


Letter

# Well Logging Based Lithology Identification Model Establishment Under Data Drift: A Transfer Learning Method

Haining Liu <sup>1,2</sup>, Yuping Wu <sup>3</sup>, Yingchang Cao <sup>1</sup>, Wenjun Lv <sup>4,\*</sup> , Hongwei Han <sup>2</sup>, Zerui Li <sup>4</sup> and Ji Chang <sup>4</sup>

<sup>1</sup> School of Geosciences, China University of Petroleum, Qingdao 266580, China; liuhaining632.slyt@sinopec.com (H.L.); caoych@upc.edu.cn (Y.C.)

<sup>2</sup> Shengli Geophysical Research Institute of SINOPEC, Dongying 257022, China; hanhongwei.slyt@sinopec.com

<sup>3</sup> School of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China; ypwu@zjut.edu.cn

<sup>4</sup> Department of Automation, University of Science and Technology of China, Hefei 230027, China; lzerui@mail.ustc.edu.cn (Z.L.); cjchange@mail.ustc.edu.cn (J.C.)

\* Correspondence: wlv@ustc.edu.cn

Received: 29 May 2020; Accepted: 23 June 2020; Published: 29 June 2020

**Abstract:** Recent years have witnessed the development of the applications of machine learning technologies to well logging-based lithology identification. Most of the existing work assumes that the well loggings gathered from different wells share the same probability distribution; however, the variations in sedimentary environment and well-logging technique might cause the data drift problem; i.e., data of different wells have different probability distributions. Therefore, the model trained on old wells does not perform well in predicting the lithologies in newly-coming wells, which motivates us to propose a transfer learning method named the data drift joint adaptation extreme learning machine (DDJA-ELM) to increase the accuracy of the old model applying to new wells. In such a method, three key points, i.e., the project mean maximum mean discrepancy, joint distribution domain adaptation, and manifold regularization, are incorporated into extreme learning machine. As found experimentally in multiple wells in Jiyang Depression, Bohai Bay Basin, DDJA-ELM could significantly increase the accuracy of an old model when identifying the lithologies in new wells.

**Keywords:** lithology identification; domain adaptation; manifold regularization; projected maximum mean discrepancy; extreme learning machine

## 1. Introduction

Well logging data have grown dramatically over the past few decades due to the widespread deployment of oil wells and the rapid development of sensing technology. A large amount of data not only brings us more opportunities to understand the underground but also brings more great challenges to the interpretation of logging data [1,2]. Well logging provides an objective and continuous method with which to observe the properties of the rocks through which the drill bit passes and to describe the deposition process quantitatively. As a bridge between surface geophysical survey and subsurface geology, well logging is an effective and irreplaceable method to understand reservoir characteristics. As conventional reservoirs dry up, oil and gas companies are turning to unconventional exploration and development in shale and low-permeability reservoirs, posing more challenges for logging interpretation [3].

In traditional logging interpretation, lithology determination, porosity, and permeability calculations are performed by specialists in exploration and geology with specialized knowledge.

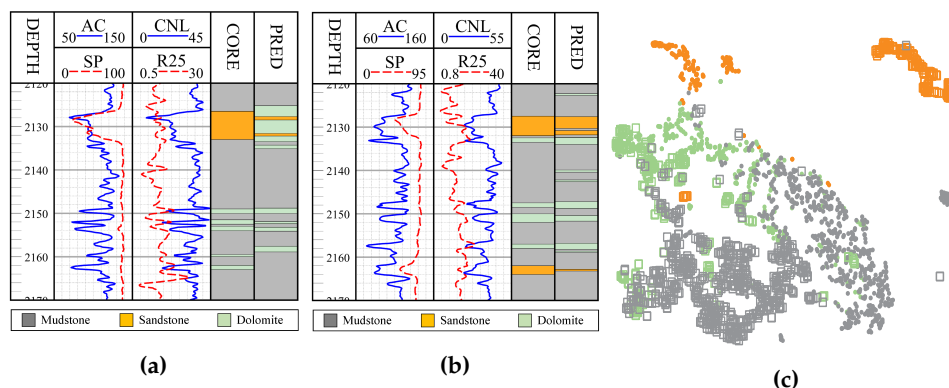
Lithology identification is a fundamental problem in well logging interpretation and is of considerable significance in petroleum exploration engineering. It is the basis of reservoir parameter calculation (such as porosity, shale volume, permeability) and geological research (such as formation correlation, sedimentary modeling, favorable zone prediction, etc.). Due to the complexity of reservoir geological conditions, the uncertainty of exploration data and the inconsistency of expert experience, the results of lithology identification are mainly dependent on expertise. With the increasing diversity of logging data, the traditional logging interpretation methods which rely on human experience have some shortcomings and limitations. As a result, researchers are turning to more advanced data analysis methods for breakthroughs in lithology identification.

In recent years, with the rapid development of machine learning technology, its application in lithology identification has also attracted full attention [4]. In order to accurately and effectively determine the lithology type, a large amount of application work based on different machine learning began to emerge [5]. For example, G. Askari [6] used satellite remote sensing data to find the lithological information in Deh-Molla sedimentary succession by principal component analysis. Al-Anazi et al. [7] proposed a support vector machine-based classification feature method and selection method based on fuzzy theory to realize the recognition of potential features and the improvement of lithology recognition performance. Wang et al. [8] proposed a novel back propagation (BP) model by modifying the self-adapting algorithm and activation function, which is proven to be effective in predicting the lithologies of the Kela-2 gas field. The experimental results show that, compared with discriminant analysis and probabilistic neural network, support vector machines can identify different lithology of heterogeneous sandstone reservoirs more accurately. Xie et al. [9] evaluated five typical machine learning methods of naive Bayes, support vector machines, artificial neural networks, random forest and gradient tree enhancement based on the formation lithology identification data of the Danudui gas field and the Hangjinqi gas field. Experimental results show that the integrated method, including random forest and gradient tree enhancement, has a lower prediction error. At the same time, the gradient tree enhancement method has the highest accuracy compared with the other four methods. In order to reduce exploration uncertainty, Bhattacharya et al. [10] compared different methods of facies division and prediction of mudstone in the current conventional logging data, and applied them to the Devonian Bakken and Mamentago–Marsilus formations in North America. In this study, support vector machines (SVM), artificial neural networks (ANN), self-organizing mapping (SFM), and multi-resolution map based clustering (MRMC) are compared experimentally. Dev et al. [11] analyzed the data from China’s Danuci gas field and Hangjinqi gas field by the gradient lifting decision tree system (i.e., XGBoost, LightGBM, and CatBoost) to study formation lithology classification, and compared their performance with random forest, AdaBoost, and gradient boosting machines. Experiments show that LightGBM and CatBoost are the preferred algorithms for lithology classification by using well logging data. A large number of similar studies can be seen in [12–16].

In addition to these directly-applied studies, more and more scholars are focusing on how to improve existing machine learning tools to solve practical problems in lithology identification. Due to the different distribution of underground lithology, there is a severe class imbalance problem in the training data set. Furthermore, Deng et al. [17] introduced a borderline-stroke technique for dealing with unbalanced data, and the results showed that this method could effectively improve the classification accuracy of SVM, especially for the minority classes. There are many free hyper-parameters in machine learning algorithms, and the settings of these hyper-parameters will have significant influences on the performance of lithology identification. Therefore, Saporetti et al. [18] adopted an evolutionary parameter tuning strategy, and combined gradient boosting (GB) with differential evolution (DE) to achieve the optimization of super parameters, thereby making the lithology identification more stable. In the study of [19], the wavelet decomposition was used to construct multi-channel images of logging data, and then the lithology identification problem based on the logging curve was skillfully transformed into an image segmentation problem. Finally, the feasibility of this method was verified by the application in the Daqing oilfield. Aiming at the

issue of data drift between different wells, Ao et al. [20] proposed a hybrid algorithm for lithology identification that combines the mean drift algorithm and the random forest algorithm in the prototype similarity space. It is pointed out that a more accurate lithology identification model can be obtained by transforming the classification problem into prototype similarity space. Li et al. [21] proposed a semi-supervised algorithm based on Generated Adversarial Network, which uses logging curves as the labeled data and seismic data as the unlabeled data. The model was trained by Adam algorithm and uses the discriminator to identify the lithologies. Compared with the experimental results of various supervised methods, the model can effectively use unlabeled data to achieve higher prediction accuracy. Similar work can be found in [22–24].

Although much work has been done, a practical problem, i.e., the well loggings from different wells are different in the probability distribution, is not taken into consideration. Hence, the model trained on old wells might not perform well on new wells. As illustrated in Figure 1, the phenomenon of data drift occurs between two wells, even when they are geospatially near. In particular, applying the model trained on well  $W_b$  to well  $W_a$  has many errors, as shown in Figure 1a. To suppress the data drift-induced accuracy decrease, we propose a transfer learning method named the data drift joint adaptation extreme learning machine (DDJA-ELM) to increase the accuracy of the old model applying to new wells. By incorporating the project mean maximum mean discrepancy, joint distribution domain adaptation, and manifold regularization into the extreme learning machine, we realize the knowledge transfer from old wells to new wells. As experimented in multiple wells in Jiyang Depression, Bohai Bay Basin, DDJA-ELM could increase the accuracy of an old model significantly when identifying the lithologies in new wells. In the remainder of the paper, Section 2 expatiates on the proposed DDJA-ELM, which is evaluated in Section 3. The last section concludes the paper.



