# Tunnel Settlement Prediction by Transfer Learning

**Qicai Zhou[1], Hehong Shen[1*], Jiong Zhao[1] & Xiaolei Xiong[2]**

[1]School of Mechanical Engineering, Tongji University,
4800 Caoan Road, Shanghai, China
[2]Tongji Zhejiang College, 168 Shangwu Road, Jiaxing, China
[*]E-mail: shenhhh@foxmail.com

**Abstract.** Tunnel settlement has a significant impact on property security and personal safety. Accurate tunnel-settlement predictions can quickly reveal problems that may be addressed to prevent accidents. However, each acquisition point in the tunnel is only monitored once daily for around two months. This paper presents a new method for predicting tunnel settlement via transfer learning. First, a source model is constructed and trained by deep learning, then parameter transfer is used to transfer the knowledge gained from the source model to the target model, which has a small dataset. Based on this, the training complexity and training time of the target model can be reduced. The proposed method was tested to predict tunnel settlement in the tunnel of Shanghai metro line 13 at Jinshajiang Road and proven to be effective. Artificial neural network and support vector machines were also tested for comparison. The results showed that the transfer-learning method provided the most accurate tunnel-settlement prediction.

**Keywords**: *deep neural network; gated recurrent unit; settlement prediction; tunneling; transfer learning.*

## 1 Introduction

In recent years, artificial intelligence, especially deep-learning technology, has achieved great success in many different fields, such as the internet industry, the financial sector and the education industry. Meanwhile, the interaction between deep learning and the manufacturing industry is also becoming more intense. Deep-learning models like Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Restricted Boltzmann Machine (RBM) and Auto Encoder (AE) have been successfully applied to prognostics and health management of mechanical equipment [1].

These methods are used to handle various problems among all sorts of machine components. Li [2] used RBM as a deep statistical feature extraction tool for both gearbox and bearing systems; the fault classification performances in the experiments were 95.17% and 91.75%, respectively. This shows that deep

Received February 19th, 2019, Revised June 10th, 2019, Accepted for publication August 20th, 2019.
 ISSN: 2337-5787, DOI: 10.5614/itbj.ict.res.appl.2019.13.2.3

learning with statistical feature extraction has interesting improvement potential for diagnosing rotating machinery faults. Obst [3] presented a distributed recurrent neural network architecture that learned spatio-temporal correlations between different sensors and made use of the learned model to detect anomalous sensors by using distributed computation and only local communication between nodes. What is more, some companies that focus on combining deep learning and manufacturing have been established, as typified by Landing.ai, which is aimed at providing AI brains for manufacturing companies.

As we all know, metro systems provide great convenience to people who need transportation. However, tunneling metro tunnels is not an easy task. Usually metro tunnels are tunneled using a shield machine, which is a kind of construction equipment dedicated to tunnel construction. A tunnel constructed by the shield method is called a shield tunnel. During the process of tunneling using the shield method there is an extremely important parameter, called surface settlement, which is used to measure the degree of ground subsidence. Large surface settlement affects the safety of the shield construction and the normal use of surrounding buildings, so monitoring and prediction of surface settlement values is essential. Many researches concerning surface settlement of tunnels have been done. Wei [4] put forward a prediction model of long-term uneven settlement in tunnels on the basis of ant colony algorithms. They tested the model by comparing the predicted data with the measured data and the results showed that the established prediction model proved to be accurate, easy to operate and adaptable.

Fan [5] proposed an adaptive multiple kernel learning (AMKL) method to improve the prediction precision of support vector machine (SVM). They used a tree structure to screen the kernels. This process could be done with manipulation of growing and cutting branches for adding and multiplying kernels in each layer. At the same time they used grid traversal and particle swarm optimization to solve optimization problems of the kernel parameters, the weight coefficient and the model parameters. The result showed that AMKL could effectively improve accuracy. Goh [6] utilized a multivariate adaptive regression spline approach to establish relationships between the maximum surface settlement and the major influencing factors. Majedi *et al.* [7] expected the estimates of ground settlement to fall into experimental, analytical and numerical groups. In their study, an estimate of ground settlement caused by tunneling was first investigated through experimental and analytic procedures.

