction. Section 3 introduces the project background, Section 4 elaborates the proposed methodology, and Section 5 presents the experimental results, while Section 6 draws conclusions.

2. Related Works

In general, tunnel settlement prediction models can be classified into model-based methods and AI-based methods. Model-based methods, including empirical solutions and numerical simulations, use professional physical conditions to build rigorous mathematical models. Additionally, empirical methods utilize professional knowledge to construct appropriate mathematical models for fitting observed data. Vorster et al. [22] developed a conservative estimation method for ground settlement by considering physical variables such as tunnel geometry, pipe stiffness, and soil properties. Fang et al. [23] established an empirical formula based on the normal distribution function to estimate underground settlement caused by shield tunneling. Lu et al. [24] designed a tunnel settlement formula based on Gaussian functions using measurements from multiple field sites. However, empirical methods may yield unstable predictions due to variations in environmental conditions. Meanwhile, theoretical approaches are usually established based on certain physical assumptions and constraints to describe, explain, and predict settlements. They mainly use computers to simulate and visualize parameter variations in different scenarios [25]. Klotoé and Bourgeois [26] simulated the impact of umbrella arches on tunnel settlement using the CESAR-LCPC 3D finite element mode. Lai et al. [27] used finite element difference methods to simulate the influence of underpass tunnel construction on settlement. Li et al. [28] conducted sophisticated simulations using FLAC3D to analyze bridge piles and tunnel lining deformations caused by shield tunnel excavation. Numerical simulation methods demand substantial computational resources for adjustments in response to environmental changes, hindering their ability to offer timely dynamic predictions. While model-based methods offer explicit explanations for tunnel settlement, they are not extensively used due to constraints in valid mathematical assumptions and available multiple influencing variables.

AI-based methods explore potential correlations among input data to generate predictive results. Ling et al. [29] employed RF to predict settlement in shield tunneling through complex geological formations and determine the relative importance of each input variable. Additionally, some researchers have compared the performance of multiple ML models on the same single dataset. Mahmoodzadeh et al. [30] compared the tunnel settlement prediction performance of SVR, RF, Gradient Boosting Machine (GBM), XGBoost, and Light Gradient Boosting Machine (LightGBM). Tang and Na [10] evaluated SVM, RF, BPNN, and Deep Neural Network (DNN), determining that SVM struck the best balance between training time and accuracy. Researchers have also integrated parameter optimization mechanisms like BO [8], genetic algorithm optimization (GA) [31], and particle swarm optimization [32] to improve training effectiveness. Previous studies have demonstrated good settlement prediction accuracy with ML models, but their simple structures may struggle to uncover complex non-linear and non-stationary relations in time-series data. Consequently, they have poor prediction performance and limited applicability. By contrast, DL models can adaptively adjust parameters, mine hidden features [33], and enhance prediction accuracy. Cao et al. [34] proposed a Recurrent Neural Network-Gappy Proper Orthogonal Decomposition (RNN-GPOD) model for tunnel surface settlement prediction, aiding accurate tunnel boring machine operation. Wu et al. [35] showed that LSTM outperforms traditional ML methods in settlement prediction tasks. Zhu et al. [36] integrated K-Means and LSTM to predict tunnel settlement under different degradation patterns, ultimately assessing tunnel structure performance. While complex network structures excel in capturing intricate input data relationships, prediction model effectiveness hinges on data quality. Sparse input datasets may result in overfitting or diminished accuracy.

The model-driven approach, based on geomechanics, mathematical assumptions, and other theoretical knowledge, can be utilized to calculate the settlement evolution of tunnels under different circumstances. However, these methods simplified the external tunnel environment to meet corresponding mathematical assumptions. Nevertheless, as operational time increases, the surrounding environment undergoes continuous changes, which have a particular impact on the tunnel structure. Therefore, this approach cannot accurately predict the dynamic changes in structural settlement. The effectiveness of data-driven prediction methods has been validated in various engineering projects during the tunnel's construction. These highly accurate AI models require abundant data samples to support model training and parameter updates. However, during the operation period, manually collected settlement data exhibit sparse characteristics of low frequency, short time series, and single variables. If these data are directly utilized for training AI models, problems like ineffective updates of model parameters or model overfitting may arise, which can result in decreased prediction accuracy. Therefore, current approaches fail to provide precise and stable settlement predictions during the operational period. Consequently, further research is needed to predict sparse settlement data during the operational period.

3. Project Overview

This study validates the proposed model using data from a cross-river tunnel in Shanghai. The tunnel spans 8950 m, with 7500 m crossing the Yangtze River Nan Harbor water area, characterized by a diameter of 15 m. As the longest cross-river tunnel in Shanghai, its structure has undergone spatial–temporal variations due to complex hydrological, geological, and human factors in its vicinity.

According to the structural characteristics of the tunnel, it is divided into five sections. A total of 227 locations within the tunnel are equipped with hydrostatic level gauges, and height is monitored semiannually. These sections and acquisition points are depicted in Figure 1. The approach sections (Sections 1 and 5) connect the tunnel with surface roads, while the buried section (Sections 2 and 4) links the approach and shield sections (Section 3). The center region of the cross-river area represents the shield-driven section. This study uses datasets from June 2011 to December 2021 for training and validation, removing the abnormal data due to device damage, resulting in a dataset dimension 203 × 22. Figure 2 illustrates accumulated settlement variations between 2011 and 2021, showing similar patterns in some areas but significant differences in others. Sections 1–3 exhibit an overall rise in settlement over time, while Sections 4 and 5 experience degradation. Notably, the area from S217 to S225 has significant long-term cumulative settlement, warranting focused maintenance.

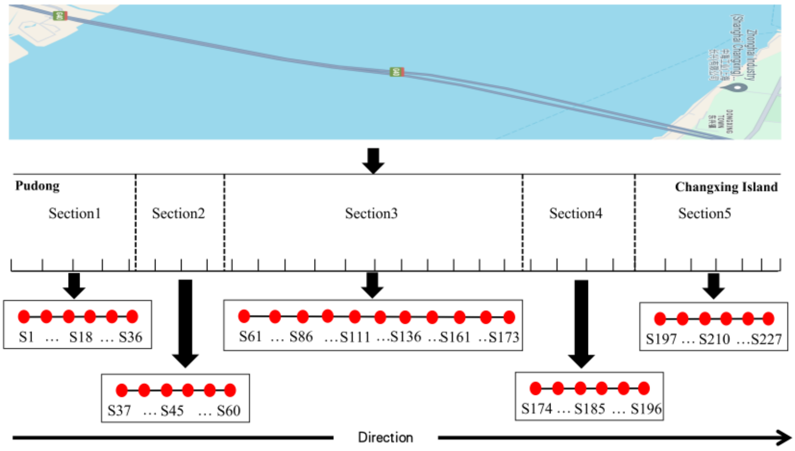


Figure 1. Distribution of key sections and acquisition points in the tunnel.

TSI, proposed by Li et al. [37], thinks settlement is a significant factor influencing the overall condition of shield tunnels. Unlike accumulated settlement, settlement can reveal timely changes in tunnel structure. As a result, this study focuses on predicting settlement to assist maintenance personnel in formulating preventative maintenance strategies and improving the operational performance of tunnels. The original monitored height is transformed into settlement data according to Equation (1). S_{jt} is the settlement of acquisition point j at time t, H_{jt} is the height of acquisition point j at t.

$$S_{jt} = (H_{jt} - H_{jt-1}) \times 1000 \quad (1)$$

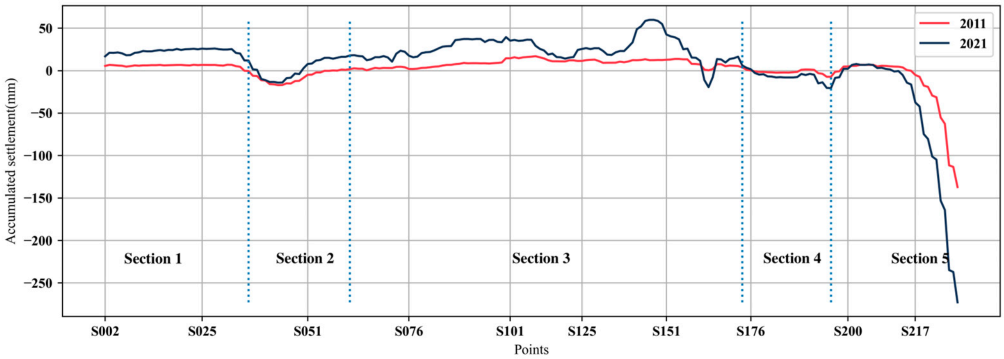


Figure 2. Comparison of accumulated settlement in different locations.

4. Methodology

4.1. Improved ML Model for Tunnel Settlement Prediction

This paper proposes an improved ML model shown in Figure 3, comprising three parts: data augmentation by DTW-K-Means, multi-feature mining by CEEMDAN, and settlement prediction by an ML-based model.

1. Augment the sparse settlement dataset. The DTW-K-Means aggregates acquisition points with similar settlement patterns in different clusters. By learning all time-series data in the same cluster, the number of training sets for the prediction model is directly increased;
2. Extract potential multi-feature information from the univariate settlement dataset. The CEEMDAN decomposes univariate settlement data into multi-dimensional data, revealing hidden influencing factors by capturing underlying non-linear and non-stationary relationships. The multi-dimensional data consist of multiple Intrinsic Mode Functions (IMFs) and residual items (Res) obtained from the decomposition process;
3. Predict the future settlement of each acquisition point. A rolling window approach generates the training set from the same IMF series, followed by the predictive model based on ML. Ultimately, the model predicts future trends for each IMF and synthesizes the predicted settlement of each acquisition point.

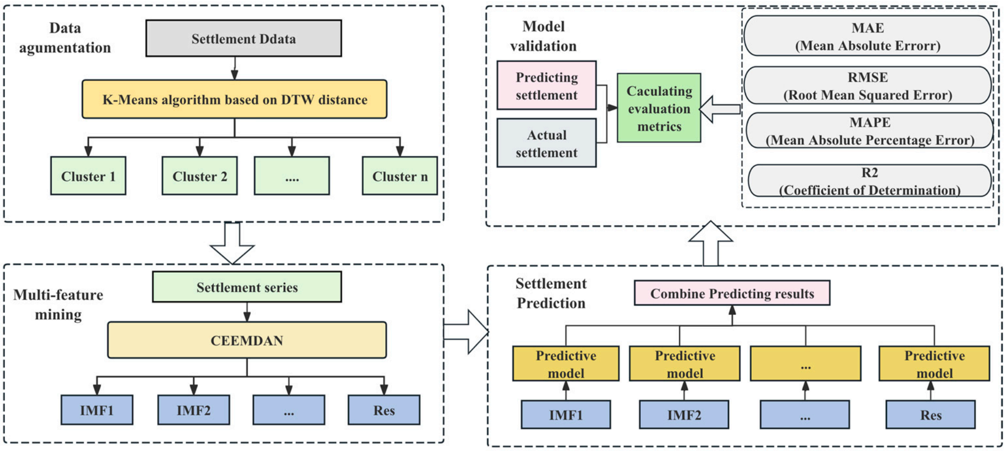


Figure 3. Flowchart of the proposed improved ML model.

