

Article

A Novel Hybrid Transfer Learning Framework for Dynamic Cutterhead Torque Prediction of the Tunnel Boring Machine

Tao Fu, Tianci Zhang and Xueguan Song *

School of Mechanical Engineering, Dalian University of Technology, No.2 Linggong Road, Ganjingzi District, Dalian 116024, China; futao_cam@126.com (T.F.); tczhang@mail.dlut.edu.cn (T.Z.)

* Correspondence: sxg@dlut.edu.cn

Abstract: A tunnel boring machine (TBM) is an important large-scale engineering machine, which is widely applied in tunnel construction. Precise cutterhead torque prediction plays an essential role in the cost estimation of energy consumption and safety operation in the tunneling process, since it directly influences the adaptable adjustment of excavation parameters. Complicated and variable geological conditions, leading to operational and status parameters of the TBM, usually exhibit some spatio-temporally varying characteristic, which poses a serious challenge to conventional data-based methods for dynamic cutterhead torque prediction. In this study, a novel hybrid transfer learning framework, namely TRLS-SVR, is proposed to transfer knowledge from a historical dataset that may contain multiple working patterns and alleviate fresh data noise interference when addressing dynamic cutterhead torque prediction issues. Compared with conventional data-driven algorithms, TRLS-SVR considers long-ago historical data, and can effectively extract and leverage the public latent knowledge that is implied in historical datasets for current prediction. A collection of in situ TBM operation data from a tunnel project located in China is utilized to evaluate the performance of the proposed framework.



Citation: Fu, T.; Zhang, T.; Song, X. A Novel Hybrid Transfer Learning Framework for Dynamic Cutterhead Torque Prediction of the Tunnel Boring Machine. *Energies* **2022**, *15*, 2907. <https://doi.org/10.3390/en15082907>

Academic Editor: Manoj Khandelwal

Received: 14 March 2022

Accepted: 11 April 2022

Published: 15 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: tunnel boring machine (TBM); cutterhead torque prediction; operation parameters; transfer learning

1. Introduction

Tunnel boring machines (TBM) are widely applied in various tunnel construction projects, such as subways, mining ores, railways, etc., due to advantages of higher reliability, safety, and environmental friendliness [1]. Figure 1 illustrates a typical structure of the TBM, which contains multiple sub-systems, such as the cutterhead driving system, thrust system, cutterhead system, etc. In real-world applications, TBMs generally work in heterogeneous and complicated geological environments, such as spalling, faulting, fracturing, rock bursting, squeezing, swelling, and high water in the flow [2], that pose severe challenges to the operation of TBMs. A schematic illustration of the geological conditions of a tunnel is demonstrated in Figure 2. To ensure construction safety and reduce energy consumption, it is desirable to accurately predict the dynamic load (generally referring to the cutterhead torque) under spatio-temporally varying geological conditions and to dynamically adjust the TBM control parameters during excavation.

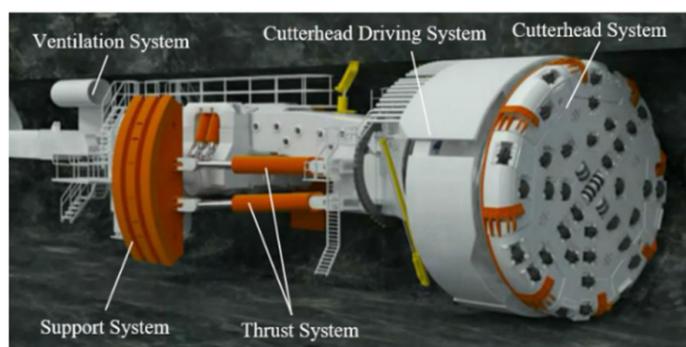


Figure 1. A typical diagram of the TBM. Reproduced with permission from [3], *Springer Nature*, 2016.

In general, the prediction methods for cutterhead torque can be roughly grouped into three types: rock–soil mechanics methods, empirical methods (combined with experiments), and soft computing methods. The rock–soil mechanics method establishes a model according to the force balance among rock, cutters, and internal machinery [4,5]. The empirical models are based on engineering experience involving a large amount of laboratory tests, field measurements, and construction records [6,7]. The soft computing methods are developed as data-based solutions for predicting the TBM’s load through mathematical mapping. Rostami [8] elaborated theoretical and empirical methods in a recent review. S. K. Shreyas [9] and Shahrour Isam [10] provided a brief retrospect of recent application of soft computing methods to predict various parameters in tunneling and underground excavations.

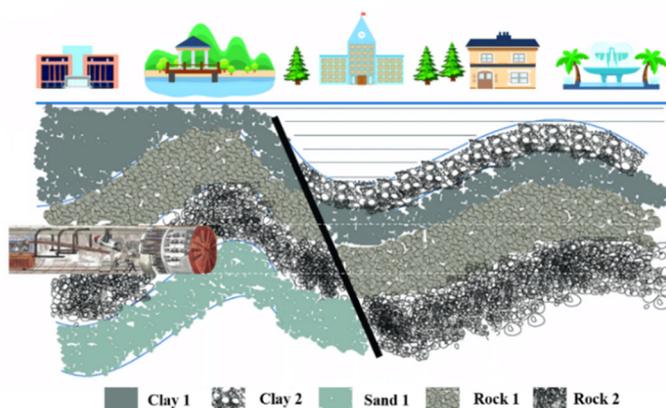


Figure 2. Longitudinal geological profile of a tunnel. Reproduced with permission from [11], *IEEE*, 2019.

By dividing the tunnel alignment into three general sections in terms of geological and geotechnical conditions, Avunduk et al. [12] proposed an empirical model for predicting excavation performance of TBM. Through the mechanical decoupling method for analyzing the cutterhead–ground interaction, Zhang et al. [13] proposed an approximate calculation method for determining the load acting on the cutterhead. Based on the interaction between the TBM and excavated material, Faramarzi et al. [14] applied the discrete element method (DEM) to evaluate the TBM torque and thrust. Rock–soil mechanics methods and empirical models are both based on the premise that the geological information is known. However, the accurate prediction of a geological profile before excavation is a hard and challenging task. In tunneling and underground excavation, the geological information is obtained through borehole sampling, and the stratum between sampling points are usually estimated by linear fitting. The distance between the sampling points is typically considerable, and the relevant result is often different from the real distribution, which may affect the accuracy of the rock–soil mechanics methods and the empirical models [15].

Assisted by the advancement of sensor and measurement technology, modern TBMs can record series of operation parameters closely related to dynamic load, which provides a basis for the practical application of soft computing methods. Sun et al. [16] utilized the random forest (RF) algorithm to design a predictor for TBM load. Kong et al. [17] took geological conditions and operational data as inputs to build a prediction model based on the RF for predicting driving forces of a TBM in a soil–rock, mixed-face ground. Li et al. [18] used the one-dimensional convolutional neural networks and long short-term memory network (CNN-LSTM) to predict cutterhead speed and penetration rate (PR). Qin et al. [19] applied a deep neural network-based method to predict dynamic cutterhead torque based on operating data and status parameters. Suwansawat et al. [20] applied the multi-layer perceptron (MLP) to determine the correlation among TBM operational data, groundmass characteristics, and surface movements. Lau et al. [21] used a radial basis function (RBF) to estimate tunneling production rates of successive cycles. Gao et al. [22] used three kinds of recurrent neural networks (RNNs) to deal with TBM operating parameters' real-time prediction. Soft calculation methods usually involve the optimization of many parameters, and the selection of parameters based on experience will reduce the accuracy of the analysis results. To deal with this problem, there have been many hybrid methods proposed in the literature. For example, Zhou et al. [23] applied three optimization algorithms to optima of the hyper-parameters of the support vector machine (SVM) technique in forecasting the advance rate (AR) of TBMs. Armaghani et al. [24,25] proposed two hybrid, intelligent systems, namely the particle swarm optimization (PSO)-artificial neural network (ANN) and the imperialism competitive algorithm (ICA)-ANN, to approximate the PR and AR of TBMs, respectively.

Although relatively accurate prediction results can be achieved by soft computing approaches, most of them generally assume that training samples and future test samples have identical distribution characteristics, and their practicability still has room for improvement. During the excavation process, TBMs encounter varying geological and working conditions, such as accelerating, turning, jamming releasing, etc., resulting in considerable changes in the underlying pattern of operation data over space and time. So, historical datasets behave as a non-stationary time series that makes the correlation among parameters in a high degree of complicated, changeable, and challenging conditions to be described by simple or fixed mathematical expressions. Hence, it is a serious challenge to extract common knowledge from historical datasets to assist in building an adaptive model which dynamically changes with geological conditions and operating parameters, for implementing dynamic cutterhead torque prediction at the current moment. To a certain degree, this problem is similar to the paradigm of transfer learning [26,27], which addresses this problem by utilizing experiences gained from source tasks to improve the learning of new related tasks. Hu et al. [28] applied the concept of transfer learning for efficient wind speed prediction. The prediction model was trained on samples from older data-rich farms to extract wind speed patterns, and then finely tuned with samples from newly built farms. Rui et al. [29] constructed a novel transfer learning paradigm for time series prediction, and the principle of transfer learning is employed. However, TBM's historical data contains a variety of geological information and working modes. So, directly adopting the most intuitive transfer learning method without distinguishing all the working modes in the historical data may result in negative transfer problems.

