



Article Solving Regression Problems with Intelligent Machine Learner for Engineering Informatics

Jui-Sheng Chou^{1,*}, Dinh-Nhat Truong^{1,2} and Chih-Fong Tsai³

- ¹ Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taipei City 106335, Taiwan; d10605806@mail.ntust.edu.tw
- ² Department of Civil Engineering, University of Architecture Ho Chi Minh City (UAH), Ho Chi Minh City 700000, Vietnam
- ³ Department of Information Management, National Central University, Taoyuan City 320317, Taiwan; cftsai@mgt.ncu.edu.tw
- * Correspondence: jschou@mail.ntust.edu.tw

Abstract: Machine learning techniques have been used to develop many regression models to make predictions based on experience and historical data. They might be used singly or in ensembles. Single models are either classification or regression models that use one technique, while ensemble models combine various single models. To construct or find the best model is very complex and time-consuming, so this study develops a new platform, called intelligent Machine Learner (iML), to automatically build popular models and identify the best one. The iML platform is benchmarked with WEKA by analyzing publicly available datasets. After that, four industrial experiments are conducted to evaluate the performance of iML. In all cases, the best models determined by iML are superior to prior studies in terms of accuracy and computation time. Thus, the iML is a powerful and efficient tool for solving regression problems in engineering informatics.

Keywords: applied machine learning; classification and regression; data mining; ensemble model; engineering informatics

1. Introduction

Machine Learning (ML)-based methods for building prediction models have attracted abundant scientific attention and are extensively used in industrial engineering [1–3], design optimization of electromagnetic devices, and other areas [4,5]. The ML-based methods have been confirmed to be effective for solving real-world engineering problems [6–8]. Various supervised ML techniques (e.g., artificial neural network, support vector machine, classification and regression tree, linear (ridge) regression, and logistic regression) are typically used individually to construct single models and ensemble models [9,10]. To construct a series of models and identify the best one among these ML techniques, users need a comprehensive knowledge of ML and spend a significant effort building advanced models.

The primary objective of this research is to develop a user-friendly and powerful ML platform, called intelligent Machine Learner (iML), to help its users to solve real-world engineering problems with a shorter training time and greater accuracy than before. The iML can automatically build and scan all regression models, and then identify the best one. Novice users with no experience of ML can easily use this system. Briefly, the iML (1) helps users to make prediction model easily; (2) provides an overview of the parameter settings for the purpose of making objective choices; and (3) yields clear performance indicators, facilitating reading and understanding of the results, on which decisions can be based.

Four experiments were carried out to evaluate the performance of iML and were compared with previous studies. In the first experiment, empirical data concerning enterprise resource planning (ERP) for software projects by a leading Taiwan software provider over the last five years were collected and analyzed [1]. The datasets in the other three



Citation: Chou, J.-S.; Truong, D.-N.; Tsai, C.-F. Solving Regression Problems with Intelligent Machine Learner for Engineering Informatics. *Mathematics* **2021**, *9*, 686. https:// doi.org/10.3390/math9060686

Academic Editor: Florin Leon

Received: 18 February 2021 Accepted: 19 March 2021 Published: 23 March 2021

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



Copyright: © 2021 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/). experiments were published on the UCI website [11–13]. Specifically, the purpose of the second experiment was to train a regression model of comparing the performance of CPU processors by using some characteristics as input. The third experiment involved forecasting the demand supporting structured productivity and high levels of customer service, and the fourth experiment involved estimating the total bikes rented per day.

The rest of this paper is organized as follows. Section 2 reviews application of machine learning techniques in various disciplines. Section 3 presents the proposed methodology and iML framework. Section 4 introduces the evaluation metrics to measure accuracy of the developed system. Section 5 demonstrates iML's interface. Section 6 shows benchmarks between iML and WEKA (a free, open source program). Section 7 exhibits the applicability of iML in numerical experiments. Section 8 draws conclusions, and provides managerial implications and suggestions for future research.

2. Literature Review

Numerous researchers in various fields, such as ecology [14,15], materials properties [16–18], water resource [19], energy management [20], and decision support [21,22], use data-mining techniques to solve regression problems, and especially project-related problems [23,24]. Artificial neural network (ANN), support vector machine/regression (SVM/SVR) classification and regression tree (CART), linear ridge regression (LRR), and logistic regression (LgR) are the most commonly used methods for this purpose and are all considered to be among the best machine learning techniques [25–27]. Similarly, four popular ensemble models, including voting, bagging, stacking and tiering [28–30], can be built based on the meta-combination rules of aforementioned single models.

Chou (2009) [31] developed a generalized linear model-based expert system for estimating the cost of transportation projects. Dandikas et al. (2018) [32] assessed the advantages and disadvantages of regression models for predicting potential of biomethane. The results indicated that the regression method could predict variations in the methane yield and could be used to rank substrates for production quality. However, least squaresbased regression usually leads to overfitting a model, failure to find unique solutions, and issues dealing with multicollinearity among the predictors [33], so ridge regression, another type of regularized regression, is favorably integrated in this study to avoid the above problems. Additionally, Sentas and Angelis (2006) [34] investigated the possibility of using some machine learning methods for estimating categorical missing values in software cost databases. They concluded that multinomial logistic regression was the best for imputation owing to its superior accuracy.

The general regression neural network was originally designed chiefly to solve regression problems [24,35]. Caputo and Pelagagge (2008) [36] compared the ANN with the parametric methods for estimating the cost of manufacturing large, complex-shaped pressure vessels in engineer-to-order manufacturing systems. Their comparison demonstrated that the ANN was more effective than the parametric models, presumably because of its better mapping capabilities. Rocabruno-Valdés et al. (2015) [37] developed models based on ANN for predicting the density, dynamic viscosity, and cetane number of methyl esters and biodiesel. Similarly, Ganesan et al. (2015) [38] used ANN to predict the performance and exhaust emissions of a diesel electricity generator.

SVM was originally developed by Vapnik (1999) for classification (SVM) and regression (SVR) [39,40]. Jing et al. (2018) [41] used SVM to classify air balancing, which is a key element for heating, ventilating, air-conditioning (HAVC), and variable air volume (VAV) system installation, and is useful for improving the energy efficiency by minimizing unnecessary fresh air to the air-conditioned zones. The results demonstrated that SVM achieved 4.6% of relative error value and is a promising approach for air balancing. García-Floriano et al. (2018) [42] used SVR to model software maintenance (SM) effort prediction. The SVR model was superior to regression, neural networks, association rules and decision trees, with 95% confidence level.

The classification and regression tree method (CART), introduced by Breiman et al. (2017) [43], is an effective method to solve classification and regression problems [42]. Choi and Seo (2018) [44] predicted the fecal coliform in the North Han River, South Korea by CART models, the test results showed the total correct classification rates of the four models ranged from 83.7% to 93.0%. Ru et al. (2016) [45] used the CART model to predict cadmium enrichment levels in reclaimed coastal soils. The results showed that cadmium enrichment levels had an accuracy of 78.0%. Similarly, Li (2006) [16] used CART to predict materials properties and behavior. Chou et al. (2014, 2017) [26,46] utilized the CART method to modeling steel pitting risk and corrosion rate and forecasting project dispute resolutions.

In addition to the aforementioned single models, Elish (2013) [47] used voting ensemble for estimating software development effort. The ensemble model outperformed all the single models in terms of Mean Magnitude of Relative Error (MMRE), and achieved competitive percentage of observations whose Magnitude of Relative Error (MRE) is less than 0.25 (PRED (25)) and recently proposed Evaluation Function (EF) results. Wang at el. (2018) demonstrated that ensemble bagging tree (EBT) model could accurately predict hourly building energy usage with MAPE ranging from 2.97% to 4.63% [48]. Comparing to the conventional single prediction model, EBT is superior in prediction accuracy and stability. However, it requires more computation time and is short of interpretability owing to its sophisticated model structure.

Chen et al. (2019) [49] showed that the stacking model outperformed the individual models, achieving the highest R² of 0.85, followed by XGBoost (0.84), AdaBoost (0.84) and random forest (0.82). For the estimation of hourly PM2.5 in China, the stacking model exhibited relatively high stability, with R² ranging from 0.79 to 0.92. Basant at el. (2016) [50] proposed a three-tier quantitative structure-activity relationship (QSAR) model. This model can be used for the screening of chemicals for future drug design and development process and safety assessment of the chemicals. In comparison with previously studies, the QSAR models on the same endpoint property showed the encouraging statistical quality of the proposed models.

According to the reviewed literature, various machine learning platforms have been developed for the past decades, such as the Scikit-Learn Python libraries, Google's TensorFlow, WEKA and Microsoft Research's CNTK. Users can find it easy to use a machine learning tool and/or framework to solve numerous problems as per their needs [51]. ML-based approaches have been confirmed to be effective in providing decisive information. Since there is no best model suitable to predict all problems (the "No Free Lunch" theorem [52,53]), a comprehensive comparison of single and ensemble models embedded within an efficient forecasting platform for solving real-world engineering problems is imperatively needed. The iML platform proposed in this study can efficiently address this issue.

3. Applied Machine Learning

3.1. Classification and Regression Model

3.1.1. Artificial Neural Network (ANN)

Neural networks (or artificial neural networks) comprise information-processing units, which are similar to the neurons in the human brain, except that a neural network is composed of artificial neurons (Figure 1) [54]. Particular, back-propagation networks (BPNNs) are widely used, and are known to be the most effective network models [55,56].



Figure 1. Artificial neural network (ANN) model.

Equation (1) uses sigmoid function to activate each neuron in a hidden output layer, and the Scaled Conjugate Gradient Algorithm is used to calculate the weights of the network. BPNNs will be trained until the stopping criteria is reached by default settings in MATLAB.

$$net_k = \sum w_{kj}O_j \text{ and } y_k = f(net_k) = \frac{1}{1 + e^{-net_k}}$$
(1)

where net_k is the activation of the k^{th} neuron; *j* is the set of neurons in the preceding layer; w_{kj} is the weight of the connection between neuron *k* and neuron *j*; O_j is the output neuron *j*; and y_k is the sigmoid or logistic transfer function.