**Figure 1.** Illustration of data drift (a): Logging curves, core, and prediction results (using the model trained on well  $W_b$ ) on well  $W_a$ . (b) Logging curves, core, and prediction results (using the model trained on  $W_a$ ) of  $W_b$ . (c) t-SNE [25] visualization of data distribution for  $W_a$  and  $W_b$ . Gray, yellow, and green indicate mudstone, sandstone, and dolomite, respectively. Point and hollow square indicate data from  $W_a$  and  $W_b$ , respectively.  $W_a$  and  $W_b$  are geospatially near.

## 2. Methodology

### 2.1. Notation

The dataset  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$  is composed of the sample  $x_i = [x_1, \dots, x_d]' \in \mathbb{R}^d$  with  $d$  dimension and the label  $y_i = \underbrace{[0, \dots, 0, 1, 0, \dots, 0]}_k \in \{0, 1\}$  if  $x_i$  belongs to the  $k$ -th ( $1 \leq k \leq c$ ) class, where  $n$  and  $c$  are the numbers of samples and classes, respectively. In well logging-based lithology identification,  $d$  generally means the number of loggings types, and  $c$  denotes the number of lithology types. A sample is composed of the logging values at a certain depth. Considering the problem of data drift, we use  $\mathcal{D}_S$  and  $\mathcal{D}_T$  to differentiate the datasets with drift; i.e.,  $\mathcal{D}_S$  and  $\mathcal{D}_T$  represent the source dataset for training and the target dataset to predict, respectively. Specifically, the source dataset is labeled and

the target dataset is unlabeled, thereby denoting  $\mathcal{D}_S = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{n_s}$  and  $\mathcal{D}_T = \{\mathbf{x}_i^t\}_{i=1}^{n_t}$  with  $n_s$  and  $n_t$  samples accordingly, and  $n_s + n_t = n_{s+t}$ , where  $\mathbf{x}_i^s$  and  $\mathbf{x}_i^t$  are source and target-dataset samples, respectively; and  $\mathbf{y}_i^s$  is a source-dataset label of  $\mathbf{x}_i^s$ . By defining  $\mathbf{X}_S = [\mathbf{x}_1^s; \dots; \mathbf{x}_{n_s}^s]$ ,  $\mathbf{Y}_S = [\mathbf{y}_1^s; \dots; \mathbf{y}_{n_s}^s]$ , and  $\mathbf{X}_T = [\mathbf{x}_1^t; \dots; \mathbf{x}_{n_t}^t]$ , then the  $\mathcal{D}_S = \{(\mathbf{X}_S, \mathbf{Y}_S)\}$  and  $\mathcal{D}_T = \{\mathbf{X}_T\}$ .

## 2.2. Problem Definition and Formulation

In general, a classifier  $f(x)$  trained on  $\mathcal{D}$  classifies well on those samples that satisfied the independent identically distributed (i.i.d.) assumption. However, the classifier  $f(x)$  trained on  $\mathcal{D}_S$  might not perform well on  $\mathcal{D}_T$  because the data drift means that the i.i.d. assumption does not hold; i.e., the knowledge learned from  $\mathcal{D}_S$  cannot transfer to  $\mathcal{D}_T$ . In this case, it is necessary to add some new constraint conditions to achieve expected performance while learning the classifier  $f$ . Exactly, the expected performance can be concluded as the following constraint conditions: (i) minimizing the structural risk to avoid overfitting; (ii) minimizing the data drift between the source dataset and the target dataset; (iii) minimizing the prediction inconsistency within the target dataset. If considering the ELM as the basic classifier  $f$ ,  $f(\mathbf{x}) = f(\mathbf{x}; \boldsymbol{\beta})$  where  $\boldsymbol{\beta}$  is the output weight matrix of ELM.

According to the structural risk minimization (SRM) and regularization techniques [26], the classifier  $f$  can be denoted as

$$f = \arg \min_{\boldsymbol{\beta}} \ell(f(\mathbf{x}; \boldsymbol{\beta}), \mathbf{y}) + R(\mathbf{x}; \boldsymbol{\beta}), \quad (1)$$

where the first term represents the empirical loss on samples (i.e., describing the fitness of applying the model to predict the training data); the second term represents the regularization term (i.e., representing the formulation of constraint conditions). Thus, combined with the constraint conditions mentioned above, the objective function is formulated mathematically as follows

$$f = \arg \min_{\boldsymbol{\beta}} \ell(f(\mathbf{x}; \boldsymbol{\beta}), \mathbf{y}) + \frac{1}{2} \|\boldsymbol{\beta}\|^2 + \frac{\lambda}{2} \Omega(\mathcal{D}_S, \mathcal{D}_T; \boldsymbol{\beta}) + \frac{\gamma}{2} \mathcal{M}(\mathcal{D}_T; \boldsymbol{\beta}), \quad (2)$$

where the empirical loss term  $\ell(f(\mathbf{x}; \boldsymbol{\beta}), \mathbf{y})$  and the structural risk regularization term  $\|\boldsymbol{\beta}\|^2$  compose the model accuracy and complexity of ELM. The term  $\Omega(\mathcal{D}_S, \mathcal{D}_T; \boldsymbol{\beta})$  indicates the extent of data drift between  $\mathcal{D}_S$  and  $\mathcal{D}_T$ . Additionally, we introduce the manifold regularization term  $\mathcal{M}(\mathcal{D}_T; \boldsymbol{\beta})$  to improve the prediction consistency within the target dataset.  $\lambda$  and  $\gamma$  are the regularization parameters accordingly.

In the remainder of this section, we will expatiate on each term of the objective function.

### 2.2.1. ELM

Recent years have witnessed the development of a promising machine learning model; i.e., extreme learning machine (ELM). ELM is actually an artificial neural network with a single hidden layer, which was first proposed by Huang et al. [27] and found its application in many domains, such as robotic perception, hyperspectral image classification, lithology identification, and human activity recognition [28–32]. Compared with support vector machine and other artificial neural networks, ELM has significant superiority in generalization performance and training time. In addition, many variants for ELM have been investigated, including semi-supervised ELM, multi-kernel ELM, rough ELM, one-class ELM, etc. [33–36].

According to the objective function (2), we first describe the mathematical model of ELM as follows [37],

$$\arg \min_{\beta \in \mathbb{R}^{z \times c}} \ell(f(\mathbf{x}; \beta), \mathbf{y}) + \frac{1}{2} \|\beta\|^2 = \frac{C}{2} \sum_{i=1}^{n_{s+t}} \|\mathbf{e}_i\|^2 + \frac{1}{2} \|\beta\|^2 \quad (3a)$$

$$\text{s.t. } j_i \mathbf{h}(\mathbf{x}_i) \beta = \mathbf{y}_i - \mathbf{e}_i, \quad i = 1, \dots, n_{s+t}, \quad (3b)$$

where  $\mathbf{e}_i \in \mathbb{R}^c$  represents the error vector with respect to the  $i$ -th training sample, and  $\sum_{i=1}^{n_{s+t}} \|\mathbf{e}_i\|^2$  is the sum of prediction errors. The tradeoff coefficient  $C$  is used to balance the contribution between the two terms. Since the target dataset  $\mathcal{D}_T$  is unlabeled,  $j_i = 1$  if  $i \leq n_s$ ; otherwise,  $j_i = 0$ ; and  $\mathbf{y}_i$  equals a zero vector if  $i > n_s$ . The output of hidden layer is equal to  $\mathbf{h}(\mathbf{x}_i) = [\mathbf{g}(\mathbf{x}_i; \mathbf{w}_1, b_1), \dots, \mathbf{g}(\mathbf{x}_i; \mathbf{w}_z, b_z)]$ , where  $\mathbf{w}_j \in \mathbb{R}^z$ ;  $z$  is the number of hidden neurons;  $b_j \in \mathbb{R}$  are the  $j$ -th random generated weight vector and bias constant;  $\mathbf{g}(\cdot)$  is a piecewise continuous nonlinear activation function, such as the sigmoid function  $\mathbf{g}(\mathbf{x}_i; \mathbf{w}_j, b_j) = 1 / \{1 + \exp(-\mathbf{w}_j^T \cdot \mathbf{x}_i - b_j)\}$  or the Gaussian function  $\mathbf{g}(\mathbf{x}_i; \mathbf{w}_j, b_j) = \exp(-b_j \|\mathbf{x}_i - \mathbf{w}_j\|^2)$ . In this paper, the sigmoid function is used as the activation function.