Herein, a novel hybrid data-mining framework based on clustering, multitask learning (MTL), transfer learning, and least-squares support vector regression machines (LS-SVR), abbreviated as TRLS-SVR, is proposed for dynamic cutterhead torque forecasting of TBMs. In this framework, LS-SVR is selected as a baseline model, which has a powerful capability to capture underlying nonlinear relationships for a complex system. The underlying patterns in historical data are effectively divided according to the relationship among attributes [30]. To take advantage of the knowledge contained in different working modes and to eliminate the damage from dataset bias, we adopt the idea of MTL [31], which explicitly exploits commonalities and differences across multiple working modes by learning them

simultaneously rather than individually, to improve knowledge extraction ability. Based on the common knowledge extracted from historical data, we utilize the newly collected operation data to continuously update the pattern-specific biases parameters for adapting to the changing geological and working conditions. This study offers the following innovations and contributions. (1) The unsupervised learning algorithm for data clustering is combined with the MTL paradigm to explore and exploit the correlations among multiple working modes by learning simultaneously rather than individually, which enhances the ability of extracting public knowledge from a diversely recorded TBM historical dataset. (2) It employs a transfer learning paradigm to reuse the public knowledge that is contained in the historical dataset to supply new data, and it alleviates random noise interference and fits the varying geological and working conditions well. (3) The TRLS-SVR performs superior performance in geologically complex and changeable locations, compared with that of conventional data-driven algorithms.

The rest of this study is organized as follows. Section 2 presents details of the proposed framework. In Section 3, the experimental verification is presented. In Section 4, some discussions on experimental results are provided. Section 5 concludes the whole study and provides future work.

2. The Proposed Dynamic Cutterhead Torque Prediction Framework

2.1. Overall Framework

The framework of dynamic cutterhead torque prediction proposed in this paper draws inspiration from various machine learning methods, including clustering, MTL, and transfer learning. The overall framework of the TRLS-SVR mainly consists of four components, namely data pre-processing, dividing of typical working modes based on unsupervised clustering algorithm, extracting implicit common knowledge by MTL algorithm, and knowledge reuse based on transfer learning, as described in Figure 3. In the first step, a large number of historical datum that have a long-time span with current sample are extracted from the database. In the second step, a clustering algorithm was used to effectively divide working modes in the historical dataset according to the relationship among attributes. Next, the MTL paradigm was used to exploit representative knowledge from multiple working modes. Based on the transfer learning paradigm, experiences extracted from the historical dataset were retained and utilized to train a fresh model. The detailed descriptions of each component are introduced below.

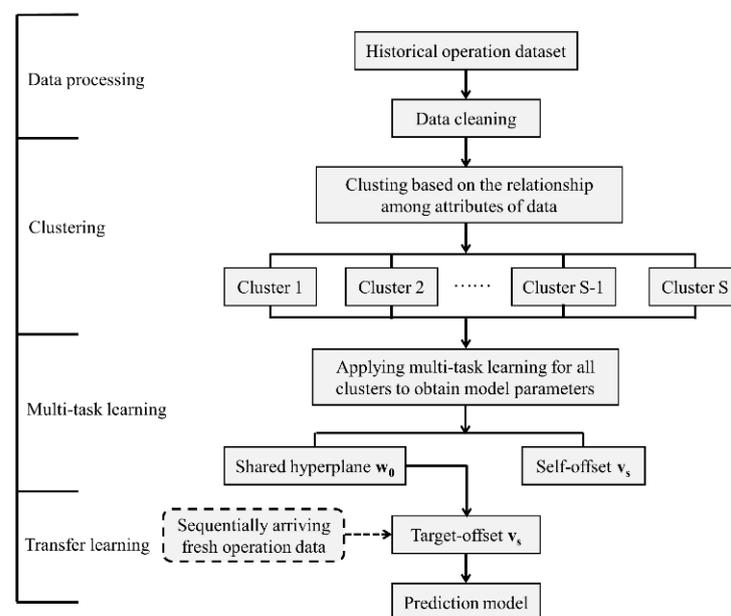


Figure 3. The proposed dynamic cutterhead torque prediction framework.

2.2. Clustering Based on the Relationship among Attributes

Due to continuous changes in geological conditions and work patterns, the historical dataset may contain multiple modes. In order to better extract public knowledge under different patterns, the first step is to divide the historical dataset into different clusters. Clustering as a pre-processing algorithm to uncover the underlying patterns and find natural partitioning within a dataset is widely utilized in engineering data analyses, such as fault detection, pattern recognition, and risk analysis. Currently, widely used clustering algorithms such as K-Nearest Neighbor (K-NN) and the fuzzy c -means algorithm (FCM) are mostly based on the spatial distribution to classify the dataset. However, the spatial distribution of different categories of TBM operation data is often similar, and conventional data clustering methods might not partition it effectively. The relationship among attributes varies considerably under different working and geological conditions, which can be used to improve clustering performance [32]. Thus, in this paper, we employ the modified FCM algorithm, namely, SVR-FCM, presented by Shi et al. [30] for TBM operation data clustering, which is designed under the architecture of FCM, but it partitions the data based on the relationship among attributes rather than their spatial distribution. The distance D_{ik} is defined as follows:

$$D_{ik} = (x_{obj,k} - SVR(x_{1,k}, \dots, x_{obj-1,k}, x_{obj+1,k}, \dots, x_{s,k}))_i^2 \quad (1)$$

The clustering objective function modified as follows:

$$J_{SVR-FCM} = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m (x_{obj,k} - SVR(x_{1,k}, \dots, x_{obj-1,k}, x_{obj+1,k}, \dots, x_{s,k}))_i^2 \quad (2)$$

The necessary conditions for minimizing (2) result in the following partition matrix:

$$u_{ik} = \left[\sum_{t=1}^c \left(\frac{(x_{obj,k} - SVR(x_{1,k}, \dots, x_{obj-1,k}, x_{obj+1,k}, \dots, x_{s,k}))_i^2}{(x_{obj,k} - SVR(x_{1,k}, \dots, x_{obj-1,k}, x_{obj+1,k}, \dots, x_{s,k}))_t^2} \right)^{\frac{1}{m-1}} \right]^{-1} \quad (3)$$

A more detailed description of the algorithm architecture can be seen in [22].

2.3. Extracting Public Knowledge from Historical Dataset

The clustering categories correspond to typical working modes, which are combined of representative working and geological conditions. It should be noted that the data distribution is distinct but similar in different working modes. To extract the public knowledge contained in typical working modes, we adopt the paradigm of MTL, which explicitly exploits commonalities and differences across multiple working modes by learning them simultaneously rather than individually to improve knowledge extraction ability. MTL reinforces each task by using the interconnections between tasks, considering both the relevance and the difference between tasks to enhance the generalization performance. There has been abundant literature on MTL, showing that learning various related tasks simultaneously can be advantageous in predictive performance relative to learning these tasks independently [33,34]. This study adopts the MTL method based on the minimization of the regularization function similar to LS-SVR, which has been successfully utilized for single-task learning [35]. The LS-SVR can be formulated as Equation (4), which solves the regression problem by optimizing the output weight vector, \mathbf{w} , and bias term, b , by minimizing a cost function with constraint, as shown in Equation (5).

$$\mathbf{y} = \mathbf{w}^T \varphi(\mathbf{x}) + b \cdot 1 \quad (4)$$

where $\varphi(\bullet)$ denotes a features map.

$$\begin{aligned} \min J(\mathbf{w}, \mathbf{e}) &= \frac{1}{2} \mathbf{w}^T \mathbf{w} + \rho \mathbf{e}^T \mathbf{e} \\ \text{s.t. } \mathbf{y} &= \mathbf{w}^T \varphi(\mathbf{x}) + b \cdot \mathbf{1} + \mathbf{e} \end{aligned} \tag{5}$$

\mathbf{e} is a vector consisting of slack variables, and the hyper-parameter ρ controls the relative weight of each term. Herein, the output weight vector of different working modes, noted as \mathbf{w}_s , can be divided into the common vector \mathbf{w}_0 , shared by all working modes and working-mode-specific bias vectors, \mathbf{v}_s , which can be formulated as follows:

$$\mathbf{w}_s = \mathbf{w}_0 + \mathbf{v}_s, \forall s \in S \tag{6}$$

We estimate all \mathbf{v}_s as well as the (common) \mathbf{w}_0 simultaneously. To this end, we solve the following optimization problem, which is analogous to the LS-SVR used for single-task learning:

$$\begin{aligned} \min_{w_0, v_s, \zeta_{s,i}, \rho_{s,i}} \left\{ J(\mathbf{w}_0, \mathbf{v}_s, \zeta_{s,i}, \rho_{s,i}) := \gamma \cdot \frac{1}{2} \sum_{s=1}^S \sum_{i=1}^{ns} \zeta_{s,i}^2 + \eta \cdot \frac{1}{2} \sum_{s=1}^S \sum_{i=1}^{ns} \rho_{s,i}^2 + \frac{1}{2} \cdot \frac{\lambda}{S} \sum_{s=1}^S \|\mathbf{v}_s\|^2 + \frac{1}{2} \|\mathbf{w}_0\|^2 \right\} \\ \text{s.t. } \phi(\mathbf{x}_{s,i})^T \cdot (\mathbf{w}_s) + b_s = y_{s,i} - \zeta_{s,i} \\ \phi(\mathbf{x}_{s,i})^T \cdot \mathbf{w}_0 + b_{s_0} = y_{s,i} - \rho_{s,i} \end{aligned} \tag{7}$$

