ELEKTROTEHNIŠKI VESTNIK 92(1-2): 13-25, 2025
ORIGINAL SCIENTIFIC PAPER

Performance evaluation of machine learning methods for ground settlement prediction

Amira Šerifović Trbalić[1], Naser Prljača[1], Ausilia Paparo[2], Martino Lorusso[2]

[1] *Faculty of Electrical Engineering, University of Tuzla, Bosnia and Herzegovina*
[2] *SECO Mind, Italy*
E-mail: mamira.serifovic-trbalic@fet.ba

Abstract. Prediction of tunneling-induced ground settlements is an important task during tunnel excavation in urban areas. Ground settlements should be limited within a tolerable threshold to avoid damages to existing buildings and infrastructures during and after the construction. Machine learning (ML) methods have been gaining an increasing popularity in many fields, including tunnel excavations, as a powerful learning and predicting technique. The paper analyzes the possibilities of different machine learning methods to predict the ground surface settlement induced by tunneling. Three different ML approaches, including support vector regression (SVR), multilayer perceptron (MLP), and long short-term memory (LSTM) networks, are utilized. Two techniques are used for the hyperparameter optimization: particle swarm optimization (PSO) and grid search (GS) methods. To assess the performance of the ML methods, three performance metrics are used: the mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE). The paper demonstrates the applicability of the three ML methods in tunneling-induced ground settlement prediction for real-world settlement datasets. The obtained experimental results indicate that the proposed ML models can accurately and efficiently predict the ground settlement.

Keywords: ground settlement, tunneling, machine learning

Ocena učinkovitosti metod strojnega učenja za napoved posedanja tal

Napoved posedanja tal, ki ga povzroča gradnja predorov, je ključnega pomena pri izkopavanju predorov v urbanih območjih. Posedanje tal mora ostati znotraj sprejemljivih mejnih vrednosti, da se preprečijo poškodbe obstoječih stavb in infrastrukture med gradnjo in po njej. Metode strojnega učenja pridobivajo vse večjo priljubljenost na različnih področjih, vključno z gradnjo predorov, saj omogočajo učinkovito učenje in napovedovanje. Prispevek analizira možnosti uporabe različnih metod strojnega učenja za napoved posedanja tal, ki ga povzroča gradnja predorov. Uporabljeni so trije pristopi strojnega učenja: regresija s podporo vektorjev, večplastni perceptron in nevronske mreže dolgega kratkoročnega spomina. Za optimizacijo hiperparametrov sta uporabljeni dve tehniki: optimizacija z rojem delcev in metoda iskanja po mreži. Za oceno učinkovitosti metod strojnega učenja so uporabljene tri metrike: povprečna absolutna napaka, kvadratna srednja napaka in povprečna absolutna odstotkovna napaka. Prispevek prikazuje uporabnost treh metod strojnega učenja za napoved posedanja tal na realnih podatkovnih zbirkah. Eksperimentalni rezultati kažejo, da predlagani modeli strojnega učenja omogočajo natančno in učinkovito napoved posedanja tal.

1 INTRODUCTION

The increasing traffic pressure has led to the construction of metro tunnels in urban areas, as the metro tunnels have become one of the most practical methods to alleviate traffic jams. During construction of metro tunnels, the ground surface settlement will be induced and can cause a significant damage to the surrounding infrastructures, during and after the construction. Settlement prediction is important for monitoring of changes and implementation of strategies for prevention of severe structural damages. The ground settlement mechanism and the undergoing processes caused by the tunneling are complex. Therefore, various methods have been proposed for prediction of the tunneling-induced ground settlement, including empirical, analytical, numerical and machine learning methods. Traditional approaches often rely on empirical or analytical methods, developed using field measurements, prior engineering knowledge and theoretical assumptions. An approach using Gaussian normal distribution to represent the tunneling-induced ground settlement was initially proposed by Peck [1]. It was further modified by numerous researches [2-4]. Analytical methods have been developed based on fundamental equations of the elastic theory. [5-9] propose multiple analytical methods for tunneling-induced ground settlement. While the empirical and analytical methods are convenient and straightforward, they are limited in their ability to capture the tunnel boring machine (TBM) operation features and complexity of geological characteristics. Numerical simulations such as the Finite Element Method (FEM) [10, 11] and Finite Difference Method (FDM) [12, 13]

Received: 13 January 2025
Accepted: 7 March 2025

Figure 1. Tunneling alignment.

provide a more sophisticated approach. They take into account the TBM operation and the geological and geometry features. In the past two decades, machine learning (ML) approaches have been applied in various studies to predict the ground settlement caused by the TBM tunneling. Examples of such ML-based prediction models include artificial neural networks (ANNs) [14], decision trees (DT), random forest (RF) [15, 16], back-propagation neural networks (BPNNs) [17], support vector machine (SVM) [14, 17], extreme gradient boosting (XGBoost) [18] and extreme learning machine (ELM) [17]. To improve the prediction accuracy, the deep learning (DL) models, such as long-short term memory (LSTM) [19], deep neural networks (DNN) [14], 1d convolutional neural networks (Conv1d) [19] and gated recurrent units (GRU) [19], have been applied. However, prediction of a tunneling-induced ground settlement, can usually use monitoring datasets of a limited scope. Also, in the literature, there is no systematic and quantitative analysis of the performance of the available ML algorithms to predict an univariate tunnel-induced ground settlement.

The paper analyzes the applicability of the ML methods in the tunneling-induced ground settlement prediction for real-world settlement datasets. It presents results of a ground settlement analysis for real settlement data induced by tunneling of a metro line tube through urban areas using TBM. Three different ML approaches, i.e. the support vector regression (SVR), multilayer perceptron (MLP), and long short-term memory (LSTM) networks, are utilized. Two techniques are used for the hyperparameter optimization: the particle swarm optimization (PSO) and the grid search (GS) method. The rest of the paper is organized as follows. Section 2 describes the analyzed data and the ML methods. Section 3 presents the experimental results. Section 4 draws conclusions.

2 METHODS

In the proposed ML-based approach to the ground settlement prediction the ground settlement data are first processed to select the relevant data by an outlier detection, resampling and interpolation. The processed data are divided into the training and test set. The two sets are further used for training and performance evaluation of the ML models. Two optimization methods, i.e. PSO and GS, are used to optimize the hyperparameters of the ML models, to fine-tune the model performance. Data processing, ML algorithms, hyperparameter optimization and evaluation metrics, are briefly described below.

2.1 Data processing

The dataset used in our study are collected from the first of the two tunnels under construction crossing beneath the densely populated area of Firenze, Italy. Since most of the tunnel excavation is conducted below the urban areas with heavy traffic roads with major transportation services, it is necessary to minimize the ground settlement and the consequent damage to the existing infrastructures. A part of the tunnel alignment and the adjacent infrastructure are presented in Fig. 1. A large number of the ground surface sensors are utilized, to measure the overall settlement during the tunnel construction. The recording frequency is once or two times per day, depending on the conditions of a particular construction work progress. Therefore, the measurement data are resampled at a one-day frequency. As the data quality significantly affects the performance of the ML models, the data processing includes the outlier detection based on a z-score before resampling and linear interpolation of the missing values. The outliers are the data points that significantly differ from other data points in the dataset. Assuming the data follows a normal distribution, the outliers are removed using the z-score method. It is a statistical measure that indicates how many standard deviations a data point is from the mean of the dataset. To remove outliers using the z-scores, each data point z-score is calculated first,

$$Z_{score} = (x_i - x_{mean}) / \sigma \tag{1}$$

The data points with the z-scores above +3 or below -3 are then marked as an outlier and filtered out from the dataset.

For the settlement prediction, in total 18 settlement measurement points (sensors) are selected. The sensors are distributed along the route of the tunnel, thus covering different parts of the urban area (e.g., tree-lined areas or purely asphalted areas).