### 2.2.2. Weighted ELM

Additionally, since each lithology in dataset  $\mathcal{D}$  usually contains the samples of different amounts, appropriate weights should be assigned to each error vectors for the samples imbalance issue, so (3) is rewritten as

$$\arg \min_{\beta \in \mathbb{R}^{z \times c}} \ell(f(\mathbf{x}; \beta), \mathbf{y}) + \frac{1}{2} \|\beta\|^2 = \frac{C}{2} \sum_{i=1}^{n_{s+t}} \omega_i \|\mathbf{e}_i\|^2 + \frac{1}{2} \|\beta\|^2 \quad (4a)$$

$$\text{s.t. } j_i \mathbf{h}(\mathbf{x}_i) \beta = \mathbf{y}_i - \mathbf{e}_i, \quad i = 1, \dots, n_{s+t}, \quad (4b)$$

where  $\omega_i = \frac{v_i}{\sum_{i=1}^{n_{s+t}} v_i}$  denotes the weight,  $v_i = \frac{1}{n_i^\tau}$  and  $\tau$  is a constant, and  $n_i$  is the number of samples belonging to  $y_i$ , thereby modifying the ELM to the weighted ELM (WELM).

Substituting the constraints (4b) into (4a) yields the equivalent unconstrained optimization problem and the matrix form:

$$\arg \min_{\beta \in \mathbb{R}^{z \times c}} \frac{C}{2} \mathbf{W} \|(\mathbf{Y} - \mathbf{JH}\beta)\|^2 + \frac{1}{2} \|\beta\|^2, \quad (5)$$

where  $\mathbf{W} = \text{blockdiag}(\mathbf{W}_S, \mathbf{0}_{n_t \times n_t}) \in \mathbb{R}^{n_{s+t} \times n_{s+t}}$ ;  $\mathbf{W}_S \in \mathbb{R}^{n_s \times n_s}$  is a diagonal matrix with each item  $\omega_i$ ; and  $\mathbf{Y} = [\mathbf{Y}_S; \mathbf{0}_{n_t \times c}] \in \mathbb{R}^{n_{s+t} \times c}$ ,  $\mathbf{H} = [\mathbf{h}(\mathbf{x}_1); \dots; \mathbf{h}(\mathbf{x}_{n_{s+t}})] \in \mathbb{R}^{n_{s+t} \times z}$ ,  $\mathbf{J} = \text{blockdiag}(\mathbf{I}_{n_s \times n_s}, \mathbf{0}_{n_t \times n_t}) \in \mathbb{R}^{n_{s+t} \times n_{s+t}}$ , where  $\mathbf{0}$  and  $\mathbf{I}$  are the zero matrix and the identity matrix with appropriate dimension, respectively. By setting the gradient of (5) over  $\beta$  to zero, we have

$$\beta - \mathbf{C}\mathbf{H}'\mathbf{J}'\mathbf{W}(\mathbf{Y} - \mathbf{JH}\beta) = 0, \quad (6)$$

where  $'$  represents the transpose of matrix. There are two forms of an optimal solution to the  $\beta^*$ . When the number of the training data is larger or smaller than the number of hidden neurons, then  $\mathbf{H}$  has more or fewer rows than columns, thereby resulting in an underdetermined or overdetermined least squares problem. The closed-form solution for (6) can be described as

$$\beta^* = \begin{cases} \mathbf{H}'\mathbf{W} \left( \mathbf{H}\mathbf{H}'\mathbf{W} + \frac{1}{C} \mathbf{I}_{n_{s+t}} \right)^{-1} \mathbf{Y}, & \text{if } n_{s+t} < z \\ \left( \mathbf{H}'\mathbf{W}\mathbf{H} + \frac{1}{C} \mathbf{I}_z \right)^{-1} \mathbf{H}'\mathbf{W}\mathbf{Y}, & \text{if } n_{s+t} \geq z \end{cases}, \quad (7)$$

where  $\mathbf{I}_{n_{s+t}}$  and  $\mathbf{I}_z$  are the identity matrix of  $n_{s+t}$  and  $z$  dimensions.

### 2.2.3. Data Drift Adaptation

In general, the data distribution properties can be used to statistically describe the correlation among the samples  $(x_i, y_i) \in \mathcal{D}$ , thus the data drift-induced distribution discrepancy can be estimated by statistical criteria. In this paper, we utilize the *projected maximum mean discrepancy* (PMMD) criterion for distribution discrepancy measure [38]. Thus, the regularization term  $\Omega(\mathcal{D}_S, \mathcal{D}_T; \beta)$  with PMMD is formulated as

$$\Omega_P(\mathcal{D}_S, \mathcal{D}_T; \beta) = \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \mathbf{h}(x_i^s) \beta - \frac{1}{n_t} \sum_{j=1}^{n_t} \mathbf{h}(x_j^t) \beta \right\|^2, \tag{8}$$

Further, (8) can be rewritten to a matrix form; that is,

$$\Omega_P(\mathcal{D}_S, \mathcal{D}_T; \beta) = \text{tr}(\beta' \mathbf{H}' \mathbf{M}_P \mathbf{H} \beta), \tag{9}$$

where  $\text{tr}(\cdot)$  denotes the trace of a matrix. The elements of matrix  $\mathbf{M}_P \in \mathbb{R}^{n_{s+t} \times n_{s+t}}$  are calculated by

$$(\mathbf{M}_P)_{pq} = \begin{cases} \frac{1}{n_s^2}, & \text{if } p, q \leq n_s \\ \frac{1}{n_t^2}, & \text{if } p, q > n_s \\ -\frac{1}{n_s n_t}, & \text{otherwise.} \end{cases} \tag{10}$$

According to (8), the samples  $x_i^s$  and  $x_j^t$  are transformed from the feature space to the mapping space by  $\mathbf{h}(\cdot)$ , where the distribution discrepancy between the source and target datasets can be reduced by adjusting  $\beta$ . Thus, the modified WELM named data drift adaptation WELM (DDA-WELM) classifier can be achieved by introducing the regularization term  $\Omega_P(\mathcal{D}_S, \mathcal{D}_T; \beta)$  to adapt the data drift.

However, the overall accuracy is increased with sacrificing the accuracies of some classes, because DDA-WELM is a method of reducing the distribution discrepancy between two datasets as a whole. Consequently, we modify the PMMD to propose the joint PMMD (JPMMD) that aims to reduce the distribution discrepancy between classes of two datasets, respectively. The regularization term  $\Omega(\mathcal{D}_S, \mathcal{D}_T; \beta)$  with JPMMD is formulated as

$$\Omega_J(\mathcal{D}_S, \mathcal{D}_T; \beta) = \sum_{k=1}^c \left\| \frac{1}{n_s^{(k)}} \sum_{y_i^s=k} \mathbf{h}(x_i^s) \beta - \frac{1}{n_t^{(k)}} \sum_{\tilde{y}_j^t=k} \mathbf{h}(x_j^t) \beta \right\|^2, \tag{11}$$

where  $y_i^s$  is the real label of  $x_i^s$  and  $\tilde{y}_j^t$  is the pseudo label of  $x_j^t$  which is generated by the DDA-WELM classifier;  $n_s^{(k)}$  and  $n_t^{(k)}$  are the number of source-dataset samples which belong to the class  $k$  and the number of target-dataset samples whose pseudo labels are  $k$ , respectively.

Similarly, (11) can be rewritten to the matrix form; that is,

$$\Omega_J(\mathcal{D}_S, \mathcal{D}_T; \beta) = \text{tr}(\beta' \mathbf{H}' \mathbf{M}_J \mathbf{H} \beta), \tag{12}$$

where the elements of matrix  $\mathbf{M}_J \in \mathbb{R}^{n_{s+t} \times n_{s+t}}$  are computed by

$$\left( \mathbf{M}_J^{(k)} \right)_{pq} = \begin{cases} \frac{1}{(n_s^{(k)})^2}, & \text{if } p, q \leq n_s \text{ and } \mathbf{y}_p^s, \mathbf{y}_q^s = k \\ -\frac{1}{n_s^{(k)} n_t^{(k)}}, & \text{if } p > n_s, q \leq n_s \text{ and } \tilde{\mathbf{y}}_{p-n_s}^t, \mathbf{y}_q^s = k \\ -\frac{1}{n_s^{(k)} n_t^{(k)}}, & \text{if } p \leq n_s, q > n_s \text{ and } \mathbf{y}_p^s, \tilde{\mathbf{y}}_{q-n_s}^t = k \\ \frac{1}{(n_t^{(k)})^2}, & \text{if } p, q > n_s \text{ and } \tilde{\mathbf{y}}_{p-n_s}^t, \tilde{\mathbf{y}}_{q-n_s}^t = k \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Thus, the data drift joint adaptation WELM (DDJA-WELM) classifier can be obtained by introducing the regularization term  $\Omega_J(\mathcal{D}_S, \mathcal{D}_T; \beta)$  and the DDA-WELM classifier to improve the WELM.

### 2.2.4. Manifold Regularization

Manifold regularization is widely used to improve the smoothness of predictions and suppress the classifier cutting through the high-density regions [39]. Specifically, we introduce the manifold regularization  $\mathcal{M}(\mathcal{D}_T; \beta)$  to assign smooth labels within the target dataset and make the classifier more adaptable to the target dataset.