In summary, the proposed model first utilizes DTW-K-Means to aggregate data samples with similar settlement patterns into a new shared training dataset, thereby directly increasing the number of training samples. Next, CEEMDAN is employed to decompose the original settlement series of each sample point, revealing information at different time frequencies. After these two processes, the training datasets consist of settlement data from all acquisition points in the same cluster, with features including multidimensional decomposed variables. Finally, different BO-ML models are constructed to learn the different decomposed features and make predictions. The prediction of each feature is then summed to form the final settlement prediction. Finally, the improved ML model is validated by the actual data based on four evaluation metrics.

4.2. Data Augmentation: DTW-K-Means

Revealing the similarity between series from the overall set of time series to form distinct groups can serve as a means of data sample augmentation [38]. Due to the temporal nature of settlement data, this study applies a K-Means model based on DTW to cluster settlement datasets. Compared to traditional Euclidean distance, DTW can effectively calculate the distance between sequences of different lengths and reflects the maximum similarity [36]. Therefore, adopting this method can improve the accuracy of time series clustering. Suppose we have a dataset consisting of m time series sharing the same length n and there are two time series $X = \{x_1, x_2, \dots x_n\}$ and $Y = \{y_1, y_2, \dots, y_n\}$. Firstly, a similarity matrix $D_{m \times n}$ is constructed where each element is the Euclidean distance between sequences defined by $d(i,j) = d(x_i, y_j) = \sqrt{(x_i - y_j)^2} \in D_{m \times n}$. Starting from the first elements of both sequences, the algorithm determines the cumulative minimum cost path through the matrix, considering possible warping paths that allow local shifts in the time axis. DTW finds the optimal alignment with the minimum cumulative cost by iteratively calculating the costs and updating the path. Subsequently, the Elbow method and Sum of Squared Distances (SSD) are used to determine the optimal number of clusters [39]. K-Means [40] as an effective and commonly used model is suitable for clustering small to medium-sized data samples. In this study, the DTW algorithm determines the similarity between acquisition points during the K-Means clustering process. The clustering process can be divided into the following steps.

1. Determination of optimal cluster numbers using the Elbow method to pre-train the settlement datasets to get the optimal number of clusters k;
2. Initialization of cluster centers. A set of samples is selected as the initial centers u_i;
3. Initialization of clustering. Each sample is assigned to the nearest cluster based on the DTW distance to the cluster center u_i;
4. Determination of the validity of clustering results by calculating the minimal SSD for the cluster to assess the completion of clustering. If the condition is not met, the above steps are repeated. Once the desired result is achieved, the final clustering results are outputted.

Time clustering can group settlement data based on their trends, overcoming constraints like geographical locations and environmental factors. Data in the same cluster show higher similarity in settlement patterns, while patterns differ significantly between clusters. Sections in similar geographical environments experience consistent external influencing factors, leading to consistent long-term settlement patterns and trends within the same area. Consequently, they are grouped into the same cluster. Subsequently, the settlement data from the same cluster are aggregated to form a uniform training dataset, enriching sparse historical settlement data and expanding training samples. This method helps reduce the possibility of overfitting. Suppose we have a set $X = \{x_1, \dots, x_m\}$ and m time series with the same length n, $x_i = \{y_{i,1}, \dots, y_{i,n}\}$. Taking x_1 as an example, before utilizing the augmentation method, the training set can be defined as $D(x_1) = \{y_{1,1}, \dots, y_{1,n}\}$. After adopting DTW-K-Means, suppose the original dataset

X is divided into $C_1 = \{x_1, x_3, \ldots, x_{m-1}\}$ and $C_2 = \{x_2, x_4, \ldots, x_m\}$. As a result, the training dataset for x_1 is augmented as follows.

$$D(x_1) = D(x_3) = \ldots = D(x_{m-1}) = \left\{ \begin{matrix} y_{1,1} & \cdots & y_{1,n} \\ y_{3,1} & \cdots & y_{3,n} \\ \cdots & \cdots & \cdots \\ y_{m-1,1} & \cdots & y_{m-1,n} \end{matrix} \right\} \tag{2}$$

4.3. Multi-Feature Mining: CEEMDAN

Time series decomposition reveals inherent non-linear and non-stationary information in the data [41]. Empirical Mode Decomposition (EMD) [42] decomposes signals based on the inherent time scale characteristics of the data, yielding a finite number of IMFs. Each IMF is the different characteristics of the original signal at different time scales. It represents various factors that lead to tunnel settlement. To ensure that each decomposed mode is not influenced by noise, thereby enhancing the accuracy and robustness of the decomposition, this study utilizes CEEMDAN [43] to perform time decomposition of the settlement data. This approach effectively eliminates mode mixing caused by EMD and improves the model's ability to capture nonlinear and non-stationary features in settlement. The decomposition process is as follows.

1. White noise $v^{\lambda}(t)$ which follows a normal distribution is added to the original time series $x(t)$ in iterations for $\lambda(\lambda = 1, 2, \ldots, I)$. The reconstructed series can be defined as $x^{\lambda}(t) = x(t) + v^{\lambda}(t)$. Performing the decomposition step of EMD on $x^{\lambda}(t)$, the first IMF IMF_1 and residual $m_1(t)$ are given as follows.

$$IMF_n = \frac{1}{I} \times \sum_{\lambda=n-1}^{1} IMF_n^{\lambda} \tag{3}$$

$$m_n(t) = m_{n-1}(t) - IMF_n \tag{4}$$

2. Then add $v^{\lambda}(t)$ to $m_1(t)$, and further decompose by EMD to calculate the IMF_2 and $m_2(t)$. After decomposing in the $n-1$ time, add $v^{\lambda}(t)$ to $m_{n-1}(t)$, and decompose $m_{n-1}^{\lambda}(t)$ using EMD in the n time. The IMF_n and $m_n(t)$ can be computed as follows.
3. After the *J-th* decomposition, $m_n(t)(n = J)$ cannot be further decomposed. Finally, the original series can be decomposed as follows, where $m_J(t)$ is the final residual.

$$x(t) = \sum_{n=1}^{J} IMF_n + m_J(t) \tag{5}$$

4.4. Settlement Prediction

The prediction method needs to account for both linear and non-linear relationships within the settlement data, which makes traditional linear regression and simpler models unsuitable. Meanwhile, due to restrictions imposed by early monitoring equipment, this study only uses settlement data from various acquisition points within the tunnel during its operational period, resulting in a sparse training dataset. However, using sparse settlement data to train complex neural network models may lead to overfitting. Studies have proved XGBoost's superiority over other algorithms on small-scale datasets, demonstrating its effectiveness in solving non-linear regression problems while mitigating overfitting [44]. Based on the previous literature review, SVR and RF have also been confirmed as effective, commonly used settlement prediction models. Thus, this study selects these three ML models as candidate models to learn and predict the settlement. Additionally, BO continuously learns from previous optimizations to identify the best-optimized parameter configuration, requiring fewer samples to derive the optimal values [45]. Therefore, BO is adopted to address the challenge of complex hyperparameter selection in ML model training when dealing with sparse data.

4.4.1. ML Models

1. Extreme gradient boosting

XGBoost [46] selects decision trees as base learners, aiming to reduce the error between predicted values and targets by adding new base learners. The final predicted output value is obtained as the summation of predictions from all the base learners, as shown in Equation (6).

$$Y_i = \sum_{m=1}^{M} f_m(x_i), f_m \in F \tag{6}$$

where f represents decision tree, F represents all functions of used decision trees, and M represents the total number of trees. In the regression process, the first regression tree is fixed with the learned data features from the first iteration. The new regression trees are added to compensate for the errors and improve accuracy. The error generated by the preceding t ensemble models will serve as a reference for building the $t + 1$ tree. The new tree can rectify model errors and continue adding trees until further improvement is not possible. Ultimately, the objective function is smaller than the desired threshold.

In XGBoost, the objective function is defined as Equation (7). The first item $l(y_i, \hat{y}_i)$ is the loss function, which is the difference between the true value y_i and predicted value $\hat{y}_i$. The second item $\Omega(f_m)$ represents the regularization penalty item, which is used to control model complexity. Regularization terms can smooth the final learning weights, thereby preventing overfitting.

$$obj(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{m=1}^{M} \Omega(f_m) \tag{7}$$

The regularization term $\Omega(f_m)$ for the decision tree is defined as Equation (8). λ controls the penalty, and T represents the number of leaves in the decision trees. γ represents the complexity of each leaf and is the vector of scores on the leaves. Next, the second-order Taylor expansion in general gradient boosting is employed in the loss function of XGBoost. Then, the objective function can be defined as Equation (9). g_i and h_i represent the first and second derivatives of the loss function, and q is a function that assigns data to the corresponding leaf. Because each data sample only belonged to one leaf node, the sum of the loss values of each leaf node can also be used to describe the loss function. Thus, the objective function can also be defined as Equation (10). Accordingly, G_i and H_i can be defined as Equations (11) and (12). I_i represents all the data samples in the leaf node j.

$$\Omega(f_m) = \gamma T + 0.5\lambda \sum_{j=1}^{T} \omega_j^2 \tag{8}$$

$$J^{(t)} \approx \textstyle\sum_{i=1}^{n} \left[g_i \omega_{q(x_i)} + \dfrac{1}{2}(h_i \omega_{q(x_i)}^2) \right] + \gamma T + \dfrac{1}{2}\lambda \textstyle\sum_{j=1}^{T} \omega_j^2 \tag{9}$$

$$J^{(t)} \approx \gamma T + \textstyle\sum_{j=1}^{T} \left[(\textstyle\sum_{i \in i_j} g_i)\omega_j \right] + \textstyle\sum_{j=1}^{T} \left[+ \dfrac{1}{2}(\textstyle\sum_{i \in i_j} h_i + \lambda)_i \omega_j^2 \right] \tag{10}$$

$$G_j = \textstyle\sum_{i \in i_j} g_i \tag{11}$$

$$H_j = \textstyle\sum_{i \in i_j} h_i \tag{12}$$

2. Support vector regression

SVR [47] aims to make all the sample points approximate the regression hyperplane while minimizing the total deviation between the sample points and the hyperplane. Suppose there is a n-dimensional training set $\{(x_1, y_1), \ldots, (x_n, y_n)\} \subset R^d \times R$, where R^d is the space of input features for x_i. Additionally, let $f(x) = \omega \cdot \phi(x) + b$ denote the regression function mapping from the output space to the input space, where the ω and b are weight vector and bias, and $\phi(x)$ represents a function mapping from input values to high-dimensional space. The objective of SVR is to find $f(x)$ such that the model bias is less than or equal to a given error threshold ε. This problem can be solved by minimizing the objective function, as in Equation (13).

