Engineering Applications of Artificial Intelligence 45 (2015) 204–219

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

A novel Boosted-neural network ensemble for modeling multi-target regression problems

Esmaeil Hadavandi [a], Jamal Shahrabi [a,*], Shahaboddin Shamshirband [b]

[a] Department of Industrial Engineering and management systems, Amirkabir University of Technology, P.O. Box 15875-4413, Tehran, Iran
[b] Department of Computer System and Technology, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia

ARTICLE INFO

Article history:
Received 21 September 2014
Received in revised form
20 May 2015
Accepted 26 June 2015
Available online 25 July 2015

Keywords:
Neural network ensemble
Constrained least mean square
Negative correlation learning
Subspace projection method
Boosting
Multi-target regression

ABSTRACT

In this paper, the concept of ensemble learning is adopted and applied to modeling multi-target regression problems with high-dimensional feature spaces and a small number of instances. A novel neural network ensemble (NNE) model is introduced, called Boosted-NNE based on notions from boosting, subspace projection methods and the negative correlation learning algorithm (NCL). Rather than using an entire feature space for training each component in the Boosted-NNE, a new cluster-based subspace projection method (CLSP) is proposed to automatically construct a low-dimensional input space with focus on the difficult instances in each step of the boosting approach. To enhance diversity in the Boosted-NNE, a new, sequential negative correlation learning algorithm (SNCL) is proposed to train negatively correlated components. Furthermore, the constrained least mean square error (CLMS) algorithm is employed to obtain the optimal weights of components in the combination module. The proposed Boosted-NNE model is compared with other ensemble and single models using four real cases of multi-target regression problems. The experimental results indicate that using the SNCL in combination with the CLSP method offers the capability to improve the diversity and accuracy of the Boosted-NNE. Thus, this model seems a promising alternative for modeling high-dimensional multi-target regression problems.

1. Introduction and literature review

In commercial decision making, planning and controlling are critical. Prediction, a basis for planning and controlling, attempts to predict future trends and is vital for industry planning and operation (Shahrabi et al., 2013). In most common prediction problem settings, the value of a single target numeric attribute is predicted. A natural generalization of this setting is to predict multi-target numeric attributes (Aho et al., 2012) simultaneously, because in many real-life predictive modeling problems, the output is structured, meaning there can be dependencies or internal relations between targets. These problems generally exhibit complex behaviors, high-dimensional feature spaces and a small number of instances (Kocev et al., 2013).

Artificial intelligence (AI) methods, such as artificial neural networks (ANNs) (Rumelhart and McClelland, 1986) are popular tools for solving complex engineering and optimization problems. One approach to deal with complex, real-world problems is to combine AI prediction models and form an ensemble of predictors that exploit the different local behaviors of the base models to improve the overall prediction system's performance (Masoudnia et al., 2012). The main objective of ensemble learning methods is to simplify a difficult prediction task by dividing it into some relatively easy prediction subtasks and formulating a consensus prediction result for the original data (García-Pedrajas et al., 2012). From another perspective, ensemble learning is an approach to enhance the prediction accuracy for complex problems, such as those involving a limited number of instances, high-dimensional feature sets, and highly complex trends and behaviors (Kotsiantis, 2011).

Among the prevalent ensemble approaches is the neural network ensemble (NNE) (Hansen and Salamon, 1990) that consists of a finite number of NNs and has been intensively studied in recent years (Tian et al., 2012; Zhai et al., 2012). Techniques using NNE models usually comprise two independent phases: using a method to create individual NN components and using a method to combine NN components. Both theoretical and experimental studies have shown that an NNE model is more effective when its components' estimates are negatively correlated, it is moderately effective when its components are uncorrelated and it is only mildly effective when its components are

* Corresponding author. Tel.: +98 21 66413034; fax: +98 21 66413025.
E-mail addresses: es.hadavandi@aut.ac.ir (E. Hadavandi), jamalshahrabi@aut.ac.ir (J. Shahrabi).

http://dx.doi.org/10.1016/j.engappai.2015.06.022
0952-1976/

positively correlated (Brown et al., 2005). There are some common approaches to produce accurate and negatively correlated components for NNE models, such as manipulating the training data set or using the penalty method, whereby a penalty term is added to the error function of the NNE model (Brown et al., 2005).