The formulation of the manifold regularization  $\mathcal{M}(\mathcal{D}_T; \beta)$  is that

$$\mathcal{M}(\mathcal{D}_T; \beta) = \frac{1}{2n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} a_{i,j} \left\| \mathbf{h}(\mathbf{x}_i^t) \beta - \mathbf{h}(\mathbf{x}_j^t) \beta \right\|^2 \tag{14}$$

where the similarity  $a_{i,j}$  between the sample  $\mathbf{x}_i^t$  and  $\mathbf{x}_j^t$  is calculated by

$$a_{i,j} = \begin{cases} \exp \left( \frac{-\|\mathbf{x}_i^t - \mathbf{x}_j^t\|^2}{4\sigma^2} \right), & \mathbf{x}_i^t \in \mathcal{N}(\mathbf{x}_j^t) \text{ or } \mathbf{x}_j^t \in \mathcal{N}(\mathbf{x}_i^t), \\ 0, & \text{otherwise,} \end{cases} \tag{15}$$

where  $\mathcal{N}(\mathbf{x}_j^t)$  is the set of  $\kappa$ -nearest neighboring samples of  $\mathbf{x}_j^t$  under the metric of Euclidean distance in feature space, and  $\sigma > 0$  is the width of Guassian kernel. Additionally, (14) can be rewritten to a matrix form

$$\mathcal{M}(\mathcal{D}_T; \beta) = \text{tr}(\beta' \mathbf{H}' \mathbf{L} \mathbf{H} \beta). \tag{16}$$

where  $\mathbf{L} = \text{diag}(\mathbf{0}_{n_s \times n_s}, \mathbf{L}_T) \in \mathbb{R}^{n_{s+t} \times n_{s+t}}$ ,  $\mathbf{L}_T = \mathbf{D} - \mathbf{A}$  is the target dataset Laplacian matrix, and  $\mathbf{A} = [a_{i,j}]_{n_t \times n_t}$ ,  $\mathbf{D}$  is a diagonal matrix with diagonal elements  $d_i = \sum_{j=1}^{n_t} a_{i,j}$ .

Thus, the classifiers DDA-WELM and DDJA-WELM can be upgraded to the DDA-S2WELM and DDJA-S2WELM by introducing the manifold regularization term  $\mathcal{M}(\mathcal{D}_T; \beta)$ . Moreover, the regularization term  $\Omega_J(\mathcal{D}_S, \mathcal{D}_T; \beta)$  with pseudo labels generated by the DDA-S2WELM classifier can further improve the data drift adaptation performance of the DDJA-WELM and DDJA-S2WELM classifiers.

### 2.3. Solution of Objective Function

The solution of the objective function (2) is introduced in this section. Especially, the regularization term  $\Omega(\mathcal{D}_S, \mathcal{D}_T; \beta)$  is divided into  $\Omega_P(\mathcal{D}_S, \mathcal{D}_T; \beta)$  with PMMD and  $\Omega_J(\mathcal{D}_S, \mathcal{D}_T; \beta)$  with JPMMD so that we will discuss them in the followings, respectively.

### 2.3.1. Solution of Objective Function with $\Omega_P$

By incorporating (5), (9), and (16) into (2), we have

$$f = \arg \min_{\beta \in \mathbb{R}^{z \times c}} \frac{C}{2} \mathbf{W} \|\mathbf{Y} - \mathbf{JH}\beta\|^2 + \frac{1}{2} \|\beta\|^2 + \frac{\lambda}{2} \text{tr}(\beta' \mathbf{H}' \mathbf{M}_P \mathbf{H} \beta) + \frac{\gamma}{2} \text{tr}(\beta' \mathbf{H}' \mathbf{L} \mathbf{H} \beta). \quad (17)$$

By setting the gradient of  $f$  with respect to  $\beta$  to be zero

$$\nabla f = \mathbf{C} \mathbf{H}' \mathbf{J}' \mathbf{W} (\mathbf{JH}\beta - \mathbf{Y}) + \beta + \lambda \mathbf{H}' \mathbf{M}_P \mathbf{H} \beta + \gamma \mathbf{H}' \mathbf{L} \mathbf{H} \beta = 0. \quad (18)$$

According to (18), the closed-form solution of the optimal  $\beta$  is

$$\beta^* = \begin{cases} \mathbf{H}' \mathbf{W} \left( \frac{1}{C} \mathbf{I}_n + \left( \mathbf{W} + \frac{\lambda}{C} \mathbf{M}_P + \frac{\gamma}{C} \mathbf{L} \right) \mathbf{H} \mathbf{H}' \right)^{-1} \mathbf{Y}, & \text{if } n_s + n_t < z \\ \left( \frac{1}{C} \mathbf{I}_z + \mathbf{H}' \left( \mathbf{W} + \frac{\lambda}{C} \mathbf{M}_P + \frac{\gamma}{C} \mathbf{L} \right) \mathbf{H} \right)^{-1} \mathbf{H}' \mathbf{W} \mathbf{Y}, & \text{if } n_s + n_t \geq z \end{cases} \quad (19)$$

where  $z$  is the number of hidden neurons.

### 2.3.2. Solution of Objective Function with $\Omega_J$

Similarly, incorporating (5), (12), and (16) into (2), and solving  $\frac{\partial f}{\partial \beta} = 0$  yields the optimal  $\beta$  as follows

$$\beta^* = \begin{cases} \mathbf{H}' \mathbf{W} \left( \frac{1}{C} \mathbf{I}_n + \left( \mathbf{W} + \frac{\lambda}{C} \mathbf{M}_J + \frac{\gamma}{C} \mathbf{L} \right) \mathbf{H} \mathbf{H}' \right)^{-1} \mathbf{Y}, & \text{if } n_s + n_t < z \\ \left( \frac{1}{C} \mathbf{I}_z + \mathbf{H}' \left( \mathbf{W} + \frac{\lambda}{C} \mathbf{M}_J + \frac{\gamma}{C} \mathbf{L} \right) \mathbf{H} \right)^{-1} \mathbf{H}' \mathbf{W} \mathbf{Y}, & \text{if } n_s + n_t \geq z \end{cases} \quad (20)$$

where  $z$  is the number of hidden neurons.

## 3. Experimental Verification

In this section, we conduct extensive experiments to verify the effectiveness of our method, using the well-logging data collected from multiple wells in Jiyang Depression, Bohai Bay Basin. The experimental datasets and settings will be described first. Then, the performance and impact of each regularization terms are shown in detail. The analysis of hyper-parameters sensibility is presented at last.

### 3.1. Experimental Settings

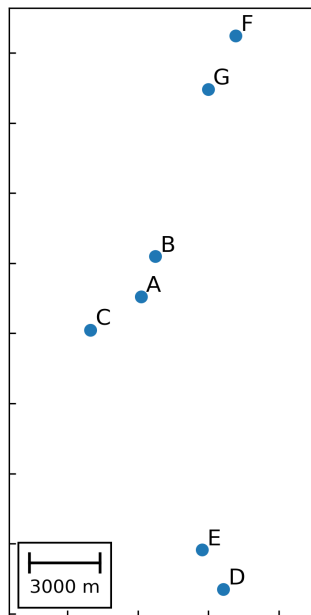
As shown in Table 1, the experimental datasets are composed of three datasets collected from different regions in the Jiyang Depression, Bohai Bay Basin and contain 3, 2 and 2 wells, respectively (The wells in Figure 1 do not appear in the experimental datasets). Their relative positions are shown in Figure 2. Table 2 describes the data drift statistically by maximum mean discrepancy. In one experiment, we set the well A as the training data and the well B as the testing data, thereby verifying the effectiveness of our method through  $A \rightarrow B$ . In this case, the other experiments are denoted as  $A \rightarrow C$ ,  $B \rightarrow A$ ,  $B \rightarrow C$ ,  $C \rightarrow A$ ,  $C \rightarrow B$ ,  $D \rightarrow E$ ,  $E \rightarrow D$ ,  $F \rightarrow G$ , and  $G \rightarrow F$ , respectively. Considering the different ranges of measurements, the value of samples are transformed to  $[0, 1]$  by a min-max normalization method. In the experiments, we adopt the index Recall (i.e., the number of classify correctly divided by the number of samples) to calculate the classification accuracy, thus denoting the average recall (Macro-R) to represent the overall classification accuracy.



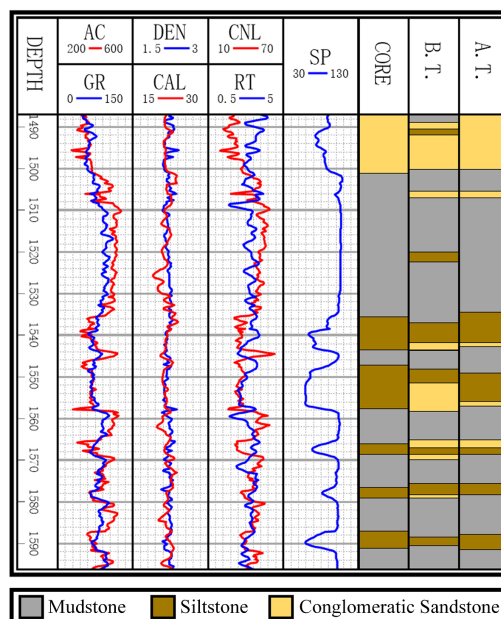
**Table 1.** Dataset description.