3.1.2. Support Vector Machine (SVM) and Support Vector Regression (SVR)

Developed by Cortes and Vapnik (1995) [57], SVM is used for binary classification problems. The SVM was created based on decision hyper-planes that determine decision boundaries in an input space or a high-dimensional feature space [40,58]. Binary classification can only classify samples into negative and positive while multi-class classification problems are complex (Figure 2). In this study, One Against All (OAA) is used to solve multiple classification problems.



Figure 2. Support Vector Machine (SVM) and Support Vector Regression (SVR) models.

The OAA-SVM constructs *m* SVM models for *m*-class classification problems, and the i^{th} SVM model is trained based on the dataset of the i^{th} class which includes a positive

class and a negative class. In training, a set of *l* data points $(x_i, y_i)_{i=1}^l$, where $x_i \in \mathbb{R}^n$ the input data, and $y_i \in (1, 2, ..., m)$ is the class label of x_i ; the *i*th SVM model is solved using the following optimization problem equation [59].

$$\min_{w^i,b,\xi} J\left(w^i,b,\xi\right) = \frac{1}{2} \left(w^i\right)^T w^i + C \sum_{i=1}^l \xi^i_j$$
⁽²⁾

subject to:
$$\begin{cases} (w^{i})^{T} \varphi(x_{j}) + b^{i} \geq 1 - \xi_{j}^{i}, y_{j} = i, \\ (w^{i})^{T} \varphi(x_{j}) + b^{i} \leq -1 + \xi_{j}^{i}, y_{j} \neq i, \\ \xi_{j}^{i} \geq 0, \ j = 1, \dots, l. \end{cases}$$
(3)

When the SVM models have been solved, the class label of example *x* is predicted as follows:

$$y(x) = \arg\max_{i=1\dots m} \left(\left(w^i \right)^T \varphi(x) + b^i \right)$$
(4)

where *i* is the *i*th SVM model; w^i is a vector normal to the hyper-plane; b^i is a bias, $\varphi(x)$ is a nonlinear function that maps *x* to a high-dimension feature space, ξ^i is the error in misclassification, and $C \ge 0$ is a constant that specifies the trade-off between the classification margin and the cost of misclassification.

To train the SVM model, radial basic function (RBF) kernel maps samples non-linearly into a feature space with more dimensions. In this study, the RBF kernel is used as SVM kernel function.

$$K(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)$$
(5)

where σ is a positive parameter that controls the radius of RBF kernel function.

Support vector regression (SVR) [40] is one version of SVM. SVR computes a linear regression function for the new higher-dimensional feature space using ε -insensitive loss while simultaneously reducing model complexity of the model by minimizing $||w||^2$. This process can be implemented by introducing (non-negative) slack variables ξ_i , ξ_i^* to measure the deviation in training samples outside the ε -insensitive zone. The SVR can be formulated as the minimization of the following equation:

$$\min_{w,b,\xi} J(w,b,\xi) = \frac{1}{2} (w)^T w + C \sum_{i=1}^{l} (\xi_i + \xi_i^*)$$
(6)

subject to :
$$\begin{cases} y_i - f(x_i, w) \le \varepsilon + \xi_i^* \\ f(x_i, w) - y_i \le \varepsilon + \xi_i \\ \xi_i^*, \xi_i \ge 0, \ i = 1, \dots, n \end{cases}$$
(7)

When SVR model has been solved, the value of example *x* is predicted as follows.

$$f(x) = \sum (\alpha_i - \alpha_i^*) K(x_i, x) + \mathbf{b}$$
(8)

where $K(x_i, x)$ is the kernel function and α_i^* , α_i are Lagrange multipliers in the dual function.

3.1.3. Classification and Regression Tree (CART)

Classification and regression tree technique is described as a tree on which each internal (non-leaf) node represents a test of an attribute, each branch represents the test result, and each leaf (or terminal) node has a class label and class result (Figure 3) [60]. The tree is "trimmed" until total error is minimized to optimize the predictive accuracy of the tree by minimizing the number of branches. The training CART is constructed through the Gini index. The formulas are as follows.



Figure 3. The classification and regression tree (CART) model.

$$g(t) = \sum_{j \neq i} p(j|t)p(i|t)$$
(9)

$$p(j|t) = \frac{p(j,t)}{p(t)}$$
(10)

$$p(j,t) = \frac{p(j)N_j(t)}{N_j} \tag{11}$$

$$p(t) = \sum_{j} p(j, t) \tag{12}$$

Gini index =
$$1 - \sum p(j, t)^2$$
 (13)

where *i* and *j* are the categorical variables in each item; $N_j(t)$ is the recorded number of nodes *t* in category *j*; and N_j . is the recorded number of the root nodes in category *j*; and p(j) is the prior probability value for category *j*.

3.1.4. Linear Ridge Regression (LRR) and Logistic Regression (LgR)

Statistical models of the relationship between dependent variables (response variables) and independent variables (explanatory variables) are developed using linear regression (Figure 4). The general formula for multiple regression models is as follows.

$$y = f(x) = \beta_o + \sum_{j=1}^n \beta_j x_j + \varepsilon$$
(14)

where *y* is a dependent variable; β_0 is a constant; β_j is a regression coefficient (j = 1, 2, ..., n), and ε is an error term.

Linear ridge regression (LRR) is a regularization technique that can be used together with generic regression algorithms to model highly correlated data [61,62]. Least squares

method is a powerful technique for training the LRR model, which denotes β to minimize the Residual Sum Squares (RSS)-function. Therefore, the cost function is presented as below.

$$Cost(\beta) = RSS(\beta) = \sum_{i=1}^{l} (y - y')^2 + \lambda \left(\sum_{j=1}^{n} \beta_j^2\right)$$
(15)

$$y' = \beta_0 + \sum \beta_j x_j \tag{16}$$

where λ is a pre-chosen constant, which is the product of a penalty term and the squared norm in the β vector of regression method, and y' is the predicted values.



Figure 4. Linear Ridge Regression (LRR) and Logistic Regression (LgR) models.

Statistician David Cox developed logistic regression in 1958 [63]. An explanation of logistic regression begins with an explanation of the standard logistic function. Equation (17) mathematically represents the logistic regression model.

$$p(x) = \frac{1}{1 + e^{-(\beta_o + \sum_{j=1}^n \beta_j x_j)}}$$
(17)

where p(x) is the probability that the dependent variable equals a "success" or "case" rather than a failure or non-case. β_0 and β_j are found by minimizing cost function defined in Equation (18).

$$Cost(\beta) = -\left(\sum_{i=1}^{l} (y_i \ln(p(\mathbf{x}_i)) + (1 - y_i) \ln(1 - p(\mathbf{x}_i)))\right) + \frac{\lambda}{2} \sum_{j=1}^{n} \beta_j^2$$
(18)

where y_i is the observed outcome of case x_i , having 0 or 1 as possible values [64]

3.2. Ensemble Regression Model

In this study, several ensemble schemes, including voting, bagging, stacking, and tiering were investigated using the input data and described as below.

- *Voting*: The voting ensemble model combines the outputs of the single models using a meta-rule. The mean of the output values is used in this study. According to the adopted ML models, 11 voting models are trained in this study, including (1) ANN + SVR, (2) ANN + CART, (3) ANN + LRR, (4) SVR + CART, (5) SVR + LRR, (6) CART + LRR, (7) ANN + SVR + CART, (8) ANN + CART + LRR, (9) ANN + CART + LRR, (10) SVR + CART + LRR, (11) ANN + SVR + CART + LRR. Figure 5a presents the voting ensemble model.
- *Bagging*: The bagging ensemble model duplicates samples at random, and each regression model predicts values from the samples independently. The meta-rule is

applied to all of the outputs in this study. Bagging ensemble model is depicted at Figure 5b.

• *Stacking*: The stacking ensemble model is a two-stage model, and Figure 5c describes the principle of the model. In stage 1, each single model predicts one output value. Then, these outputs are used as inputs to train a model by these machine learning techniques again to make a meta-prediction in stage 2. There are four stacking models herein, including ANN (ANN, SVR, CART, LRR); SVR (ANN, SVR, CART, LRR); CART (ANN, SVR, CART, LRR); LRR (ANN, SVR, CART, LRR).



Figure 5. Ensemble models.

• *Tiering*: Figure 5d illustrates the tiering ensemble model. There are two tiers inside a tiering ensemble model in this study. The first tier is to classify data into *k* classes on the basis of T value [18]. Machine learning technique in the first tier for classifying data needs to be identified. After classifying the data, the regression machine learning is used to train each data (Sub Data) of each class (second tier) to predict results. In the iML, we developed three types of models, including 2-class, 3-class, and 4-class. The equation for calculating T value is:

$$T = \frac{y_{max} + y_{min}}{k}$$
(19)

where T is standard value, k is the number of classes, and y_{max} and y_{min} are the maximum and minimum of actual values, respectively.

3.3. K-Fold Cross Validation

K-fold cross validation is used to compare two or more prediction models. This method randomly divides a sample into a training sample and a test sample by splitting into K subsets. K-1 subsets are selected to train the model while the other is used to test, and this training process is repeated K times (Figure 6). To compare models, the average of performance results (e.g., RMSE, and MAPE) is computed. Kohavi (1995) stated that K = 10 provides analytical validity, computational efficiency, and optimal deviation [65]. Thus, K = 10 is used in this study. Performance metrics will be explained in details Section 4.



Figure 6. K-fold cross-validation method.

3.4. Intelligent Machine Learner Framework

Figure 7 presents the structure of iML. In stage 1 (data preprocessing), the data is classified distinctly for particular use in the Tiering ensemble model. Meanwhile, all data is divided into two main data groups, namely, learning data and test data, and the learning data is duplicated for training ensemble models.