The number of tasks is S , which is equal to the number of clustering results. Specifically, $\mathbf{x}_{s,i}$ represents the i th sample of the s th task, λ is the constraint coefficient, γ and η are penalty coefficient, and $\zeta_{s,i}$ and $\rho_{s,i}$ represent the training error vector of the s th task. According to the Lagrangian multiplier method, to solve Equation (7) is equivalent to solving the corresponding Lagrangian problem:

$$\begin{aligned} L_D = \frac{1}{2} \|\mathbf{w}_0\|^2 + \frac{1}{2} \cdot \frac{\lambda}{S} \sum_{s=1}^S \|\mathbf{v}_s\|^2 + \gamma \cdot \frac{1}{2} \sum_{s=1}^S \sum_{i=1}^{ns} \zeta_{s,i}^2 + \eta \cdot \frac{1}{2} \sum_{s=1}^S \sum_{i=1}^{ns} \rho_{s,i}^2 - \\ \sum_{s=1}^S \sum_{i=1}^{ns} \alpha_{s,i} \times \left\{ (w_0 + v_s)^T \cdot \phi(\mathbf{x}_{s,i}) + b_s + \zeta_{s,i} - y_{s,i} \right\} - \sum_{s=1}^S \sum_{i=1}^{ns} \beta_{s,i} \times \left\{ w_0^T \cdot \phi(\mathbf{x}_{s,i}) + b_{s_0} + \rho_{s,i} - y_{s,i} \right\} \end{aligned} \tag{8}$$

where $\alpha_{s,i}$ and $\beta_{s,i}$ are the i th Lagrangian multiplier for the s th task. Based on the Karush–Kuhn–Tucker (KKT) conditions, setting the first partial derivatives of L_D to zero,

$$\left\{ \begin{aligned} \frac{\partial L_D}{\partial \mathbf{w}_0} = 0 &\Rightarrow \mathbf{w}_0 = \sum_{s=1}^S \sum_{i=1}^{ns} (\alpha_{s,i} + \beta_{s,i}) \cdot \phi(\mathbf{x}_{s,i}) \\ \frac{\partial L_D}{\partial \mathbf{v}_s} = 0 &\Rightarrow \mathbf{v}_s = \frac{S}{\lambda} \sum_{i=1}^{ns} \alpha_{s,i} \cdot \phi(\mathbf{x}_{s,i}), \forall s \in S \\ \frac{\partial L_D}{\partial b_s} = 0 &\Rightarrow \sum_{i=1}^{ns} \alpha_{s,i} = 0, \forall s \in S \\ \frac{\partial L_D}{\partial b_{s_0}} = 0 &\Rightarrow \sum_{s=1}^S \sum_{i=1}^{ns} \beta_{s,i} = 0 \\ \frac{\partial L_D}{\partial \zeta_{s,i}} = 0 &\Rightarrow \alpha_{s,i} = \gamma \cdot \zeta_{s,i}, \forall s \in S \\ \frac{\partial L_D}{\partial \rho_{s,i}} = 0 &\Rightarrow \beta_{s,i} = \eta \cdot \rho_{s,i}, \forall s \in S \\ \frac{\partial L_D}{\partial \alpha_{s,i}} = 0 &\Rightarrow (\mathbf{w}_0 + \mathbf{v}_s)^T \cdot \phi(\mathbf{x}_{s,i}) + b_s + \zeta_{s,i} - y_{s,i} = 0, \forall s \in S \\ \frac{\partial L_D}{\partial \beta_{s,i}} = 0 &\Rightarrow \mathbf{w}_0^T \cdot \phi(\mathbf{x}_{s,i}) + b_{s_0} + \rho_{s,i} - y_{s,i} = 0, \forall s \in S \end{aligned} \right. \tag{9}$$

Eliminating \mathbf{w}_0 , $\{\mathbf{v}_i\}_{i=1}^S$, $\{\zeta_{s,i}\}_{s=1, i=1}^{S, n_i}$, and $\{\rho_{s,i}\}_{s=1, i=1}^{S, n_i}$ results in the solution of (9), being $\boldsymbol{\alpha}^* = (\boldsymbol{\alpha}_1^{*T}, \boldsymbol{\alpha}_2^{*T}, \dots, \boldsymbol{\alpha}_S^{*T})^T$ and $\boldsymbol{\beta}^* = (\boldsymbol{\beta}_1^{*T}, \boldsymbol{\beta}_2^{*T}, \dots, \boldsymbol{\beta}_S^{*T})^T$, where

$\alpha_s^* = (\alpha_{s,1}^*, \alpha_{s,2}^*, \dots, \alpha_{s,ns}^*)^T$ and $\beta_s^* = (\beta_{s,1}^*, \beta_{s,2}^*, \dots, \beta_{s,ns}^*)^T$. The working mode-specific bias vectors can be mathematically formulated as follows:

$$\mathbf{v}_s = \frac{S}{\lambda} \sum_{i=1}^{ns} \alpha_{s,i}^* \cdot \phi(\mathbf{x}_{s,i}), \forall s \in S \quad (10)$$

The extracted public knowledge is denoted as the following vector:

$$\mathbf{w}_0 = \sum_{s=1}^S \sum_{i=1}^{ns} (\alpha_{s,i}^* + \beta_{s,i}^*) \cdot \phi(\mathbf{x}_{s,i}) \quad (11)$$

2.4. Dynamic Cutterhead Torque Prediction Based on Transfer Learning

Transfer learning is an emerging framework that aims to provide a paradigm to utilize previously acquired experience to solve new but similar problems faster and more effectively [33]. There are some commonalities and associations between transfer learning and MTL. Both of them aim to improve the performance of learners via knowledge transfer. Transfer learning has been studied extensively for different applications in recent years, providing many opportunities for applying data-based methods to assist in design and analysis of complex engineering systems.

During the excavation process, geological information and operating parameters generally change continuously, so operation data around the excavation point have more reference significance for subsequent dynamic cutterhead torque prediction. In addition, vibration and shock often occur during excavation, and random noise interference inevitably exists in the measurement of fresh data, which may have a substantial impact on the prediction performance. Hence, training a new model by utilizing the knowledge contained in the historical dataset to reduce the requirement of number of new samples and alleviate the interference of random noise is always considered advisable. To leverage experiences extracted from the historical dataset, the output weight vector of the fresh model, noted as \mathbf{w}_t , is feasible to minimize the difference with the public vector, \mathbf{w}_0 , that can be regarded as the public knowledge transferred from the historical dataset. We intend to train an approximator which has the minimal norm parameter vector and training errors for available fresh samples, that can be written as,

$$\begin{aligned} \min L &= \frac{1}{2} \|\mathbf{w}_t\|^2 + \frac{1}{2} \mu \|\mathbf{w}_t - \mathbf{w}_0\|^2 + \frac{C}{2} \sum_{j=1}^{m_t} \tilde{\zeta}_j^2 \\ \text{s.t. } &\phi(\mathbf{x}_j)^T \mathbf{w}_t + b_t = y_j - \tilde{\zeta}_j \end{aligned} \quad (12)$$

where \mathbf{w}_t is the output weight vector over the fresh data, μ denotes the penalty parameter, C is the regularization parameter, $\tilde{\zeta}_j$ is the training error, and m_t is number of fresh training sets around the excavation point. According to Lagrangian multiplier method, to solve Equation (12) is equivalent to solving the corresponding Lagrangian problem:

$$L_D = \frac{1}{2} \|\mathbf{w}_t\|^2 + \frac{1}{2} \mu \|\mathbf{w}_t - \mathbf{w}_0\|^2 + \frac{C}{2} \sum_{j=1}^{m_t} \tilde{\zeta}_j^2 - \sum_{j=1}^{m_t} \alpha_j (\phi(\mathbf{x}_j)^T \mathbf{w}_t + b_t - y_j + \tilde{\zeta}_j) \quad (13)$$

where α_j is the j th Lagrangian multiplier, and based on the KKT conditions, the problem can be solved with the Lagrangian multiplier method,

$$\begin{cases} \frac{\partial L_D}{\partial \mathbf{w}_t} = 0 \Rightarrow \mathbf{w}_t = \frac{1}{1+\mu} \left(\mu \mathbf{w}_0 + \sum_{j=1}^{m_t} \alpha_j \phi(\mathbf{x}_j) \right) \\ \frac{\partial L_D}{\partial \tilde{\zeta}_j} = 0 \Rightarrow \alpha_j = C \tilde{\zeta}_j \\ \frac{\partial L_D}{\partial b_t} = 0 \Rightarrow \sum_{j=1}^{m_t} \alpha_j = 0 \\ \frac{\partial L_D}{\partial \alpha_j} = 0 \Rightarrow \phi(\mathbf{x}_j)^T \cdot \mathbf{w}_t + b_t - y_j + \tilde{\zeta}_j = 0 \end{cases} \quad (14)$$