Dataset	1				2				3			
LOGS	AC, CAL, CNL, GR, RT, SP				AC, CAL, GR, R25, SP				AC, CAL, GR, R25, SP			
	LITH	Mu	Si	Co	LITH	Mu	Sa	Do	LITH	Mu	Sa	Sh
Samples	Well A	3820	2048	452	Well D	4508	3076	1000	Well F	6072	2852	1644
	Well B	2164	1584	432	Well E	5000	1464	1112	Well G	6996	3404	1548
	Well C	2368	1452	400	-	-	-	-	-	-	-	-

[1] Logging curves. AC: acoustic log, CAL: caliper log, CNL: compensated neutron log, GR: gamma ray log, RT: true formation resistivity, SP: spontaneous potential log, R25: 2.5 m bottom gradient resistivity. [2] Lithology. Mu: mudstone, Si: siltstone, Co: conglomeratic sandstone, Sa: sandstone, Do: dolomite, Sh: shale.



**Figure 2.** Relative positions of experimental wells.



**Figure 3.** Logging curves, core, and classification results of B.T. (before transfer) and A.T. (after transfer).

**Table 2.** Data drift description.

		<b>A and B</b>	<b>A and C</b>	<b>B and C</b>	<b>D and E</b>	<b>F and G</b>
MMD	Class 1	0.107	0.182	0.281	0.109	0.090
MMD	Class 2	0.230	0.067	0.277	0.238	0.251
MMD	Class 3	0.203	0.026	0.384	0.189	0.601
MMD	Overall	0.540	0.275	0.942	0.536	0.942

[1] MMD is short for maximum mean discrepancy, which can be found in [40]. [2] For (A,B), (A,C), and (B,C), classes 1–3 mean Mu, Si, and Co, respectively. For (D,E), classes 1–3 mean Mu, Sa, and Do, respectively. For (F,G), classes 1–3 mean Mu, Sa, and Sh, respectively.

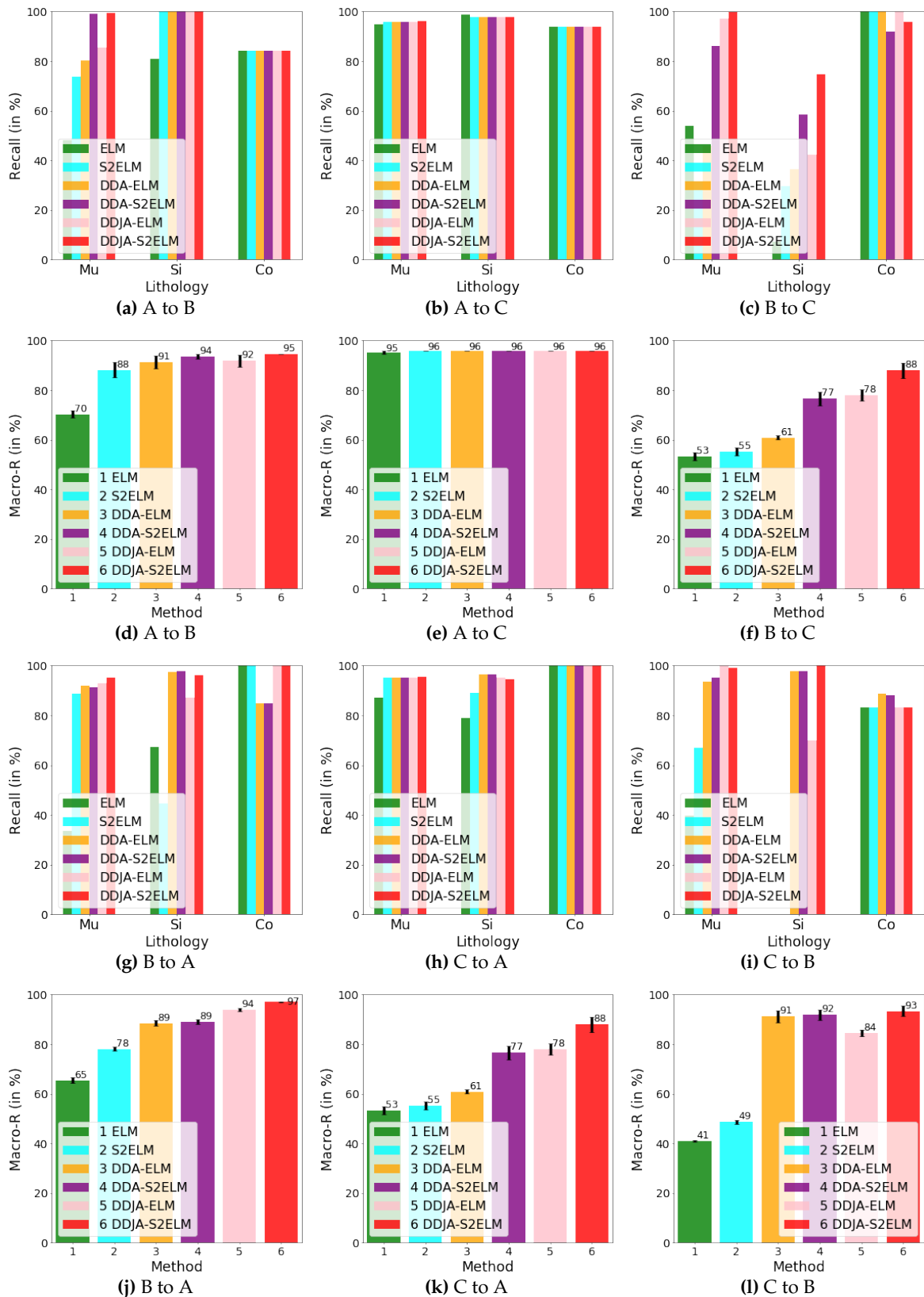
### 3.2. Experimental Results

Figure 3 exhibits the logging curves, core (i.e., label), classification performance before  $B \rightarrow A$  transfer (i.e., using model trained on data B to directly predict data A), and classification performance after  $B \rightarrow A$  transfer. It is observed that transfer learning could eliminate the data drift-induced accuracy loss. As shown in Figures 4, 5, and 6, the classification performances of applying the classifiers ELM, S2ELM, DDA-ELM, DDA-S2ELM, DDJA-ELM, and DDJA-S2ELM to datasets 1, 2, and 3 are presented, respectively. It can be obviously observed from Figures 4a–c,g–i, 5a,c, and 6a,c, that the accuracy of each class is gradually increased with the introducing of DDJA regularization terms, especially when it comes to the accuracy of Si and Sh that increase from 0 to more than 80 in Figure 4i, 6a,c, respectively. Additionally, the ELM and S2ELM classifiers without DDJA regularization term are insufficient to the tasks with data drift.