$$\begin{aligned} &\min[\tfrac{1}{2}\|\omega\|^2 + C\cdot\sum_{i=1}^{n}(\zeta_i+\zeta^*)] \\ &s.t.\begin{cases} y_i - f(x_i) \le \varepsilon + \zeta_i, \zeta_i \ge 0 \\ y_i - f(x_i) \ge \varepsilon + \zeta_i^*, \zeta_i^* \ge 0 \end{cases} \end{aligned} \tag{13}$$

In Equation (13), C is a regularization parameter, which controls the model's complexity; ε is the Vapnik's intensive loss defining the margin of acceptable error around the predicted value; ζ_i and ζ_i^* are slack variables measuring the training errors calculated by the ε-loss function. Next, the Equation (13) can be rewritten as Equation (14), where $K(x_i, x_j)$ is the kernel function which transforms the data point from the low-dimensional to high-dimensional space; α_i and α_i^* are the Lagrange multipliers; n_{sv} represents the number of support vectors.

$$\begin{aligned} &f(x) = \sum_{i=1}^{n_{sv}}(\alpha_i - \alpha_i^*)K(x_i, x_j) + b \\ &s.t.\begin{cases} 0 \le \alpha_i \le C \\ 0 \le \alpha_i^* \le C \end{cases} \end{aligned} \tag{14}$$

3. Random forest

RF regression is an ensemble method consisting of multiple decision trees [48]. Initially, bootstrap samples are randomly selected from the original training dataset with replacement. Next, each decision tree is evolved by using the bootstrap sample. Firstly, a randomly selected subset of input features from the bootstrap sample is chosen. Then, the best splitting method within this subset is determined at each node split in the decision tree. The splitting process continues until further subdivision no longer reduces the Gini index. The tree grows to the maximum size and remains unpruned. This process is repeated until the defined number of trees is reached. Finally, the ensemble algorithm generates multiple outputs corresponding to each tree. The final prediction result is obtained by averaging the prediction results of multiple decision trees, as shown in Equation (15), where y represents the prediction results, K is the number of decision trees, and $f_i(x)$ is the prediction result of the i-th decision tree for an input vector x.

$$y = \frac{1}{K}\sum_{i=1}^{K} f_i(x) \tag{15}$$

4.4.2. Bayesian Optimization

In ML-based models, multiple hyperparameters control the learning behavior, and their combinations significantly impact the final training performance of the model. Unlike classical optimization scenarios with well-defined mathematical objective functions, hyperparameter selection is computationally demanding and lacks a clear objective function. BO surpasses conventional approaches such as grid search in effectiveness. BO is a probabilistic distribution-based global optimization algorithm [49]. It predicts the next sampled hyperparameter combinations based on the previously experimented distribution. Applying BO to optimize hyperparameters can significantly enhance tuning efficiency, especially in settlement prediction with limited sample points.

Supposing a function $x^* = \mathrm{argmin}x_{\chi \in R^d} f(x)$ presents the relationship between the hyperparameters and objective loss function, where x^* represents optimal parameter combination, χ represents decision space, and $f(x)$ is the objective loss function. Because settlement prediction is a regression question, the objective loss function is Root Mean Square Error (*RMSE*), which can be defined as $f(x_j) = \sqrt{\sum_{i=1}^{N}\left(\widehat{y}_i(x_j) - y_i\right)^2/N}$, where $\widehat{y}_i(x_j)$ is the model results by using the hyperparameter combination x_j.

During the optimization, firstly, the dataset of hyperparameters $D = (X, y)$ is established, where $X = \{x_1, x_2, \ldots, x_i\}$ and $y = \{f(x_1), f(x_2), \ldots, f(x_i)\}$. Next, Gaussian Process Regression (GPR) composed of a mean function and a covariance function is trained to learn data and update the posterior distribution of the objective function. In other words, BO constructs a probabilistic regression model M for the objective function $f : X \to R^d$ and calculates the probability distribution function $P(f(x)|x, D)$. Then, the next observation is calculated by an Acquisition Function (AC) $\alpha(x_t)$. The AC uses the current $P(f(x)|x, D)$ as a cheap surrogate for the loss function and adopts the Probability of Improvement (PI) to guide the selection of the next evaluation point, which aims to find the optimal solution for the hyperparameters that minimizes the objective loss function. Through each iteration, the dataset of hyperparameters D is updated according to the parameters and losses from the previous stage.

4.4.3. Prediction Based on BO-ML Model

This paper employs the BO-ML model for settlement prediction, as shown in Figure 4. Because the time series split requires the testing dataset to be later than the training dataset, each settlement data is divided initially, with the first 80% of the data used for training and the remaining 20% for testing. Additionally, similar settlement pattern data are grouped using the mentioned data augmentation method to create a new training set. Each data sample in this set contains the same number of decomposed features. Next, separate ML-based predictive models are constructed to learn and predict each component within the same cluster. Finally, according to the decomposition principle of CEEMDAN, the settlement prediction can be calculated by Equation (16), where the $\hat{C}_i(t)$ is the prediction result of each IMF, $\hat{R}_M(t)$ is the prediction result of Res, and $\hat{S}(t)$ is the final prediction of settlement.

$$\hat{S}(t) = \sum_{i=1}^{M} \hat{C}_i(t) + \hat{R}_M(t) \tag{16}$$

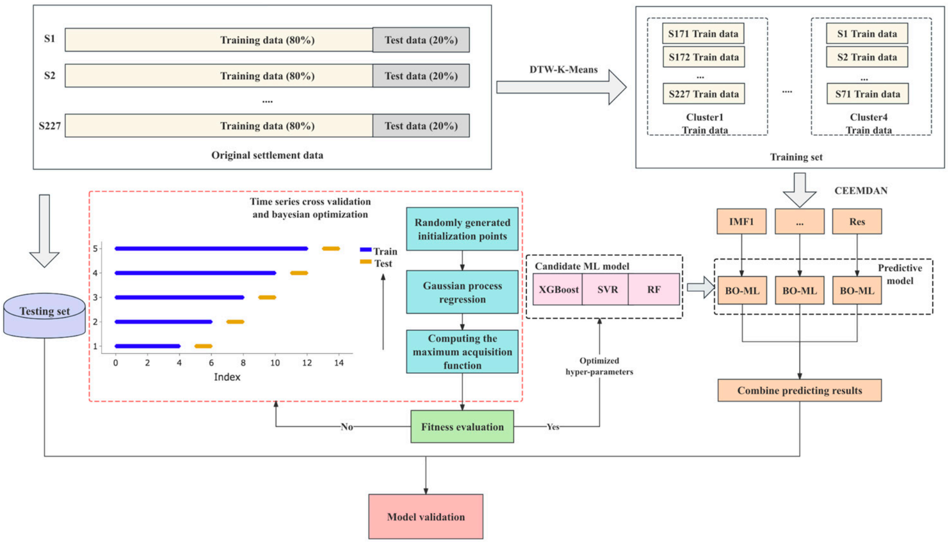


Figure 4. Framework of the BO-ML model.

When constructing the predictive model, this paper selects three efficient ML models as candidate models: XGBoost, SVR, and RF. To enhance the models' accuracy, BO is used to optimize parameter selection. Table 1 shows each ML model's explanation and search scope of hyperparameters. Meanwhile, cross-validation is used to prevent over-fitting. Rather than using the common K-fold cross-validation [50], this study adopts Time Series Cross-Validation (TSCV). Specifically, the TSCV ensures that in each split, the indices of testing set for every temporal settlement sequence remain later than those of the training set (based on sklearn.TimeSeriesSplit v1.0.2). During the *k-th* iteration, the first k folds are designated as the training set, while the $(k+1)$-th fold serves as the testing set. After iterations, the K number of models' $RMSE$ is calculated using the testing set. This study sets the mean results of the 5-fold TSCV as the objective function to guide the parameter updates using BO. In the optimization process, the model hyperparameters and the range of hyperparameters are first initialized to generate random initialization points. Then, the training set and the initialized parameters are input variables for the GPR. The mean TSCV results of the ML model under each parameter set are employed as the objective function, and the parameters are adjusted to improve the GPR. Subsequently, based on AC, the next parameter set, which is most likely to enhance the objective function, is selected to be further evaluated. The new sample points are then incorporated into the model for training, and the GPR is updated to predict the objective function values more accurately for unsampled points in the parameter space. When the preset number of iterations is reached, the optimization iteration is stopped. Subsequently, the BO-ML model with the best performance in original settlement data prediction will be used as the basic predictive model to train on the datasets processed by DTW-K-Means and CEEMDAN. Finally, based on predefined evaluation metrics, the improved ML model is evaluated using the testing set.

Table 1. Explanation and search scope of hyperparameters for each ML model.

Model	Hyperparameter	Hyperparameters Explanation	Search Scope
XGBoost	colsample_bytree	Subsample ratio of features used for fitting a tree	[0.01, 1]
	gamma	Minimum loss reduction required for further partitioning	[0.001, 10]
	learning_rate	Step size shrinkage used in model update	[0.01, 0.3]
	max_depth	Maximum depth of a tree	[1, 10]
	min_child_weight	Minimum weights of the instances required in a leaf	[0, 20]
	n_estimators	Number of trees to be used in the boosted ensemble.	[1, 1000]
	reg_alpha	L1 regularization term on weights	[0.001, 1]
	reg_lambda	L2 regularization term on weights	[0.001, 1]
	subsample	Subsample ratio of the training instances	[0.001, 1]
SVR	C	Penalty parameter of the error term	[0.1, 10]
	epsilon	Specifies the tolerance margin where no penalty is given to errors	[0.001, 1]
	gamma	Coefficient for Gaussian Radial Basis Function (RBF), polynomial, and sigmoid kernels	[0.01, 0.1]
RF	max_depth	The maximum depth of the tree	[3, 10]
	max_features	The number of features to consider when searching the best split	[0.1, 0.999]
RF	min_samples_leaf	The minimum number of samples required to be at a leaf node	[1, 30]
	min_samples_split	The minimum number of samples required to split an internal node	[2, 30]
	n_estimators	The number of trees in the forest	[10, 300]

5. Case Study

5.1. Evaluation Metrics

In this paper, four evaluation metrics were used: Mean Absolute Error (MAE), $RMSE$, Mean Absolute Percentage Error ($MAPE$), and Coefficient of Determination (R^2). The formulas for these metrics are shown in Equations (17)–(20), where n represents the total number of samples; y_i represents the actual measured value of settlement; $\hat{y}_i$ represents the predicted value; $\overline{y}$ is the average of the measured values. MAE, $RMSE$, and $MAPE$ measure the prediction error: the smaller the value is, the better the prediction is. R^2 measures the goodness of fit for the prediction model: the larger the value is, the better the performance is.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{17}$$

$$RMSE = \sqrt{\frac{\sum\limits_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{18}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \tag{19}$$

$$R^2 = 1 - \frac{\sum\limits_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum\limits_{i=1}^{n}(y_i - \overline{y}_i)^2} \tag{20}$$

5.2. Results of Augmentation for Settlement Data

XGBoost is selected as the basic predictive model based on the comparative results of candidate models. Thus, all subsequent analyses of the proposed model are based on the improved BO-XGBoost. This section employs DTW-K-Means (based on tslearn v0.6.2) to classify settlement data into clusters, identifying similar settlement patterns for unified training. First, we use the Elbow criteria to determine the appropriate value of k. Figure 5 demonstrates the SSD curve exhibits significant change at the values 3 to 6. Therefore, the DTW-K-means models with $k = 3$ to $k = 6$ are tested to determine the optimal number. Additionally, when k is larger, the similarity between data increases, but the number of samples clustered together decreases, reducing data augmentation's benefit. Therefore, when training the clustering model, it is necessary to consider both the impacts on training samples and data similarity. In this study, different values of k within the range of 3 to 6 are used to pre-train the proposed model, and the corresponding evaluation metrics are calculated. As shown in Figure 6, at $k = 4$, the *MAE*, *RMSE* and *MAPE* are the lowest, and R^2 is highest. Therefore, it is thought that at $k = 4$, the model simultaneously considers both sample similarity and the quantity of training samples, leading to optimal predictive results. Consequently, $k = 4$ is selected for further discussions.