On analysis of Equation (14), it can be concluded that:

$$\begin{cases} \frac{1}{1+\mu} \left(\phi(\mathbf{x}_j)^T \cdot \mu \cdot w_0 + \sum_{k=1}^{m_t} \alpha_k \phi(\mathbf{x}_k)^T \cdot \phi(\mathbf{x}_j) \right) + b_t - y_j + \frac{\alpha_j}{C} = 0 \\ \sum_{j=1}^{m_t} \alpha_j = 0 \end{cases} \tag{15}$$

Plugging Equations (12) and (13) into Equation (15) can we obtain:

$$\begin{aligned} & \frac{1}{1+\mu} \left(\mu \sum_{s=1}^S \sum_{i=1}^{ns} (\alpha_{s,i} + \beta_{s,i}) \cdot \phi(\mathbf{x}_{s,i}) \cdot \phi(\mathbf{x}_j^T) + \sum_{k=1}^{m_t} \alpha_k \phi(\mathbf{x}_k) \cdot \phi(\mathbf{x}_j^T) \right) + b_t - y_j + \frac{\alpha_j}{C} = 0 \\ \Rightarrow & \left(\frac{1}{1+\mu} \begin{bmatrix} \phi(\mathbf{x}_1) \cdot \phi(\mathbf{x}_1^T) & \cdots & \phi(\mathbf{x}_1) \cdot \phi(\mathbf{x}_{m_t}^T) \\ \vdots & \ddots & \vdots \\ \phi(\mathbf{x}_{m_t}) \cdot \phi(\mathbf{x}_1^T) & \cdots & \phi(\mathbf{x}_{m_t}) \cdot \phi(\mathbf{x}_{m_t}^T) \end{bmatrix} + \frac{1}{C} \right) \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{m_t} \end{bmatrix} + b_t = \begin{bmatrix} \frac{-\mu}{1+\mu} \cdot \sum_{s=1}^S \sum_{i=1}^{ns} (\alpha_{s,i} + \beta_{s,i}) \cdot \phi(\mathbf{x}_{s,i}) \cdot \phi(\mathbf{x}_1^T) + y_1 \\ \frac{-\mu}{1+\mu} \cdot \sum_{s=1}^S \sum_{i=1}^{ns} (\alpha_{s,i} + \beta_{s,i}) \cdot \phi(\mathbf{x}_{s,i}) \cdot \phi(\mathbf{x}_2^T) + y_2 \\ \vdots \\ \frac{-\mu}{1+\mu} \cdot \sum_{s=1}^S \sum_{i=1}^{ns} (\alpha_{s,i} + \beta_{s,i}) \cdot \phi(\mathbf{x}_{s,i}) \cdot \phi(\mathbf{x}_{m_t}^T) + y_{m_t} \end{bmatrix} \tag{16} \\ \Rightarrow & \left(\frac{1}{1+\mu} \begin{bmatrix} \phi(\mathbf{x}_1) \cdot \phi(\mathbf{x}_1^T) & \cdots & \phi(\mathbf{x}_1) \cdot \phi(\mathbf{x}_{m_t}^T) & 1+\mu \\ \vdots & \ddots & \vdots & \vdots \\ \phi(\mathbf{x}_{m_t}) \cdot \phi(\mathbf{x}_1^T) & \cdots & \phi(\mathbf{x}_{m_t}) \cdot \phi(\mathbf{x}_{m_t}^T) & 1+\mu \\ 1 & \cdots & 1 & \frac{-1-\mu}{C} \end{bmatrix} + \frac{1}{C} \right) \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{m_t} \\ b_t \end{bmatrix} = \begin{bmatrix} \frac{-\mu}{1+\mu} \cdot \sum_{s=1}^S \sum_{i=1}^{ns} (\alpha_{s,i} + \beta_{s,i}) \cdot \phi(\mathbf{x}_{s,i}) \cdot \phi(\mathbf{x}_1^T) + y_1 \\ \frac{-\mu}{1+\mu} \cdot \sum_{s=1}^S \sum_{i=1}^{ns} (\alpha_{s,i} + \beta_{s,i}) \cdot \phi(\mathbf{x}_{s,i}) \cdot \phi(\mathbf{x}_2^T) + y_2 \\ \vdots \\ \frac{-\mu}{1+\mu} \cdot \sum_{s=1}^S \sum_{i=1}^{ns} (\alpha_{s,i} + \beta_{s,i}) \cdot \phi(\mathbf{x}_{s,i}) \cdot \phi(\mathbf{x}_{m_t}^T) + y_{m_t} \\ 0 \end{bmatrix} \end{aligned}$$

Let the solution of (16) be $\alpha_t^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_{m_t}^*)^T$ and b_t^* . In addition, the dynamic cutterhead torque prediction of fresh data can be mathematically formulated as follows:

$$\begin{aligned} f_t(\mathbf{x}) &= \phi(\mathbf{x})^T \mathbf{w}_t^* + b_t^* \\ &= \frac{\mu}{1+\mu} \sum_{s=1}^S \sum_{i=1}^{ns} (\alpha_{s,i}^* + \beta_{s,i}^*) \phi(\mathbf{x}_{s,i}) \phi(\mathbf{x}^T) + \frac{1}{1+\mu} \sum_{k=1}^{m_t} \alpha_k^* \phi(\mathbf{x}_k) \phi(\mathbf{x}^T) + b_t^* \end{aligned} \tag{17}$$

3. Numerical Experiments

In this section, a collection of real-world operational and status parameters of TBM is utilized to demonstrate the superiority and applicability of the framework.

3.1. Experimental Settings

The tunnel project studied in this study is located in Shenzhen, China, which is about 2000 m long and 6.4 m in diameter. As described in Figure 4a, from the ground surface to the tunnel floor, various geological layers, such as clay, sand, and rock, are unevenly distributed. The tunneling equipment used in this tunnel is shown in Figure 4b, and has an earth pressure balance shield TBM with 500 T of total mass and 120 knives on its cutter head. The basic equipment parameters are listed in Table 1. During the tunneling process, the operational and state data of the TBM were recorded by a PLC, which was further read by an industrial computer at regular intervals and stored in the database. Thus, the fresh data in the database were added in batches during the tunneling process. The collected operation dataset represents the operational information and status parameters along the length of the tunnel, which contains about 44 attributes, such as cutterhead torque, chamber pressure, and advance velocity, etc. Please refer to the appendix for a detailed list of these attributes (see Table A1). In the process of dynamic cutterhead torque prediction, data come in batches. We selected five sets of sequence data to construct the test datasets, covering various working and geological conditions. Each collection of data contained approximately 5000 rows and 44 columns; the first 80% of the dataset were used as training samples and the last 20% were used as test samples. Each row of data represents the data of all physical quantities at a certain moment, and each column of data represents the data of a physical quantity at any moment.

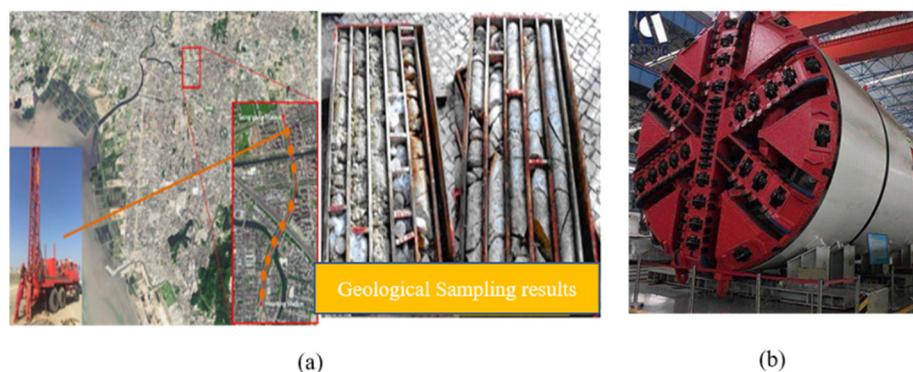


Figure 4. Geological sampling results and the TBM used. (a) Geological sampling results. (b) The TBM used.

Table 1. Basic parameters of the TBM used.

Parameters	Value	Unit
Cutterhead diameter	6680	mm
Maximum torque	8322	kNm
Rated power of drive motor	160	kW
Number of drive motors	8	1

To improve prediction accuracy, in this paper, we first normalized the samples with a normalization method, which is an essential pre-processing step in the field of machine learning. It is commonly referred to simply as “normalization,” or sometimes as “feature scaling,” and can be formulated as:

$$\min - \max = \frac{x - X_{\min}}{X_{\max} - X_{\min}} \quad (18)$$

where x is the current value and X_{\min} and X_{\max} are the minimum and maximum values of the entire dataset, respectively. The min–max method rescales values and confines samples to an interval between 0 and 1.