However, there is a difficulty in predicting tunnel settlement that can never be ignored. Because there is a complex nonlinear relationship between tunnel settlement and many uncertain random factors, it is difficult to use a certain relationship for the description. Fortunately, deep-learning technology has a strong non-linear fitting capability and thus offers new ideas when trying to predict tunnel settlement more accurately. This paper presents a new method to predict tunnel settlement via transfer learning. The target prediction model built by deep learning can be trained by knowledge transferred from the source model. The method was applied to predict tunnel settlement in the Shanghai metro line 13 tunnel at Jinshajiang Road. Comparing the predictions among transfer learning, ANN, and SVM, the transfer learning method was proven to be highly effective in accurately and timely predicting tunnel settlement.

## 2 Transfer-Learning Method

Usually deep learning requires a large amount of high-quality data, which are not always available, such as settlement values in the case under study. Under this circumstance, transfer learning may provide a solution. Transfer learning is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem [8]. The differences between deep learning and transfer learning is that deep learning tries to learn each task from scratch, while transfer learning tries to transfer knowledge from a previous task to a target task when the latter has fewer training data. Combining deep-learning technology with transfer-learning technology can be useful even with small data volumes. For example, knowledge gained while learning to recognize cats could be applied when trying to recognize dogs.

### 2.1 Transfer-Learning Theory

First we give an explanation of the transfer-learning process in simple words. The core thought of transfer learning is to help improve the learning of the target task using knowledge gained in dealing with a related problem. As shown in Figure 1, there is a target dataset, including feature vectors and labels. The target task is to establish the mapping relation between the feature vectors and labels; this mapping relation is called the target model. Unfortunately, in some cases we do not have enough data to establish an accurate mapping relation, i.e. the target task cannot be executed successfully with only a small amount of target data. Thanks to a related problem that has a large amount of data, the target task can still be handled. Knowledge gained from the training process of the source model is transferred to deal with the target task. In this way we can get a satisfactory target model.

According to Ref. [9], there are four kinds of transfer learning: instance transfer, feature-representation transfer, parameter transfer, and relational-knowledge transfer. Instance transfer is used to re-weight labeled data in the source domain for use in the target domain. Feature-representation transfer is used to find a 'good' feature representation that reduces the difference between the source and the target domains and the error of classification and regression models. Parameter transfer is used to discover shared parameters or priors between the source domain and the target domain models, which can be beneficial for transfer learning. Relational-knowledge transfer is used to build a mapping of the relational knowledge between the source domain and the related target domain.

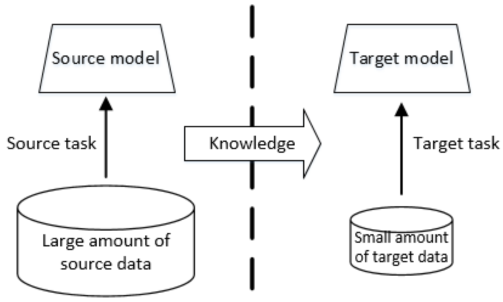


**Figure 1** Transfer learning.

Our study was carried out using parameter transfer. Using parameter transfer, there is no need to train the model with a large amount of data because the parameters transferred from the source model allow for an accurate prediction model with a small training dataset. Additionally, parameter-transfer can be conveniently combined with deep learning to decrease the training complexity and shorten the training time of the deep neural network.

## 2.2 The Reason for using Transfer Learning

When monitoring tunnel settlement, data acquisition is the first step. Usually, a tunnel has many acquisition points and these acquisition points are measured once a day. The number of acquisition points grows during the tunneling process and each acquisition point is monitored for around two months until its settlement value point tends to be stable. What we wanted to achieve in this study was to predict tunnel settlement using data that have already been obtained.