At the next stage, all retrieved data is automatically used for training models, which include single models (ANN, SVR, LRR, and CART), and ensemble models (voting, bagging, stacking, and tiering). Notably, the tiering ensemble model needs to employ a classification technique to assign a class label to the original input at the first tier. A corresponding regression model for the particular class is then adopted at the second tier to obtain the predictive value [17,26].



Figure 7. Intelligent machine leaner framework.

Finally, in stage 3 (find the best model), the predictive performances of all the models learned (trained) in stage 2 using test dataset are compared to identify the best models. Section 4 describes the performance evaluation metrics in detail.

4. Mathematical Formulas for Performance Measures

To measure the performance of classification models, the accuracy, precision, sensitivity, specificity and the area under the curve (AUC) are calculated. For the regression models, five-performance measures, (i.e., correlation coefficient (R), mean absolute error (MAE), mean absolute percentage error (MAPE), root mean squared error (RMSE), and total error rate (TER)) are calculated. Table 1 presents a confusion matrix and Table 2 exhibits those performance measures [17,66].

Table 1. Confusion matrix.

		Actual Class				
		Positive	Negative			
Predicted class	Positive	True positive	False Negative			
	Negative	False positive	True negative			

In Table 2, MAE is the mean absolute difference between the prediction and the actual value. MAPE represents the mean percentage error between prediction and actual value, the smaller value of MAPE, the better prediction result achieved by the model. The MAPE is the index typically used to evaluate the accuracy of prediction models. RMSE represents the dispersion of errors by a prediction model. The statistical index that shows the linear correlation between two variables is denoted as R. Lastly, TER is the total difference of predicted and actual values [17].

Table 2. Mathematical formulas for performance measures.

Measure	Formula	Measure	Formula
Accuracy	Accuracy = $\frac{tp+tn}{tp+fp+tn+fn}$	Mean absolute error	$MAE = \frac{1}{n} \sum_{i=1}^{n} y_i - y'_i $
Precision	$Precision = \frac{tp}{tp+fp}$	Mean absolute percentage error	$MAPE = \frac{1}{n} \left \frac{y_i - y'_i}{y_i} \right $
Sensitivity	Sensitivity = $\frac{tp}{tp+fn}$	Root mean square error	$ ext{RMSE} = \sqrt{rac{1}{n}\sum\limits_{i=1}^n \left(y_i - y_i' ight)^2}$
Specificity	Specificity = $\frac{tn}{tn+fp}$	Correlation coefficient	$\mathbf{R} = \frac{n \sum y_i \cdot y'_i - (\sum y_i)(\sum y'_i)}{\sqrt{n(y_i^2) - (\sum y_i)^2} \sqrt{n(y_i'^2) - (\sum y'_i)^2}}$
Area under the curve	$AUC = \frac{1}{2} \Big[\Big(\frac{tp}{tp+fn} \Big) + \Big(\frac{tn}{tn+fp} \Big) \Big]$	Total error rate	$\text{TER} = \frac{ \sum_{i=1}^{n} y_i' - \sum_{i=1}^{n} y_i }{\sum_{i=1}^{n} y_i}$

tp is the true positives (number of correctly recognized class examples); tn is the true negatives (number of correctly recognized examples that do not belong to the class); fp is the number of false positives (number of examples that were incorrectly assigned to a class); fn is the number of false negatives (number of examples that were not assigned to a class); y_i is actual value; y'_i is predicted value; n is sample size.

The goal is to identify the model that yields the lowest error of test data. To obtain a comprehensive performance measure, the five statistical measures (RMSE, MAE, MAPE, 1-R, and TER) were combined into a synthesis index (SI) using Equation (20). Based on the SI values, the best model is identified.

$$SI = \frac{1}{m_p} \sum_{i=1}^{m_p} \left(\frac{P_i - P_{min,i}}{P_{max,i} - P_{min,i}} \right)$$
(20)

where m_p = number of performance measures; $P_i = i^{\text{th}}$ performance measure; and $P_{min,i}$ and $P_{max,i}$ are the maximum and minimum of i^{th} measure. The SI range is 0–1; the SI value close to 0 indicates a better accuracy of the predictive model.

5. Design and Implementation of iML Interface

The iML was developed in MATLAB R2016a on a PC with an Intel Core i5-750 CPU, a clock speed of 3.4 GHz, and 8 GB of RAM, running Windows 10. Figure 8 presents a user-friendly interface for iML. First, users select models on setting-parameters board and set the parameters for the chosen models, which will be trained and analyzed. Next, users choose whether to test with either "K-Fold Validation" or "Percentage Split" before uploading the data. Notably, if "Percentage Split" is selected, the user only has to input percentage value of learning data. Then, users click on the "Run" button to train the model. Finally, the "Make Report" function is to create a report containing performance metrics of all selected models and the identified best model. Figure 9 displays a snapshot of report file in notepad.





REPORT_EXAMPLE_rev4 - Notepad	-		×
File Edit Format View Help			
INL RESULT This program is written by Pro. Jui-Sheng Chou and Ph.D. student: Dinh-Nhat Truong PiM Lab, Taipei, Taiwan, March 2020 Compution time : 238.8664 (seconds) KFold validation : OFF Rate of learn: 70 Percent The best model : BAGGING ENSEMBLE MODEL :ANNS MODEL Performance results			- ^
Model RMSE_learn RMSE_test MAE_learn MAE_test MAPE_learn MAPE_test R_learn R_test TER_learn	ITER_	test	ī
model1 0.046277 0.10445 0.030087 0.073216 0.066684 0.14118 0.97886 0.88264 1.0813e-15 model2 0.051805 0.079972 0.03707 0.056192 0.078495 0.1075 0.9743 0.92904 0.0010585 model3 0.048683 0.06932 0.034289 0.046312 0.071524 0.086673 0.9766 0.9479 0.0018796 mode14 0.0046783 0.21427 0.0026944 0.16924 0.0069949 0.38678 0.99979 0 7.2088e-16 mode15 0.57987 0.58904 0.4323 0.45098 0.57622 0.61409 -0.58237 -0.56974 0.68722 mode16 0.63942 0.64601 0.56765 0.56528 0.84422 0.82113 -0.68026 -0.71723 0.9124 mode17 0.64791 0.652 0.59556 0.58429 0.92369 0.874 -0.51246 -0.62971 0.96428	0.0 0.0 0.0 0.0 0.6 0.9 0.9	054693 026114 018722 0086396 8288 0262 4298	- - - - - - - - - - - - - - - - - - -
Note: Model1 : SINGLE MODEL : CART MODEL Model2 : VOTING ENSEMBLE MODEL : ANN_CART_LR Model3 : BAGGING ENSEMBLE MODEL : ANN® MODEL Model4 : STACKING ENSEMBLE MODEL : CART MODEL Model5 : TIERING ENSEMBLE MODEL : 2-CLASS:CART(ANN,ANN) Model6 : TIERING ENSEMBLE MODEL : 3-CLASS:CART(ANN,ANN) Model7 : TIERING ENSEMBLE MODEL : 4-CLASS:CART(LRR,LRR,LRR)			_

Figure 9. Snapshot of report file.

6. Benchmarks between iML and WEKA

6.1. Publicly Available Datasets

Table 3 shows the publicly available datasets from the UCI Machine Learning Repository (https://archive.ics.uci.edu/mL/index.php; accessed 1 March 2021). The iML is benchmarked with WEKA (a free, open source program) using hold-out validation and K-fold cross-validation on the target datasets. All algorithm parameters are set default for both iML and WEKA platforms.

Table 3.	Characteristic o	f data from	uCI Machine	Learning Re	epository.

UCI Data Set	No. of Samples	No. of Attributes	Output Information
Concrete Compressive Strength (Yeh (2006) [67])	1030	8	Concrete compressive strength (MPa)
Real estate valuation (Yeh and Hsu (2018) [68])	414	6	Y = house price of unit area (10,000 New Taiwan Dollar/Ping, where Ping is a local unit, 1 Ping = 3.3 m squared)
Energy efficiency (Tsanas and	768	8	y1 Heating Load (kW)
Xifara (2012) [69])	700	0	y2 Cooling Load (kW)
Airfoil Self-Noise (Lau and López (2009) [70])	1503	5	Scaled sound pressure level (dB).

6.2. Hold-Out Validation

In this test, datasets are randomly partitioned into 80% and 20% for learning and test, respectively. Tables 4–8 show the one-time performance results on these five datasets. A model with a normalized SI value of 0.000 is the best prediction model among all the models tested by iML and WEKA. Notably, the best model can be automatically identified by iML with "one-click". To train models with WEKA, the users need to build each model individually. Moreover, iML gives better test results of single, voting and bagging models than those of WEKA. Based on the benchmark results, iML is effective to find the best model in the hold-out validation.

Table 4. Test results by WEKA and iML on concrete compressive strength dataset via hold-out validation.

	WEKA						iN	1L		
Model	R	RMSE (MPa)	MAE (MPa)	MAPE (%)	SI (Ranking)	R	RMSE (MPa)	MAE (MPa)	MAPE (%)	SI (Ranking)
I. Single		CA	RT			ANN				
0	0.927	6.546	5.170	18.770	0.142 (7)	0.946	5.302	3.728	12.673	0.023 (3)
II. Voting		ANN +	- CART				ANN +	- CART		
Ũ	0.936	6.202	4.930	19.090	0.124 (6)	0.956	4.771	3.550	12.723	0.000 (1)
III. Bagging		CA	RT				AN	JN		
00 0	0.960	5.044	3.983	15.130	0.032 (4)	0.951	5.056	3.647	12.249	0.010 (2)
IV. Stacking		(*) C	ART				(*) I	LRR		
Ū.	0.939	5.986	4.792	17.520	0.104 (5)	0.444	14.829	11.779	56.775	1.000 (8)

Note: (*) is (ANN + SVR + CART + LRR); bold value denotes the best overall performance.