Two popular methods of constructing ensembles by manipulating the instances to independently and sequentially train individual NN components are bagging (bootstrap aggregating) (Breiman, 1996) and boosting (Freund and Schapire, 1996), respectively. Bagging only generates different bootstrap samples from the original training set for training various components. Boosting adaptively changes the distribution of the training set based on the performance of the previously added components to the ensemble. The most widely employed boosting method is AdaBoost. It is based on adaptively increasing the probability of sampling instances with greater prediction errors using previous components. Different versions of AdaBoost available for regression problems include AdaBoost. RT (Solomatine and Shrestha, 2004). The idea behind it is to filter out examples with higher relative estimation error than a pre-set threshold value, and then to follow the AdaBoost procedure. Asymmetric-AdaBoost is another extension of AdaBoost, which only alters the procedure for weight updating. It is a cost-sensitive boosting algorithm based on the statistical interpretation of boosting to enable deriving a principled asymmetric boosting loss, which, much like the original AdaBoost, is then minimized by gradient descent in the functional space of convex combinations of weak learners (Masnadi-Shirazi and Vasconcelos, 2007). One of the causes of boosting failure is putting too much emphasis on correctly classifying all instances. Outliers, or noisy instances become too relevant in the training set, undermining the ensemble's performance (García-Pedrajas et al., 2012).

The feature space can be manipulated by randomly selecting feature subspaces from the original feature space (Garc´ıa-Pedrajas et al., 2007). The most widely applied ensemble learning method that manipulates the feature space is the Random Subspaces Method (RSM) (Ho, 1998). Some ensemble methods are based on constructing each component using features obtained by rotating subspaces of the original dataset (Rodriguez et al., 2006; Xiong et al., 2015). The notion of using subspace projection methods to construct NNE models has been applied in different works (Garc´ıa-Pedrajas et al., 2007; Garc´ıa-Pedrajas and Ortiz-Boyer, 2008). More recently, García-Pedrajas et al. (2012) proposed a new boosting-based method for designing classifier ensembles according to unsupervised and supervised projection of random subspaces using different projection methods. The projections were constructed using only misclassified instances to focus the next classifier on the most difficult dataset instances.

A popular penalty method for creating explicitly diverse components is negative correlation learning (NCL) (Liu and Yao, 1999). The key idea behind NCL is to introduce a correlation penalty term to the cost function of individual NN components so that each component minimizes its mean square error (MSE) together with the ensemble's error correlation (Masoudnia et al., 2012). Alhamdoosh and Wang (2014) incorporated random vector functional link (RVFL) networks as base components with the NCL strategy to build neural network ensembles. Lee et al. (2012) proposed a new selective neural network ensemble with negative correlation. A set of component networks were chosen to build an ensemble so the generalization error would be minimized and the negative correlation maximized. The mentioned advantages and limitations of NCL and subspace projection methods are summarized in Table 1.

In this paper, as a first study in the literature, a novel Boosted-NNE model is developed for high-dimensional multi-target regression problems with a small number of instances based on several ideas from boosting, subspace projection and a new NCL algorithm. The proposed model is constructed according to the following contributions:

(1) The first contribution is to develop a new cluster-based subspace projection method (CLSP) to construct a low-dimensional feature space for training each component network rather than using the entire input feature space to train the components. CLSP uses a clustering algorithm on the original input features, which is combined with the boosting principle of directing learning on the difficult instances. As for subspaces, a smaller number of input features allows obtaining diverse and negatively correlated components for the Boosted-NNE.
(2) As the second contribution, the sequential negative correlation learning algorithm (SNCL) is proposed to train the Boosted-NNE model components in order to improve the diversity among individual component networks. In SNCL, the penalty term in the error function of component t measures the error correlation between itself and previous components $(1, 2\ldots, t-1)$. This way, it is hoped the NNE model components trained by SNCL are pushed away from each other so that the total distance between them expands.
(3) Moreover, the constrained least mean square error (CLMS) algorithm is utilized to obtain the optimal weights of the components in the Boosted-NNE model's combination module.

The Boosted-NNE model features are compared with other ensemble models to demonstrate its advantages (see Table 2). The listed models are constructed based on combinations of a number of ensemble creation methods.

The rest of the paper is organized as follows. In Section 2, the Boosted-NNE model is presented in detail. In Sections 3 and 4, the proposed model is compared with the existing ensemble models using four real datasets of multi-target regression problems. Finally, conclusions and future works are discussed in Section 5.