Deep-learning algorithms, especially RNNs, are good at dealing with time-series data such as tunnel-settlement values because of its strong non-linear fitting capability. Modified RNNs such as long short-term memory (LSTM) and gated recurrent unit (GRU) have been proposed for application to time-series prediction [10,11]. Deep-learning algorithms require a large amount of data, but each acquisition point only generates about 60 data points, which is not enough for training a deep neural network. Luckily we have plenty of acquisition points and they have the same task, so transfer learning can solve this problem effectively owing to the fact that the related problem has a large amount of data. On account of all acquisition points having the same task, it is preferred to transfer parameters rather than instances or features. Thus, parameter transfer is the better choice.

## 3 Modeling Approach

The modeling process of tunnel-settlement prediction using transfer learning is shown in Figure 2. In transfer learning there is a source task and a target task. The source task is used to aid in the learning of the target task [12]. The first step is collecting as many data as possible from the acquisition points to build the source dataset.

Then an appropriate deep neural network is constructed as the prediction model. The model is trained using the source dataset until it meets the precision requirements and we can get the source model. After that, the parameters gained are transferred from the source task, including weights and bias, to the target task. These parameters are kept unchanged and the target model is trained using the target dataset until it meets the precision requirements.

Finally, the target model can be used to predict tunnel settlement. Figure 2 shows a detailed explanation of this process.

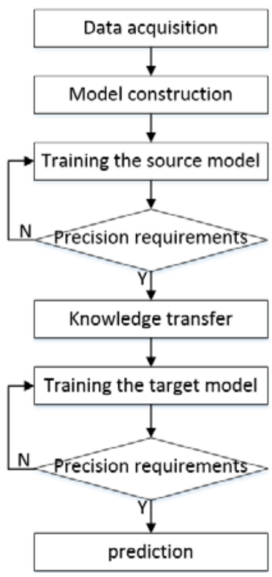


**Figure 2** Modeling process.

### 3.1 Data Acquisition

The number of acquisition points becomes increasingly bigger as the tunneling process proceeds, but the measurement of each single acquisition point will be stopped when the settlement value tends to be stable, which takes about two months. It is necessary to collect source data from as many acquisition points as possible. It is equally important to collect source data from the beginning to the stable state of each single acquisition point in order to ensure their integrity and comprehensiveness. Only in this way can we get a high-quality source dataset.

### 3.2 Model Construction

When using deep learning to predict tunnel settlement it is fundamental to build an appropriate deep neural network. A deep neural network is an artificial neural network (ANN) with multiple hidden layers between the input and the output layer that can model complex non-linear relationships. For constructing a deep neural network, a number of elements have to be determined: basic architecture of the network, number of hidden layers, number of neuron cells in each layer, and activation functions between different layers.

Tunnel-settlement values are time-series data, so using RNN as the basic architecture of the deep neural network is a good choice. RNN is characterized by the presence of cycles in a graph of interconnections; it models temporal dependencies of unspecified duration between the inputs and the associated desired outputs by using an internal memory that captures what has already been calculated. The memory is coded by recurrent connections and the outputs of the neurons themselves.

### 3.3 Training the Source Model

Using the source dataset to train the source model is the goal in this step. The source model is built based on the deep neural network architecture, so the model is trained using the training methods for deep learning. First, add the initialization parameters to the model and then do forward propagation to calculate the output. For comparing this output with labeled data, a loss function has to be determined. A loss function is a function that maps an event or values of one or more variables onto a real number, intuitively representing some 'cost' associated with the event. The training process will not stop until the 'cost' is below a predefined threshold value.

The most commonly used loss function is mean squared error. After getting the 'cost', do backward propagation to update the parameters. During backward propagation an optimizer is necessary, which aims to find the best available values of some objective function, given a defined domain. Frequently used optimizers are stochastic gradient descent optimizer, RMSprop optimizer, and Adam optimizer. After updating the parameters, repeat forward propagation, loss calculation, backward propagation, and parameter updating until the model meets the precision requirements or reaches the maximum number of iterations. This training process can be accomplished more efficiently by means of deep-learning libraries like Tensorflow or Keras.