In the first experiment, the four sensors are used (with approximately 80 m distance between them). For each sensor, approximately 2/3 of the total recorded length is taken as the training dataset for the three ML models: SVR, MLP and LSTM. The remaining 1/3 of the total recorded length is used for testing. In the second experiment, the measurement data are assigned to the training and the test sets which receive the data from 14 and 4 sensors, respectively. In both experiments, the settlement prediction is done a day ahead.

The sampling frequency being one day, the measurement data are a short and sparse one-dimensional time series. Therefore, it is necessary to select the appropriate size of the rolling window for each ML method. The size of the rolling window determines the length of the input data samples, i.e., it determines the number of the previous time steps used for prediction. As such, it is a very important parameter. By using the rolling windows, the original single-dimensional data is expanded into multidimensional data ML models. According to the data analysis and conducted experiments, the suitable rolling window size is set to 5.

2.2 *Machine learning algorithms*

After an extensive analysis of the ML algorithms used in other studies for the tunnel-induced ground settlement prediction, we analyze three ML algorithms which have been proven to be efficient and are frequently used prediction models, SVR, MLP, and LSTM.

2.2.1 SVR

The Support Vector Machine (SVM) [20] is an effective technique used to solve the classification (Support Vector Classification, SVC) and regression (Support Vector Regression, SVR) problems. SVR finds the regression function that can adequately map given input dataset x and target value y as follows:

$$f(x) = \omega\phi(x) + b \tag{2}$$

where ω is the function weight vector, b is the bias and ϕ is the nonlinear mapping from the input space to the output space. To avoid overfitting the training data samples, SVR finds function $f(x)$ such that the model bias is less than or equal to given error threshold ε and that can be achieved by minimizing the objective function (Eq. (3)):

$$\min \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}\left(\xi_i^- + \xi_i^+\right) \tag{3}$$

subject to

$$\begin{cases} y_i - \left(\omega\phi(x) + b\right) \le \varepsilon + \xi_i \\ \left(\omega\phi(x) + b\right) - y_i \le \varepsilon + \xi_i^* \\ \xi_i \ge 0, \xi_i^* \ge 0, \quad i = 1, \dots, n \end{cases} \tag{4}$$

where ε defines the margin of an acceptable error around the predicted value, C is a regularization parameter defined by the user to minimize the associated error and maximize the margin. ξ_i and ξ_i^* are the positive numbers and are the measured distances between the data points to the regression margins.

Eq. (3) can be rewritten as Eq. (5) where $K(x_i, x_j)$ is the kernel function which transforms the data point from the low-dimensional to the high-dimensional space; α_i and α_i^* are the Lagrange multipliers and n_{sv} is the number of the support vectors.

$$f(x) = \sum_{i=1}^{n_{sv}}\left(\alpha_i - \alpha_i^*\right)K\left(x_i, x_j\right) + b \tag{5}$$

subject to:

$$\begin{cases} 0 \le \alpha_i \le C \\ 0 \le \alpha_i^* \le C \end{cases} \tag{6}$$

The common kernel types in SVR include the linear, polynomial, sigmoid and radial basis function (RBF) kernels. The different kernel functions can be denoted as follows:

- Linear kernel:

$$K(x_i, x_j) = x_i^T x_j \tag{7}$$

- Polynomial kernel:

$$K(x_i, x_j) = \left(\gamma x_i^T x_j + r\right)^d \tag{8}$$

- Sigmoid kernel:

$$K(x_i, x_j) = \left(\tanh\left(\gamma x_i^T x_j + r\right)\right) \tag{9}$$

- RBF kernel:

$$K(x_i, x_j) = \exp\left(-\gamma\left\|x_i - x_j\right\|^2\right) \tag{10}$$

Prior to model training, the hyperparameters, i.e., the type of the kernels (either linear, polynomial, sigmoid, Gaussian or Gaussian Kernel Radial Basis Function (RBF)) and their corresponding parameters, i.e. C, gamma and epsilon values, are optimized using the PSO and Grid Search algorithm of the same range of the values of the above hyperparameters. Since the ground settlement data exhibits a complex, non-linear relationship, it is logical to use a non-linear kernel, such as the RBF kernel, for the SVR modeling. As expected, the RBF kernel is used as a result of the two hyperparameter optimization techniques.

2.2.2 MLP

A multilayer perceptron (MLP) is a type of the feedforward neural network architecture that consists of multiple layers of neurons, including an input layer, one or more hidden layers, and an output layer, that are connected from one layer to the next one [21]. MLP is a universal function approximator. It is a widely used type of the neural networks, particularly for supervised learning tasks, such as classification and regression. Connections between the adjacent layers are characterized with weights and biases, while the activation functions are used to introduce the non-linearity allowing the network to learn non-linear relationships between the input and output vectors. MLP employs a supervised learning technique called the backpropagation (BP) algorithm for training the network. It is an optimization procedure based on the gradient descent. The BP learning involves feeding a data from the training set to the input of the network, propagating it across the layers from the input layer to the output layer via hidden layers and then calculating the output of the network. The difference between the output to the desired output gives an error. From the error, the gradient of the error is calculated. It is then propagated back, from the output layer to the input layer (backpropagation). This makes it possible to modify the weights values of the network, always intending to decrease the network error and therefore the learning. Both hyperparameters optimization techniques tend to optimize the MLP hyperparameters, like the neuron counts in hidden layers, learning rate, maximum number of iterations, alpha value and activation functions, to improve the MLP performance.

2.2.3 LSTM

The LSTM algorithm is a type of RNN (Recurrent Neural Network), which can learn long-term dependencies in the sequence data [22]. The LSTM algorithm introduces the LSTM cell, where each cell consists of a memory cell and input, forget and output gates. The memory cell added to the LSTMs cell remembers the previous steps. The input gate controls the information flow from a previous step to the memory cell. The forget gate controls whether the information in the previous step is remembered or forgotten. The output gate controls the information flow to be the output, which is relative to the vectors of the cell memory output, previous output, and current input. When the information is inputted to the LSTM algorithm, the gates judge the information. The information that conforms to the rules is left, otherwise it is forgotten. Thus, by selectively remembering the information, the problem of long sequence dependencies in the neural network can be solved. The corresponding functions of each gate structure are given as:

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \tag{11}$$

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{12}$$

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right) \tag{13}$$

where i_t, f_t and o_t are the vector of the input, forget and output gate, respectively, of a LSTM cell at the *t*-th time; σ is the sigmoid activation function mapping the real number to $[0,1]$; b_i, b_f and b_o are the bias weights for the input, forget and output gate, respectively; h_{t-1} denotes the past hidden state and W_i, W_f and W_o are the weight matrices.
The output of the hidden layer at the *t*-th time step can be written as:

$$h_t = o_t * \tanh(C_t) \tag{14}$$

where tanh is a hyperbolic tangent function mapping the real number to $[-1,1]$ and C_t is the vector of the memory cell at a *t*-th time step:

$$\tilde{C}_t = \tanh\left(W_c \cdot \left[h_{t-1}, x_t\right] + b_c\right) \tag{15}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{16}$$

where W_c is the weight matrix and b_c are the bias weights for the memory cell.
Finally, output of the LSTM cell at the *t*-th time step can be obtained by:

$$y_t = Wh_t + b \tag{17}$$

where W is the weight matrix between the hidden nodes and the output vectors and b are the bias weights of W.

The ground settlement may be significantly affected by the time-dependent impacting factors. For example, impacting excavation works may lead to a large daily ground settlement which may change slowly during and after the construction of tunnel structures. As LSTM can learn long-term dependencies in the sequence data and the ground settlement at the predicted point depends on the tunneling information of the surrounding section, the LSTM captures the settlement information at long intervals. When LSTM outputs through the output gate, it considers not only the current input, but also the information beyond the current input through the input and forget gates, i.e., it simultaneously considers the tunneling information of the surrounding sections.