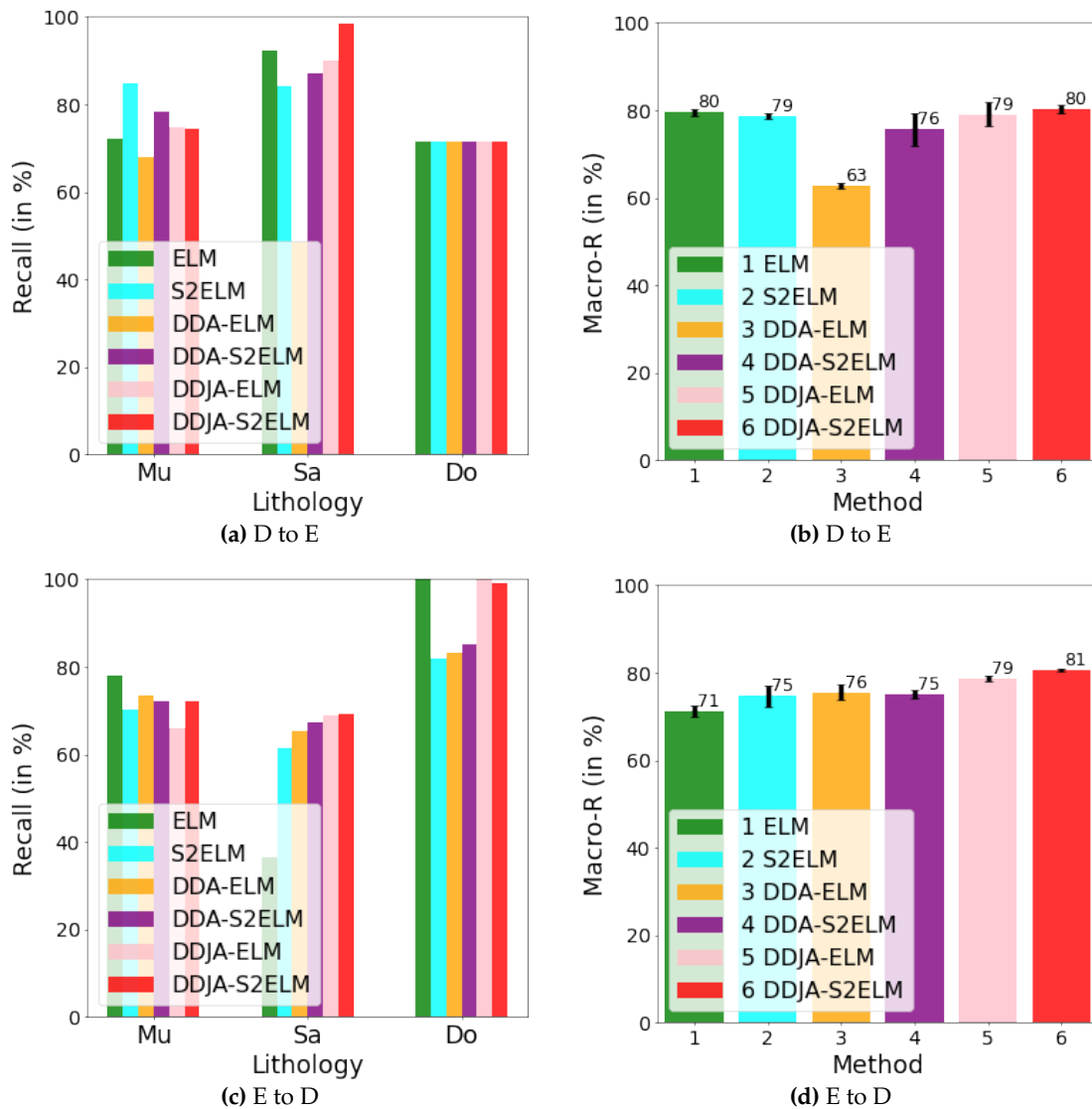
Considering the ELM-based classifier with the random weights and biases parameters, we conduct multiple experiments by setting different random seeds to generate different weights and biases parameters which yield the experimental results in Figures 4d–f,j–l, 5b,d and 6b,d. According to the results, we have the following observations: (i) The overall Macro-R shows a trend of increasing step by step and the DDJA-S2ELM classifiers achieve the highest accuracy. Moreover, the Macro-R of DDJA-S2ELM are increased to 88% at least compared with the ELM on dataset 1 ( $B \rightarrow C$  and  $C \rightarrow A$ ), 80% on dataset 2 ( $D \rightarrow E$ ), and 69% on dataset 3 ( $F \rightarrow G$  and  $G \rightarrow F$ ). Additionally, compared the ELM classifier with the DDJA-S2ELM classifier, the Macro-R of our method are increased 52% on dataset 1 ( $C \rightarrow B$ ), 10% on dataset 2 ( $E \rightarrow D$ ), and 42% on dataset 3 ( $F \rightarrow G$ ). (ii) Although the Macro-R is not increased on dataset 1 ( $A \rightarrow C$ ) and dataset 2 ( $D \rightarrow E$ ), the standard deviation can be kept at a lower level. Thus, the stability can be improved by introducing the DDJA-S2ELM. (iii) Comparing the experimental results on dataset 3 with datasets 1 and 2, the ELM classifiers only achieve 27% ( $F \rightarrow G$ ) and 35% ( $G \rightarrow F$ ). The performance is significantly enhanced with our method regarding the data drift-induced accuracy decrease surges.

### 3.3. Parameters' Sensitivity

In this section, we aim to analyze the sensitivity on the key hyper-parameters of the DDJA-S2ELM: the trade-off coefficient  $C$ , the contribution coefficient of DDJA regularization term  $\lambda$ , and the contribution coefficient of semi-supervised regularization term  $\gamma$ . By analyzing the Macro-R over different settings of hyper-parameters, the configuration of these parameters is given. To avoid repetition, we only show the results of  $B \rightarrow C$ .

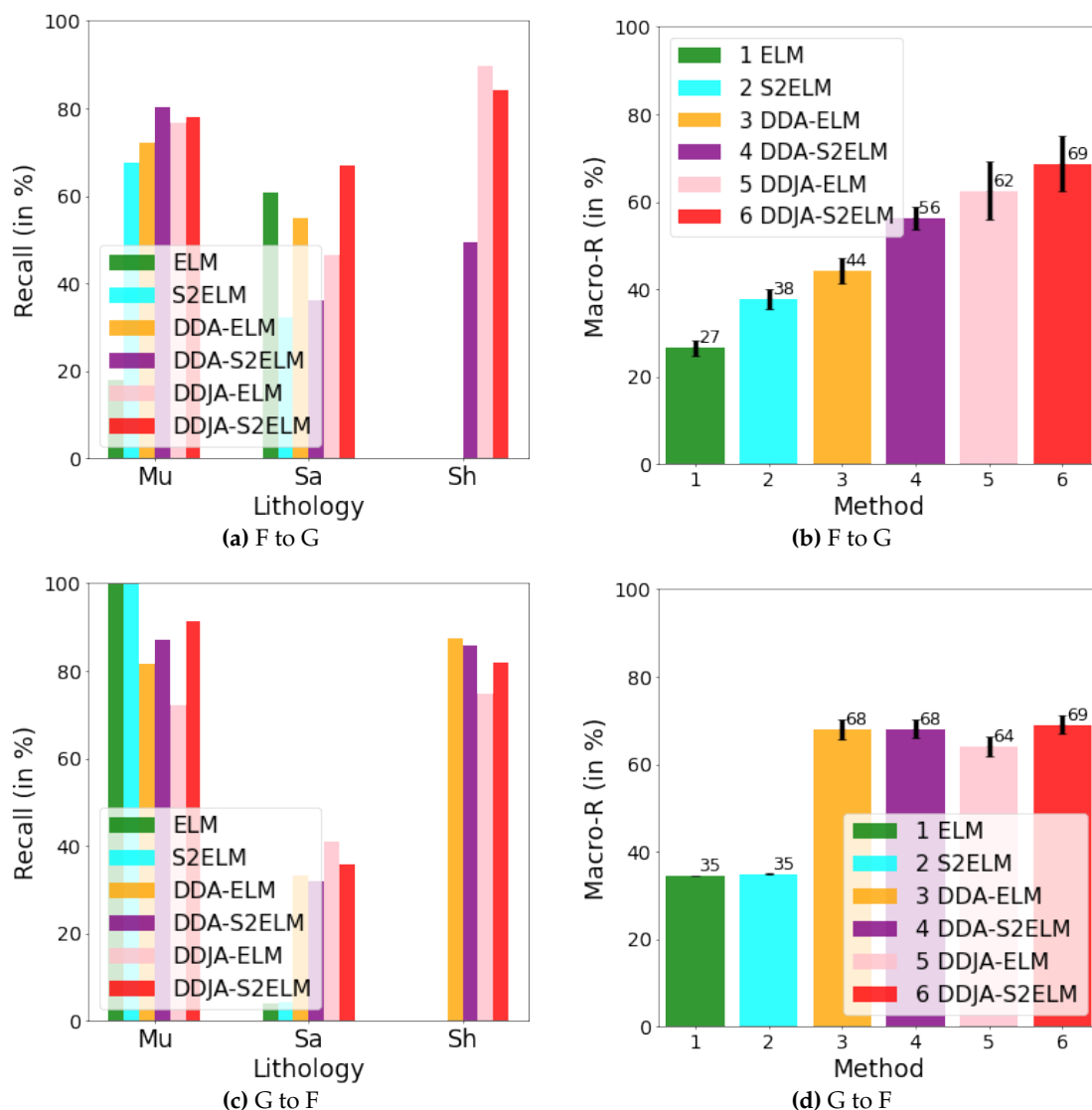


**Figure 4.** The results of performing our method on dataset 1. In (a)–(c), (g)–(i), we show the recalls for the classes of Mu, Si, and Co by using different methods. In (d)–(f), (j)–(l), we show the macro average recalls by using different methods.