The operational data modeling was conducted with a personal computer (CPU: Intel Core i7-10700; RAM: 32 G). The framework was coded by the author with Matlab and set as follows: the clustering algorithm parameter set refers to the setting of references [30], where the fuzzification parameter, m , was 2, threshold value, ε , was 10^{-6} , number of clusters was 4, and maximum iteration was 1000. The radial basis function (RBF) was selected as a kernel function for LS-SVR. Compared with ordinary LS-SVR models, the framework proposed in this paper has more hyper-parameters, such as η , λ , and μ , that determine the information extracted from historical data and knowledge transferred for constructing a new model. In this section, we set $\eta = 100$, $\lambda = 1$, and μ was determined according to the forecast accuracy of the previous batch, varying with the value of μ as {1, 5, 10, 15}. Other hyper-parameters were set to the same values with the baseline model LS-SVR, i.e., $\gamma = C = 100$.

3.2. Experiments and Results

To verify the efficacy and superiority, the performance of TRLS-SVR was compared to that of existing data-driven methods, such as RF, SVR, Lasso, and deep neural networks, i.e., long short-term memory (LSTM) networks [22] and online learning methods (i.e., online support vector regression (OSVR) [36]). The fitness of these prediction models was evaluated with four error criteria, i.e., the coefficient of determination (R^2), mean absolute

error (*MAE*), root mean square error (*RMSE*), and mean absolute percentage error (*MAPE*). These metrics have the following formula.

$$R^2 = \frac{\sum_{i=1}^n (y'_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (19)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - y'_i| \quad (20)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2} \quad (21)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - y'_i|}{y_i} \times 100\% \quad (22)$$

where n is total number of samples, \bar{y}_i is the average of all actual values, and y'_i is the predicted value of y_i . The closer R^2 is to 1, the better the performance is. The *MAE* and *RMSE* measure the disparity between actual values and predicted values, which reflects the dispersion of models. The *RMSE* is more sensitive to large errors than *MAE* because the errors are squared, and the large errors are amplified further. *MAPE* is the ratio between errors and actual values. It can be considered as a relative error function, and the smaller the value, the higher the prediction accuracy. These four error criterions can be applied to evaluate the fitness of these prediction models from various viewpoints.

The evaluation results of the proposed TRLS-SVR and other five data-driven models on the five test datasets are shown in Tables 2–5. In general, the results show that three indicators of TRLS-SVR, i.e., *MAE*, *RMSE*, and *MAPE*, are lower than the other five data-driven models, and the coefficient of determination, R^2 , is higher than others. The average value of R^2 , *MAE*, *RMSE*, and *MAPE* in the five datasets are 0.83, 63.34, 95.11, and 3.61% for the proposed TRLS-SVR; −3.03, 321.60, 385.72, and 16.38% for RF; −0.391, 199.86, 235.22, and 10.58% for LSTM; −0.48, 194.75, 236.27, and 10.52% for SVR; −0.146, 160.55, 211.86, and 8.89% for Lasso; 0.56, 88.97, 148.40, and 5.28% for OSVR, respectively. Hence, the average *MAE* of TRLS-SVR is 80.3% less than RF; 68.31% less than LSTM; 67.48% less than SVR; 60.55% less than Lasso; 28.8% less than OSVR, respectively. In addition, the average *RMSE* of TRLS-SVR is 75.34% less than RF; 59.56% less than LSTM; 59.74% less than SVR; 55.11% less than Lasso; 35.91% less than OSVR, respectively. Moreover, the prediction precision of TRLS-SVR is 77.95% higher than RF; 65.85% higher than LSTM; 65.67% high than SVR; 59.35% higher than Lasso; 31.61% higher than OSVR, respectively.

Table 2. R^2 of different methods in five datasets.

Datasets	RF	LSTM	SVR	Lasso	OSVR	TRLS-SVR
1	0.43	0.68	0.73	0.55	0.58	0.83
2	−9.86	−1.60	0.31	−1.33	0.55	0.85
3	−2.10	−1.21	−3.27	−0.84	0.46	0.85
4	−0.43	0.56	0.46	0.74	0.77	0.85
5	−3.17	−0.39	−0.64	0.15	0.45	0.77

Table 3. *MAE* of different methods in five datasets.

Datasets	RF	LSTM	SVR	Lasso	OSVR	TRLS-SVR
1	172.49	123.69	116.54	135.93	98.61	81.37
2	493.13	231.81	108.57	213.93	71.23	38.00
3	264.91	218.95	298.31	186.23	77.73	49.12
4	215.99	152.91	163.10	93.97	75.85	58.91
5	461.49	271.95	287.26	172.69	121.41	89.31

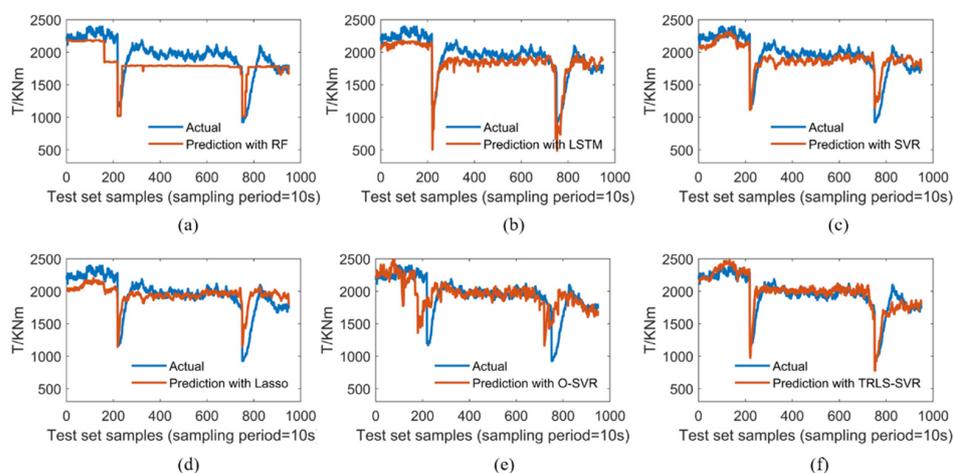
Table 4. RMSE of different methods in five datasets.

Datasets	RF	LSTM	SVR	Lasso	OSVR	TRLS-SVR
1	205.35	153.85	143.10	183.64	176.36	111.59
2	549.17	268.61	138.01	254.49	111.35	65.46
3	309.10	260.87	362.76	238.04	128.90	67.80
4	311.40	173.49	190.99	133.39	124.72	101.36
5	553.55	319.28	346.48	249.76	200.65	129.36

Table 5. MAPE of different methods in five datasets.

Datasets	RF	LSTM	SVR	Lasso	OSVR	TRLS-SVR
1	9.18%	6.53%	6.51%	7.85%	6.31%	4.45%
2	22.19%	10.45%	5.04%	9.69%	3.38%	1.82%
3	15.89%	14.05%	18.18%	11.58%	5.14%	3.09%
4	13.56%	10.53%	10.64%	7.26%	5.78%	4.65%
5	21.09%	11.33%	12.22%	8.05%	5.79%	4.04%

For visual comparison, the real cutterhead torque values and predicted cutterhead torque values with these models are also provided in Figures 5–9. It can be observed that the prediction accuracy of existing data-driven models, i.e., RF, LSTM, SVR, and Lasso, is relatively low, and can only predict the average value and changing trend but cannot achieve prediction dynamically and accurately. The main reason may lie in that the cutterhead torque sequence is nonlinear and non-stationary, and it may contain several different working conditions simultaneously. Therefore, it is not advisable to describe the cutterhead torque sequence data by a simple or fixed mathematical formula. The in situ monitoring data are spatio-temporally coupled, and the data close to the excavation point have more reference significance for subsequent load prediction. Using these fresh data to update the model parameters dynamically can capture the load data sequence's changing trend with the geological parameters and the working parameters. Therefore, online learning-based methods' prediction accuracy is higher than traditional statistical data-driven models. In addition, in spite of online learning-based methods, OSVR has high prediction accuracy in some samples; its accuracy is still less than TRLS-SVR on the entire dataset, mainly because there is random noise interference in the measurement of cutterhead torque data. Only using a small amount of fresh data that are close to the excavation point to update model parameters will inevitably overfit random noises and introduce model bias, which leads to performance degradation.

**Figure 5.** Comparisons between real and predicted cutterhead torque for dataset 1. (a) Prediction result of RF. (b) Prediction result of LSTM. (c) Prediction result of SVR. (d) Prediction result of Lasso. (e) Prediction result of OSVR. (f) Prediction result of TRLS-SVR.