When implementing the LSTM network, numerous LSTM hyperparameters, such as the number of the neurons, batch size, epoch, learning rate, and dropout rate, can importantly affect the network performance and are therefore tuned by both hyperparameters optimization techniques

2.3 Hyperparameters optimization

The hyperparameters optimization techniques find the optimal combination of the hyperparameters for the ML models which achieve the best performance on the data in a reasonable amount of the time. Hyperparameters are different from the internal model parameters, such as the neural network weights for MLP or kernel function for SVR, which can be learned from the data during the model training phase. Therefore, the choice of the hyperparameters optimization method becomes a key issue in the ML algorithm. Among the many different hyperparameter optimization techniques, we use an evolutionary algorithm, i.e. the Particle Swarm Optimization (PSO) and a deterministic algorithm, i.e. the Grid Search (GS) algorithm.

2.3.1 Particle swarm optimization

The Particle Swarm Optimization (PSO) is an optimization algorithm designed to solve the problem of finding an optimal target value by iteratively improving the candidate solutions, here termed particles [23]. The PSO algorithm consists of a swarm of particles and each particle is represented by its position vector X_i^k, velocity vector V_i^k and fitness, where *k* is the current generation and *i* is the *i*th particle. The predominant objective of the PSO algorithm is to find the optimum fitness and the corresponding location. For the PSO algorithm, it is necessary to define the fitness function, and the PSO parameters, such as the swarm size, generations, initial velocity vectors and position vectors. In every iteration of the algorithm, for each particle, the velocity and position vectors will be updated as well as the corresponding best fitness value and position of the swarm, until the termination criteria are reached, using the following equations:

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 \left(P_i^k - X_i^k\right) + c_2 r_2 \left(P_g^k - X_i^k\right) \quad (18)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (19)$$

where c_1 is the cognitive learning factor, c_2 is the social learning factor, ω is a constant called the momentum that regulates how much the previous velocity value affects the velocity at the present step, r_1, r_2 are the random numbers in the range, P_i is the personal best location of the *i*-th particle and P_g is the global best among all particles. The PSO method is employed to optimize the hyperparameters of the ML algorithms by defining the hyperparameters of the ML algorithms as particles. It reduces the error of the settlement prediction model through a constant updating the particles.

PSO is more complex than the GS algorithm, but it supports all types of the hyperparameters and is particularly efficient for large configuration spaces. The main limitation of PSO is that it requires an appropriate population initialization, to avoid converging slowly or only identifying a local instead of a global optimum. Therefore, a proper population initialization requires a prior programmer experience or the use of population initialization techniques.

2.3.2 Grid Search

The Grid Search (GS) is a hyperparameter optimization method, which is a simple and exhaustive searching through a user-specified subset of the hyperparameter space. The GS algorithm first creates a matrix of all possible combinations of the hyperparameter values. Each combination of the hyperparameters values is used for training and evaluation of the ML model, and the performance metric is recorded. The performance metric can be any metric relevant to the specific problem, such as the accuracy or mean square error. After the performances of all possible combinations are determined, the combination that results in the best performance is chosen. Then, the model is trained again, this time using the best combination of the hyperparameters, and the final model is used for prediction.

GS can be easily implemented and parallelized, and it guarantees the identification of the best combination within the specified search space. The method is particularly beneficial for models with a limited number of the hyperparameters, such as SVR in our analysis where only four hyperparameters are tuned.

2.4 K-fold cross-validation

The k-fold cross-validation is a robust technique used to improve the generalization performance of the ML model. It divides the original training data set randomly into k subsets (known as folds), where they are used as a new training set, and the remaining subset is used as a new test set. The process iterates k times with a different subset reserved for a testing purpose each time. The model performance is evaluated by the mean prediction error of the k subsets. The fitness function is given by

$$Fitness = \frac{1}{k}\sum_{i=1}^{k} MAE_i \quad (20)$$

where MAE_i is the prediction error for the *i*-th validation set.

Because of the limited amount of the data, the three-fold cross-validation method is used in combination with the hyperparameters optimization algorithms.

2.5 Evaluation metrics

To assess the performance of the ML methods, three performance metrics are used: the mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE). Their values are calculated as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \left| r_i - p_i \right| \quad (21)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \left(r_i - p_i \right)^2} \quad (22)$$

$$MAPE = \frac{100}{n}\sum_{i=1}^{n} \left| \frac{r_i - p_i}{r_i} \right| \quad (23)$$

where r_i is the actual measured value of settlement, p_i is the predicted value of the settlement, and *n* is the number of the data samples.

3 EXPERIMENTAL RESULTS

The three ML algorithms, i.e. SVR, MLP and LSTM, in combination with the hyperparameters optimization algorithms, i.e. PSO and GS, are applied to the tunneling-induced ground settlement prediction problem with a real-world settlement dataset obtained from the tunnel construction site in Firenze, Italy. The same set of ML model hyperparameters and the same range of hyperparameters values are used for both hyperparameters optimization methods. The three-fold cross-validation method overcomes the data scarcity data and improves the robustness of the prediction model. For the settlement prediction, 18 settlement measurement points (sensors) are selected. The sensors are distributed along the route of the tunnel, thus covering different parts of the urban area (e.g., tree-lined areas or purely asphalted areas). A Python code is generated to develop an ML model with a hyperparameters optimization for the prediction of the tunneling-induced settlement.

In the first experiment, four measurement points some 80 m distant from each other are selected. For each measurement point, approximately 2/3 of the total recorded length is taken as a training dataset for the ML models (SVR, MLP and LSTM). The remaining 1/3 of the total recorded length is used for testing.

Fig. 2 and 3 show the tunneling-induced ground settlement for the measurement point 04-006. There are 120 data points for the sensor. After preprocessing the data, the first 80 data samples are used for training and the rest for testing. All ML models, with hyperparameters optimized with PSO and GS, are analyzed. Figs. 4 – 9 show the prediction results for the other three measurement points. Table 1 presents the values of the evaluation metrics: MAE, RMSE and MAPE for each measurement point, ML model and hyperparameters optimization method. It can be noticed that for each ML model, a better performance is mainly achieved when using the GS optimization. The value of the test data for the 06-003 sensor is significantly lower compared to the value of the training data, resulting in a slightly worse performance of each ML models. The sensor is located at different types of the urban area unlike the other three sensors (in an alley with trees), and it exhibits a bit different settlement curve. Among the analyzed ML models, SVR shows a good stability and a good prediction accuracy (Table 1). The traditional ML algorithms, such as SVR, tend to achieve a higher accuracy for small datasets compared to the MLP and deep learning algorithms, like LSTM. The LSTM network is prone to overfitting, especially when working with small datasets like in our experiment, where LSTM shows to a poor performance. Figs. 2 – 11 show that most of the settlement measurement data in one phase have positive values that may significantly decrease during later phases of the tunnel construction. A proper choice of the rolling window size assures enough historical data to allow the ML models to accurately predict the ground settlement values. In our experiments, the best performance is obtained when using the 5 size window.

In our second experiment, we analyze the generalizability of the ML model. The measurement data from a group of 14 consecutive sensors are assigned to the training set, while the data from another group of 4 sensors located at other locations along the route are assigned to the test set. Table 2 presents the values of the evaluation metrics: MAE, RMSE and MAPE for the test data, ML model and hyperparameters optimization method. Figs. 10 and 11 show the prediction results obtained when using the ML models in combination with the PSO and GS hyperparameters optimization techniques, respectively. As seen, each ML model predicts the future settlement along the route within a reasonable accuracy, confirmed also by the performance metrics values presented in Table 2.

It should be noted that the ML model performance is significantly affected by the size of the rolling window, and of the training dataset and the hyperparameter values of the models, and the value of k used for the k-fold cross-validation.

Table 1. Prediction results for the test data for four measurement points.