## 3.4 Knowledge Transfer

Knowledge transfer is vital to train the target model. Parameter transfer plays the role of a porter who transfers the parameters containing weights and bias from the source model to the target model in order to fulfill the target task. As shown in Figure 3, a weight matrix and a bias matrix of the source model are available after training. Each layer except the output layer has its own weight and bias, for example, the first layer's weights and bias are w1 and b1, respectively. Because the source model and the target model share the same model construct, the weight and bias values from some layers of the source model can be transferred directly to the corresponding layers of the target model. Thus, the first layer's weights and bias of the target model are also w1 and b1, and the same applies for the other layers.

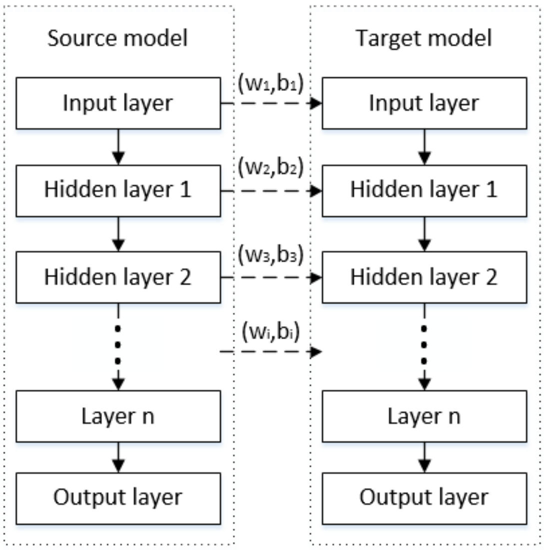


**Figure 3** Parameter transfer.

## 3.5 Training the Target Model

Via parameter transfer, the target model has been given the weights and bias of a small number of layers, so the training complexity and training time can be reduced significantly. The parameters transferred from the source model are constants through the whole training process, which is to say that only a few layers need to be trained. The training method is the same as the method for training the source model, which consists of four steps: forward propagation, loss calculation, backward propagation, and parameter updating, and repeating until the target model meets the precision requirements or reaches the maximum number of iterations.

By parameter transfer, the target model can be trained using a small amount of data owing to the related problem, which shares the same task and the same model construct. Once the remaining parameters have been determined, the target model can be used to predict values of tunnel settlement.

## 4 Case Implementation

In this study, tunnel-settlement data were measured at the Shanghai metro line 13 tunnel at Jinshajiang Road. Shanghai is located in China’s soft soil region, which features high moisture content, large voids, and high compressibility; settlement and deformation phenomena are highly likely in its tunnels [13]. The source dataset consisted of settlement data from 2000 acquisition points, which were collected once daily for 50 consecutive days. As an example, Table 1 shows the settlement data from one acquisition point. Our goal was to predict another set of acquisition point data, which were only measured for 20 consecutive days (Table 2) using the knowledge from the source model.

**Table 1** Tunnel settlement data from one acquisition point.

| No. | Measured value (mm) | No. | Measured value (mm) | No. | Measured value (mm) | No. | Measured value (mm) |
|---|---|---|---|---|---|---|---|
| 1 | 0.70 | 14 | 35.56 | 27 | 48.87 | 39 | 60.34 |
| 2 | 1.07 | 15 | 37.38 | 28 | 50.02 | 40 | 61.78 |
| 3 | 1.18 | 16 | 38.47 | 29 | 51.48 | 41 | 62.60 |
| 4 | 6.78 | 17 | 39.52 | 30 | 53.02 | 42 | 63.05 |
| 5 | 13.68 | 18 | 41.59 | 31 | 54.89 | 43 | 63.45 |
| 6 | 16.74 | 19 | 42.33 | 32 | 56.87 | 44 | 65.87 |
| 7 | 20.58 | 20 | 42.69 | 33 | 57.27 | 45 | 66.19 |
| 8 | 23.11 | 21 | 42.77 | 34 | 58.34 | 46 | 66.90 |
| 9 | 25.89 | 22 | 44.69 | 35 | 59.09 | 47 | 67.63 |
| 10 | 28.45 | 23 | 44.90 | 36 | 59.20 | 48 | 68.97 |
| 11 | 30.20 | 24 | 45.80 | 37 | 59.82 | 49 | 70.41 |
| 12 | 31.82 | 25 | 46.65 | 38 | 59.97 | 50 | 70.43 |
| 13 | 33.87 | 26 | 47.97 | | | | |