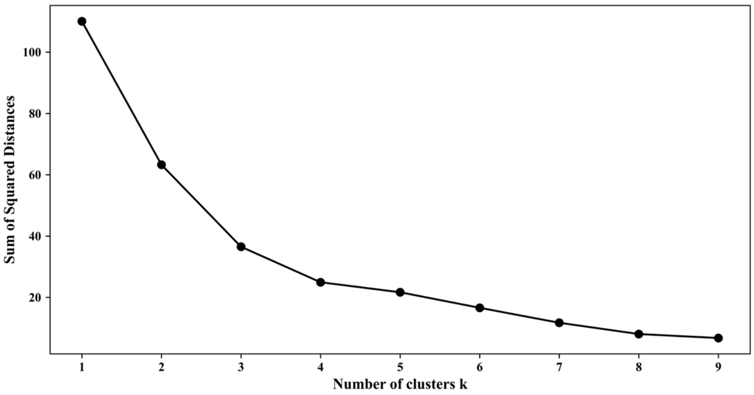
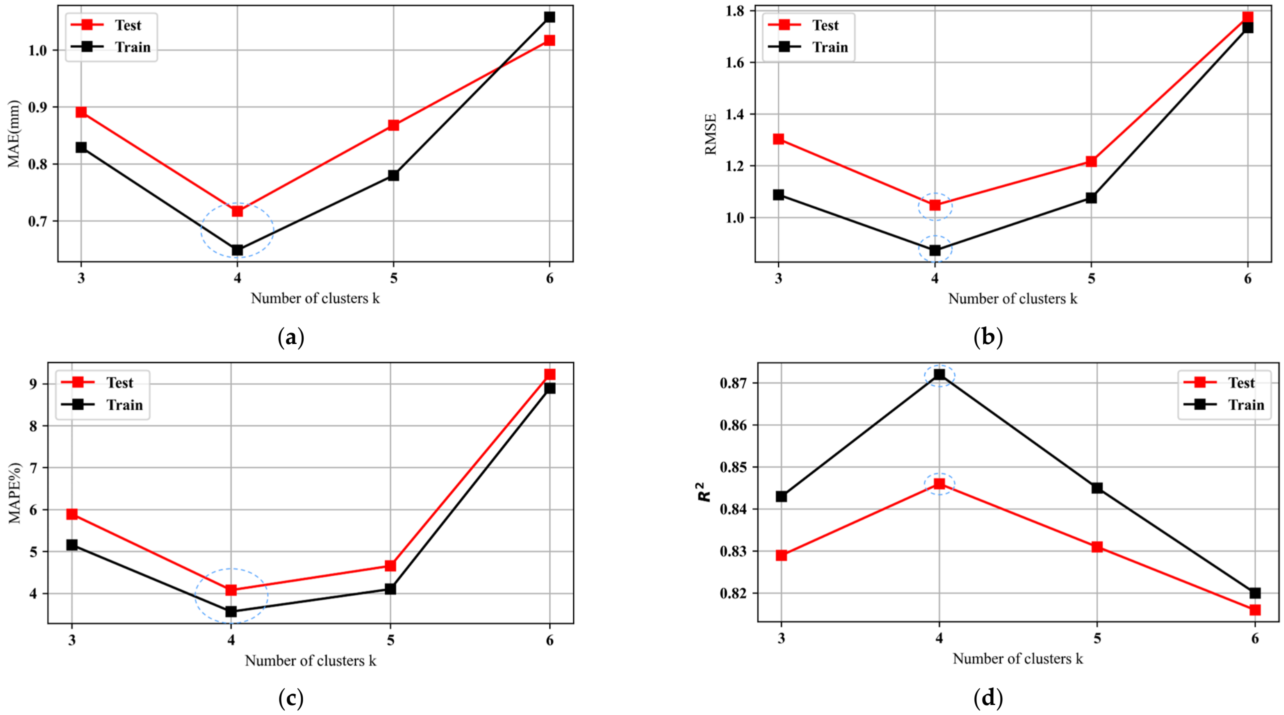


Figure 5. SSD curve of DTW-K-Means.

Figure 7 illustrates the clustering outcomes and Figure 8 shows settlement series of four clusters, showcasing spatial proximity among acquisition points within the same cluster despite lacking explicit spatial features in the dataset. These points exhibit similar settlement patterns due to their similar geological environments and structures, verifying the effectiveness of the clustering model. Cluster 1 primarily encompasses acquisition points in Sections 4 and 5, closer to the island susceptible to ground activities like adjacent construction, resulting in more severe settlements. Conversely, Cluster 2 includes points in Section 3, influenced by factors like tides and water pressure, potentially exhibiting upward trends due to periodic water level rises. Clusters 3–4, situated in more stable areas with fewer nearby constructions, display distinct settlement trends. Notably, data from disparate sections may be grouped together based on temporal trends rather than spatial proximity alone. For example, some acquisition points' geographical locations in Sections 1 and 3 are not adjacent. From a temporal trend perspective, most acquisition points in Section 3 show a rising trend. However, some acquisition points have a shorter duration of this trend, lasting only one monitoring period before degrading. These acquisition points' settlement patterns align with those in Section 1. Thus, they are clustered together. If only spatial information is considered, the similarity between these data points would be overlooked.



(**a**) (**b**)

(**c**) (**d**)

Figure 6. Evaluation metrics of proposed model with different number of k in DTW-K-Means: (**a**) *MAE*; (**b**) *RMSE*; (**c**) *MAPE*; (**d**) R^2.

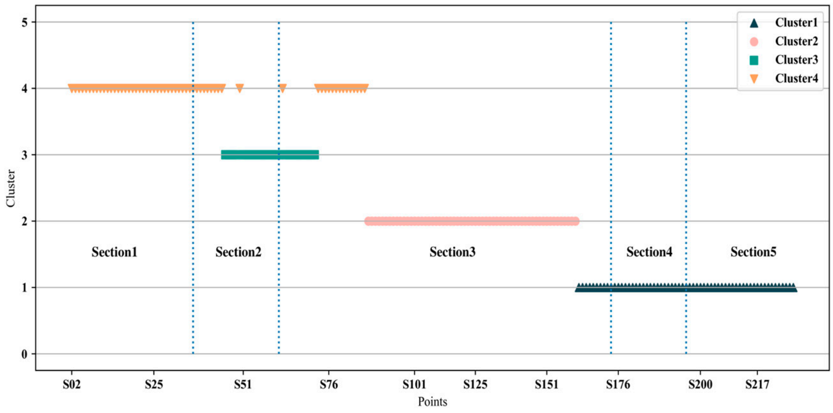
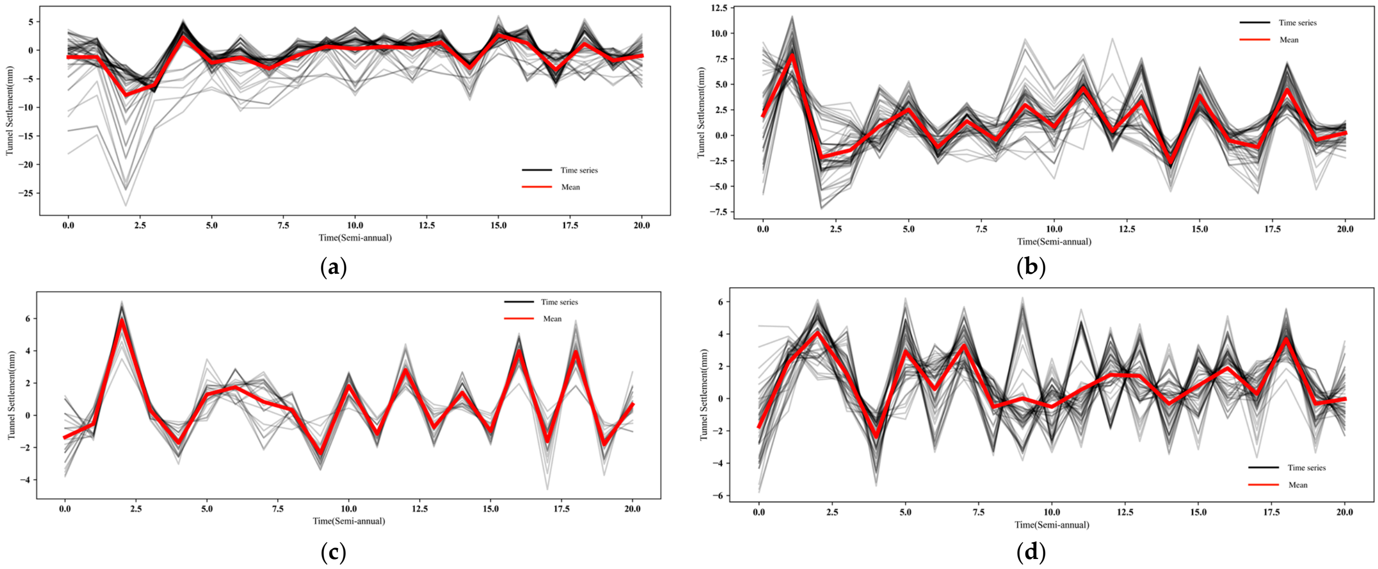


Figure 7. Clustering results of settlement data at different sections.