Hyperparameters optimization methods	**Measurement points**	**Machine learning model**								
		SVR			**MLP**			**LSTM**		
		MAE	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
PSO	04-006	0.017027	0.022363	0.001299	0.024737	0.030606	0.001893	0.044738	0.055646	0.003431
	05-021	0.033032	0.034982	0.003839	0.029308	0.031728	0.003429	0.228177	0.238505	0.026426
	06-003	0.558196	0.673347	0.155314	0.643909	0.708595	0.147152	1.614758	1.912369	0.301144
	07-016	0.177437	0.237796	0.032794	0.326333	0.348464	0.057831	0.422524	0.448524	0.072715
GS	04-006	0.026032	0.037614	0.001989	0.017350	0.023350	0.001324	0.036941	0.047521	0.002822
	05-021	0.017663	0.021123	0.002042	0.025566	0.026781	0.002988	0.093183	0.104332	0.010764
	06-003	0.260884	0.302590	0.063892	0.417931	0.451850	0.097591	1.585103	1.863784	0.297303
	07-016	0.137926	0.240986	0.028278	0.277085	0.341541	0.051752	0.255465	0.27916	0.045282

Table 2. Prediction results for the test data for 18 measurement points.

Hyperparameters optimization methods	**Machine learning model**								
	SVR			**MLP**			**LSTM**		
	MAE	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
PSO	0.017027	0.022363	0.001299	0.024737	0.030606	0.001893	0.044738	0.055646	0.003431
GS	0.137926	0.240986	0.028278	0.277085	0.341541	0.051752	0.255465	0.27916	0.045282

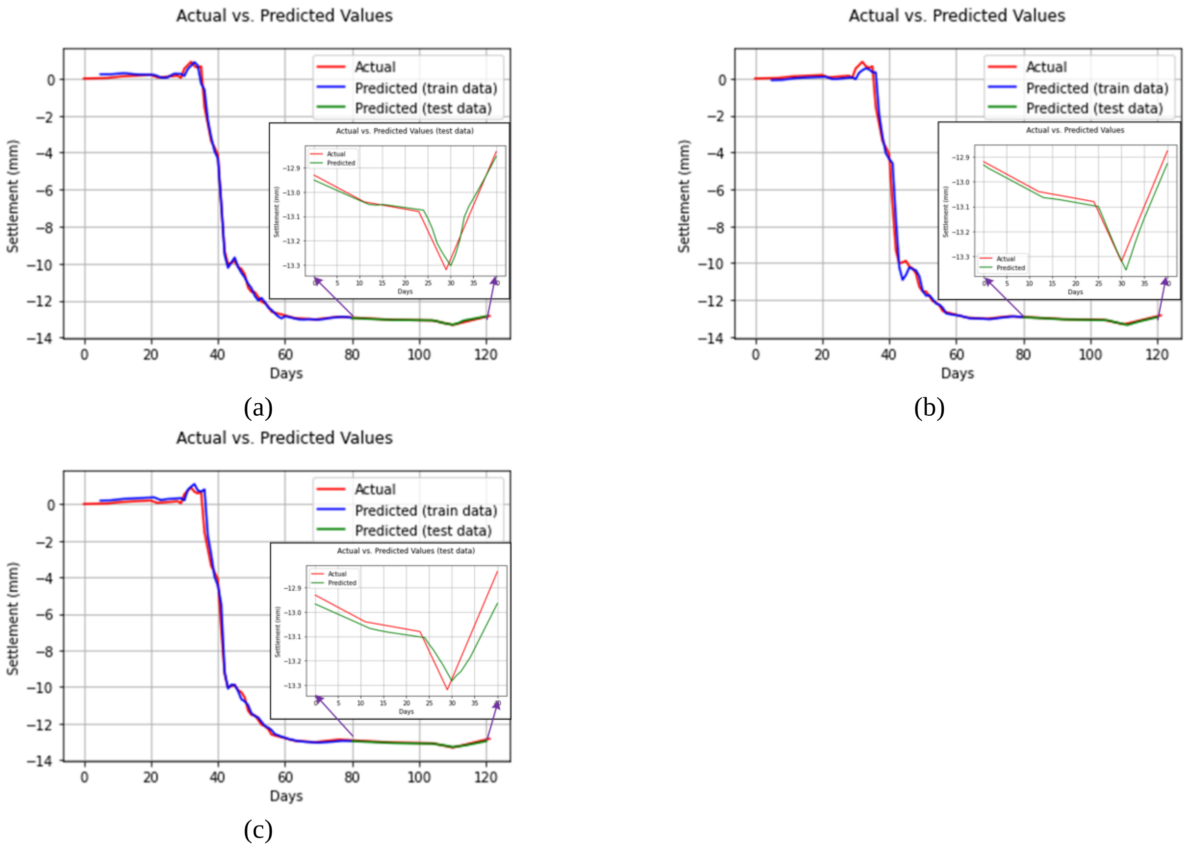


Figure 2. Prediction results for the 04-006 measurement point. PSO and: (a) SVR, (b) MLP and (c) LSTM model.

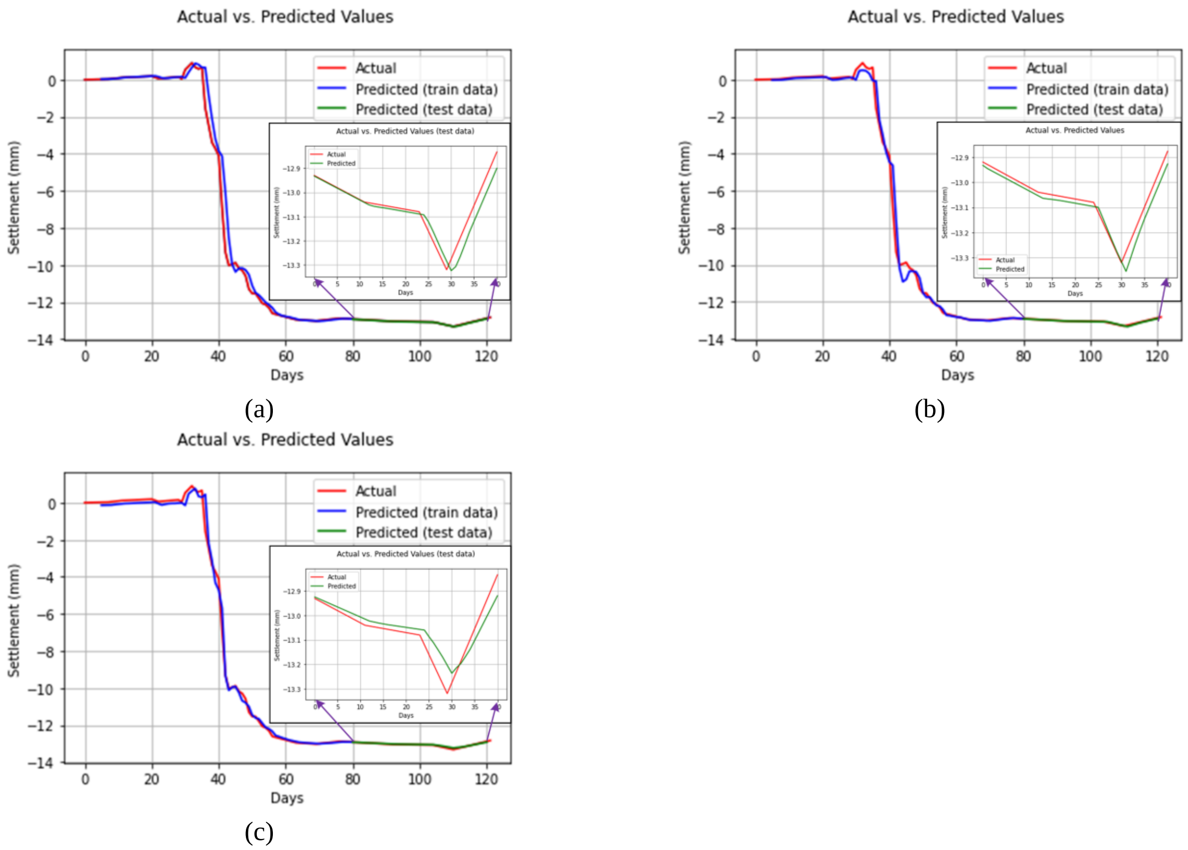
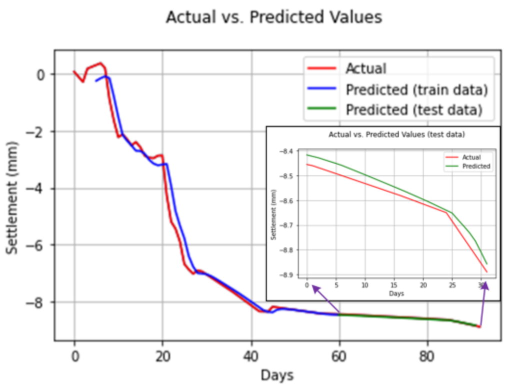
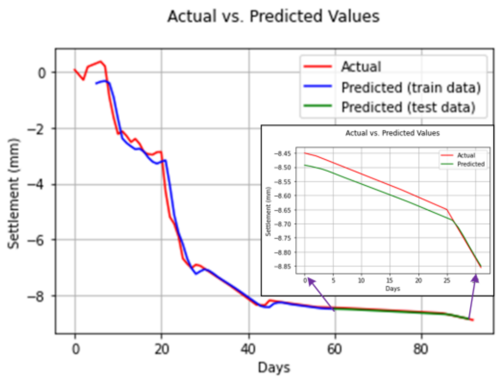