**Table 2** Tunnel settlement data from the target task.

| No. | Measured value (mm) | No. | Measured value (mm) | No. | Measured value (mm) | No. | Measured value (mm) |
|---|---|---|---|---|---|---|---|
| 1 | 37.9 | 6 | 43.7 | 11 | 48.8 | 16 | 55.7 |
| 2 | 38.7 | 7 | 43.4 | 12 | 49.6 | 17 | 56.6 |
| 3 | 40.3 | 8 | 46.0 | 13 | 50.0 | 18 | 57.9 |
| 4 | 41.8 | 9 | 46.3 | 14 | 52.5 | 19 | 58.8 |
| 5 | 43.0 | 10 | 47.9 | 15 | 54.6 | 20 | 59.9 |

As shown in Figure 4, we next distilled each six consecutive data points as a single instance of the sample dataset. After securing each single instance, we removed the first data point and added a data point after the sixth data point to form a new single instance, and so on. This ensured that at least five data points of the previous single instance were the same as the first five data points of the last single instance.

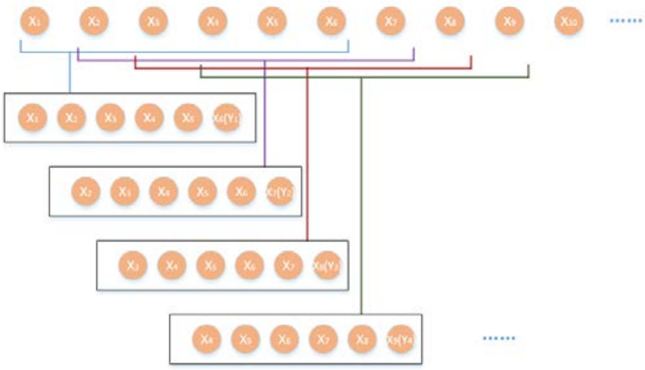

**Figure 4** Dataset partition.

We established the prediction model based on the RNN architecture and replaced the simple unit of RNN by GRU in order to enhance the accuracy of the prediction. GRU has two gates: reset gate r and update gate z. The function of the reset gate is to determine how to combine the new input with the previous memory. The update gate functions to determine how much of the previous memory has to be retained. Each hidden unit has separate reset gates and update gates and can learn to capture dependencies over different time scales. These characteristics allow GRU to make each recurrent unit adaptively capture dependencies of different time scales.

GRU's structure is shown in Figure 5, where $x_t$ represents the input matrix at time t, $r_t$ represents the reset gate at time t, $z_t$ represents the update gate at time t, $\widetilde{h_t}$ represents the candidate activation at time t, and $h_t$ represents the actual activation at time t. These parameters can be calculated by

$$z_t = \emptyset(W_z x_t + U_z h_{t-1}) \quad (1)$$

$$r_t = \emptyset(W_r x_t + U_r h_{t-1}) \quad (2)$$

$$\widetilde{h_t} = \emptyset[W x_t + U(r_t \odot h_{t-1})] \quad (3)$$

$$h_t = z_t h_{t-1} + (1 - z_t)\widetilde{h_t} \quad (4)$$

where W and U are the weight matrices, and $\emptyset(*)$ refers to the activation function.