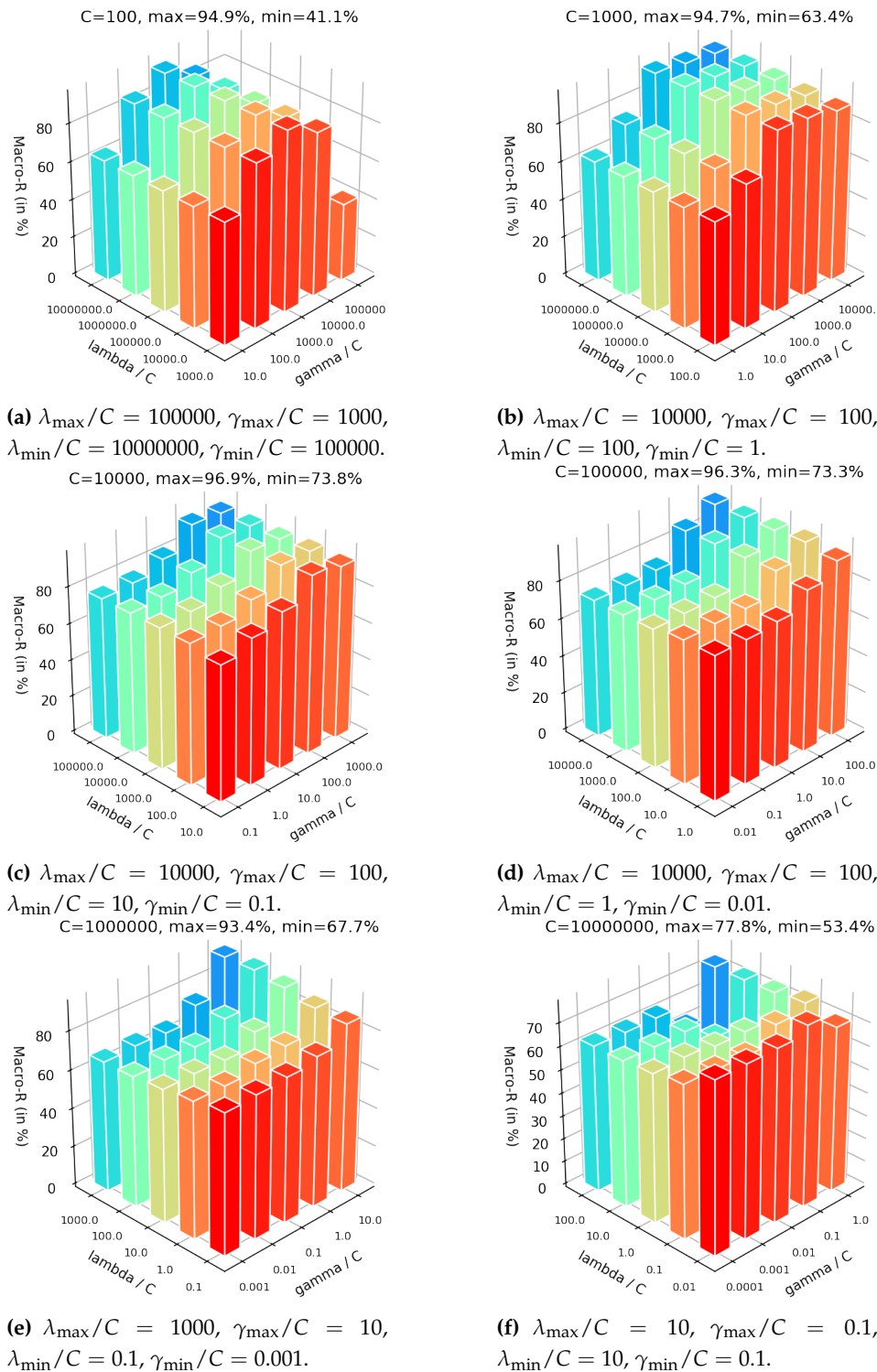
**Figure 5.** The results of performing our method on dataset 2. In (a) and (b), we show the recalls for the classes of Mu, Sa, and Do and their macro average recall by transferring D to E. In (c) and (d), we show those by transferring E to D.

In the Figure 7, we set the hyper-parameter  $C$  range from 100 to 10,000,000. It can be seen from these figures in Figure 7 that the maximum and minimum Macro-R are increased first and then decreased with the increasing of  $C$ , indicating that a small  $C$  incurs under-fitting in classification and a high  $C$  causes the over-fitting. Additionally, when  $C$  is set too small or too big, the overall accuracies are preserved in the large variation range highest 94.9% in Figure 7a and lowest 41.1% in Figure 7a or the overall just more than 70% slightly in Figure 7f. According to Figure 7c,d with  $C = 1000$  and  $C = 10000$ , respectively, the overall Macro-R are held steady highest 96.9% in Figure 7c and lowest 73.8% in Figure 7c and the accuracies are presented a trend of increasing gradually. Therefore, it is important to configure a moderate  $C$  first.



**Figure 6.** The results of performing our method on dataset 3. In (a) and (b), we show the recalls for the classes of Mu, Sa, and Sh and their macro average recall by transferring F to G. In (c) and (d), we show those by transferring G to F.

The influence of adjusting  $\lambda$  and  $\gamma$  are given by fixing the  $C$ . It can be seen from Figure 7 that the maximum Macro-R is achieved 96.9% in Figure 7c when set  $\lambda/C = 10,000$  and  $\gamma/C = 100$ . Additionally, the Macro-R almost increase first and then decrease with  $\gamma$  getting larger under fixed  $\lambda$  showing in Figure 7a–c. Moreover, these results show that the maximum is achieved almost when setting the  $\lambda$  larger than  $\gamma$  by two to four orders of magnitude. Since the  $\gamma$  control the contribution of semi-supervised regularization term that based on the manifold assumption, but the assumption is invalid when the dataset with data drift. We introduce the DDJA regularization term to suppress the data drift, so the coefficient  $\lambda$  should be more than  $\gamma$ .



**Figure 7.** Classification accuracies of JDA-S2ELM on dataset 1 over the variation of  $C$ ,  $\lambda$  and  $\gamma$ .

According to the analysis of setting  $C$ ,  $\lambda$ , and  $\gamma$  mentioned above, it can be concluded that: (i)  $C$  should be set a relatively larger range first to configure a moderate value; (ii) the setting of  $\lambda$  should be bigger than  $\gamma$  by two to four orders of magnitude. Here are some suggested settings:  $C \in [10^3, 10^6]$ ,  $\frac{\lambda}{C} \in [10^3, 10^4]$ ,  $\frac{\gamma}{C} \in [10^2, 10^3]$ .

Furthermore, we use Sobol method (In our experiment, the python tool "SALib" is employed, which can be found at <https://salib.readthedocs.io/en/latest/index.html>). Ref. [41] to implement a global sensitivity analysis where all parameters are varied simultaneously over the entire parameter

space. The sensitivities shown in Table 3 demonstrate that: (i)  $C$  and  $\lambda$  contribute mainly compared with  $\gamma$ , (ii)  $C$  contributes more than  $\lambda$  slightly, and (iii) the interactions between these parameters are weak, so they are relatively independent.

**Table 3.** Sensitivity analysis using Sobol method.

Param.	First-Order Sensitivity	Total Sensitivity	Param. 1	Param. 2	Second-Order Sensitivity
$C$	0.511	0.575	$C$	$\lambda$	0.074
$\lambda$	0.337	0.495	$C$	$\gamma$	0.008
$\gamma$	0.051	0.102	$\lambda$	$\gamma$	0.085

#### 4. Conclusions

In this paper, we have investigated the well logging-based lithology identification under data drift, thus proposing a transfer Extreme Learning Machine method to handle it. According to the projected maximum mean discrepancy (PMMD) criterion and extreme learning machine, we introduce a new PMMD criterion and then propose the DDJA-ELM to minimize the data drift between the source dataset and the target dataset. Additionally, in order to improve the prediction consistency within the target dataset, the manifold regularization is introduced to promote the DDJA-ELM to the DDJA-S2ELM. Extensive experiments have validated the high and stable accuracy of our method.

**Author Contributions:** Conceptualization, H.L. and W.L.; data curation, Y.C.; formal analysis, H.L.; funding acquisition, H.H.; investigation, Y.W. and T.X.; methodology, Y.W.; project administration, H.H.; Resources, Z.L.; Software, Z.L.; supervision, W.L.; validation, Y.C. and J.C.; visualization, J.C.; writing—original draft, H.L.; writing—review and editing, W.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the National Natural Science Foundation of China (61903353), SINOPEC Programmes for Science and Technology Development (PE19008-8), and Fundamental Research Funds for the Central Universities (WK210000013).