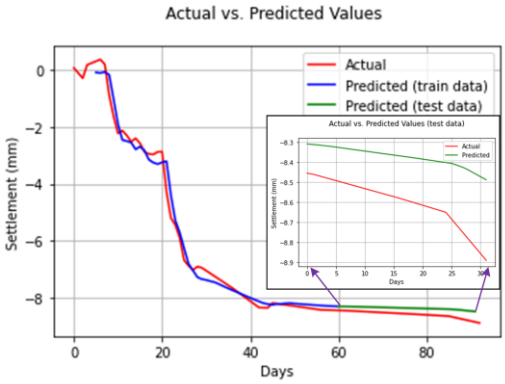
Figure 3. Prediction results for the 04-006 measurement point. GS and: (a) SVR, (b) MLP and (c) LSTM model.



(a)

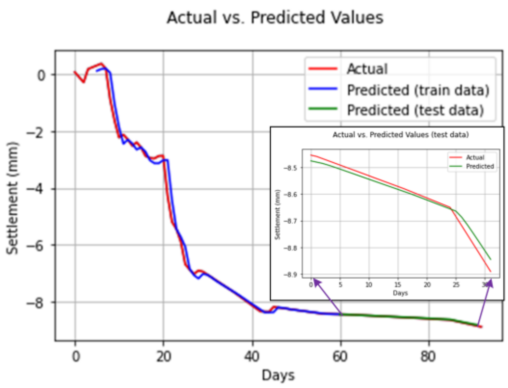


(b)

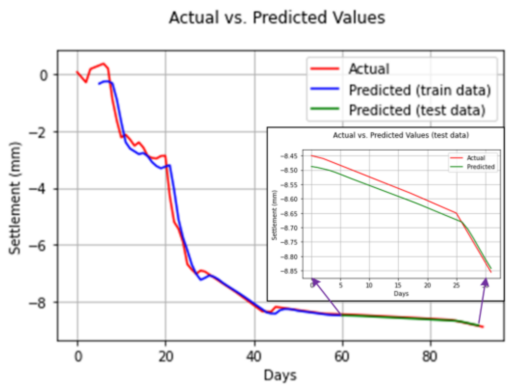


(c)

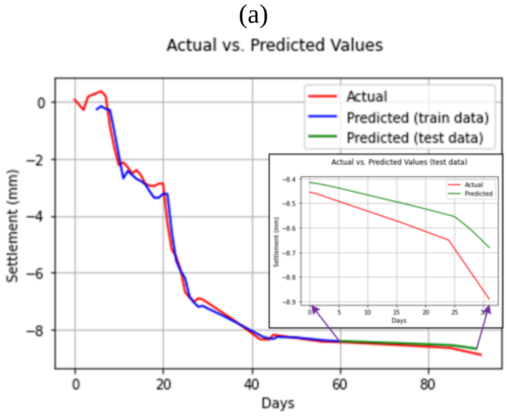
Figure 4. Prediction results for the 05-021 measurement point. PSO and: (a) SVR, (b) MLP and (c) LSTM model.



(a)

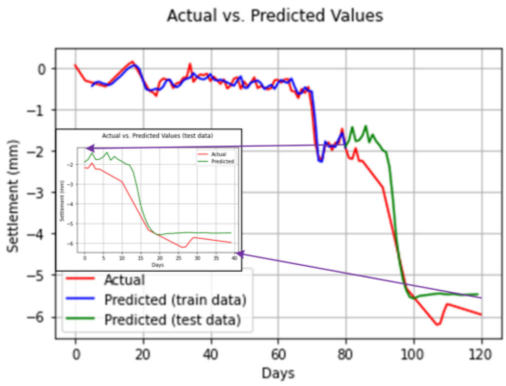


(b)

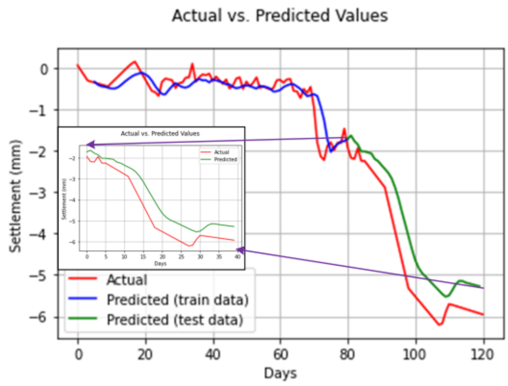


(c)

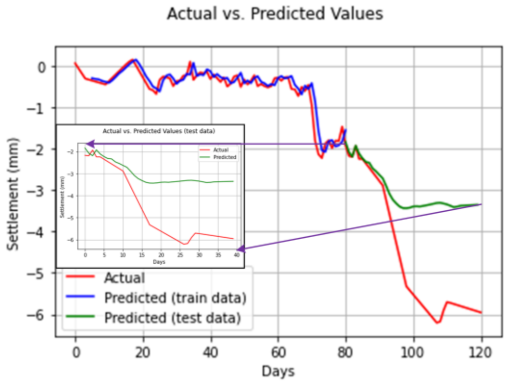
Figure 5. Prediction results for the 05-021 measurement point. GS and: (a) SVR, (b) MLP and (c) LSTM model.