Figure 9 displays the mean settlement data for the four clusters, while Table 2 summarizes the clustering results for different sections and the percentage of acquisition points in each cluster. Cluster 1 initially experienced degradation in the first half of 2012, followed by a gradual rise with minor fluctuations, showing an upward trend by 2021. This cluster shows lower settlement values, indicating severe settlement in the corresponding areas. It accounts for 30.05% of acquisition points, so demanding particular attention, especially in Sections 4 and 5, which reflect more severe settlement due to complex traffic and geographical factors. Cluster 2 showed gradual degradation with periodic fluctuations and an upward trend in 2021. This cluster is characterized by prominent peaks and the broadest range of settlement variations. Cluster 3 started with a rise in the first half of 2012, subsequently experiencing intermediate degradation, and a downward trend in 2021. Lastly, Cluster 4 primarily showed an upward trend in 2011, followed by variable changes and a slight increase in 2021, suggesting stable yet significant settlement patterns in the corresponding area.



(**a**)

(**b**)

(**c**)

(**d**)

Figure 8. Settlement data of four clusters: (**a**) Cluster 1; (**b**) Cluster 2; (**c**) Cluster 3; (**d**) Cluster 4.

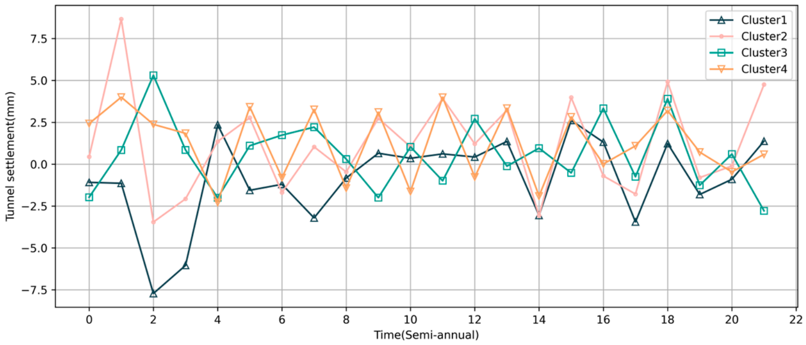


Figure 9. Mean settlement data of four clusters.

Table 2. Clustering results of acquisition points in each cluster.

Cluster	Section 1	Section 2	Section 3	Section 4	Section 5	Total	Percent
Cluster 1	0	0	11	21	29	61	30.05%
Cluster 2	0	0	47	0	0	47	23.15%
Cluster 3	27	23	12	0	0	62	30.54%
Cluster 4	8	0	25	0	0	33	16.26%
Total	35	23	95	21	29	203	100%

Time series clustering identifies similar settlement patterns in tunnel structures, consolidating data within clusters for unified training. The training data increase from the historical data of individual acquisition points to all acquisition points' historical data with similar settlement patterns, thereby increasing the number of training data. Table 3 presents the augmentation results of one acquisition point randomly selected from each cluster.

Table 3. Augmentation results of training data.

Acquisition Point	Cluster	Number of Original Train Data	Number of Augmented Trained Data
S007	Cluster 4	14	854
S012	Cluster 3	14	658
S135	Cluster 2	14	868
S217	Cluster 1	14	462

5.3. Results of Feature Mining for Settlement Data

In this section, CEEMDAN (based on PyEMD v1.5.1) decomposes the original data to extract additional features. Initially, an unconstrained CEEMDAN model is used to pre-train the data for each acquisition point to ensure sufficient feature extraction. Results indicate a minimum of two unrestricted decomposed IMFs. So, the parameter "max_imf" is set to 2, which ensures that the clustered data can be used together to train the same prediction model. Meanwhile, it means the original time series data are decomposed into two IMFs and a Res, resulting in a dataset that includes three different time-scale settlement features.

S103, a critical location in the tunnel structure, is selected for further analysis. Table 4 and Figure 10 illustrate the decomposition results, with each IMF arranged from high to low frequency. The horizontal axis represents data collection points every six months from June 2011 to December 2021. The average of IMF1 and IMF2 is approximately zero, with their respective curves fluctuating around zero. Conversely, the Res's average significantly deviates from zero, exhibiting a lower frequency of fluctuations. Moreover, the Pearson correlation coefficient (Pearson) between IMF1 and original settlement is 0.889, indicating the highest similarity in trend. IMF1 effectively removes data noise caused by monitoring instrument malfunctions, recording errors, and other factors, offering a more accurate representation of settlement variations and contributing to higher predictive accuracy. In contrast, IMF2 represents the low-frequency component, revealing the long-term periodic fluctuations in tunnel settlement influenced by external environmental factors. The average fluctuation period is five years. Notably, at t = 3 (December 2012), the IMF2 curve shows a degrading trend, likely associated with the structural impact of unauthorized tunnel loading detected above the area in 2012. These curve fluctuations affirm the decomposition algorithm's ability to uncover underlying information. Additionally, Res reflects the overall trend of long-term structural changes in the tunnel. The curve illustrates a historical downward trend, yet more recently indicates an upward trend.

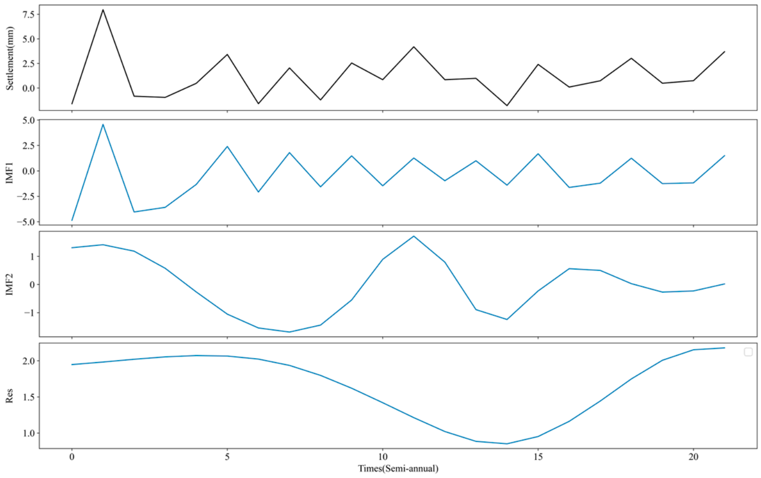


Figure 10. Decomposition results of the settlement data for S103.

Table 4. Statistical information of decomposition results for S103.

IMFs	Settlement	IMF1	IMF2	Res
Max	7.970	4.573	1.718	2.179
Min	−1.790	−4.855	−1.687	0.850
Average	1.209	−0.435	−0.017	1.662
Person	1.000	0.889	0.269	0.069

Through CEEMDAN, the original settlement sequence is decomposed into denoised settlement feature (IMF1), periodic variation feature (IMF2), and long-term trend feature (Res). By learning features at different time scales, the model can better capture the nonlinear trend changes caused by unknown external factors.

5.4. Analysis of Settlement Prediction

5.4.1. Parameter Selection

1. Number of input sequence

The number of historical inputs, representing the time window, significantly affects the learning performance of the model. Longer inputs facilitate capturing temporal trends but may reduce the training sample size; shorter inputs may decrease learning effectiveness. To determine the optimal input length, this study keeps the proposed model structure consistent and only adjusts the input data length for comparative experiments. The experimental results are shown in Figure 11; when the input number is 4, the model performs best on both the training and testing set. Therefore, this study set the input time step to 4 for further discussions.

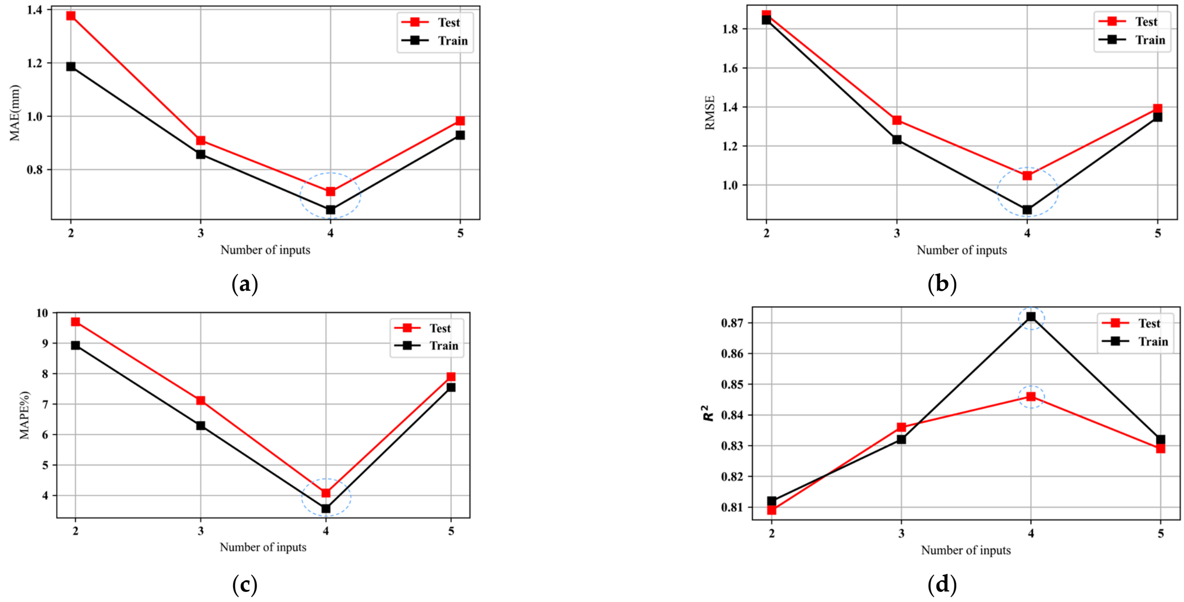


Figure 11. Evaluation metrics of proposed model with different number of inputs: (**a**) *MAE*; (**b**) *RMSE*; (**c**) *MAPE*; (**d**) R^2.

2. Hyperparameter selection for predictive model

To assess the performance of BO, this study builds proposed models with GA based on deap v1.4.1, grid search algorithm (GS) based on scikit-learn v1.0.2, and BO based on bayesian-optimization v1.2.0 for parameter optimization, separately, and compares them with an unoptimized model. The experimental results are summarized in Table 5. All optimized models show superior predictive capabilities compared to the unoptimized model, thus highlighting the importance of parameter tuning. Moreover, in contrast to the limited improvements with GS, both GA and BO exhibit higher prediction performance.

Additionally, BO shows the lowest prediction errors and highest prediction accuracy, verifying its efficiency in parameter selection when dealing with small sample datasets.

Table 5. Evaluation metrics of models using different parameter optimization methods.

Model	*MAE* (mm)		*RMSE* (mm)		*MAPE*%		R^2	
	Train	Test	Train	Test	Train	Test	Train	Test
Model with GA	0.690	0.754	0.984	1.052	3.768	4.184	0.860	0.843
Model with GS	0.704	0.804	1.032	1.154	4.059	4.669	0.851	0.836
Unoptimized model	0.712	0.818	1.041	1.208	4.077	4.672	0.847	0.834
Model with BO	0.649	0.717	0.873	1.048	3.566	4.080	0.872	0.846

Consequently, in this section, the paper constructs BO to optimize the hyperparameters of ML models. The objective function is defined as the minimization of the RMSE from 5-fold TSCV. The optimal parameter values for the 12 models are shown in Table 6.

Table 6. Optimal hyperparameters for each cluster.