2. Methodology formulation

A multi-target regression problem is defined as follows:

An input space X that consists of multiple features (in our case these are continuous), i.e., $\forall X_n \in X$, $X_n = (x_{n1}, x_{n2}, \ldots, x_{nD})$, where D is the number of input features.
A target space Y that consists of multiple features (in our case these are continuous), i.e., $\forall Y_n \in Y, Y_n = (y_{n1}, y_{n2}, \ldots, y_{nP})$, where P is the number of target features.
A set of instances S, where each instance is a pair of input-target features, i.e., $S = \{(X_n, Y_n) | X_n \epsilon X,\ Y_n \epsilon Y\ ,\ 1 \le n \le N\}$ and N is the number of instances of S $(N = |S|)$.
A quality criterion such as error function E rewards models with low predictive error.

Table 1
Summary of the advantages and limitations of NCL and subspace projection methods.

Method	Advantages	Disadvantages
Boosting	✓ It improves diversity among components of an ensemble model.	✓ Sensitive to outliers, or noisy instances. ✓ Inefficient in high-dimensional feature space problems.
Negative Correlation Learning (Chen and Yao, 2009)	✓ There is a regularization term that provides a convenient way to balance the bias–var–cov trade-off, thus improving the generalization ability.	✓ Inefficient in high-dimensional feature space problems.
Subspace projection methods ADDIN EN.CITE (García-Pedrajas et al., 2012)	✓ Using a set of disjoint subspaces promotes diversity, as the subspace projection is carried out using different non-overlapping subsets of inputs. ✓ Efficient for high-dimensional feature space problems.	✓ Using the random subspace methods may decrease the diversity in respect to the other ensemble approaches. ✓ This method does not consider the correlation among input features in constructing the projection subspace.

Table 2
The features of the proposed Boosted-NNE model versus other NNE models.

Paper	Penalty method	Manipulating training data				Other descriptions
	NCL	RSM	Bagging	Boosting	Subspace projection methods	
Tang et al. (2009)	✓					✓ Inefficient for problems with high-dimensional feature spaces and a small number of instances
Mas'ud et al. (2014)		✓	✓			
Chen and Yao (2009)	✓		✓			
Alhamdoosh and Wang (2014)	✓		✓			
García-Pedrajas et al. (2012)				✓	✓	✓ Does not consider correlation among features in subspace projection
Ayerdi and Graña (2014), Pardo et al. (2013), Rodriguez et al. (2006)			✓		✓	✓ There is no control over the bias–var–cov trade-off
Garc´ıa-Pedrajas et al. (2007)		✓		✓		
The proposed Boosted-NNE	✓	✓		✓	✓	✓ There is control over the bias–var–cov trade-off using SNCL ✓ Considers moving average in NCL ✓ Considers the correlation among input features in constructing the projection subspace by CLSP method ✓ The smaller number of input features allows the NN components of the NNE model to approximate the projection more accurately

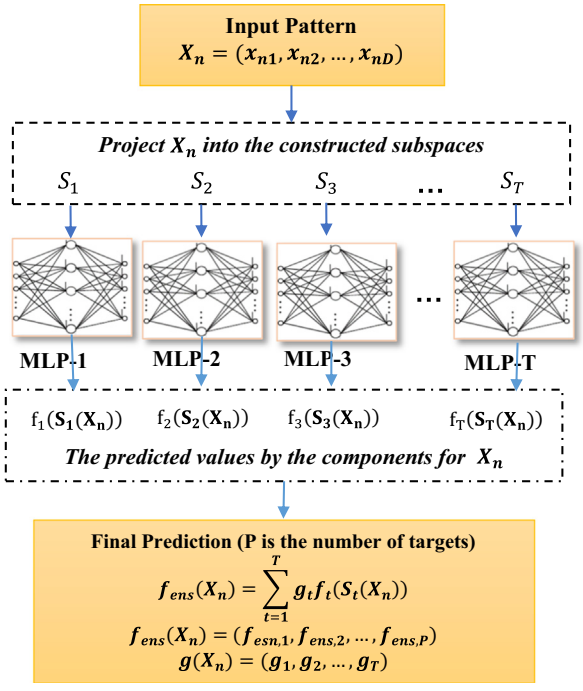

Fig. 1. Schematic of the proposed Boosted-NNE model with T components in testing phase.