(a)

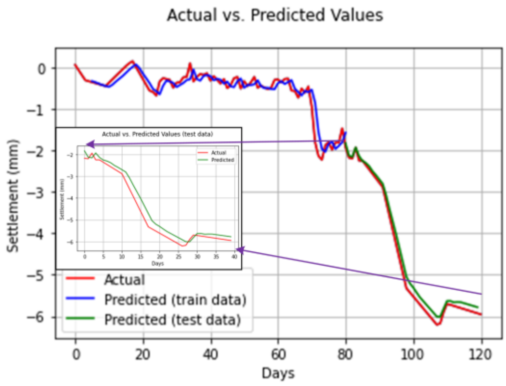


(b)

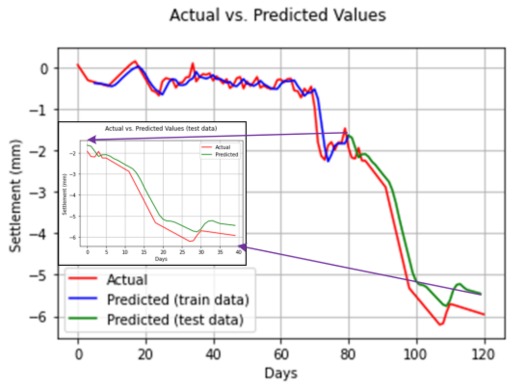


(c)

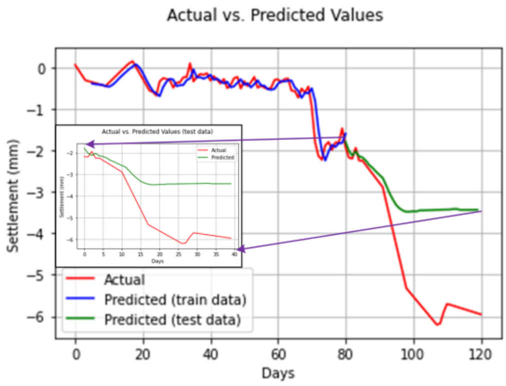
Figure 6. Prediction results for the 06-003 measurement point. PSO and: (a) SVR, (b) MLP and (c) LSTM model.



(a)

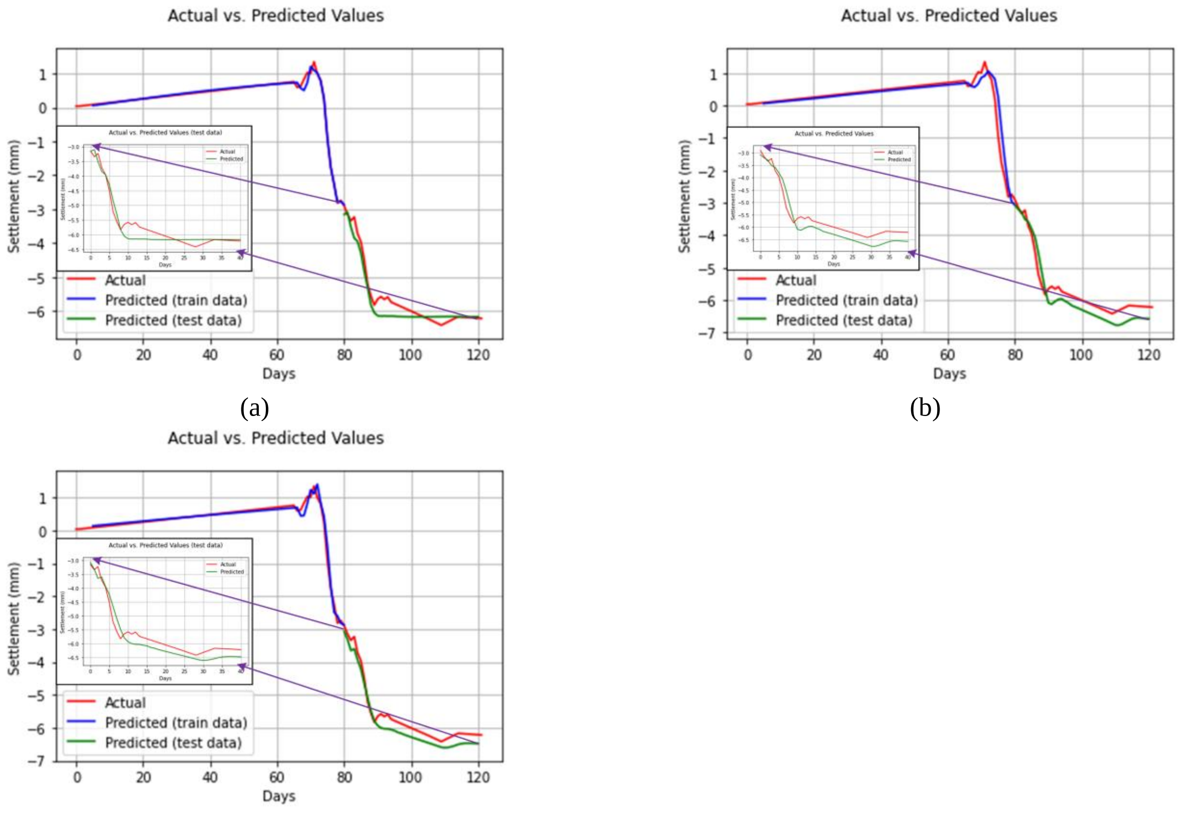


(b)



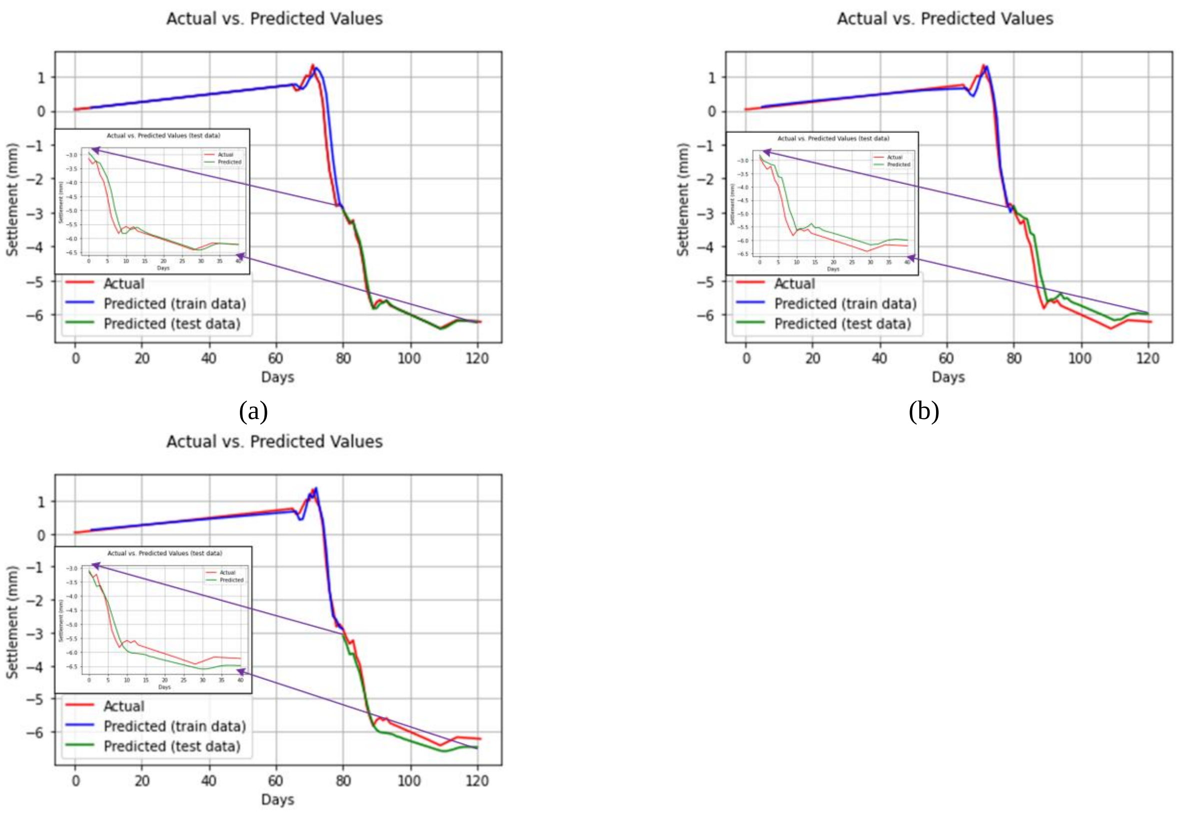
(c)

Figure 7. Prediction results for the 06-003 measurement point. GS and: (a) SVR, (b) MLP and (c) LSTM model.