	WEKA iML									
Model	R	RMSE (U)	MAE (U)	MAPE (%)	SI (Ranking)	R	RMSE (U)	MAE (U)	MAPE (%)	SI (Ranking)
I. Single		CA	RT				AN	JN		
0	0.740	10.762	5.882	13.210	0.321 (6)	0.871	6.630	4.912	13.591	0.049 (3)
II. Voting		ANN + CA	ART + LRI	R			ANN +	- CART		
Ū.	0.745	11.054	5.908	12.780	0.327 (7)	0.877	6.615	4.739	12.867	0.030 (2)
III. Bagging		CA	RT				CA	.RT		
00 0	0.770	10.321	5.281	11.760	0.246 (4)	0.884	6.485	4.381	12.305	0.000 (1)
IV. Stacking		(*) C.	ART				(*) A	NN		
Ũ	0.744	10.748	5.774	12.940	0.311 (5)	0.391	12.638	10.411	33.807	1.000 (8)

Table 5. Test results by WEKA and iML on real estate dataset via hold-out validation.

Note: (*) is (ANN + SVR + CART + LRR); bold value denotes the best overall performance; U: house price of unit area (10,000 New Taiwan Dollar/Ping, where Ping is a local unit, 1 Ping = 3.3 m squared).

Table 6. Test results by WEKA and iML on energy efficiency data set (Heating load) via hold-out validation.

	WEKA iML							1L		
Model	R	RMSE (kW)	MAE (kW)	MAPE (%)	SI (Ranking)	R	RMSE (kW)	MAE (kW)	MAPE (%)	SI (Ranking)
I. Single		CA	RT				AN	JN		
0	0.996	0.914	0.646	3.300	0.418 (6)	0.999	0.488	0.354	1.700	0.046 (3)
II. Voting		ANN +	CART				ANN +	CART		
0	0.996	0.929	0.729	3.820	0.449 (7)	0.999	0.495	0.336	1.617	0.045 (2)
III. Bagging		CA	RT				AN	JN		
	0.997	0.870	0.619	3.210	0.354 (5)	0.999	0.426	0.311	1.519	0.000 (1)
IV. Stacking		(*) C	ART				(*) I	.RR		
	0.998	0.754	0.524	2.480	0.231 (4)	0.998	3.454	3.226	17.658	1.000 (8)

Note: (*) is (ANN + SVR + CART + LRR); bold value denotes the best overall performance.

Table 7. Test results by WEKA and iML on energy efficiency dataset (Cooling load) via hold-out validation.

		WE	KA							
Model	R	RMSE (kW)	MAE (kW)	MAPE (%)	SI (Ranking)	R	RMSE (kW)	MAE (kW)	MAPE (%)	SI (Ranking)
I. Single		CA	RT				AN	JN		
0	0.986	1.524	1.006	3.900	0.320 (6)	0.992	1.231	0.884	3.577	0.038 (2)
II. Voting		ANN +	CART				ANN +	CART		
0	0.987	1.504	1.064	4.330	0.293 (4)	0.988	1.509	0.982	3.544	0.222 (3)
III. Bagging		CA	RT				AN	JN		
00 0	0.986	1.565	1.046	4.030	0.358 (7)	0.993	1.177	0.809	3.165	0.000 (1)
IV. Stacking		(*) S	SVR				(*) I	LRR		
0	0.986	1.537	0.979	3.700	0.314 (5)	0.989	4.290	3.762	17.305	1.000 (8)

Note: (*) is (ANN + SVR + CART + LRR); bold value denotes the best overall performance.

	WEKA iML									
Model	R	RMSE (dB)	MAE (dB)	MAPE (%)	SI (Ranking)	R	RMSE (dB)	MAE (dB)	MAPE (%)	SI (Ranking)
I. Single	CART ANN									
0	0.898	3.185	2.339	1.880	0.502 (6)	0.953	2.149	1.577	1.259	0.044 (2)
II. Voting		ANN +	CART				ANN +	CART		
Ũ	0.893	3.471	2.649	2.100	0.591 (7)	0.952	2.163	1.633	1.301	0.058 (3)
III. Bagging		CA	RT				AN	JN		
00 0	0.922	2.902	2.135	1.710	0.332 (4)	0.958	2.031	1.494	1.194	0.000 (1)
IV. Stacking		(*) C	ART				(*) I	RR		
	0.905	3.082	2.271	1.820	0.450 (5)	0.952	7.050	5.648	4.613	1.000 (8)

Table 8. Test results by WEKA and iML on airfoil self-noise dataset via hold-out validation.

Note: (*) is (ANN + SVR + CART + LRR); bold value denotes the best overall performance.

6.3. K-Fold Cross-Validation

Tenfold cross-validation is used to evaluate the generalized performance of WEKA and iML. Tables 9–13 show the average performance measures of five datasets, respectively. Similarly, iML identifies better models in single, voting, and bagging schemes than those trained by WEKA. The best model for each dataset is automatically determined by iML. Therefore, iML is a powerful tool to find the best model in the cross-fold validation.

Table 9. Performance of WEKA and iML on concrete compressive strength dataset via tenfold cross-validation.

	WEKA iML									
Model	R	RMSE (MPa)	MAE (MPa)	MAPE (%)	SI (Ranking)	R	RMSE (MPa)	MAE (MPa)	MAPE (%)	SI (Ranking)
I. Single		CA	.RT				AN	JN		
0	0.923	6.434	4.810	15.510	0.228 (5)	0.946	5.411	4.003	13.866	0.154 (3)
II. Voting		ANN +	- CART				ANN +	- CART		
Ū.	0.917	6.823	5.213	17.230	0.265 (7)	0.955	4.903	3.506	12.397	0.111 (2)
III. Bagging		CA	.RT				CA	.RT		
	0.932	6.082	4.598	15.030	0.205 (4)	0.980	3.359	2.432	8.356	0.000 (1)
IV. Stacking		(*) 5	SVR				(*) A	NN		
	0.924	6.436	4.852	15.530	0.229 (6)	0.613	14.381	10.867	44.759	1.000 (8)

Note: (*) is (ANN + SVR + CART + LRR); bold value denotes the best overall performance.

Table 10. Performance of WEKA and iML on real estate valuation dataset via tenfold cross-validation.

	WEKA iML									
Model	R	RMSE (U)	MAE (U)	MAPE (%)	SI (Ranking)	R	RMSE (U)	MAE (U)	MAPE (%)	SI (Ranking)
I. Single		CA	.RT				AN	JN		
0	0.807	8.021	5.197	15.270	0.314 (5)	0.813	8.011	5.388	14.991	0.315 (6)
II. Voting		SVR +	CART				ANN + CA	ART + LRI	R	
0	0.805	8.091	5.198	15.090	0.315 (7)	0.821	7.878	5.376	15.116	0.308 (4)
III. Bagging		CA	.RT				CA	.RT		
00 0	0.828	7.637	5.017	14.930	0.280 (2)	0.925	4.774	3.201	8.974	0.000 (1)
IV. Stacking	(*) SVR (*) ANN									
0	0.819	7.823	4.969	14.440	0.284 (3)	0.432	12.309	9.526	32.267	1.000 (8)

Note: (*) is (ANN + SVR + CART + LRR); bold value denotes the best overall performance; U: house price of unit area (10,000 New Taiwan Dollar/Ping, where Ping is a local unit, 1 Ping = 3.3 m squared).

	WEKA									
Model	R	RMSE (kW)	MAE (kW)	MAPE (%)	SI (Ranking)	R	RMSE (kW)	MAE (kW)	MAPE (%)	SI (Ranking)
I. Single		CA	RT				AN	JN		
	0.995	1.046	0.712	3.200	0.459 (7)	0.999	0.484	0.360	1.722	0.049 (2)
II. Voting		ANN +	CART				ANN +	CART		
Ū.	0.997	0.853	0.641	3.190	0.309 (4)	0.999	0.497	0.352	1.602	0.053 (3)
III. Bagging		CA	RT				AN	JN		
00 0	0.997	0.915	0.633	2.890	0.324 (5)	0.999	0.384	0.291	1.409	0.000 (1)
IV. Stacking		(*) S	SVR			(*) LRR				
0	0.996	0.872	0.639	2.990	0.337 (6)	0.998	3.522	3.226	18.181	1.000 (8)

Table 11. Performance of WEKA and iML on energy efficiency dataset (Heating load) via tenfold cross-validation.

Note: (*) is (ANN + SVR + CART + LRR); bold value denotes the best overall performance.

Table 12. Performance of WEKA and iML on energy efficiency dataset (Cooling load) via tenfold cross-validation.

	WEKA									
Model	R	RMSE (kW)	MAE (kW)	MAPE (%)	SI (Ranking)	R	RMSE (kW)	MAE (kW)	MAPE (%)	SI (Ranking)
I. Single		CA	RT				AN	JN		
0	0.982	1.812	1.183	4.160	0.460 (5)	0.993	1.140	0.799	3.161	0.150 (2)
II. Voting		ANN +	CART				ANN +	CART		
0	0.982	1.831	1.276	4.770	0.491 (7)	0.989	1.415	0.900	3.206	0.250 (3)
III. Bagging		CA	RT				AN	JN		
00 0	0.983	1.785	1.160	4.070	0.444 (4)	0.997	0.808	0.556	2.129	0.000 (1)
IV. Stacking		(*) 5	SVR				(*) I	.RR		
	0.982	1.827	1.195	4.210	0.465 (6)	0.989	4.108	3.619	17.253	1.000 (8)

Note: (*) is (ANN + SVR + CART + LRR); bold value denotes the best overall performance.

Table 13. Performance of WEKA and iML on airfoil self-noise dataset via tenfold cross-validation.