The goal is to find a function (using the Boosted-NNE) $f: X \to Y$, such that f minimizes E.

The proposed training algorithm works in boosting-like manner (in T iterations, where T is the number of components) to train the components of the Boosted-NNE model on subspaces projected by the CLSP method. In each step t ($2 \le t \le T$), CLSP considers only $A\%$ of the difficult instances (IS_t) that have greater prediction error by the component added in step $t-1$ and implements the clustering algorithm on the input features and IS_t to obtain the clusters of features and to estimate the main factor of principal component analysis (PCA) as the latent component of that cluster. The union of the latent components of all clusters yields the input space S_t for training component t of the Boosted-NNE. The original training set is subsequently projected to S_t, after which the next component is trained on the projected training data and the proposed SNCL algorithm. After training the Boosted-NNE components, the CLMS algorithm is applied to obtain the component's optimal weights in the model's combination module.

In many NNE model implementations, the components are multi-layer perceptron neural networks (MLP) (Chao et al., 2014; Erdal et al., 2013). Thus we use MLP networks as the components of the Boosted-NNE. The pseudo code of the proposed algorithm to train and test the Boosted-NNE model is shown in Algorithm 1. The Boosted-NNE model in the testing phase is presented in Fig. 1.

Algorithm 1. The proposed algorithm to train and test the Boosted-NNE model

Training phase

Input:

S: A set of multi-target regression instances $\{(X_n, Y_n)|X_n \epsilon X, Y_n \epsilon Y\ , 1 \leq n \leq N\}$ with original input space $X = (x_1, x_2, \ldots, x_D)$

T: Number of components of Boosted-NNE model
***A*:** The percentage of difficult instances for implementing the CLSP method.

Output: The Boosted-NNE model

1. Train component network 1 using S_1 (a random subspace of X) and the BP algorithm.
2. ***For*** *t=2* to *T* ***do***
   - 2.1. Construct subspace S_t for training component t using the CLSP method:
     - 2.1.1. Prepare IS_t (a set of A% difficult instances in S that have higher prediction error by the component added in step $t-1$).
     - 2.1.2. Divide original input space X into a number of disjoint subspaces $C_1, C_2, \ldots, C_K$, using the proposed clustering algorithm and IS_t.
     - 2.1.3. ***For each*** C_i ***do***
       Obtain the main factor of PCA (F_i) using IS_t.
       End
     - 2.1.4. Prepare $S_t = (F_1, \ldots, F_K)$ that is the union of all F_is
   - 2.2. Prepare $P_t(S)$: Project the training data S into S_t.
   - 2.3. Train component t using $P_t(S)$ and the SNCL algorithm.

End

3. Estimation of the combination weight parameters of the MLP components using the CLMS algorithm.

Prediction Phase

- For a given input x, project it into subspaces $(S_1, S_2, \ldots, S_T)$
- Obtain the prediction results of the network components of the Boosted-NNE: $f_t(S_t(x))(t = 1, \ldots, T)$
- Calculate the value predicted by the Boosted-NNE model: $f_{ens}(x) = \sum_{t=1}^{T} g_t \times f_t(S_t(x))$

2.1. The proposed CLSP method

Considering there is correlation between features in the original input space, it is better to use clustering algorithms on input features to identify the disjoint subsets of features with high correlation (positive or negative) to one another and low correlation with features in other subsets. The clustering algorithm attempts to divide a set of input features into non-overlapping clusters, such that each cluster can be interpreted as an essentially one-dimensional component (called a latent component) using unsupervised projection such as principal component analysis (PCA) (Hadavandi et al., 2015; Vigneau and Qannari, 2003).

In this paper, we propose the CLSP method be used in each step t of the boosting approach $(2 \leq t \leq T)$ to construct a low-dimensional input space for training component network t from a set of D input features $(x_1, x_2, \ldots, x_D)$ (the first component is trained on a random subspace of original feature space). Instead of using a standard boosting scheme, which involves weighting the instances, the CLSP constructs a new subspace for training each component of the Boosted-NNE using only the most difficult instances. Thus, learning is biased towards difficult instances while avoiding extra emphasis on those difficult instances. The CLSP consists of two stages:

Stage 1. Only A% of the difficult instances (IS_t) that have greater average error by the component added in step $t-1$ $(2 \leq t \leq T)$ are selected. To select the difficult instances, the average error of component t for instance n ($E_{Pre_t}(X_n)$) that has P target features is defined as Eq. (1), where A is a parameter:

$$E_{Pre_t}(X_n) = \frac{1}{P}\sum_{k=1}^{P} |f_{tk}(X_n) - y_k(X_n)| \quad (1)$$

where $f_{tk}(X_n)$ and $y_k(X_n)$ are the predicted (by component t) and actual values of target k for instance n.