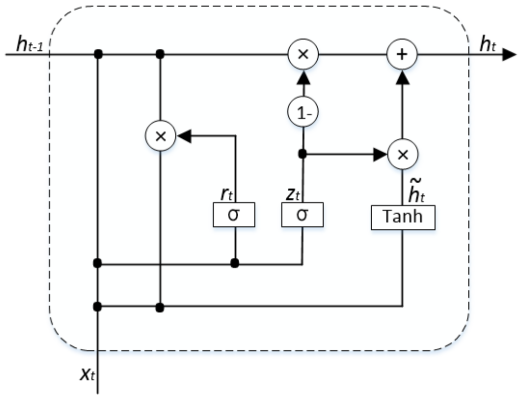


**Figure 5** Gated recurrent unit.

We built a network for tunnel-settlement prediction with one input layer, two hidden layers, and one output layer. As Figure 6 shows, the first hidden layer is a GRU layer of ten neurons with a rectified linear unit (ReLU) activation function and the second hidden layer is also a GRU layer of five neurons with a ReLU activation function. The output layer is a fully connected layer of one neuron. The input layer feeds into the first GRU layer, which in turn feeds into the second GRU layer and then feeds into the output layer, which finally outputs the prediction of the next time step. The sample data set is input into the input layer. Then nonlinearity is applied and high-dimensional features are extracted by the hidden layers. Through this model, any settlement value can be predicted by the previous five.

The proposed tunnel-settlement prediction model was established in the Keras library for Python. Keras is a high-level neural network API capable of running on top of TensorFlow, CNTK or Theano; it enables fast experimentation and is minimal, modular, and extensible. A loss function and optimizer are necessary for training a model. We used mean squared error as the loss function and

Adam as the optimizer. Meanwhile we set the batch size equal to 64 and used 100 epochs after testing several parameters. After the training step, we got the parameters of the source model, including a weight matrix and a bias matrix, and these parameters were the knowledge that would be transferred to the target model.

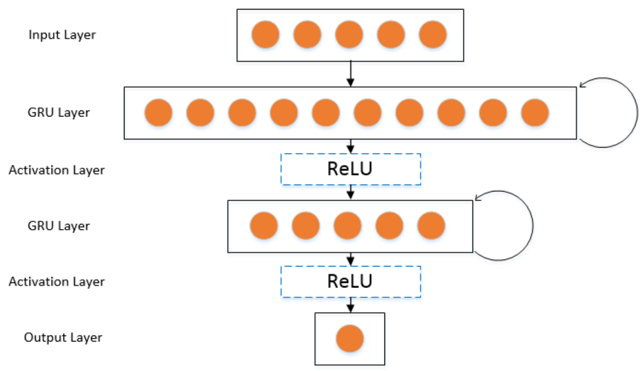


**Figure 6** Deep neural network for tunnel settlement prediction.

We collected tunnel-settlement values of the target acquisition points for 20 consecutive days. In this case, we took the first 16 settlement values as known and the latter 4 settlement values as to be predicted. Because there is a high similarity in the settlement at these acquisition points, the source model and the target model can share the same task. Thus, the knowledge learned from the source task can be applied to the target task [14]. Considering the small target dataset, we assigned the parameters transferred from the source model to the input layer and the first hidden layer of the target model during the training process of the target model. Hence, we only had to train the second hidden layer. For the second hidden layer and output layer, the weights were initialized by Glorot initialization and the bias was initialized by all-zero initialization. The number of layers that did not use knowledge from the source model was determined by the control variable method. Similarly, we used Keras to establish the prediction model and used mean squared error as the loss function and Adam as the optimizer. This time we set the batch size to 1 and used 100 epochs to train the target model. We then applied the trained model to predict the last 4 settlement values of the target acquisition point.

## 5 Results and Discussion

As described above, we obtained the predicted values from the target model, which had been trained through parameter transfer. We also tested Artificial Neural Network and Support Vector Machine to predict tunnel settlement based on the same target dataset for comparison against the proposed method. ANN is a computing system that can learn tasks by considering examples, generally without task-specific programming. SVM is a supervised learning model with associated learning that analyzes data for classification and regression analysis. Table 3 shows the tunnel-settlement measurements and predictions obtained by the three methods as well as their corresponding errors.