		WE	KA				iN	ſL		
Model	R	RMSE (dB)	MAE (dB)	MAPE (%)	SI (Ranking)	R	RMSE (dB)	MAE (dB)	MAPE (%)	SI (Ranking)
I. Single		CA	RT				AN	JN		
0	0.877	3.314	2.381	1.910	0.497 (5)	0.946	2.239	1.660	1.331	0.152 (2)
II. Voting		ANN +	CART				ANN +	CART		
0	0.851	3.685	2.747	2.220	0.641 (7)	0.946	2.246	1.664	1.334	0.152 (3)
III. Bagging		CA	RT				CA	RT		
	0.911	2.906	2.160	1.730	0.352 (4)	0.971	1.727	1.271	1.023	0.000 (1)
IV. Stacking		(*) I	LRR				(*) I	RR		
	0.874	3.374	2.494	1.990	0.525 (6)	0.946	6.894	5.587	4.562	1.000 (8)

Note: (*) is (ANN + SVR + CART + LRR); bold value denotes the best overall performance.

6.4. Discussion

Single, voting, bagging, and stacking models are compared using WEKA and iML, except for the tiering method, which is not available in WEKA. Additionally, unlike manual construction of individual models in WEKA interface, iML can automatically build and identify the best model for the imported datasets. Hold-out validation and tenfold cross-validation are used to evaluate the performance results (R, MAE, RMSE, and MAPE) in each scheme (single, voting, bagging, and stacking). The analytical results of either validation

show that most of the models trained by iML are superior to those trained by WEKA using the same datasets. Hence, iML is an effective platform to solve regression problems.

7. Numerical Experiments

This section validates iML by using various industrial datasets, including (1) enterprise resource planning data [1], (2) CPU computer performance data [12], (3) customer data for a logistics company [13], and (4) daily data bike rentals [11]. Table 14 presents the initial parameter settings for these problems.

		ANN	S	VR and S	VM	LRR and LgR	CART
Experiment	Model	Number of Hidden Node	С	Sigma	Epsilon	Lambda	Min Leaf
	Single regression model	30	$7.3 imes10^6$	45.67	$1.0 imes 10^{-5}$	$1.0 imes10^{-8}$	1
1	Single classification model	30	41,703	3.67	-	$1.0 imes10^{-5}$	1
	Ensemble regression model	30	$7.3 imes10^6$	45.67	$1.0 imes 10^{-5}$	$1.0 imes 10^{-8}$	1
	Single regression model	30	$7.3 imes10^6$	20.03	$1.0 imes10^{-5}$	$1.0 imes 10^{-8}$	1
2	Single classification model	30	4200	3.40	-	$1.0 imes10^{-5}$	1
	Ensemble regression model	30	$7.3 imes10^6$	45.67	$1.0 imes 10^{-5}$	$1.0 imes 10^{-8}$	1
	Single regression model	30	$7.3 imes10^6$	30.00	$1.0 imes 10^{-5}$	$1.0 imes 10^{-8}$	1
3	Single classification model	20	41,703	3.67	-	$1.0 imes10^{-5}$	1
	Ensemble regression model	30	$7.3 imes10^6$	30.00	$1.0 imes 10^{-5}$	$1.0 imes10^{-8}$	1
	Single regression model	15	$7.3 imes10^6$	45.67	$1.0 imes 10^{-5}$	$1.0 imes10^{-8}$	1
4	Single classification model	15	41,703	3.67	-	$1.0 imes10^{-5}$	1
	Ensemble regression model	15	$7.3 imes 10^6$	45.67	$1.0 imes 10^{-5}$	$1.0 imes 10^{-8}$	1

Table 14. Parameter setting.

7.1. Enterprise Resource Planning Software Development Effort

Enterprise Resource Planning (ERP) data for 182 software projects of a leading Taiwan software provider over the last five years was collected, analyzed, and tested with K-fold cross validation.

7.1.1. Variable Selection

Experienced in-house project managers were interviewed to identify factors that affect the ERP software development effort (SDE). There are 182 samples and 17 attributes, and Table 15 summarizes the descriptive statistical data in details. The input and output attributes are defined by Chou el at. (2012) [1].

7.1.2. iML Results

iML automatically trains the models and calculates the performance values. Then, it compares the SI values (SI_{local} and SI_{globlal}) among the selected modeling type (singe, voting ensemble, bagging ensemble, stacking ensemble and tiering ensemble). Table 16 presents the detailed results of iML and Figure 10 plots the RMSE of best models for the studied case. Both SI_{local} and SI_{global} values of bagging ANN ensemble are equal to zero, which indicate that the bagging ANN ensemble is the best model in terms of prediction accuracy.

_

Variable	Min.	Max.	Mean	Standard Deviation	Data Type
Y: Software development effort (person-hour)	4	2694	258.55	394.69	Numerical
X_1 : Program type entry	0	1	Dumm	ıy variable	Boolean
X_2 : Program type report	0	1	Dumm	y variable	Boolean
X_3 : Program type batch	0	1	Dumm	y variable	Boolean
X_4 : Program type query	0	1	Dumm	y variable	Boolean
X ₅ : Program type transaction	0	0	Referent	ial category	Boolean
X_6 : Number of programs	1	88	16.73	19.12	Numerical
X_7 : Number of zooms	0	2028	100.22	255.40	Numerical
X_8 : Number of columns in form	3	3216	397.75	548.06	Numerical
X ₉ : Number of actions	0	1645	288.44	339.61	Numerical
X_{10} : Number of signature tasks	0	15	0.39	1.77	Numerical
X_{11} : Number of batch serial numbers	0	11	0.31	1.50	Numerical
X ₁₂ : Number of multi-angle trade tasks	0	22	0.55	2.66	Numerical
X_{13} : Number of multi-unit tasks	0	21	1.10	3.41	Numerical
X_{14} : Number of reference calls	0	528	13.96	49.92	Numerical
X_{15} : Number of confirmed tasks	0	21	1.50	3.99	Numerical
X_{16} : Number of post tasks	0	12	0.23	1.33	Numerical
X_{17} : Number of industry type tasks	0	21	0.80	2.97	Numerical

Table 15. Variables and descriptive statistics for predicting enterprise resource planning (ERP) software development effort.

 Table 16. Performances of predictive models for ERP software development effort.

No.	Model			Learn					Test			SI and I	Ranking
1101	1120401	RMSE	MAE	MAPE	R	TER	RMSE	MAE	MAPE	R	TER	SI _{local}	SIglobal
Ι	Single												
1 2 3 4	ANN SVR CART LRR	68.81 0.00 86.89 250.21	24.92 0.00 40.38 221.15	19.91% 0.00% 19.56% 617.02%	0.98 1.00 0.97 0.84	1.49% 0.00% 0.00% 48.77%	115.24 361.88 196.48 255.43	61.85 255.35 107.02 227.80	30.65% 611.63% 47.29% 647.08%	0.95 Inf 0.85 0.78	9.86% 36.81% 12.77% 63.59%	0.00 (1) 0.89 (4) 0.15 (2) 0.72 (3)	0.13 (2)
II	Voting												
$ \begin{array}{r} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 11 \end{array} $	(*) ANN + CART + LRR ANN + SVR + CART ANN + SVR + LRR SVR + CART + LRR ANN + CART ANN + LRR ANN + SVR SVR + CART CART + LRR SVR + LRR	73.30 97.74 39.95 87.66 93.17 59.92 131.49 34.40 43.45 139.76 125.10	62.90 83.87 19.33 76.90 80.47 28.99 115.35 12.46 20.19 120.71 110.57	157.96% 210.62% 11.22% 207.61% 208.60% 16.83% 311.42% 9.96% 9.78% 312.91% 308.51%	$\begin{array}{c} 0.99\\ 0.98\\ 0.99\\ 0.99\\ 0.99\\ 0.99\\ 0.97\\ 1.00\\ 0.99\\ 0.96\\ 0.96\\ 0.97\end{array}$	$\begin{array}{c} 12.39\% \\ 16.52\% \\ 0.50\% \\ 16.52\% \\ 16.26\% \\ 0.75\% \\ 24.78\% \\ 0.75\% \\ 0.00\% \\ 24.38\% \\ 24.38\% \end{array}$	$\begin{array}{c} 185.39\\ 139.70\\ 181.63\\ 202.91\\ 236.10\\ 123.43\\ 152.57\\ 199.24\\ 254.27\\ 190.91\\ 287.41 \end{array}$	$\begin{array}{c} 139.02\\ 110.32\\ 117.20\\ 158.85\\ 178.73\\ 68.46\\ 129.46\\ 135.42\\ 166.51\\ 151.58\\ 230.03 \end{array}$	321.77% 227.32% 215.13% 419.21% 427.29% 33.90% 326.55% 307.89% 320.26% 337.64% 626.83%	$\begin{array}{c} 0.94 \\ 0.94 \\ 0.94 \\ 0.94 \\ 0.89 \\ 0.94 \\ 0.94 \\ 0.95 \\ 0.85 \\ 0.85 \\ 0.89 \\ 0.78 \end{array}$	23.90% 22.01% 14.45% 30.90% 32.97% 10.10% 32.67% 18.21% 21.09% 32.09% 47.93%	$\begin{array}{c} 0.34 \ (5) \\ 0.21 \ (2) \\ 0.22 \ (3) \\ 0.46 \ (7) \\ 0.60 \ (10) \\ 0.01 \ (1) \\ 0.34 \ (6) \\ 0.31 \ (4) \\ 0.56 \ (9) \\ 0.48 \ (8) \\ 1.00 \ (11) \end{array}$	0.15 (3)
III	Bagging												
1 2 3 4	ANN SVR CART LRR	70.28 174.56 123.05 249.00	33.48 92.31 52.45 222.81	21.45% 231.82% 25.94% 666.29%	0.98 0.91 0.94 0.88	2.51% 3.98% 2.19% 59.61%	65.58 162.38 127.21 309.18	40.51 99.02 79.97 257.31	19.50% 87.79% 20.11% 269.67%	0.99 0.87 0.96 0.71	5.59% 11.69% 8.58% 10.90%	0.00 (1) 0.47 (3) 0.21 (2) 0.97 (4)	0.00 (1)
IV	Stacking												
1 2 3 4	(*) ANN (*) SVR (*) CART (*) LRR	$0.07 \\ 0.00 \\ 52.00 \\ 132.73$	0.02 0.00 17.40 97.48	0.03% 0.00% 5.69% 187.26%	1.00 1.00 0.99 0.95	0.00% 0.00% 0.00% 23.62%	361.54 361.76 360.24 289.20	255.19 255.53 252.61 206.32	611.55% 612.78% 593.46% 494.18%	0.71 NaN NaN 0.62	36.77% 36.76% 34.86% 34.11%	0.80 (2) 1.00 (4) 0.81 (3) 0.03 (1)	1.00 (5)
V	Tiering												
1 2 3	2-Class (**) 3-Class (***) 4-Class (****)	317.04 383.02 414.30	71.38 115.48 151.32	24.45% 26.68% 29.96%	0.58 0.30 0.10	20.48% 36.54% 49.11%	176.77 278.46 347.05	65.42 111.74 147.22	23.57% 26.94% 29.76%	0.79 0.51 0.26	16.13% 28.64% 43.00%	0.00 (1) 0.54 (2) 1.00 (3)	0.31 (4)

Note: (*) is (ANN + SVR + CART + LRR); (**) SVM-(ANN, SVR); (***) CART-(ANN, SVR, SVR); (****) CART-(CART, SVR, SVR, SVR); (*No.*): Ranking.