(a) (b) (c)

Figure 8. Prediction results for the 07-016 measurement point. PSO and: (a) SVR, (b) MLP and (c) LSTM model.



(a) (b) (c)

Figure 9**.** Prediction results for the 07-016 measurement point. GS and: (a) SVR, (b) MLP and (c) LSTM model.

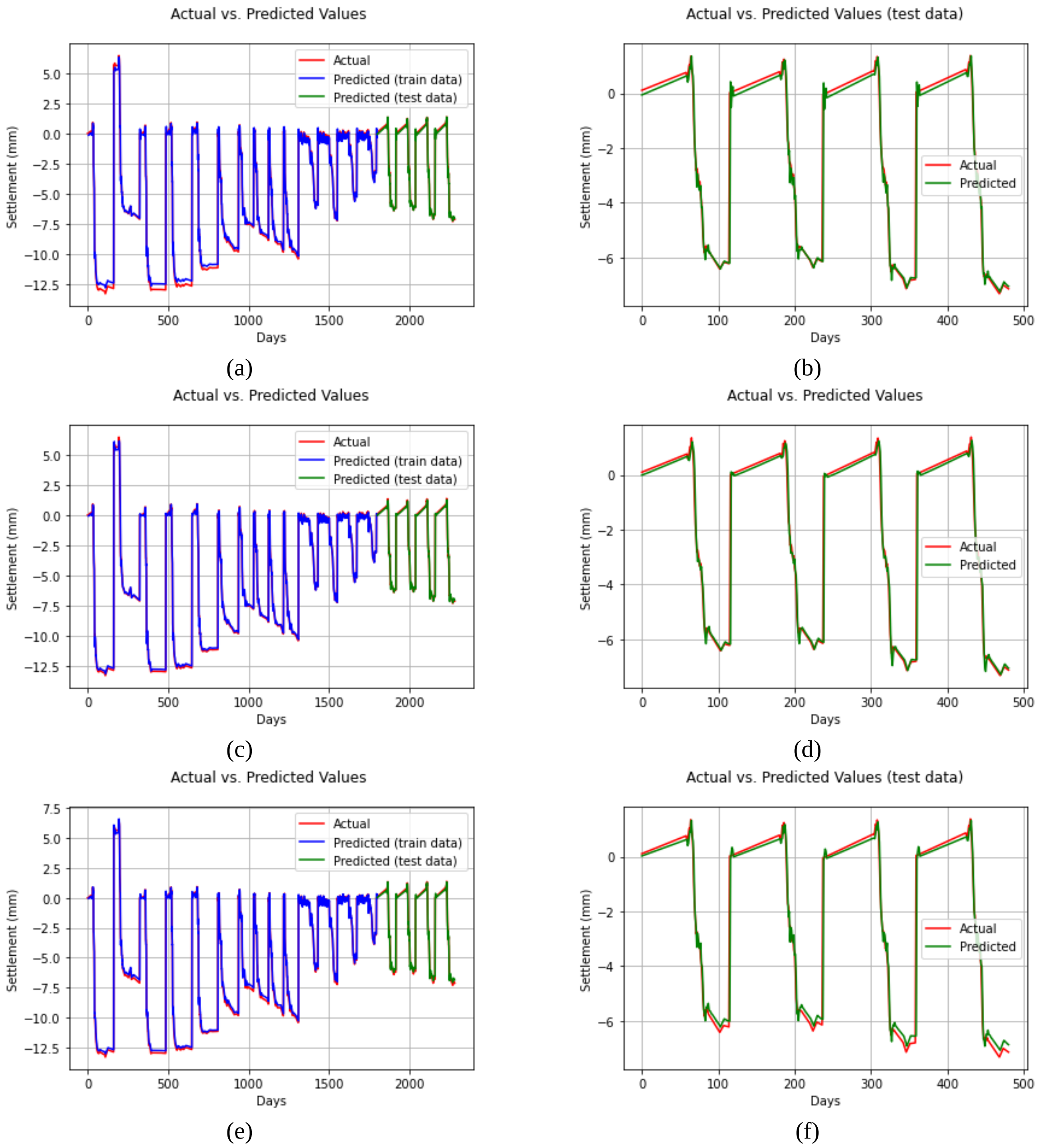


Figure 10. Prediction results for the measurement data for the PSO optimization. Prediction results for the entire dataset: (a) SVR, (c) MLP and (e) LSTM model. Prediction results for the test set: (b) SVR, (d) MLP and (f) LSTM model.

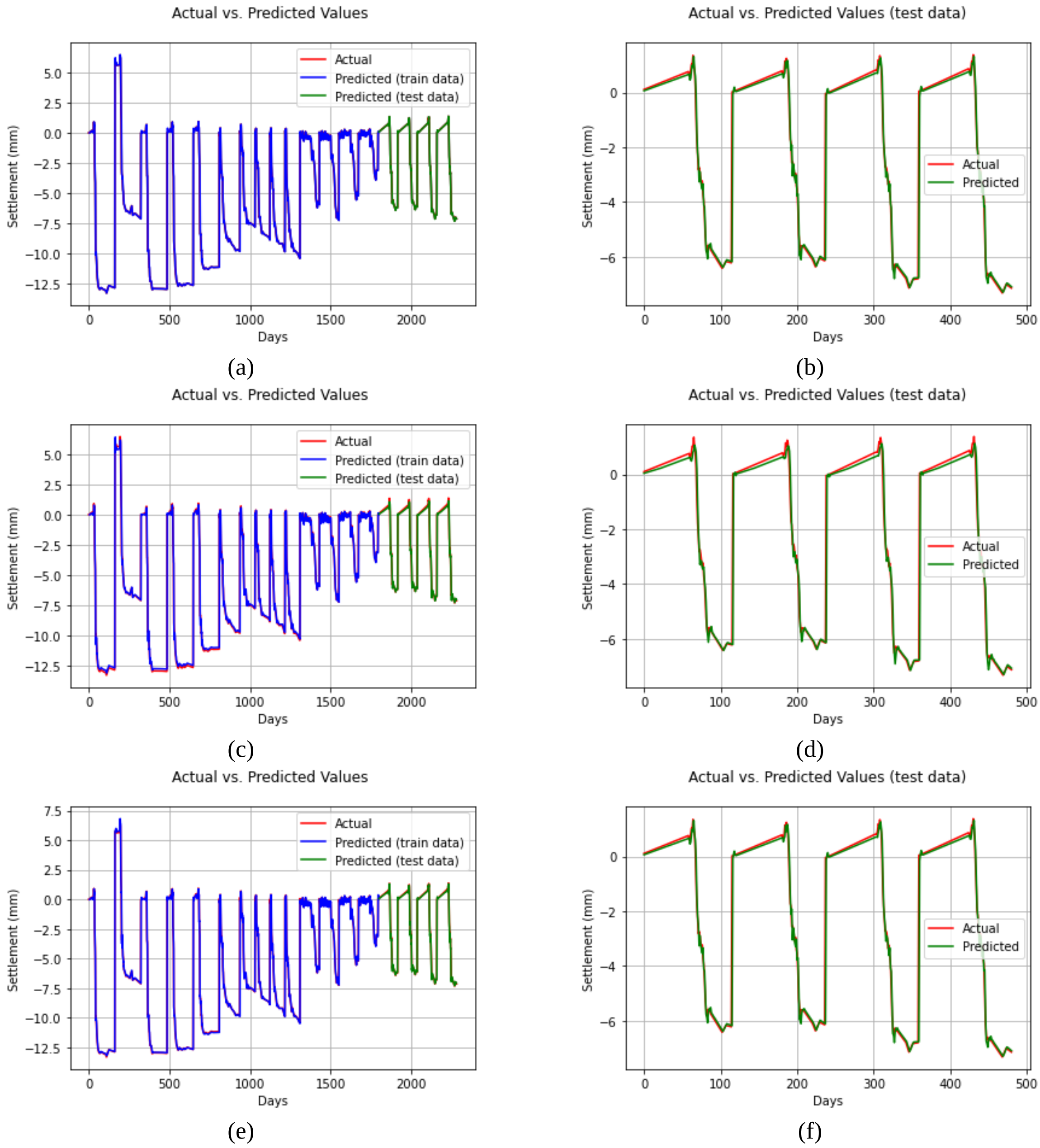


Figure 11. Prediction results for the measurement data the GS optimization. Prediction results for the entire dataset: (a) SVR, (c) MLP and (e) LSTM model. Prediction results for the test set: (b) SVR, (d) MLP and (f) LSTM model.