Hyperparameters	Cluster1			Cluster2		
	IMF1	IMF2	Res	IMF1	IMF2	Res
colsample_bytree	0.778	0.917	0.001	0.765	1.000	1.000
gamma	3.967	2.023	0.285	0.487	0.001	0.001
learning_rate	0.065	0.251	4.319	0.111	0.300	0.300
max_depth	9.313	7.716	0.697	9.573	5.925	3.576
min_child_weight	7.836	17.608	701.325	10.115	13.540	18.331
n_estimators	186.159	617.115	0.001	266.392	478.287	160.659
reg_alpha	0.358	0.721	0.445	0.584	1.000	1.000
reg_lambda	0.917	0.265	1.000	0.660	0.001	1.000
subsample	0.609	0.752	0.001	0.739	1.000	1.000
Hyperparameters	**Cluster3**			**Cluster4**		
	IMF1	**IMF2**	**Res**	**IMF1**	**IMF2**	**Res**
colsample_bytree	0.843	1.000	1.000	0.789	0.822	1.000
gamma	3.017	0.001	0.001	0.879	0.031	0.001
learning_rate	0.099	0.300	0.010	0.077	0.113	0.010
max_depth	5.449	3.357	6.173	6.952	8.009	10.000
min_child_weight	5.411	7.157	0.000	18.287	19.636	6.343
n_estimators	379.028	173.910	680.769	167.286	609.162	680.025
reg_alpha	0.179	0.001	0.001	0.554	0.331	0.001
reg_lambda	0.245	0.001	0.001	0.569	0.215	1.000
subsample	0.847	1.000	1.000	0.990	0.813	1.000

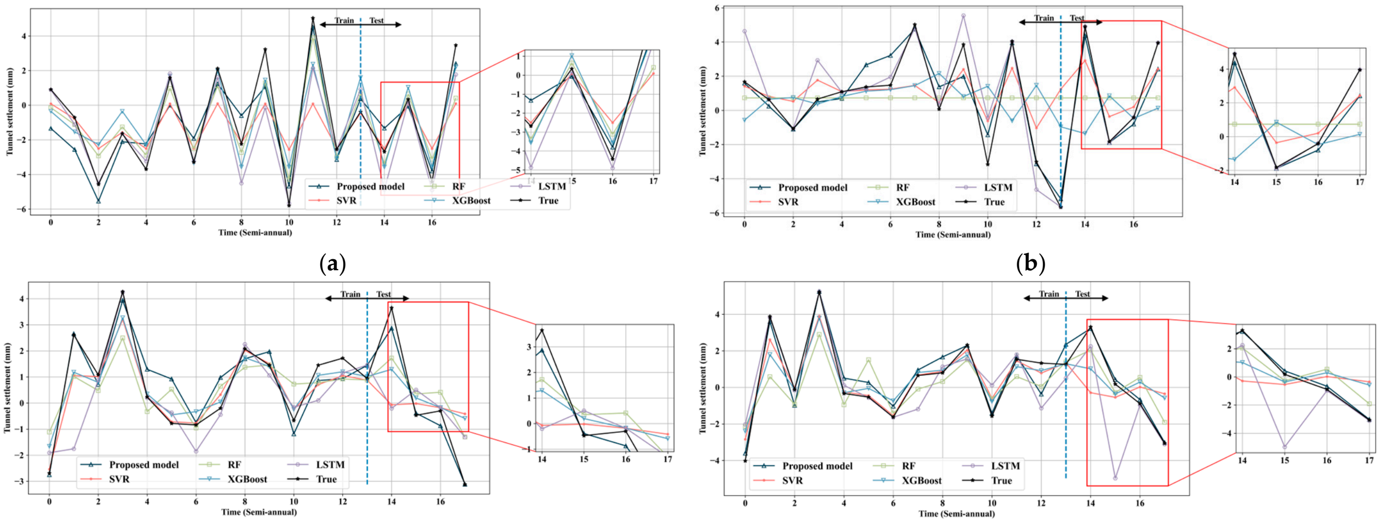
5.4.2. Model Evaluation

This study utilizes Python (v3.7) to construct predictive models. The proposed improved ML model is based on XGBoost (v1.5.1), with comparative models including single XGBoost, SVR (based on scikit-learn v1.0.2), RF (based on scikit-learn v1.0.2), and LSTM (based on keras v2.3.1). Additionally, BO is employed to optimize parameter selection for all ML models based on the method described in Section 4.4.3. As for LSTM, GS based on 5-fold TSCV is used to adjust learning_rate and dropout to prevent overfitting. Table 7 outlines the final parameter values for the comparison models, using S217 as an example. All compared models use four historical settlement data from time $t-4$ to $t-1$ as inputs, and the output is the settlement value at time t. Meanwhile, a separate compared model is built for each acquisition point's settlement data. The proposed model builds different BO-XGBoost, which respectively learn and predict historical IMFs series and Res to ultimately synthesize the predicted settlement.

Table 7. Parameters of comparison model for S217.

Model	Parameters
XGBoost	colsample_bytree = 0.63; gamma = 4.12; learning_rate = 0.176; max_depth = 5.943; min_child_weight = 1.793; n_estimators = 268.058; reg_alpha = 0.022; reg_lambda = 0.209; subsample = 0.485
SVR	kernel = “rbf”; C = 1.297; epsilon = 0.075; gamma = 0.952
RF	max_depth = 6; max_features = 0.1; min_samples_leaf = 25; min_samples_split = 2; n_estimators = 82
LSTM	units = 50; dropout = 0.7; activation = “Adam”; epochs = 100; batch_size = 1; learning_rate = 0.001

Among the 203 acquisition points, this study randomly selects S007, S012, S135, and S217 from each cluster for comparison. Using an input size of four and an output size of one for the settlement datasets, four-fifth of the total recorded length is designated as the training set (samples 0–13), with the remaining one-fifth used for testing (samples 14–17). Figure 12 depicts the predictive performance of the improved ML model and comparison models on the same dataset. While all models fit well during steady settlement changes (Figure 12a), the improved ML model consistently outperforms others in capturing unstable movement (Figure 12b–d). When the early fluctuations of the time series are gentle but become violent in the later stages (Figure 12b), all single ML models fail to promptly capture the changes in settlement, resulting in significant errors. As the curve exhibits long-period fluctuating changes (Figure 12c,d), the fitting performance of single ML models improves, leading to reduced errors. Compared to the other two ML models, the predictive accuracy of XGBoost is higher. However, LSTM demonstrates poor robustness, resulting in significant deviations in the testing set. Notably, among the four datasets, the predicted values generated by the proposed model closely match the actual settlement curves on both the training and testing set.



(**a**)

(**b**)

(**c**)

(**d**)

Figure 12. Prediction results of different points: (**a**) S217 (belonging to Cluster 1); (**b**) S135 (belonging to Cluster 2); (**c**) S012 (belonging to Cluster 3); (**d**) S007 (belonging to Cluster 4).

To assess predictive performance and generalization, this study utilized settlement data from 203 acquisition points in the tunnel and computed evaluation metrics, as shown in Table 8. Firstly, among the single ML models, RF exhibits the highest prediction errors and lowest R^2 on the training and testing sets. This model cannot effectively capture the temporal trends of settlement effectively. Meanwhile, XGBoost's predictive performance slightly outperforms SVR and RF, verifying the efficiency of this model in handling sparse datasets. Thus, this model is chosen as the basic predictive model for further improvement. Although LSTM demonstrates better fitting performance on the training set, its performance on the testing set is poor, indicating distinct overfitting. In contrast, the proposed improved model based on BO-XGBoost exhibits the lowest *MAE*, *RMSE*, *MAPE*, and highest R^2 on both the training and testing set, indicating its superior predictive performance. Furthermore, compared to XGBoost, the proposed model displays a 29% reduction in *MAE*, a 38% reduction in *RMSE*, a 48% reduction in *MAPE*, and a 3% increase in R^2 on the testing set. Therefore, the proposed improved ML model can effectively enhance the predictive performance of traditional ML models in dealing with sparse tunnel settlement during the operation period.

Table 8. Average evaluation metrics of different models.

Model	*MAE* (mm)		*RMSE* (mm)		*MAPE*%		R^2	
	Train	Test	Train	Test	Train	Test	Train	Test
RF	1.651	1.666	2.125	2.133	9.740	10.397	0.781	0.742
SVR	1.319	1.333	1.904	1.917	8.975	9.211	0.804	0.798
XGBoost	1.065	1.010	1.637	1.710	7.657	7.973	0.824	0.821
LSTM	0.858	2.429	1.287	2.794	6.248	17.623	0.830	0.693
Proposed model	0.649	0.717	0.873	1.048	3.566	4.080	0.872	0.846

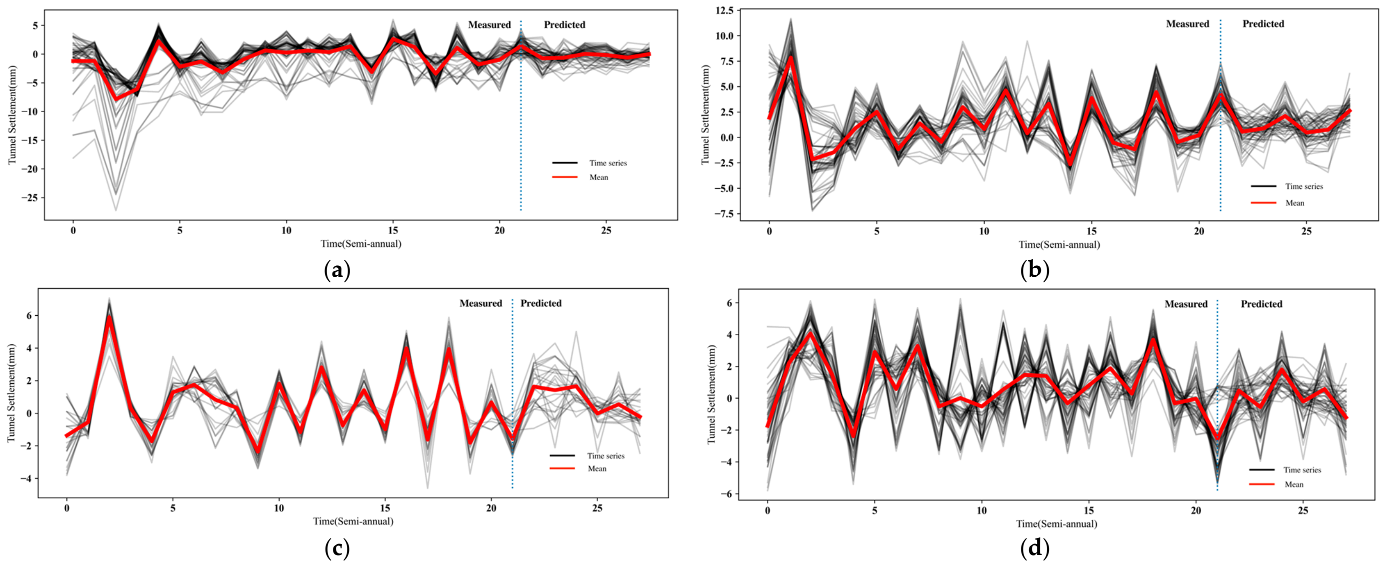
Three models are constructed for comparative analysis to explore the contributions of different parts to the proposed model. The No-clustered model does not utilize DTW-K-Means for data augmentation; the No-decomposed model does not employ CEEMDAN to explore hidden temporal features within univariate data; the No-BO model indicates the absence of BO for parameter optimization in XGBoost. Experimental results are presented in Table 9, showing that removing any single part will decrease model accuracy. This indicates that the proposed model can comprehensively improve model performance. The No-clustered model exhibits the poorest predictive performance, suggesting that data augmentation directly increases the number of samples for model training and significantly contributes to model improvement. Next, the No-decomposed model failed to explore potential trend features, resulting in lower efficiency in feature extraction and increased prediction errors. The No-Bo model shows that using BO for parameter optimization can effectively enhance model performance, although its contribution is the lowest. In summary, each part contributes to improving ML prediction performance. However, there is a discrepancy in the predictive performance of the proposed model between the training and testing set, indicating significant overfitting. This model only considers historical temporal trends while neglecting recent changes in the external environment. Consequently, its performance on the testing set is inferior, indicating a relatively lower generalizability of the model, which requires further improvement.