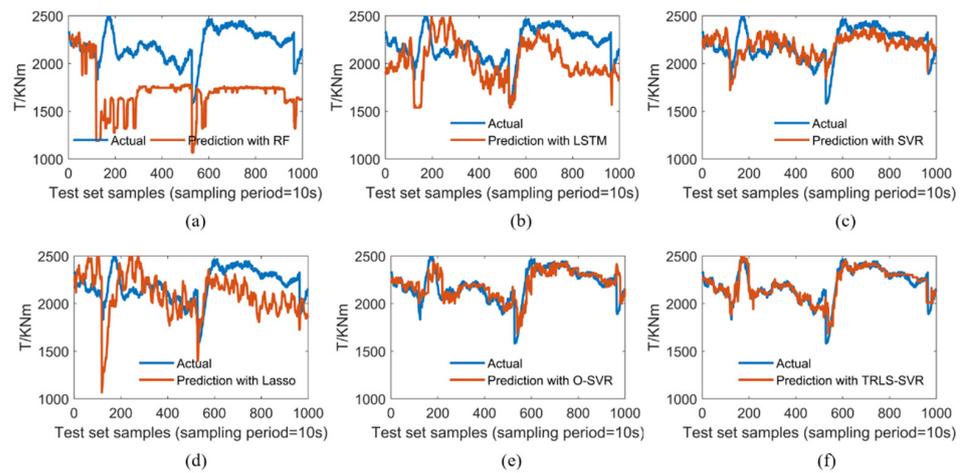


Figure 6. Comparisons between real and predicted cutterhead torque for dataset 2. (a) Prediction result of RF. (b) Prediction result of LSTM. (c) Prediction result of SVR. (d) Prediction result of Lasso. (e) Prediction result of OSVR. (f) Prediction result of TRLS-SVR.

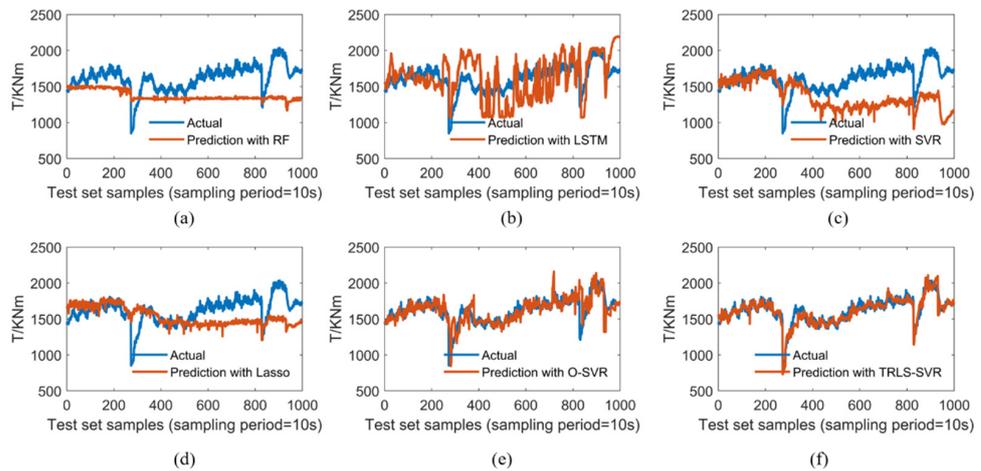


Figure 7. Comparisons between real and predicted cutterhead torque for dataset 3. (a) Prediction result of RF. (b) Prediction result of LSTM. (c) Prediction result of SVR. (d) Prediction result of Lasso. (e) Prediction result of OSVR. (f) Prediction result of TRLS-SVR.

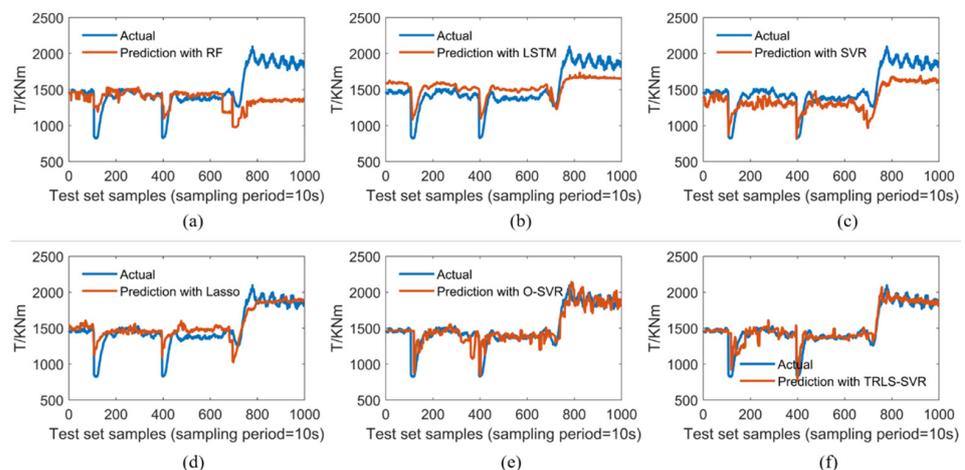


Figure 8. Comparisons between real and predicted cutterhead torque for dataset 4. (a) Prediction result of RF. (b) Prediction result of LSTM. (c) Prediction result of SVR. (d) Prediction result of Lasso. (e) Prediction result of OSVR. (f) Prediction result of TRLS-SVR.

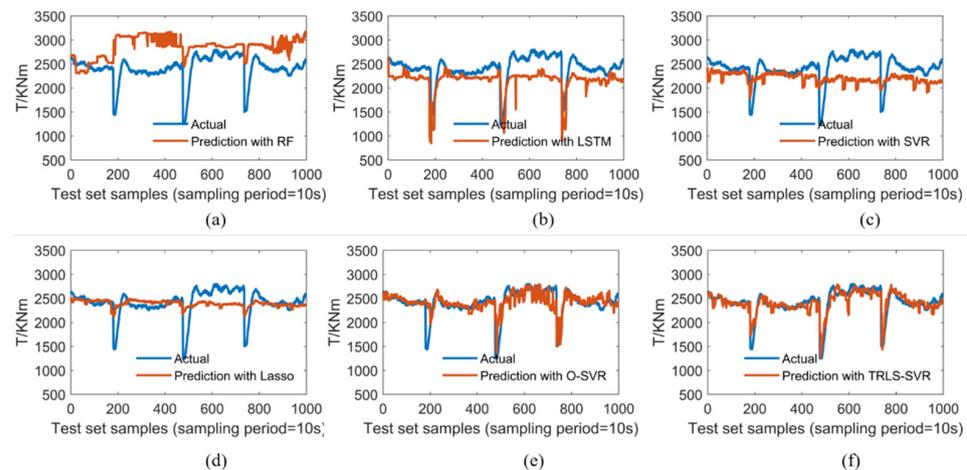


Figure 9. Comparisons between real and predicted cutterhead torque for dataset 5. (a) Prediction result of RF. (b) Prediction result of LSTM. (c) Prediction result of SVR. (d) Prediction result of Lasso. (e) Prediction result of OSVR. (f) Prediction result of TRLS-SVR.

The TRLS-SVR can effectively divide different working and geological conditions of historical data, and learn the cutterhead torque sequence's changing rule under different working modes. When the new coming data are disturbed by random noises or the excavation section's geological conditions, the implicit knowledge contained in historical data is explicitly transferred to reduce over-fitting of random noise, and to avoid introducing model bias. As a result, the proposed TRLS-SVR can achieve better prediction performance than that of existing data-driven methods.

4. Discussion

Compared with those of the baseline data-driven method, LS-SVR, the TRLS-SVR has more hyper-parameters, for example, η , λ , μ , and the number of fresh training sets, m_t . These hyper-parameters determine the amount of information extracted from historical data and the proportion of this information in the model update, which may affect the performance of the algorithm. As mentioned in Section 3.1, regularization parameter μ is determined according to the prediction accuracy of the previous batch. In this section, we focus on how the hyper-parameters η , μ , and number of fresh training sets, m_t , influence the prediction accuracy of the TRLS-SVR framework.

4.1. Analysis of the Number of Fresh Training Sizes

In these experiments, we select 10, 20, 50, 100, 200, 300, and 400 of the datum which are near the excavation point as fresh training sets. The prediction accuracy of the different number of fresh training sets is compared in Figure 10. It can be seen that when the number of training sizes, m_t , is small, the performance of TRLS-SVR improves faster as the number of samples increases, and when the number of training size, m_t , is relatively high, the performance decreases as the number of samples increases. When the number of training sizes, m_t , is 50, the proposed framework tends to provide the best prediction performance. This is because too little training data cannot reduce the interference of noise, which will lead to over-fitting of the noise and affect the prediction accuracy, while too much training data will smooth the changing characteristics of the continuous data to obtain average statistical characteristics and reduce the prediction accuracy.

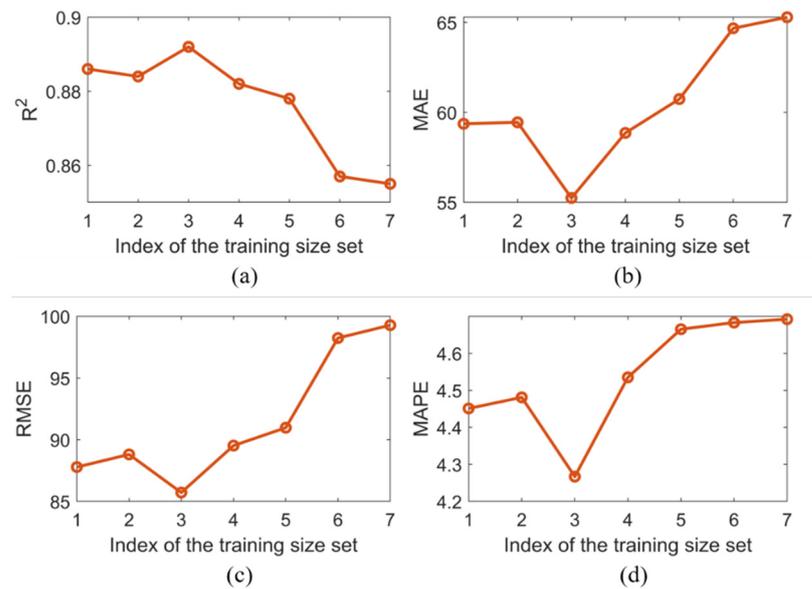


Figure 10. Sensitivity analysis on training size. (a) R^2 . (b) MAE. (c) RMSE. (d) MAPE.