4 CONCLUSION

An accurate prediction of future ground settlements during a metro tunnel construction can enable constructors to implement timely and appropriate control measures to mitigate the ground settlement and thereby prevent accidents and infrastructural damages resulting from excessive settlements. The paper analyzes the use of the machine learning algorithms in combination with hyperparameters optimization techniques for the tunneling-induced settlement prediction. The mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE), are used as an evaluation metrics. The obtained results indicate that the ML algorithms have a great potential to predict the tunneling-induced settlement. SVR, MLP and LSTM algorithms show a good performance and they accurately predict the evolution of the tunneling-induced settlements. In both experiments, the SVR method is assessed to be a useful solution for predicting the tunneling-induced settlements when the datasets are small.

In our future work, more ML and hyperparameter optimization methods will be analyzed, to improve the multi-step settlement prediction.

REFERENCES

[1] B.B. Peck, “Deep excavation and tunnelling in soft ground, State of the art volume”. In Proceedings of the 7th ICSMFE, Mexico City, Mexico, 25–29 August 1969; pp. 225–29

[2] C. Oteo, J.F. Moya, Estimation of the soil parameters of Madrid in relation to the tunnel construction. Proc 7th Euro conf. on soil mechanics and foundation engineering, vol. 3, Brighton, 1979. pp 239–47.

[3] P.B. Attewell, J. Yeates, A.R. Selby, A.R., 1986. Soil Movement Induced by Tunnelling and Their Effects on Pipelines and Structures. London.

[4] F. Wang, X. Du, P. Li, 2023. Predictions of ground surface settlement for shield tunnels in sandy cobble stratum based on stochastic medium theory and empirical formulas. Undergr. Space 11, 189–203. https://doi.org/10.1016/j.undsp.2023.01.003.

[5] A. Verruijt, J.R. Booker. Surface settlements due to deformation of a tunnel in an elastic half plane. Géotechnique 1996;46(4):753–6.

[6] N. Loganathan, H.G. Poulos HG. Analytical prediction for tunneling-induced ground movements in clays. J. Geotech. Geoenviron. Eng. ASCE 1998;124(9):846–56.

[7] W.I. Chou, A. Bobet, Prediction of round deformations in shallow tunnels in clay. Tunnell. Underground Space Technol. 2002;17:3–19.

[8] K.H. Park, Analytical solution for tunneling-induced ground movements in clays. Tunnell. Underground Space Technol. 2005; 20:249–61.

[9] Z. Zhang, M. Huang, C. Zhang, K. Jiang, Q Bai, 2020. Analytical prediction of tunneling-induced ground movements and liner deformation in saturated soils considering influences of shield air pressure. Appl. Math. Model. 78, 749–772. https://doi.org/10.1016/j.apm.2019.10.025.

[10] H. Huang, W. Gong, S. Khoshnevisan, C.H. Juang, D. Zhang, L. Wang, 2015. Simplified procedure for finite element analysis of the longitudinal performance of shield tunnels considering spatial soil variability in longitudinal direction. Comput. Geotech. 64, 132–145. https://doi.org/10.1016/j.compgeo.2014.11.010.

[11] C. Zhao, A. Alimardani Lavasan, T. Barciaga, C. Kämper, P. Mark, T. Schanz, 2017. Prediction of tunnel lining forces and deformations using analytical and numerical solutions. Tunn. Undergr. Space Technol. 64, 164–176. https://doi.org/10.1016/j.tust.2017.01.015.

[12] J.Z. Zhang, H.W. Huang, D.M. Zhang, M.L. Zhou, C. Tang, D.J. Liu, Effect of ground surface surcharge on deformational performance of tunnel in spatially variable soil, Comput. Geotech., 136 (2021), 10.1016/j.compgeo.2021.104229

[13] C. Kwak, Changwon, I..J. Park, Numerical simulation for surface settlement considering face vibration of TBM tunnelling in mixed-face condition. Journal of Korean Tunnelling and Underground Space Association [Internet]. 2015 May 31;17(3):333–9. Available from: https://doi.org/10.9711/KTAJ.2015.17.3.333

[14] L. Liu, W. Zhou, M. Gutierrez, Effectiveness of Predicting Tunneling-Induced Ground Settlements Using Machine Learning Methods with Small Datasets, Journal of Rock Mechanics and Geotechnical Engineering. 2022, 14, 1028–1041

[15] L. Tang, S. Na, Comparison of Machine Learning Methods for Ground Settlement Prediction with Different Tunneling Datasets. J. Rock Mech. Geotech. Eng. 2021, 13, 1274–1289

[16] R. Chen, P. Zhang, H.Wu, Z. Wang, Z. Zhong, (2019). Prediction of shield tunneling-induced ground settlement using machine learning techniques. Frontiers of Structural and Civil Engineering, 13(6), 1363-1378.

[17] M. Hu, W. Li, K. Yan, Z. Ji, H. Hu, Modern Machine Learning Techniques for Univariate Tunnel Settlement Forecasting: A Comparative Study, *Mathematical Problems in Engineering*, 2019, 7057612, 12 pages, 2019. https://doi.org/10.1155/2019/7057612

[18] J. Su, Y. Wang, X. Niu, S. Sha, J. Yu, 2022. Prediction of ground surface settlement by shield tunneling using XGBoost and Bayesian Optimization. Eng. Appl. Artif. Intell. 114, C (Sep 2022). https://doi.org/10.1016/j.engappai.2022.105020

[19] N. Zhang, A. Zhou, P. Yutao, S. Shui-Long. (2021). Measurement and prediction of tunnelling-induced ground settlement in karst region by using expanding deep learning method. Measurement. 183. 109700. 10.1016/j.measurement.2021.109700.

[20] C. Cortes, V.L. Vapnik, (1995). Support-vector networks, Machine Learning. 20 (3): 273–297. CiteSeerX 10.1.1.15.9362. doi:10.1007/BF00994018. S2CID 206787478.

[21] S. Haykin, Neural Networks and Learning Machine, (2006) Pearson; 3rd edition.

[22] S. Hochreiter, J. Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[23] J. Kennedy, R. Eberhart, Particle swarm optimization, in Proceedings of the IEEE International Conference on Neural Networks (ICNN ’95), vol. 4, pp. 1942–1948, Perth, Western Australia, November-December 1995.

Amira Šerifović Trbalić is an Associate Professor at the Faculty of Electrical Engineering, University of Tuzla, Bosnia and Herzegovina. She received her PhD degree from the same faculty in 2011. Her research interests are in machine learning and computer vision.

Naser Prljača is a Full Professor and Head of the Department of Control Systems, Robotics and Industrial Informatics at the Faculty of Electrical Engineering, University of Tuzla, Bosnia and Herzegovina. He holds BSc, MSc and PhD degrees in Electrical and Electronics Engineering. He has been taking part in numerous research and industrial projects, and is the author of a number of academic publications. His research and teaching interests include control systems, embedded systems, robotics, machine learning and computer vision.

Ausilia Paparo is a Chief Data Scientist at the SECO Mind Italy. She received her PhD degree from the University of Bologna in 2013.

Martino Lorusso is a Senior Data Scientist at the SECO Mind Italy. He holds a Master's Degree in Environmental & Territory Engineering from Politecnico di Bari.