Figure 10. Root mean square errors of best models.

Three models (single, voting, and bagging) provided better results in terms of R (0.94 to 0.99) than the tiering and stacking ensemble models, which had the R values of 0.58 to 0.95. Among these three best models, in terms of MAPE, the bagging model exhibited the best balance of MAPE results from learning and test data (21.45% and 19.50%, respectively). The single and voting models depicted un-balanced MAPEs for training and test data (19.91% and 30.65% for the single model; 16.83% and 33.90% for the voting model). Thus, the bagging model was the best model to predict ERP.

The first experiment indicates that, the iML not only identifies the best model, but also reports the performance values of all the training models. Chou et al. (2012) obtained training and testing MAPEs of 26.8% and 27.3%, and RMSEs of 234.0157 h and 97.2667 h using Evolutionary Support Vector Machine Inference Model (ESIM) [1]. The iML yields the bagging ensemble model with MAPEs of 21.45% and 19.50%, and RMSEs of 70.28hr and 65.58 h for the same training and test data, respectively. As a result, the iML is effective to find the best model among the popular regression models.

7.2. Experiments on Industrial Datasets

Three additional experiments were performed to evaluate iML. To ensure a fair comparison, 70 % of the data was used for learning whereas the remaining 30% was utilized for testing.

7.2.1. Performance of CPU Processors

This experiment is about the comparison of performance of CPU processors. The data for this experiment was taken from Maurya and Gupta (2015) [12]. This dataset contained 209 samples with a total of 6 attributes (Table 17). The descriptions of the attributes are as follows: X_1 : Machine cycle time in nanoseconds (integer, input); X_2 : Minimum main memory in kilobytes (integer, input); X_3 : Maximum main memory in kilobytes (integer, input); X_3 : Maximum main memory in kilobytes (integer, input); X_5 : Minimum channels in units (integer, input); X_6 : Maximum channels in units (integer, input); and Y: Estimated relative performance (integer, output).

		Input								
Statistic value	X ₁	X2	X ₃	X4	X_5	X ₆	Y			
Min	17	64	64	0	0	0	15			
Max	1500	32,000	64,000	256	52	176	1238			
Mean	203.82	2867.98	11,796.2	25.21	4.7	18.27	99.33			
Std.	260.26	3878.74	11,726.6	40.63	6.82	26	154.76			

7.2.2. Daily Demand Forecasting Orders

This experiment is about the daily demand forecasting orders. The data used in this experiment was taken from Ferreira et al. (2016) [13]. Table 18 shows a statistical analysis of the data. There were 60 samples with 12 attributes, including X₁: Week of the month (first week, second, third or fourth week of month, input); X₂: Day of the week (Monday to Friday, input); X₃: Urgent orders (integer, input); X₄: Non-urgent orders (integer, input); X₅: Type A orders (integer, input); X₆: Type B orders (integer, input); X₇: Orders of type C (integer, input); X₈: Orders from the tax sector (integer, input); X₉: Orders from the traffic controller sector (integer, input); X₁₀: Orders from the banking sector 1 (integer, input); X₁₁: Orders from the banking sector 2 (integer, input); X₁₂: Banking orders 3 (integer, input); and Y: Total orders (integer, output).

Table 18. Variables and descriptive statistics for daily demand forecasting orders.

Statistic Value							Input						Output
Statistic value	X ₁	X ₂	X ₃	X_4	X_5	X ₆	X_7	X ₈	X9	X ₁₀	X ₁₁	X ₁₂	Y
Min Max Mean Std.	1 5 -	2 6 -	43.65 435.30 172.55 69.51	77.37 223.27 118.92 27.17	21.83 118.18 52.11 18.83	25.13 267.34 109.23 50.74	74.37 302.45 139.53 41.44	0 865 77.4 186.5	11,992 71,772 44,504.4 12,197.9	3452 210,508 46,640.8 45,220.7	16,411 188,411 79,401.5 40,504.4	7679 73,839 23,114.6 13,148	129.41 616.45 300.87 89.6

7.2.3. Total Hourly-Shared Bike Rental per Days

The experiment is about the total hourly-shared bike rental per days. The data was adopted from Fanaee-T and Gama (2014) [11], and statistically analyzed in Table 19. In total, there were 731 samples and 11 attributes, defined as follows: X_1 : Season (1: spring, 2: summer, 3: fall, 4: winter, input); X_2 : Month (1 to 12, input); X_3 : Year (0:2011, 1:2012, input); X_4 : Weather day is holiday or not (input); X_5 : Day of the week (input); X_6 : Working day if day is neither weekend nor holiday is 1, otherwise is 0 (input); X_7 : Weather condition (1: Clear, Few clouds, partly cloudy; 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist; 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds; 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog, input); X_8 : Normalized temperature in Celsius. The values are divided to 41 (max) (input); X_{10} : Normalized humidity. The values are divided to 100 (max) (input); X_{11} : Normalized wind speed. The values are divided to 67 (max) (input); and Y: Count of total rental bikes including both casual and registered (output).

Table 19. Variables and descriptive statistics for total hourly-shared bike rental per days.

Clatter Value	Input											Output
Statistic value	X ₁	X2	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X9	X ₁₀	X ₁₁	Y
Min	1	0	1	0	0	0	1	0.059	0.079	0	0.022	22
Max	4	1	12	1	6	1	3	0.862	0.841	0.973	0.507	8714
Mean	-	-	-	0.029	2.997	0.684	1.395	0.495	0.474	0.628	0.19	4504.35
Std.	-	-	-	0.167	2.005	0.465	0.545	0.183	0.163	0.142	0.077	1937.21

In this study, to calculate MAPE, the output was normalized and 0.1 was added to prevent a zero value.

$$y_i = \frac{y_i - y_{min}}{y_{max} - y_{min}} + 0.1 \tag{21}$$

where y_i , y_{min} , and y_{max} are actual value, minimum and maximum of actual value, respectively.

7.2.4. Performance Results

Table 20 presents the performance results of all models for the three additional datasets. Using the same dataset in the experiment No. 2, Maurya and Gupta (2015) [12] trained ANN models with the maximum R-learn and R-test values of 0.98146 and 0.98662, respectively. Meanwhile, the iML identifies the ANN single model as the best model with R-learn and R-test values of 0.99990 and 0.99629, respectively. The iML gives out a slightly better model than those of the previous research in this numerical experiment.

Table 20. Performance results of three additional numerical experim	nents.
---	--------

No	Model			Learn					Test		
INO.	Widdel	RMSE	MAE	MAPE	R	TER	RMSE	MAE	MAPE	R	TER
2	Single	2.462	0.569	1.015%	1.000	0.236%	8.738	3.683	3.775%	0.996	2.730%
	Voting	17.509	4.965	2.808%	0.995	0.118%	13.087	5.173	5.685%	0.989	0.921%
	Bagging	40.127	8.893	2.360%	0.981	3.015%	13.484	3.930	2.489%	0.996	3.795%
	Stacking	40.428	9.030	3.389%	0.973	0.000%	64.782	43.615	104.992%	0.842	9.852%
	Tiering-2class	163.338	26.947	3.229%	0.383	24.818%	18.196	5.822	4.629%	0.986	3.220%
	Tiering-3class	167.093	29.784	3.856%	0.340	27.446%	76.406	12.698	5.268%	0.639	13.094%
	Tiering-4class	182.572	44.722	7.970%	0.112	41.332%	91.701	21.089	8.342%	0.422	24.368%
3	Single	0.349	0.080	0.023%	1.000	0.021%	0.317	0.231	0.093%	1.000	0.042%
	Voting	17.417	10.754	3.089%	0.985	0.010%	12.162	10.157	3.993%	0.951	0.867%
	Bagging	0.917	0.399	0.110%	1.000	0.020%	0.296	0.221	0.087%	1.000	0.074%
	Stacking	0.338	0.090	0.026%	1.000	0.014%	0.335	0.251	0.101%	1.000	0.042%
	Tiering-2class	169.296	63.580	14.294%	-0.399	21.483%	214.674	86.711	16.747%	-0.704	27.688%
	Tiering-3class	273.303	212.047	62.384%	-0.664	71.449%	295.065	223.209	62.186%	-0.570	71.054%
	Tiering-4class	329.001	312.397	97.619%	-0.304	99.023%	51.164	45.122	18.684%	0.706	11.339%
4	Single	0.046	0.030	6.670%	0.979	10.450%	0.105	0.073	14.120%	0.883	0.550%
	Voting	0.052	0.037	7.850%	0.974	8.000%	0.080	0.056	10.750%	0.929	0.260%
	Bagging	0.049	0.034	7.150%	0.977	6.930%	0.069	0.046	8.870%	0.948	0.190%
	Stacking	0.005	0.003	0.700%	1.000	21.430%	0.214	0.169	38.680%	0.000	0.086%
	Tiering-2class	0.580	0.432	57.620%	-0.582	58.900%	0.589	0.451	61.410%	-0.570	68.290%
	Tiering-3class	0.639	0.568	84.420%	-0.680	64.600%	0.646	0.565	82.110%	-0.717	90.260%
	Tiering-4class	0.648	0.596	92.370%	-0.513	65.200%	0.652	0.584	87.400%	-0.630	94.300%

Note: No. 2: CPU experiment dataset; No. 3: Customer experiment dataset; No. 4: Rental bike experiment dataset; the bold denotes the best model in each experiment.