**Table 3** Prediction results of tunnel settlement.

| No. | Measured value (mm) | Transfer learning | | ANN | | SVM | |
|---|---|---|---|---|---|---|---|
| | | Predictions (mm) | Relative error (%) | Predictions (mm) | Relative error (%) | Predictions (mm) | Relative error (%) |
| 17 | 56.6 | 56.835 | 0.415 | 56.218 | 0.675 | 57.453 | 1.507 |
| 18 | 57.9 | 57.734 | 0.287 | 57.667 | 0.402 | 56.115 | 3.083 |
| 19 | 58.8 | 59.023 | 0.379 | 59.339 | 0.917 | 59.706 | 1.541 |
| 20 | 59.9 | 59.925 | 0.042 | 60.450 | 0.918 | 59.371 | 0.883 |

Table 2 shows information that can be used to judge the performance of the three methods. The maximum relative error of the 4 predicted values by using parameter-transfer was 0.415% and the minimum relative error was 0.042%. The maximum and minimum relative errors by using ANN were 0.918% and 0.402%. The effects of SVM were the worst among all three methods, with maximum and minimum relative errors of 3.083% and 0.883%, respectively. The average relative errors of parameter-transfer, ANN, and SVM were 0.281%, 0.728%, and 1.753%, respectively. Obviously, the parameter-transfer method yielded the lowest average predictive error of all the methods tested. The data were plotted as shown in Figure 7 and the relative error of the three methods was plotted as shown in Figure 8 to display these differences clearly and intuitively.

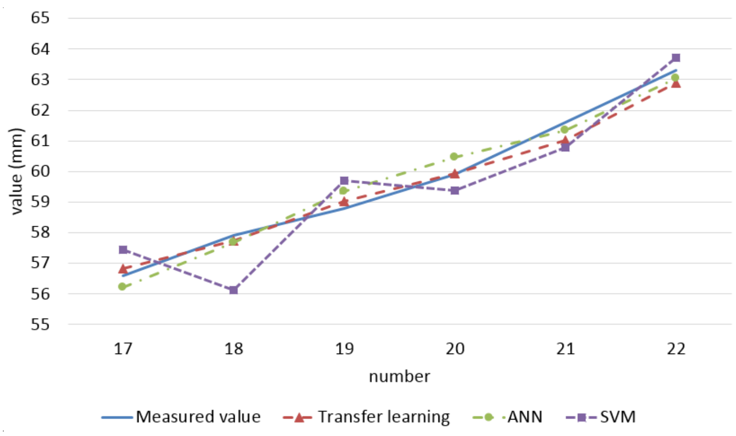


**Figure 7** Comparison of tunnel settlement prediction results.

The measured values are marked in blue, the transfer-learning predictions in red, the ANN predictions in green, and the SVM predictions in purple. The red curve and the blue curve almost coincide with each other, which illustrates the accuracy of the transfer-learning predictions. The green curve performance after the red curve means that the ANN predictions are not bad. The fluctuating range of the purple curve is higher, which indicates that the SVM predictions are relatively inaccurate.

In the bar graph below, the relative error of the transfer-learning model is shown in blue, that of ANN is in red, and that of SVM is in green. The parameter-transfer method got the best results as the total relative errors of parameter-transfer, ANN, and SVM were 1.123%, 2.912%, and 7.014%, respectively.