4.2. Analysis of Regularization Parameters

We conduct experiments on the TBM dataset to discuss the sensitivity of the two regularization parameters η and λ . We fix the number of fresh training sets, m_t , as 50, hyper-parameters as $C = \gamma = 100$, and regularization parameter μ is determined according to the prediction accuracy of the previous batch. For the sensitivity analysis of the regularization parameter, η , we fix $\lambda = 1$ and vary the value of η as $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100, 200, 500, 1000, 2000\}$. For the sensitivity analysis of the regularization parameter, λ , we fix $\eta = 100$ and vary the value of λ as $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100, 1000\}$. The prediction accuracy of different values of regularization parameters are compared in Figures 11 and 12. In Figure 11, it can be seen that the optimal prediction accuracy by TRLS-SVR is achieved by setting $\eta = 100$ when $\lambda = 1$ is fixed. From Figure 12, it can be seen that the optimal prediction accuracy by TRLS-SVR is achieved by setting the value of λ as a small value. In addition, the prediction accuracy of TRLS-SVR changes slightly when the value of λ is in the range of $[10^{-5}, 1]$.

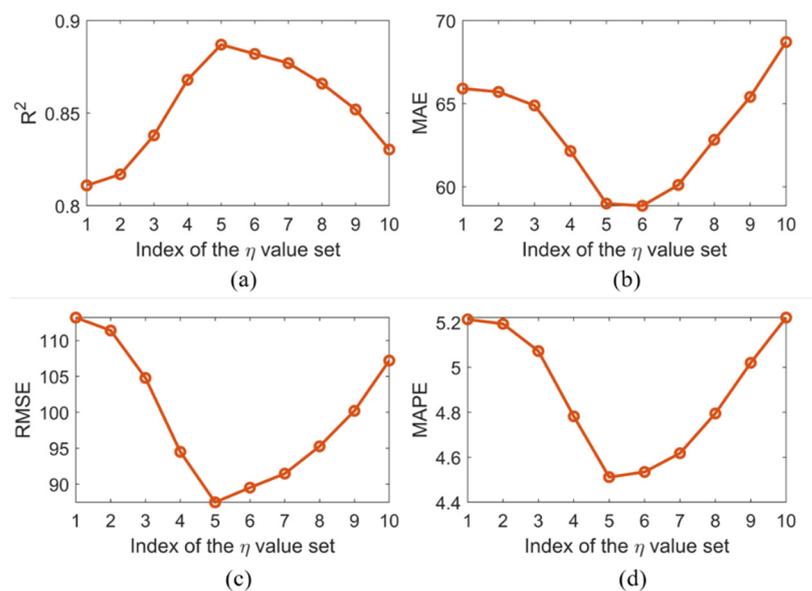


Figure 11. Sensitivity analysis on the regularization parameter, η . (a) R^2 . (b) MAE. (c) RMSE. (d) MAPE.

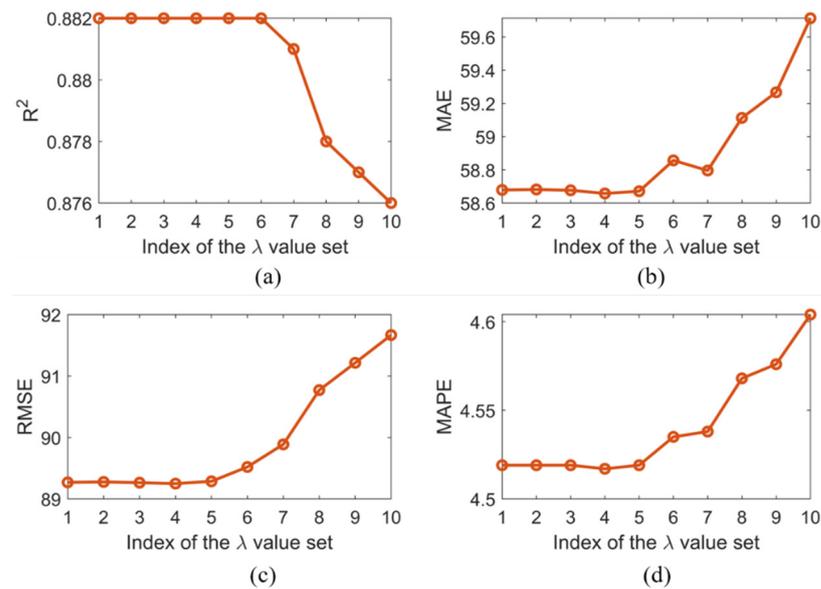


Figure 12. Sensitivity analysis on the regularization parameter, λ . (a) R^2 . (b) MAE. (c) RMSE. (d) MAPE.

4.3. Limitations and Recommendations

The heterogeneous in situ data of the TBM include not only numerical data but also categorical data, such as the geological data. The heterogeneous in situ data have one special characteristic that is different for the sizes of the geological data and the operation data, which limits the application of data-driven techniques on them. Thus, in this paper, we only consider the operational data and ignore the geological data. In the future, to further improve the prediction accuracy of the framework, it is necessary to integrate geological data through multi-source heterogeneous data fusion.

5. Conclusions

In this study, a novel hybrid transfer learning framework named TRLS-SVR, that aims to enhance the accuracy of TBM dynamic cutterhead torque prediction, is proposed. In the proposed framework, the underlying patterns in historical datasets were effectively divided according to the relationship among attributes. The idea of MTL was adopted to exploit commonalities and differences across various working modes by learning them simultaneously rather than individually, to capture the public knowledge from historical datasets. In order to cope with the changing geological and working conditions, the idea of transfer learning was adopted and the newly collected operation data were utilized to continuously update the parameters of the forecasting model as a supplement. Real-world, in situ operational and status parameters from a tunnel located in Shenzhen, China, were utilized to evaluate the efficacy and superiority of the proposed framework. Experimental results demonstrated that the TRLS-SVR alleviated the shortcoming of traditional statistical data-driven methods, which can only predict the average value and changing trend of the cutterhead torque but cannot achieve dynamically and accurately the prediction of the load. Additionally, compared with the method of an online learning paradigm, which puts more attention to data closer to the excavation point, the framework has stronger robustness. This is because the model can use the knowledge contained in historical data to reduce the impact of random noise and alleviate over-fitting issues. In summary, the major novelty of this study is to provide a first test of merging MTL and transfer learning for TBM dynamic cutterhead torque prediction. Though the framework is presented in the context of dynamic cutterhead torque prediction of TBM, it can be easily extended to the status monitoring of other engineering systems, such as wind power equipment, automobiles, etc. In the near future, we plan to further investigate the adaptable adjustment of TBM's

operating status based on the proposed framework, which is of great significance to the operation safety and energy consumption.

Author Contributions: Conceptualization, T.F.; Funding acquisition, X.S.; Investigation, T.Z.; Methodology, T.F. and T.Z.; Project administration, X.S.; Resources, X.S.; Software, T.F. and T.Z.; Writing—original draft, T.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China (Grant No. 2018YFB1702502) and the National Natural Science Foundation of China (Grant No. 52075068).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A

Table A1. List of operating parameters.

Parameter (Unit)	Parameter (Unit)
Temperature of oil tank (°C)	Temperature of gear oil (°C)
Rotation speed of cutterhead(r/min)	Cutter power (kw)
Propelling pressure (bar)	Propelling pressure of A group (bar)
Propelling pressure of B group (bar)	Propelling pressure of C group (bar)
Propelling pressure of D group (bar)	Pressure of equipment bridge (bar)
Pressure of articulation system (bar)	Pressure of shield tail seal at top right front (bar)
Pressure of shield tail seal at right front (bar)	Pressure of shield tail seal at left front (bar)
Pressure of shield tail seal at top right back (bar)	Pressure of shield tail seal at right back (bar)
Pressure of shield tail seal at bottom back (bar)	Pressure of shield tail seal at left front (bar)
Pressure of shield tail seal at top left front (bar)	Pressure of shield tail seal at left back (bar)
Pressure of shield tail seal at top left back (bar)	Pressure of shield tail seal at right back (bar)
Rolling angle (°)	Pressure of screw pump at back (bar)
Pressure of chamber at top left (bar)	Pressure of chamber at bottom left (bar)
Pressure of chamber at bottom right (bar)	Bentonite pressure (bar)
Temperature of screw conveyor (°C)	Pitch angle (°)
Thrust of cutterhead (kN)	Advance velocity (mm/min)
Torque of cutterhead (kNm)	Displacement of A group of thrust cylinders (mm)
Displacement of B group of thrust cylinders (mm)	Displacement of C group of thrust cylinders (mm)
Displacement of D group of thrust cylinders (mm)	Displacement of articulated system at top right (mm)
Displacement of articulated system at left (mm)	Displacement of articulated system at top left (mm)
Displacement of articulated system at right (mm)	Bentonite pressure of shield shell (bar)
Pressure of screw conveyor at front (bar)	Pressure of screw pump (bar)