In the experiment No. 3, Ferreira et al. (2016) had an analytical result of MAPE 3.45% and iML confirms ANN single model as the best model, with MAPE values for learning and test of 0.023% and 0.093%, respectively [13]. The stacking ANN ensemble also performs well with the MAPEs for the learning and test data by 0.026% and 0.010%, respectively.

Finally, in the experiment No. 4, iML achieves R-learn and R-test values of 0.97660 and 0.94790, with bagging ANN as the best model. In contrast, Fanaee-T and Gama (2014) obtained a maximum R value of 0.91990 [11].

As shown in the above numerical experiments, iML trains and identifies the best models which are better than those in the previous studies.

8. Conclusions and Future Work

This study develops an iML platform to efficiently operate data-mining techniques. The iML is designed to be user-friendly, so users can get the results with only "One-Click". The numerical experiments have demonstrated that iML is a powerful soft computing to identify the best prediction model by automating comparison among diverse machine learning techniques.

To benchmark the effectiveness of iML with WEKA, five datasets collected from the UCI Machine Learning Repository were analyzed via hold-out validation and tenfold cross validation. The performance results indicate that iML can find a more accurate model than that of WEKA in the publicly available datasets. The best prediction model identified by iML is also the best model among all the models trained by iML and WEKA. Notably, iML requires minimal effort from the users to build single, voting, bagging, and stacking models in comparison with WEKA.

Four industrial experiments were carried out to validate the performance of iML. The first experiment involved training a model for prediction of ERP development effort, in which iML yielded an RMSE for learning data with 70.28 h and for testing data with 65.58 h, by using the bagging ANN ensemble (best model). In contrast, Chou et al. (2012) [1] obtained training and testing RMSE values of 234.0157 h and 97.2667 h, respectively.

In the second experiment on performance of CPU processors, iML yielded 0.99990 for R-learning and 0.99629 for R-testing, which are better than those reported in Maurya and Gupta (2015) [12], and confirmed that single ANN was the best model. In the third experiment of daily demand forecasting orders, iML achieved MAPE values of 0.026% (learning) and 0.010% (testing). The results are as excellent as those obtained in Ferreira et al. (2016) [13]. In the fourth experiment for total hourly-shared bike rental, R-learning and R-testing values of 0.97660 and 0.94790 were reached using iML. The test performance was 6% better than that obtained by Fanaee-T and Gama (2014) [11]. In addition to the enhanced prediction performance, the iML possesses ability to determine the best models on the basis of multiple evaluation metrics.

In conclusion, the iML is a powerful and promising prediction platform for solving diverse engineering problems. Since the iML platform can only deal with regression problems, future research should upgrade iML for solving complex classification and time series problems by automatically presenting the alternative models for practical use in engineering applications, as well as adding some other advanced ML methods (such as deep learning models). Moreover, metaheuristic optimization algorithms could be integrated with the iML to help the users finetune the hyperparameters of chosen machine learning models.

Author Contributions: Conceptualization, J.-S.C.; data curation, D.-N.T.; formal analysis, J.-S.C. and D.-N.T.; funding acquisition, J.-S.C.; investigation, J.-S.C., D.-N.T. and C.-F.T.; methodology, J.-S.C. and C.-F.T.; project administration, J.-S.C.; resources, J.-S.C. and C.-F.T.; software, D.-N.T.; supervision, J.-S.C.; validation, J.-S.C., D.-N.T. and C.-F.T.; visualization, J.-S.C. and D.-N.T.; writing—original draft, J.-S.C., D.-N.T. and C.-F.T.; writing—review and editing, J.-S.C. and D.-N.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Technology, Taiwan, under grants 108-2221-E-011-003-MY3 and 107-2221-E-011-035-MY3.

Data Availability Statement: The data that support the findings of this study are available from the UCI Machine Learning Repository or corresponding author upon reasonable request.

Acknowledgments: The authors would like to thank the Ministry of Science and Technology, Taiwan, for financially supporting this research.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

- 1. Chou, J.-S.; Cheng, M.-Y.; Wu, Y.-W.; Wu, C.-C. Forecasting enterprise resource planning software effort using evolutionary support vector machine inference model. *Int. J. Proj. Manag.* **2012**, *30*, 967–977. [CrossRef]
- Pham, A.-D.; Ngo, N.-T.; Nguyen, Q.-T.; Truong, N.-S. Hybrid machine learning for predicting strength of sustainable concrete. Soft Comput. 2020. [CrossRef]
- 3. Cheng, M.-Y.; Chou, J.-S.; Cao, M.-T. Nature-inspired metaheuristic multivariate adaptive regression splines for predicting refrigeration system performance. *Soft Comput.* **2015**, *21*, 477–489. [CrossRef]
- 4. Li, Y.; Lei, G.; Bramerdorfer, G.; Peng, S.; Sun, X.; Zhu, J. Machine Learning for Design Optimization of Electromagnetic Devices: Recent Developments and Future Directions. *Appl. Sci.* **2021**, *11*, 1627. [CrossRef]
- 5. Piersanti, S.; Orlandi, A.; Paulis, F.d. Electromagnetic Absorbing Materials Design by Optimization Using a Machine Learning Approach. *IEEE Trans. Electromagn. Compat.* **2018**, 1–8. [CrossRef]
- 6. Chou, J.S.; Pham, A.D. Smart artificial firefly colony algorithm-based support vector regression for enhanced forecasting in civil engineering. *Comput.-Aided Civ. Infrastruct. Eng.* **2015**, *30*, 715–732. [CrossRef]

- 7. Cheng, M.-Y.; Prayogo, D.; Wu, Y.-W. A self-tuning least squares support vector machine for estimating the pavement rutting behavior of asphalt mixtures. *Soft Comput.* **2019**, *23*, 7755–7768. [CrossRef]
- Al-Ali, H.; Cuzzocrea, A.; Damiani, E.; Mizouni, R.; Tello, G. A composite machine-learning-based framework for supporting low-level event logs to high-level business process model activities mappings enhanced by flexible BPMN model translation. *Soft Comput.* 2019. [CrossRef]
- 9. López, J.; Maldonado, S.; Carrasco, M. A novel multi-class SVM model using second-order cone constraints. *Appl. Intell.* 2016, 44, 457–469. [CrossRef]
- 10. Bogawar, P.S.; Bhoyar, K.K. An improved multiclass support vector machine classifier using reduced hyper-plane with skewed binary tree. *Appl. Intell.* **2018**, *48*, 4382–4391. [CrossRef]
- 11. Fanaee-T, H.; Gama, J. Event labeling combining ensemble detectors and background knowledge. *Prog. Artif. Intell.* 2014, 2, 113–127. [CrossRef]
- Maurya, V.; Gupta, S.C. Comparative Analysis of Processors Performance Using ANN. In Proceedings of the 2015 5th International Conference on IT Convergence and Security (ICITCS), Kuala Lumpur, Malaysia, 24–27 August 2015; pp. 1–5.
- 13. Ferreira, R.P.; Martiniano, A.; Ferreira, A.; Ferreira, A.; Sassi, R.J. Study on Daily Demand Forecasting Orders using Artificial Neural Network. *IEEE Lat. Am. Trans.* 2016, 14, 1519–1525. [CrossRef]
- 14. De'ath, G.; Fabricius, K.E. Classification and regression trees: A powerful yet simple technique for ecological data analysis. *Ecology* **2000**, *81*, 3178–3192. [CrossRef]
- 15. Li, H.; Wen, G. Modeling reverse thinking for machine learning. Soft Comput. 2020, 24, 1483–1496. [CrossRef]
- 16. Li, Y. Predicting materials properties and behavior using classification and regression trees. *Mater. Sci. Eng. A* 2006, 433, 261–268. [CrossRef]
- 17. Chou, J.-S.; Yang, K.-H.; Lin, J.-Y. Peak Shear Strength of Discrete Fiber-Reinforced Soils Computed by Machine Learning and Metaensemble Methods. *J. Comput. Civ. Eng.* **2016**, *30*, 04016036. [CrossRef]
- 18. Qi, C.; Tang, X. Slope stability prediction using integrated metaheuristic and machine learning approaches: A comparative study. *Comput. Ind. Eng.* **2018**, *118*, 112–122. [CrossRef]
- 19. Chou, J.-S.; Ho, C.-C.; Hoang, H.-S. Determining quality of water in reservoir using machine learning. *Ecol. Inform.* **2018**, *44*, 57–75. [CrossRef]
- Chou, J.-S.; Bui, D.-K. Modeling heating and cooling loads by artificial intelligence for energy-efficient building design. *Energy Build.* 2014, 82, 437–446. [CrossRef]
- 21. Alkahtani, M.; Choudhary, A.; De, A.; Harding, J.A. A decision support system based on ontology and data mining to improve design using warranty data. *Comput. Ind. Eng.* **2018**. [CrossRef]
- 22. Daras, G.; Agard, B.; Penz, B. A spatial data pre-processing tool to improve the quality of the analysis and to reduce preparation duration. *Comput. Ind. Eng.* 2018, 119, 219–232. [CrossRef]
- 23. Chou, J.-S.; Tsai, C.-F. Preliminary cost estimates for thin-film transistor liquid–crystal display inspection and repair equipment: A hybrid hierarchical approach. *Comput. Ind. Eng.* **2012**, *62*, 661–669. [CrossRef]
- 24. Chen, T. An ANN approach for modeling the multisource yield learning process with semiconductor manufacturing as an example. *Comput. Ind. Eng.* **2017**, *103*, 98–104. [CrossRef]
- Wu, X.; Kumar, V.; Quinlan, J.R.; Ghosh, J.; Yang, Q.; Motoda, H.; McLachlan, G.J.; Ng, A.; Liu, B.; Philip, S.Y. Top 10 algorithms in data mining. *Knowl. Inf. Syst.* 2008, 14, 1–37. [CrossRef]
- Chou, J.-S.; Ngo, N.-T.; Chong, W.K. The use of artificial intelligence combiners for modeling steel pitting risk and corrosion rate. Eng. Appl. Artif. Intell. 2017, 65, 471–483. [CrossRef]
- 27. Das, D.; Pratihar, D.K.; Roy, G.G.; Pal, A.R. Phenomenological model-based study on electron beam welding process, and input-output modeling using neural networks trained by back-propagation algorithm, genetic algorithms, particle swarm optimization algorithm and bat algorithm. *Appl. Intell.* **2018**, *48*, 2698–2718. [CrossRef]
- 28. Tewari, S.; Dwivedi, U.D. Ensemble-based big data analytics of lithofacies for automatic development of petroleum reservoirs. *Comput. Ind. Eng.* **2018**. [CrossRef]
- 29. Priore, P.; Ponte, B.; Puente, J.; Gómez, A. Learning-based scheduling of flexible manufacturing systems using ensemble methods. *Comput. Ind. Eng.* **2018**, *126*, 282–291. [CrossRef]
- 30. Fang, K.; Jiang, Y.; Song, M. Customer profitability forecasting using Big Data analytics: A case study of the insurance industry. *Comput. Ind. Eng.* **2016**, *101*, 554–564. [CrossRef]
- 31. Chou, J.-S. Generalized linear model-based expert system for estimating the cost of transportation projects. *Expert Syst. Appl.* **2009**, *36*, 4253–4267. [CrossRef]
- 32. Dandikas, V.; Heuwinkel, H.; Lichti, F.; Drewes, J.E.; Koch, K. Predicting methane yield by linear regression models: A validation study for grassland biomass. *Bioresour. Technol.* 2018, 265, 372–379. [CrossRef] [PubMed]
- 33. Ngo, S.H.; Kemény, S.; Deák, A. Performance of the ridge regression method as applied to complex linear and nonlinear models. *Chemom. Intell. Lab. Syst.* 2003, 67, 69–78. [CrossRef]
- 34. Sentas, P.; Angelis, L. Categorical missing data imputation for software cost estimation by multinomial logistic regression. *J. Syst. Softw.* **2006**, *79*, 404–414. [CrossRef]