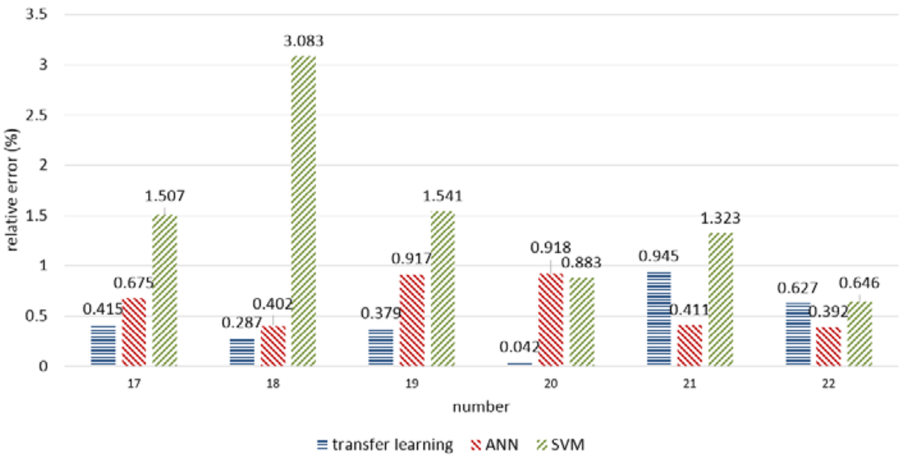


**Figure 8** Comparison of relative error among the three methods.

## 6 Conclusion

This paper presented a tunnel-settlement prediction method based on transfer learning. The use of a deep neural network trained via parameter transfer was found to provide highly accurate settlement predictions; the parameter-transfer method transfers knowledge gained from the source model to the task model in order to decrease the training complexity and reduce the training time. The target dataset was used to train the target model, which could then output predictions. We tested ANN and SVM for comparison and found that the transfer-learning method provided the most accurate tunnel-settlement predictions, with an average relative error of only 0.281%.

## Acknowledgements

The authors want to thank Dr. Liu for his contributions to this article. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

## References

[1] Wang, J.J. & Ma, Y.L., *Deep Learning for Smart Manufacturing: Methods and Applications*, Journal of Manufacturing Systems, **48**, pp. 144-156, 2018.

[2] Li, C., Sánchez, R.V., Zurita, G., Cerrada, M. & Cabrera, D., *Fault Diagnosis for Rotating Machinery Using Vibration Measurement Deep Statistical Feature Learning*, Sensors, **16**(6), Article No. 895, 2016.

[3] Obst, O., *Distributed Fault Detection in Sensor Networks Using a Recurrent Neural Network*, Neural Processing Letters, **40**, pp. 261-273, 2014.

[4] Wei, K., Gong, Q.M. & Zhou, S.H., *Ant Colony Algorithms of Long-Term Uneven Settlement Prediction in Tunnel*, Journal of Tongji University, **37**, pp. 993-998, 2009.

[5] Fan, S.X., Zhou, Q.C., Xiong, X.L. & Zhao, J., *Settlement Prediction of Tunnel Based on Multiple Kernels Learning Mode*, Rock and Soil Mechanics, **34**, pp. 291-298, 2013.

[6] Goh, A.T.C., Zhang, W.G., Zhang, Y.M., Xiao, Y. & Xiang Y.Z., *Determination of Earth Pressure Balance Tunnel-related Maximum Surface Settlement: A Multivariate Adaptive Regression Splines Approach*, Bulletin of Engineering Geology & the Environment, **77**, pp. 489-500, 2018.

[7] Maiedi, P., Akbulut, S. & Celik, S., *The Experimental, Analytical and Numerical Estimation of Ground Surface Settlement Caused by Tunnel Excavation*, 2nd International Conference on Advanced Engineering Technologies, pp. 21-23, 2017.

[8] Pan, W.K., *A Survey of Transfer Learning for Collaborative Recommendation with Auxiliary Data*, Neurocomputing, **177**, pp. 447-453, 2016.

[9] Pan, S.J. & Yang, Q., *A Survey on Transfer Learning*, IEEE Transactions on Knowledge & Data Engineering, **22**, pp. 1354-1359, 2010.

[10] Hochreiter, S. & Schmidhuber, J., *LSTM Can Solve Hard Long Time Lag Problems*, Advances in Neural Information Processing Systems, **9**, pp. 473-479, 2003.

[11] Cho, K., Merrienboer, B.V., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. & Bengio, Y. *Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation*, Proceedings of the

Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Association for Computational Linguistic, pp. 1724-1734, 2014.

[12] Galanti, T., Wolf, L. & Hazan, T., *A Theoretical Framework for Deep Transfer Learning*, Information & Inference A Journal of the IMA, **5**, pp. 159-209, 2016.

[13] Bai, Y. & Tang, J., *Experience and Lessons from Underground Construction in Shanghai*, China Civil Engineering Journal, **40**, pp. 105-110, 2007.

[14] Bengio, Y., *Deep Learning of Representations for Unsupervised and Transfer*, MLR: Workshop and Conference Proceedings. Workshop on Unsupervised and Transfer Learning, **27**, pp. 17-37, 2012.