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

- Bergen, K.J.; Johnson, P.A.; Maarten, V.; Beroza, G.C. Machine learning for data-driven discovery in solid Earth geoscience. *Science* **2019**, *363*, eaau0323. [[CrossRef](#)] [[PubMed](#)]
- Van Natiyne, A.; Lindenbergh, R.C.; Bogaard, T.A. Machine Learning: New Potential for Local and Regional Deep-Seated Landslide Nowcasting. *Sensors* **2020**, *20*, 1425. [[CrossRef](#)] [[PubMed](#)]
- Verma, S.; Zhao, T.; Marfurt, K.J.; Devegowda, D. Estimation of total organic carbon and brittleness volume. *Interpretation* **2016**, *4*, T373–T385. [[CrossRef](#)]
- Keynejad, S.; Sbar, M.L.; Johnson, R.A. Assessment of machine-learning techniques in predicting lithofluid facies logs in hydrocarbon wells. *Interpretation* **2019**, *7*, SF1–SF13. [[CrossRef](#)]
- Zhang, T.F.; Tilke, P.; Dupont, E.; Zhu, L.C.; Liang, L.; Bailey, W. Generating geologically realistic 3D reservoir facies models using deep learning of sedimentary architecture with generative adversarial networks. *Pet. Sci.* **2019**, *16*, 541–549. [[CrossRef](#)]
- Askari, G.; Pour, A.B.; Pradhan, B.; Sarfi, M.; Nazemnejad, F. Band Ratios Matrix Transformation (BRMT): A Sedimentary Lithology Mapping Approach Using ASTER Satellite Sensor. *Sensors* **2018**, *18*, 3213. [[CrossRef](#)]
- Al-Anazi, A.; Gates, I. On the capability of support vector machines to classify lithology from well logs. *Nat. Resour. Res.* **2010**, *19*, 125–139. [[CrossRef](#)]
- Wang, K.; Zhang, L. Predicting formation lithology from log data by using a neural network. *Pet. Sci.* **2008**, *5*, 242–246. [[CrossRef](#)]
- Xie, Y.; Zhu, C.; Zhou, W.; Li, Z.; Liu, X.; Tu, M. Evaluation of machine learning methods for formation lithology identification: A comparison of tuning processes and model performances. *J. Pet. Sci. Eng.* **2018**, *160*, 182–193. [[CrossRef](#)]

10. Bhattacharya, S.; Carr, T.R.; Pal, M. Comparison of supervised and unsupervised approaches for mudstone lithofacies classification: Case studies from the Bakken and Mahantango-Marcellus Shale, USA. *J. Nat. Gas Sci. Eng.* **2016**, *33*, 1119–1133. [[CrossRef](#)]
11. Dev, V.A.; Eden, M.R. Formation lithology classification using scalable gradient boosted decision trees. *Comput. Chem. Eng.* **2019**, *128*, 392–404. [[CrossRef](#)]
12. Wang, X.; Yang, S.; Zhao, Y.; Wang, Y. Lithology identification using an optimized KNN clustering method based on entropy-weighted cosine distance in Mesozoic strata of Gaoqing field, Jiyang depression. *J. Pet. Sci. Eng.* **2018**, *166*, 157–174. [[CrossRef](#)]
13. Gu, Y.; Bao, Z.; Song, X.; Patil, S.; Ling, K. Complex lithology prediction using probabilistic neural network improved by continuous restricted Boltzmann machine and particle swarm optimization. *J. Pet. Sci. Eng.* **2019**, *179*, 966–978. [[CrossRef](#)]
14. Zhong, Y.; Zhao, L.; Liu, Z.; Xu, Y.; Li, R. Using a support vector machine method to predict the development indices of very high water cut oilfields. *Pet. Sci.* **2010**, *7*, 379–384. [[CrossRef](#)]
15. Dong, S.; Wang, Z.; Zeng, L. Lithology identification using kernel Fisher discriminant analysis with well logs. *J. Pet. Sci. Eng.* **2016**, *143*, 95–102. [[CrossRef](#)]
16. Guo, D.; Zhu, K.; Wang, L.; Li, J.; Xu, J. A new methodology for identification of potential pay zones from well logs: Intelligent system establishment and application in the Eastern Junggar Basin, China. *Pet. Sci.* **2014**, *11*, 258–264. [[CrossRef](#)]
17. Deng, C.; Pan, H.; Fang, S.; Konaté, A.A.; Qin, R. Support vector machine as an alternative method for lithology classification of crystalline rocks. *J. Geophys. Eng.* **2017**, *14*, 341–349. [[CrossRef](#)]
18. Saporetti, C.M.; da Fonseca, L.G.; Pereira, E. A Lithology Identification Approach Based on Machine Learning With Evolutionary Parameter Tuning. *IEEE Geosci. Remote. Sens. Lett.* **2019**, *16*, 1819–1823. [[CrossRef](#)]
19. Zhu, L.; Li, H.; Yang, Z.; Li, C.; Ao, Y.; others. Intelligent Logging Lithological Interpretation With Convolution Neural Networks. *Petrophysics* **2018**, *59*, 799–810. [[CrossRef](#)]
20. Ao, Y.; Li, H.; Zhu, L.; Ali, S.; Yang, Z. Logging lithology discrimination in the prototype similarity space with random forest. *IEEE Geosci. Remote. Sens. Lett.* **2018**, *16*, 687–691. [[CrossRef](#)]
21. Li, G.; Qiao, Y.; Zheng, Y.; Li, Y.; Wu, W. Semi-Supervised Learning Based on Generative Adversarial Network and Its Applied to Lithology Recognition. *IEEE Access* **2019**, *7*, 67428–67437. [[CrossRef](#)]
22. Li, Z.; Kang, Y.; Lv, W.; Zheng, W.X.; Wang, X.M. Interpretable Semisupervised Classification Method Under Multiple Smoothness Assumptions With Application to Lithology Identification. *IEEE Geosci. Remote. Sens. Lett.* **2020**, doi:10.1109/LGRS.2020.2978053. [[CrossRef](#)]
23. Dunham, M.W.; Malcolm, A.; Welford, J.K. Improved well log classification using semi-supervised algorithms. In *SEG Technical Program Expanded Abstracts 2019*; Society of Exploration Geophysicists: Houston, TX, USA, 2019; pp. 2398–2402.
24. Saleem, A.; Choi, J.; Yoon, D.; Byun, J. Facies classification using semi-supervised deep learning with pseudo-labeling strategy. In *SEG Technical Program Expanded Abstracts 2019*; Society of Exploration Geophysicists: San Antonio, TX, USA, 2019; pp. 3171–3175.
25. Maaten, L.v.d.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
26. Vapnik, V.N.; Vapnik, V. Statistical learning theory. *New York* **1998**, *1*, doi:10.1007/978-1-4419-1428-6\_5864. [[CrossRef](#)]
27. Huang, G.B.; Zhou, H.; Ding, X.; Zhang, R. Extreme learning machine for regression and multiclass classification. *IEEE Trans. Syst. Man Cybern. (Cybern.)* **2011**, *42*, 513–529. [[CrossRef](#)]
28. Lv, W.; Kang, Y.; Zheng, W.X.; Wu, Y.; Li, Z. Feature-temporal semi-supervised extreme learning machine for robotic terrain classification. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**. doi:10.1109/TCSII.2020.2990661. [[CrossRef](#)]
29. Mei, M.; Chang, J.; Li, Y.; Li, Z.; Li, X.; Lv, W. Comparative study of different methods in vibration-based terrain classification for wheeled robots with shock absorbers. *Sensors* **2019**, *19*, 1137. [[CrossRef](#)]
30. Ding, S.; Zhao, H.; Zhang, Y.; Xu, X.; Nie, R. Extreme learning machine: algorithm, theory and applications. *Artif. Intell. Rev.* **2015**, *44*, 103–115. [[CrossRef](#)]
31. Fang, X.; Cai, Y.; Cai, Z.; Jiang, X.; Chen, Z. Sparse Feature Learning of Hyperspectral Imagery via Multiobjective-Based Extreme Learning Machine. *Sensors* **2020**, *20*, 1262. [[CrossRef](#)]



32. Tian, Y.; Zhang, J.; Chen, L.; Geng, Y.; Wang, X. Selective Ensemble Based on Extreme Learning Machine for Sensor-Based Human Activity Recognition. *Sensors* **2019**, *19*, 3468. [[CrossRef](#)]
33. Huang, G.; Song, S.; Gupta, J.N.; Wu, C. Semi-supervised and unsupervised extreme learning machines. *IEEE Trans. Cybern.* **2014**, *44*, 2405–2417. [[CrossRef](#)]
34. Zhang, Y.; Wang, Y.; Zhou, G.; Jin, J.; Wang, B.; Wang, X.; Cichocki, A. Multi-kernel extreme learning machine for EEG classification in brain-computer interfaces. *Expert Syst. Appl.* **2018**, *96*, 302–310. [[CrossRef](#)]
35. Feng, L.; Xu, S.; Wang, F.; Liu, S.; Qiao, H. Rough extreme learning machine: A new classification method based on uncertainty measure. *Neurocomputing* **2019**, *325*, 269–282. [[CrossRef](#)]
36. Dai, H.; Cao, J.; Wang, T.; Deng, M.; Yang, Z. Multilayer one-class extreme learning machine. *Neural Netw.* **2019**, *115*, 11–22. [[CrossRef](#)] [[PubMed](#)]
37. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [[CrossRef](#)]
38. Quanz, B.; Huan, J. Large margin transductive transfer learning. In Proceedings of the 18th ACM Conference on Information and Knowledge Management, Hong Kong, China, 2–6 November 2009; pp. 1327–1336.
39. Shi, W.; Li, Z.; Lv, W.; Wu, Y.; Chang, J.; Li, X. Laplacian Support Vector Machine for Vibration-Based Robotic Terrain Classification. *Electronics* **2020**, *9*, 513. [[CrossRef](#)]
40. Gretton, A.; Borgwardt, K.; Rasch, M.; Schölkopf, B.; Smola, A.J. A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems*; NIPS: Vancouver, BC, Canada, 2007; pp.513–520.
41. Sobol, I.M. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Math. Comput. Simul.* **2001**, *55*, 271–280. [[CrossRef](#)]



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).