Table 9. Average evaluation metrics of models without different parts.

Model	*MAE* (mm)		*RMSE* (mm)		*MAPE*%		R^2	
	Train	Test	Train	Test	Train	Test	Train	Test
No-decomposed model	0.769	0.956	1.056	1.367	4.248	6.256	0.842	0.827
No-clustered model	0.835	1.004	1.216	1.459	4.762	6.767	0.833	0.825
No-BO model	0.712	0.818	1.041	1.208	4.077	4.672	0.847	0.834
Proposed model	0.649	0.717	0.873	1.048	3.566	4.080	0.872	0.846

5.4.3. Tunnel Settlement Prediction

To estimate settlement trend of the tunnel structure, this study applied the proposed model to predict the next 3-year settlement comprising six data points. The results are presented in Figure 13. The predicted settlement data generally coincide with the historical patterns of each cluster. Specifically, acquisition points belonging to Clusters 1 and 2 exhibit a stable fluctuating movement, while those in Clusters 3 and 4 show slight upward trends compared to the latest measured data. Notably, all predicted settlement data for the next three years are within the safety range of ±10 mm set by the maintenance company. Thus, the overall service performance of the tunnel remains stable without significant structural changes.



(**a**)

(**b**)

(**c**)

(**d**)

Figure 13. Future settlement prediction results of different points: (**a**) Cluster 1; (**b**) Cluster 2; (**c**) Cluster 3; (**d**) Cluster 4.

Furthermore, the accumulated settlement for December 2024 of each acquisition point was calculated based on the predicted results and compared with the historical data for December 2021, as shown in Figure 14. The overall differential settlement condition between tunnel sections remains unchanged, but there is a further trend of increasing cumulative settlement in Section 5, requiring special attention.

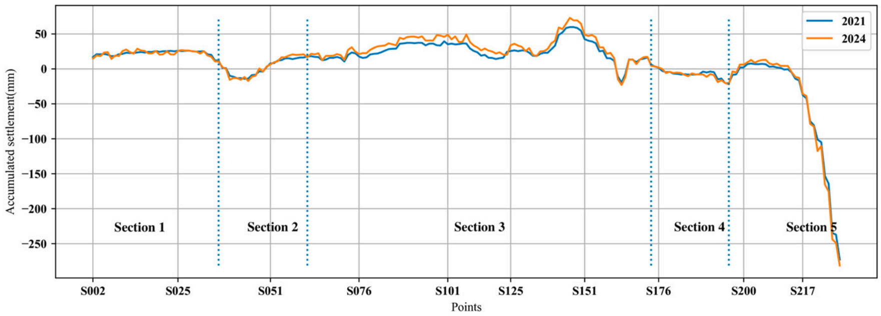


Figure 14. Comparison of future accumulated settlement in different locations.

Based on the above predictions, it is suggested that the maintenance company continuously monitor the overall settlement movement of the tunnel over the upcoming three years. For Section 5 with significant severe cumulative settlement, necessary maintenance measures should be taken to mitigate degradation. By reasonably formulating maintenance strategies based on predictive results, the tunnel can maintain regular operation, effectively reduce economic costs, and achieve the sustainability of management.

6. Conclusions

As a vital component of urban infrastructure, proper management and maintenance of tunnels are crucial for various industries [51]. In the life cycle of urban tunnels, excessive settlement poses the risk of severe structural damage, jeopardizing tunnel safety, stability, and longevity. Due to the sparsity of tunnel settlement data during the operation period, the accuracy and generalization ability of AI-based prediction models is reduced. This paper proposes an improved ML model that combines time series clustering, time series decomposition, and BO-XGBoost to achieve better predictive performance with sparse settlement data during this period. The main contributions of this paper can be summarized as follows.

1. By using a DTW-K-Means time series clustering model, the data with similar settlement patterns are aggregated for unified training. This augmentation approach directly increases the number of training samples, thereby facilitating the learning of the future trends in tunnel structures and reducing the likelihood of overfitting.
2. By utilizing the CEEMDAN time series decomposition model, the univariate settlement data are decomposed into multi-dimensional data containing different temporal frequency information. This method allows the model to effectively reveal the underlying influencing features in the univariate data, enhancing prediction accuracy.
3. By adopting BO, the high-performing XGBoost model is able to search for the optimal combination of hyperparameters even with limited sample sizes. It enhances the predictive capability of the model.

In summary, the proposed model exhibits accurate and stable predictive performance when facing sparse univariate settlement data during the operational period. Compared to traditional ML models and LSTM, the proposed model achieves the lowest prediction error and highest accuracy on both the training and testing set. The study also utilizes the proposed model to forecast the next 3-year settlement trends of the tunnel in Shanghai. Based on the prediction results, preventive maintenance strategies are suggested. This facilitates the long-term operational performance, safety, and reliability of tunnels, which supports realizing SDG 9 (Industry, Innovation, and Infrastructure) and SDG 11 (Sustainable Cities and Communities). However, due to limitations in data acquisition methods during the operation period, this study only considers the temporal patterns of settlement without incorporating other important environmental factors such as traffic flow and tides. Consequently, the model's performance on the testing set is inferior, and its generalization capability is limited.

Some further research directions may improve model performance. Firstly, tunnel structural settlement is susceptible to various external factors. Therefore, it is worth considering how to quantify and select multiple data sources to improve the model's learning ability. Additionally, while existing research has mainly focused on the temporal dependency of settlement, there is explicit spatial dependency in settlement between adjacent areas. Therefore, incorporating spatiotemporal factors can also enhance the model's performance.

Author Contributions: Conceptualization, G.Y. and M.H.; methodology, Y.J. and G.Y.; formal analysis, Y.J. and G.Y.; investigation, G.Y. and Y.J.; resources, G.Y., M.H., Z.L. and R.C.; data curation, Y.J.; verification, Z.L. and R.C.; writing—original draft preparation, Y.J., G.Y. and R.Z.; writing—review and editing, Y.J., G.Y., R.Z. and V.S.; visualization, G.Y. and Y.J.; supervision, G.Y.; project administration, M.H. and G.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the Science and Technology Commission of Shanghai Municipality, grant number 18DZ1205502; the Natural Science Foundation of Shanghai, China, grant numbers 21ZR1423800, 21ZR1423900 and 20ZR1460500.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are not publicly available due to the confidentiality requirement of the related project.