References

- Zheng, Y.L.; Zhang, Q.B.; Zhao, J. Challenges and opportunities of using tunnel boring machines in mining. *Tunn. Undergr. Space Technol.* **2016**, *57*, 287–299. [[CrossRef](#)]
- Delisio, A.; Zhao, J.; Einstein, H.H. Analysis and prediction of TBM performance in blocky rock conditions at the Löttschberg Base Tunnel. *Tunn. Undergr. Space Technol.* **2013**, *33*, 131–142. [[CrossRef](#)]
- Sun, W.; Wang, X.; Wang, L.; Zhang, J.; Song, X. Multidisciplinary design optimization of tunnel boring machine considering both structure and control parameters under complex geological conditions. *Struct. Multidiscip. Optim.* **2016**, *54*, 1073–1092. [[CrossRef](#)]
- Wang, L.; Gong, G.; Shi, H.; Yang, H. Modeling and analysis of thrust force for EPB shield tunneling machine. *Autom. Constr.* **2012**, *27*, 138–146. [[CrossRef](#)]
- Hassanpour, J.; Rostami, J.; Khamehchiyan, M.; Bruland, A. Developing new equations for TBM performance prediction in carbonate-argillaceous rocks: A case history of Nowsood water conveyance tunnel. *Geomech. Geoeng.* **2009**, *4*, 287–297. [[CrossRef](#)]
- Delisio, A.; Zhao, J. A new model for TBM performance prediction in blocky rock conditions. *Tunn. Undergr. Space Technol.* **2014**, *43*, 440–452. [[CrossRef](#)]
- Yagiz, S. New equations for predicting the field penetration index of tunnel boring machines in fractured rock mass. *Arab. J. Geosci.* **2017**, *10*, 33. [[CrossRef](#)]

8. Rostami, J. Performance prediction of hard rock Tunnel Boring Machines (TBMs) in difficult ground. *Tunn. Undergr. Space Technol.* **2016**, *57*, 173–182. [[CrossRef](#)]
9. Shreyas, S.K.; Dey, A. Application of soft computing techniques in tunnelling and underground excavations: State of the art and future prospects. *Innov. Infrastruct. Solut.* **2019**, *4*, 46. [[CrossRef](#)]
10. Shahrour, I.; Zhang, W. Use of soft computing techniques for tunneling optimization of tunnel boring machines. *Undergr. Space* **2021**, *6*, 233–239. [[CrossRef](#)]
11. Zhao, J.; Shi, M.; Hu, G.; Song, X.; Zhang, C.; Tao, D.; Wu, W. A Data-Driven Framework for Tunnel Geological-Type Prediction Based on TBM Operating Data. *IEEE Access* **2019**, *7*, 66703–66713. [[CrossRef](#)]
12. Avunduk, E.; Copur, H. Empirical modeling for predicting excavation performance of EPB TBM based on soil properties. *Tunn. Undergr. Space Technol.* **2018**, *71*, 340–353. [[CrossRef](#)]
13. Zhang, Q.; Hou, Z.; Huang, G.; Cai, Z.; Kang, Y. Mechanical characterization of the load distribution on the cutterhead-ground interface of shield tunneling machines. *Tunn. Undergr. Space Technol.* **2015**, *47*, 106–113. [[CrossRef](#)]
14. Faramarzi, L.; Kheradmandian, A.; Azhari, A. Evaluation and Optimization of the Effective Parameters on the Shield TBM Performance: Torque and Thrust—Using Discrete Element Method (DEM). *Geotech. Geol. Eng.* **2020**, *38*, 2745–2759. [[CrossRef](#)]
15. Leng, S.; Lin, J.R.; Hu, Z.Z.; Shen, X. A Hybrid Data Mining Method for Tunnel Engineering Based on Real-Time Monitoring Data from Tunnel Boring Machines. *IEEE Access* **2020**, *8*, 90430–90449. [[CrossRef](#)]
16. Sun, W.; Shi, M.; Zhang, C.; Zhao, J.; Song, X. Dynamic load prediction of tunnel boring machine (TBM) based on heterogeneous in-situ data. *Autom. Constr.* **2018**, *92*, 23–34. [[CrossRef](#)]
17. 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](#)]
18. Li, L.; Liu, Z.; Zhou, H.; Zhang, J.; Shen, W.; Shao, J. Prediction of TBM cutterhead speed and penetration rate for high-efficiency excavation of hard rock tunnel using CNN-LSTM model with construction big data. *Arab. J. Geosci.* **2022**, *15*, 280. [[CrossRef](#)]
19. Qin, C.; Shi, G.; Tao, J.; Yu, H.; Jin, Y.; Lei, J.; Liu, C. Precise cutterhead torque prediction for shield tunneling machines using a novel hybrid deep neural network. *Mech. Syst. Signal Process.* **2021**, *151*, 107386. [[CrossRef](#)]
20. Suwansawat, S.; Einstein, H.H. Artificial neural networks for predicting the maximum surface settlement caused by EPB shield tunneling. *Tunn. Undergr. Space Technol.* **2006**, *21*, 133–150. [[CrossRef](#)]
21. Lau, S.C.; Lu, M.; Ariaratnam, S.T. Applying radial basis function neural networks to estimate next-cycle production rates in tunnelling construction. *Tunn. Undergr. Space Technol.* **2010**, *25*, 357–365. [[CrossRef](#)]
22. Gao, X.; Shi, M.; Song, X.; Zhang, C.; Zhang, H. Recurrent neural networks for real-time prediction of TBM operating parameters. *Autom. Constr.* **2019**, *98*, 225–235. [[CrossRef](#)]
23. Zhou, J.; Qiu, Y.; Zhu, S.; Armaghani, D.J.; Li, C.; Nguyen, H.; Yagiz, S. Optimization of support vector machine through the use of metaheuristic algorithms in forecasting TBM advance rate. *Eng. Appl. Artif. Intell.* **2021**, *97*, 104015. [[CrossRef](#)]
24. Armaghani, D.J.; Mohamad, E.T.; Narayanasamy, M.S.; Narita, N.; Yagiz, S. Development of hybrid intelligent models for predicting TBM penetration rate in hard rock condition. *Tunn. Undergr. Space Technol.* **2017**, *63*, 29–43. [[CrossRef](#)]
25. Armaghani, D.J.; Koopialipoor, M.; Marto, A.; Yagiz, S. Application of several optimization techniques for estimating TBM advance rate in granitic rocks. *J. Rock Mech. Geotech. Eng.* **2019**, *11*, 779–789. [[CrossRef](#)]
26. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A Comprehensive Survey on Transfer Learning. *Proc. IEEE* **2021**, *109*, 43–76. [[CrossRef](#)]
27. Lu, J.; Behbood, V.; Hao, P.; Zuo, H.; Xue, S.; Zhang, G. Transfer learning using computational intelligence: A survey. *Knowl.-Based Syst.* **2015**, *80*, 14–23. [[CrossRef](#)]
28. Hu, Q.; Zhang, R.; Zhou, Y. Transfer learning for short-term wind speed prediction with deep neural networks. *Renew. Energy* **2016**, *85*, 83–95. [[CrossRef](#)]
29. Ye, R.; Dai, Q. A novel transfer learning framework for time series forecasting. *Knowl.-Based Syst.* **2018**, *156*, 74–99. [[CrossRef](#)]
30. Shi, M.; Zhang, L.; Sun, W.; Song, X. A fuzzy c-means algorithm guided by attribute correlations and its application in the big data analysis of tunnel boring machine. *Knowl.-Based Syst.* **2019**, *182*, 104859. [[CrossRef](#)]
31. Zhang, Y.; Yang, Q. A Survey on Multi-Task Learning. *IEEE Trans. Knowl. Data Eng.* **2021**, *4347*, 1–20. [[CrossRef](#)]
32. Song, X.; Shi, M.; Wu, J.; Sun, W. A new fuzzy c-means clustering-based time series segmentation approach and its application on tunnel boring machine analysis. *Mech. Syst. Signal Process.* **2019**, *133*, 106279. [[CrossRef](#)]
33. Li, Y.; Tian, X.; Liu, T.; Tao, D. On better exploring and exploiting task relationships in multitask learning: Joint model and feature learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 1975–1985. [[CrossRef](#)] [[PubMed](#)]
34. Xu, S.; An, X.; Qiao, X.; Zhu, L. Multi-task least-squares support vector machines. *Multimed. Tools Appl.* **2014**, *71*, 699–715. [[CrossRef](#)]
35. Evgeniou, T.; Pontil, M. Regularized Multi-Task Learning. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 22–25 August 2004.
36. Ma, J.; Theiler, J.; Perkins, S. Accurate On-line Support Vector Regression. *Neural Comput.* **2003**, *15*, 2683–2703. [[CrossRef](#)]