- 35. Slowik, A. Application of an Adaptive Differential Evolution Algorithm With Multiple Trial Vectors to Artificial Neural Network Training. *IEEE Trans. Ind. Electron.* **2011**, *58*, 3160–3167. [CrossRef]
- 36. Caputo, A.C.; Pelagagge, P.M. Parametric and neural methods for cost estimation of process vessels. *Int. J. Prod. Econ.* 2008, 112, 934–954. [CrossRef]
- 37. Rocabruno-Valdés, C.I.; Ramírez-Verduzco, L.F.; Hernández, J.A. Artificial neural network models to predict density, dynamic viscosity, and cetane number of biodiesel. *Fuel* **2015**, 147, 9–17. [CrossRef]
- 38. Ganesan, P.; Rajakarunakaran, S.; Thirugnanasambandam, M.; Devaraj, D. Artificial neural network model to predict the diesel electric generator performance and exhaust emissions. *Energy* **2015**, *83*, 115–124. [CrossRef]
- 39. Vapnik, V.N. An overview of statistical learning theory. IEEE Trans. Neural Netw. 1999, 10, 988–999. [CrossRef]
- 40. Vapnik, V. The Nature of Statistical Learning Theory, 2nd ed.; Springer: New York, NY, USA, 2013.
- 41. Jing, G.; Cai, W.; Chen, H.; Zhai, D.; Cui, C.; Yin, X. An air balancing method using support vector machine for a ventilation system. *Build. Environ.* **2018**, *143*, 487–495. [CrossRef]
- García-Floriano, A.; López-Martín, C.; Yáñez-Márquez, C.; Abran, A. Support vector regression for predicting software enhancement effort. *Inf. Softw. Technol.* 2018, 97, 99–109. [CrossRef]
- 43. Breiman, L.; Friedman, J.; Stone, C.J.; Olshen, R.A. *Classification and Regression Trees*; Routledge: New York, NY, USA, 2017; p. 368. [CrossRef]
- 44. Choi, S.Y.; Seo, I.W. Prediction of fecal coliform using logistic regression and tree-based classification models in the North Han River, South Korea. J. Hydro-Environ. Res. 2018, 21, 96–108. [CrossRef]
- 45. Ru, F.; Yin, A.; Jin, J.; Zhang, X.; Yang, X.; Zhang, M.; Gao, C. Prediction of cadmium enrichment in reclaimed coastal soils by classification and regression tree. *Estuar. Coast. Shelf Sci.* 2016, 177, 1–7. [CrossRef]
- 46. Chou, J.-S.; Tsai, C.-F.; Pham, A.-D.; Lu, Y.-H. Machine learning in concrete strength simulations: Multi-nation data analytics. *Constr. Build. Mater.* **2014**, *73*, 771–780. [CrossRef]
- 47. Elish, M.O. Assessment of voting ensemble for estimating software development effort. In Proceedings of the 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Singapore, 16–19 April 2013; pp. 316–321.
- Wang, Z.; Wang, Y.; Srinivasan, R.S. A novel ensemble learning approach to support building energy use prediction. *Energy Build*. 2018, 159, 109–122. [CrossRef]
- 49. Chen, J.; Yin, J.; Zang, L.; Zhang, T.; Zhao, M. Stacking machine learning model for estimating hourly PM2.5 in China based on Himawari 8 aerosol optical depth data. *Sci. Total Environ.* **2019**, *697*, 134021. [CrossRef] [PubMed]
- 50. Basant, N.; Gupta, S.; Singh, K.P. A three-tier QSAR modeling strategy for estimating eye irritation potential of diverse chemicals in rabbit for regulatory purposes. *Regul. Toxicol. Pharmacol.* **2016**, *77*, 282–291. [CrossRef]
- 51. Lee, K.M.; Yoo, J.; Kim, S.-W.; Lee, J.-H.; Hong, J. Autonomic machine learning platform. *Int. J. Inf. Manag.* 2019, 49, 491–501. [CrossRef]
- 52. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. IEEE Trans. Evol. Comput. 1997, 1, 67–82. [CrossRef]
- 53. Wolpert, D.H.; Macready, W.G. *No Free Lunch Theorems for Search*; Technical Report SFI-TR-95-02-010; Santa Fe Institute: Santa Fe, NM, USA, 1995.
- 54. Cheng, D.; Shi, Y.; Gwee, B.; Toh, K.; Lin, T. A Hierarchical Multiclassifier System for Automated Analysis of Delayered IC Images. *IEEE Intell. Syst.* 2019, 34, 36–43. [CrossRef]
- 55. Basheer, I.A.; Hajmeer, M. Artificial neural networks: Fundamentals, computing, design, and application. *J. Microbiol. Methods* **2000**, *43*, 3–31. [CrossRef]
- 56. Jain, A.K.; Jianchang, M.; Mohiuddin, K.M. Artificial neural networks: A tutorial. Computer 1996, 29, 31–44. [CrossRef]
- 57. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
- Chamasemani, F.F.; Singh, Y.P. Multi-class Support Vector Machine (SVM) Classifiers—An Application in Hypothyroid Detection and Classification. In Proceedings of the 2011 Sixth International Conference on Bio-Inspired Computing: Theories and Applications, Penang, Malaysia, 27–29 September 2011; pp. 351–356.
- 59. Yang, X.; Yu, Q.; He, L.; Guo, T. The one-against-all partition based binary tree support vector machine algorithms for multi-class classification. *Neurocomputing* **2013**, *113*, 1–7. [CrossRef]
- 60. Tuv, E.; Runger, G.C. Scoring levels of categorical variables with heterogeneous data. IEEE Intell. Syst. 2004, 19, 14–19. [CrossRef]
- 61. Chiang, W.; Liu, X.; Zhang, T.; Yang, B. A Study of Exact Ridge Regression for Big Data. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 3821–3830.
- 62. Marquardt, D.W.; Snee, R.D. Ridge Regression in Practice. *Am. Stat.* **1975**, *29*, 3–20. [CrossRef]
- 63. Cox, D.R. The regression analysis of binary sequences. J. R. Stat. Society. Ser. B 1958, 20, 215–242. [CrossRef]
- 64. Jiang, F.; Guan, Z.; Li, Z.; Wang, X. A method of predicting visual detectability of low-velocity impact damage in composite structures based on logistic regression model. *Chin. J. Aeronaut.* **2021**, *34*, 296–308. [CrossRef]
- 65. Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the International Joint Conference on Artificial Intelligence 1995, Montreal, QC, Canada, 20–25 August 1995; pp. 1137–1143.
- Chou, J.; Truong, D.; Le, T. Interval Forecasting of Financial Time Series by Accelerated Particle Swarm-Optimized Multi-Output Machine Learning System. *IEEE Access* 2020, *8*, 14798–14808. [CrossRef]
- 67. Yeh, I.-C. Analysis of Strength of Concrete Using Design of Experiments and Neural Networks. J. Mater. Civ. Eng. 2006, 18, 597–604. [CrossRef]

- 68. Yeh, I.C.; Hsu, T.-K. Building real estate valuation models with comparative approach through case-based reasoning. *Appl. Soft Comput.* **2018**, *65*, 260–271. [CrossRef]
- 69. Tsanas, A.; Xifara, A. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy Build*. **2012**, *49*, 560–567. [CrossRef]
- 70. Lau, K.; López, R. A Neural Networks Approach to Aerofoil Noise Prediction; International Center for Numerical Methods in Engineering: Barcelona, Spain, 2009.