Conflicts of Interest: Authors Zhisheng Li and Rongbin Cai were employed by Shanghai Urban Construction City Operation (Group) Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Zhang, D.-M.; Huang, Z.-K.; Wang, R.-L.; Yan, J.-Y.; Zhang, J. Grouting-Based Treatment of Tunnel Settlement: Practice in Shanghai. *Tunn. Undergr. Space Technol.* **2018**, *80*, 181–196. [CrossRef]
2. Cao, Y.; Zhou, X.; Yan, K. Deep Learning Neural Network Model for Tunnel Ground Surface Settlement Prediction Based on Sensor Data. *Math. Probl. Eng.* **2021**, *2021*, 9488892. [CrossRef]
3. Chen, J.; Lu, W.; Lou, J. Automatic Concrete Defect Detection and Reconstruction by Aligning Aerial Images onto Semantic-Rich Building Information Model. *Comput.-Aided Civ. Infrastruct. Eng.* **2023**, *38*, 1079–1098. [CrossRef]
4. Chen, L.; Hashiba, K.; Liu, Z.; Lin, F.; Mao, W. Spatial-Temporal Fusion Network for Maximum Ground Surface Settlement Prediction during Tunnel Excavation. *Autom. Constr.* **2023**, *147*, 104732. [CrossRef]
5. Gao, Y.; Liu, Y.; Tang, P.; Mi, C. Modification of Peck Formula to Predict Surface Settlement of Tunnel Construction in Water-Rich Sandy Cobble Strata and Its Program Implementation. *Sustainability* **2022**, *14*, 14545. [CrossRef]
6. Mahmoodzadeh, A.; Zare, S. Probabilistic Prediction of Expected Ground Condition and Construction Time and Costs in Road Tunnels. *J. Rock Mech. Geotech. Eng.* **2016**, *8*, 734–745. [CrossRef]
7. Machado, L.B.; Futai, M.M. Tunnel Performance Prediction through Degradation Inspection and Digital Twin Construction. *Tunn. Undergr. Space Technol.* **2024**, *144*, 105544. [CrossRef]
8. Su, J.; Wang, Y.; Niu, X.; Sha, S.; Yu, J. Prediction of Ground Surface Settlement by Shield Tunneling Using XGBoost and Bayesian Optimization. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105020. [CrossRef]
9. Liu, L.; Zhou, W.; Gutierrez, M. Effectiveness of Predicting Tunneling-Induced Ground Settlements Using Machine Learning Methods with Small Datasets. *J. Rock Mech. Geotech. Eng.* **2022**, *14*, 1028–1041. [CrossRef]
10. Tang, L.; Na, S. Comparison of Machine Learning Methods for Ground Settlement Prediction with Different Tunneling Datasets. *J. Rock Mech. Geotech. Eng.* **2021**, *13*, 1274–1289. [CrossRef]
11. Chen, R.; Zhang, P.; Wu, H.; Wang, Z.; Zhong, Z. Prediction of Shield Tunneling-Induced Ground Settlement Using Machine Learning Techniques. *Front. Struct. Civ. Eng.* **2019**, *13*, 1363–1378. [CrossRef]
12. Ye, X.-W.; Jin, T.; Chen, Y.-M. Machine Learning-Based Forecasting of Soil Settlement Induced by Shield Tunneling Construction. *Tunn. Undergr. Space Technol.* **2022**, *124*, 104452. [CrossRef]
13. Shi, S.; Zhao, R.; Li, S.; Xie, X.; Li, L.; Zhou, Z.; Liu, H. Intelligent Prediction of Surrounding Rock Deformation of Shallow Buried Highway Tunnel and Its Engineering Application. *Tunn. Undergr. Space Technol.* **2019**, *90*, 1–11. [CrossRef]
14. Zhang, W.; Li, H.; Wu, C.; Li, Y.; Liu, Z.; Liu, H. Soft Computing Approach for Prediction of Surface Settlement Induced by Earth Pressure Balance Shield Tunneling. *Undergr. Space* **2021**, *6*, 353–363. [CrossRef]
15. Moghaddasi, M.R.; Noorian-Bidgoli, M. ICA-ANN, ANN and Multiple Regression Models for Prediction of Surface Settlement Caused by Tunneling. *Tunn. Undergr. Space Technol.* **2018**, *79*, 197–209. [CrossRef]
16. Zhang, P.; Wu, H.-N.; Chen, R.-P.; Dai, T.; Meng, F.-Y.; Wang, H.-B. A Critical Evaluation of Machine Learning and Deep Learning in Shield-Ground Interaction Prediction. *Tunn. Undergr. Space Technol.* **2020**, *106*, 103593. [CrossRef]
17. Xu, W.; Cheng, M.; Xu, X.; Chen, C.; Liu, W. Deep Learning Method on Deformation Prediction for Large-Section Tunnels. *Symmetry* **2022**, *14*, 2019. [CrossRef]
18. Pourtaghi, A.; Lotfollahi-Yaghin, M. Wavenet Ability Assessment in Comparison to ANN for Predicting the Maximum Surface Settlement Caused by Tunneling. *Tunn. Undergr. Space Technol.* **2012**, *28*, 257–271. [CrossRef]
19. Ge, S.; Gao, W.; Cui, S.; Chen, X.; Wang, S. Safety Prediction of Shield Tunnel Construction Using Deep Belief Network and Whale Optimization Algorithm. *Autom. Constr.* **2022**, *142*, 104488. [CrossRef]
20. Zhang, N.; Zhou, A.; Pan, Y.; Shen, S.-L. Measurement and Prediction of Tunnelling-Induced Ground Settlement in Karst Region by Using Expanding Deep Learning Method. *Measurement* **2021**, *183*, 109700. [CrossRef]
21. Yan, K.; Dai, Y.; Xu, M.; Mo, Y. Tunnel Surface Settlement Forecasting with Ensemble Learning. *Sustainability* **2019**, *12*, 232. [CrossRef]
22. Vorster, T.; Klar, A.; Soga, K.; Mair, R. Estimating the Effects of Tunneling on Existing Pipelines. *J. Geotech. Geoenviron. Eng.* **2005**, *131*, 1399–1410. [CrossRef]
23. Fang, Y.-S.; Wu, C.-T.; Chen, S.-F.; Liu, C. An Estimation of Subsurface Settlement Due to Shield Tunneling. *Tunn. Undergr. Space Technol.* **2014**, *44*, 121–129. [CrossRef]
24. Lu, D.; Lin, Q.; Tian, Y.; Du, X.; Gong, Q. Formula for Predicting Ground Settlement Induced by Tunnelling Based on Gaussian Function. *Tunn. Undergr. Space Technol.* **2020**, *103*, 103443. [CrossRef]
25. Paternesi, A.; Schweiger, H.F.; Scarpelli, G. Numerical Analyses of Stability and Deformation Behavior of Reinforced and Unreinforced Tunnel Faces. *Comput. Geotech.* **2017**, *88*, 256–266. [CrossRef]

26. Klotoé, C.H.; Bourgeois, E. Three Dimensional Finite Element Analysis of the Influence of the Umbrella Arch on the Settlements Induced by Shallow Tunneling. *Comput. Geotech.* **2019**, *110*, 114–121. [CrossRef]
27. Lai, H.; Zheng, H.; Chen, R.; Kang, Z.; Liu, Y. Settlement Behaviors of Existing Tunnel Caused by Obliquely Under-Crossing Shield Tunneling in Close Proximity with Small Intersection Angle. *Tunn. Undergr. Space Technol.* **2020**, *97*, 103258. [CrossRef]
28. Li, Z.; Chen, Z.; Wang, L.; Zeng, Z.; Gu, D. Numerical Simulation and Analysis of the Pile Underpinning Technology Used in Shield Tunnel Crossings on Bridge Pile Foundations. *Undergr. Space* **2021**, *6*, 396–408. [CrossRef]
29. Ling, X.; Kong, X.; Tang, L.; Zhao, Y.; Tang, W.; Zhang, Y. Predicting Earth Pressure Balance (EPB) Shield Tunneling-Induced Ground Settlement in Compound Strata Using Random Forest. *Transp. Geotech.* **2022**, *35*, 100771. [CrossRef]
30. Mahmoodzadeh, A.; Mohammadi, M.; Daraei, A.; Ali, H.F.H.; Al-Salihi, N.K.; Omer, R.M.D. Forecasting Maximum Surface Settlement Caused by Urban Tunneling. *Autom. Constr.* **2020**, *120*, 103375. [CrossRef]
31. Feng, L.; Zhang, L. Enhanced Prediction Intervals of Tunnel-Induced Settlement Using the Genetic Algorithm and Neural Network. *Reliab. Eng. Syst. Saf.* **2022**, *223*, 108439. [CrossRef]
32. Song, Z.; Liu, S.; Jiang, M.; Yao, S. Research on the Settlement Prediction Model of Foundation Pit Based on the Improved PSO-SVM Model. *Sci. Program.* **2022**, *2022*, 1921378. [CrossRef]
33. Qi, Z.; Liu, D.; Zhang, J.; Chen, J. Micro-Concrete Crack Detection of Underwater Structures Based on Convolutional Neural Network. *Mach. Vis. Appl.* **2022**, *33*, 74. [CrossRef]
34. Cao, B.-T.; Freitag, S.; Meschke, G. A Hybrid RNN-GPOD Surrogate Model for Real-Time Settlement Predictions in Mechanised Tunnelling. *Adv. Model. Simul. Eng. Sci.* **2016**, *3*, 5. [CrossRef]
35. Wu, H.; Wang, Z.; Hu, W.; Liu, J.; Shi, X. Research on Deformation Prediction of Subway Tunnel Structure Based on LSTM Network. In Proceedings of the International Conference on Advanced Manufacturing Technology and Manufacturing Systems (ICAMTMS 2022), Shijiazhuang, China, 27–29 May 2022; SPIE: Bellingham, WA, USA; Volume 12309, pp. 580–585.
36. Zhu, H.; Chen, X.; Zhang, L. Similarity Search and Performance Prediction of Shield Tunnels in Operation through Time Series Data Mining. *Autom. Constr.* **2020**, *114*, 103178. [CrossRef]
37. Li, X.; Lin, X.; Zhu, H.; Wang, X.; Liu, Z. Condition Assessment of Shield Tunnel Using a New Indicator: The Tunnel Serviceability Index. *Tunn. Undergr. Space Technol.* **2017**, *67*, 98–106. [CrossRef]
38. Bandara, K.; Bergmeir, C.; Smyl, S. Forecasting Across Time Series Databases Using Recurrent Neural Networks on Groups of Similar Series: A Clustering Approach. *Expert Syst. Appl.* **2019**, *140*, 112896. [CrossRef]
39. Pham, D.T.; Dimov, S.S.; Nguyen, C.D. Selection of K in K-Means Clustering. *Proc. Inst. Mech. Eng.-Part C* **2005**, *219*, 103–119. [CrossRef]
40. Na, S.; Xumin, L.; Yong, G. Research on K-Means Clustering Algorithm: An Improved k-Means Clustering Algorithm. In Proceedings of the Third International Symposium on Intelligent Information Technology & Security Informatics, Jian, China, 2–4 April 2010.
41. Bandara, K.; Hewamalage, H.; Liu, Y.H.; Kang, Y.; Bergmeir, C. Improving the Accuracy of Global Forecasting Models Using Time Series Data Augmentation. *Pattern Recognit.* **2020**, *120*, 108148. [CrossRef]
42. Zhang, Y.; Li, C.; Jiang, Y.; Sun, L.; Zhao, R.; Yan, K.; Wang, W. Accurate Prediction of Water Quality in Urban Drainage Network with Integrated EMD-LSTM Model. *J. Clean. Prod.* **2022**, *354*, 131724. [CrossRef]
43. Cao, J.; Li, Z.; Li, J. Financial Time Series Forecasting Model Based on CEEMDAN and LSTM. *Phys. A Stat. Mech. Its Appl.* **2018**, *519*, 127–139. [CrossRef]
44. Zou, M.; Jiang, W.-G.; Qin, Q.-H.; Liu, Y.-C.; Li, M.-L. Optimized XGBoost Model with Small Dataset for Predicting Relative Density of Ti-6Al-4V Parts Manufactured by Selective Laser Melting. *Materials* **2022**, *15*, 5298. [CrossRef] [PubMed]
45. Frazier, P.I. A Tutorial on Bayesian Optimization. *arXiv* **2018**, arXiv:1807.02811.
46. Chen, T.; He, T.; Benesty, M.; Khotilovich, V.; Tang, Y.; Cho, H.; Chen, K.; Mitchell, R.; Cano, I.; Zhou, T.; et al. *R Package*; Version 0.4-2; Xgboost: Extreme Gradient Boosting; The R Foundation: Vienna, Austria, 2015; Volume 1, pp. 1–4.
47. Sun, Y.; Ding, S.; Zhang, Z.; Jia, W. An Improved Grid Search Algorithm to Optimize SVR for Prediction. *Soft Comput.* **2021**, *25*, 5633–5644. [CrossRef]
48. Kong, X.; Ling, X.; Tang, L.; Tang, W.; Zhang, Y. Random Forest-Based Predictors for Driving Forces of Earth Pressure Balance (EPB) Shield Tunnel Boring Machine (TBM). *Tunn. Undergr. Space Technol.* **2022**, *122*, 104373. [CrossRef]
49. Victoria, A.H.; Maragatham, G. Automatic Tuning of Hyperparameters Using Bayesian Optimization. *Evol. Syst.* **2021**, *12*, 217–223. [CrossRef]
50. Wakjira, T.G.; Ibrahim, M.; Ebead, U.; Alam, M.S. Explainable Machine Learning Model and Reliability Analysis for Flexural Capacity Prediction of RC Beams Strengthened in Flexure with FRCM. *Eng. Struct.* **2022**, *255*, 113903. [CrossRef]
51. Lu, W.; Chen, J.; Fu, Y.; Pan, Y.; Ghansah, F.A. Digital Twin-Enabled Human-Robot Collaborative Teaming towards Sustainable and Healthy Built Environments. *J. Clean. Prod.* **2023**, *412*, 137412. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.