Stage 2. The features in the original input space X are clustered using Algorithm 2 and the instances in IS_t. Combining the latent components of all clusters produces the input space S_t for training component network t of the Boosted-NNE.

As a feature clustering method, hierarchical agglomerative clustering is employed, which starts with D clusters, each containing a single feature, i.e. $C_j^0 = \{x_j\}$, $j = 1, 2, \ldots, D$. This means there are as many clusters as there are input features. In the Kth stage $(0 \le K \le D-1)$ there are $D-K$ clusters available denoted by C_j^K, $j = 1, 2, \ldots, D-K$. It is assumed that the features are already grouped into K clusters and the kth cluster is composed of n_k features for $k = 1, 2, \ldots, K$ and $\sum_{k=1}^{K} n_k = D$. Let R_k be the $n_k \times n_k$ correlation matrix of the n_k features in the kth cluster C_k and let $w_k = (w_{k,1}, w_{k,2}, \ldots, w_{k,n_k})^t$ denote the eigenvector corresponding to its largest eigenvalue. Now, let v_k be the $n_k \times 1$ vector containing the indices of the original features clustered into the kth cluster in ascending order, i.e. $v_1 \le v_2 \le \ldots \le v_{n_k}$. Define the $D \times n_k$ indicator matrix IM_k for $k = 1, 2, \ldots, K$, such that IM_k has 1 at position (v_j, j) for $j = 1, 2, \ldots, n_k$ and 0 otherwise. Finally, for $k = 1, 2, \ldots, K$, define $D \times 1$ vectors b_k, such that $b_k = IM_k w_k$. The main factor of PCA for n_k features in cluster k serves as the latent cluster component and is obtained by the following linear combination:

$$F_k = IS_t b_k, \qquad k = 1, 2, \ldots, K \tag{2}$$

The proposed clustering method tries to maximize the objective function G_K for K clusters. Algorithm 2 represents the steps of the proposed clustering method in CLSP.

$$G_K = N \sum_{k=1}^{K} \sum_{j=1}^{D} \delta_{kj} Covariance^2(x_j, F_k)$$

$$\delta_{kj} = \begin{cases} 1 & \text{the } j\text{th feature belongs to cluster } C_k \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Algorithm 2. The proposed clustering method in CLSP

1. Start with D clusters, each containing a single feature C_j^0, $j = 1, 2, \ldots, D$
2. Merge two clusters C_j^0, C_i^0 with the largest squared coefficient of covariance between x_j and x_i. If $(i^*, j^*) = \underset{i \ne j}{Arg\ max} \{Covariance^2(x_i, x_j)\}$ then clusters $C_{i^*}^0, C_{j^*}^0$ are merged, and update the set of clusters to $\{C_1^1, C_2^1, \ldots, C_{D-1}^1\}$, which contains $D-1$ clusters.
3. In the Kth stage, there are $D-K$ clusters with the latent components F_k, $k = 1, 2, \ldots, D-K$. Merge two clusters C_j^K, C_i^K with the largest squared coefficient of covariance between their latent components. If $(i^*, j^*) = \underset{i \ne j}{\text{Arg max}} \{Covariance^2(F_i, F_j)\}$, then clusters $C_{i^*}^K, C_{j^*}^K$ are merged; update the set of clusters.
4. Continue merging and updating the clusters until a suitable number of clusters is achieved.

In PCA, the eigenvalues represent the amount of variance associated with each component. Since only the latent component of each cluster is used in constructing the new subspace, we use the variance explained by the latent component F_k as the variance explained by cluster C_k (the total variance for each cluster corresponds to the number of features in the cluster). To determine the most suitable number of clusters, we define the "total variance explained" index that gives the sum of the explained variance over all clusters. The "proportion of variance explained" (PV) is also defined, which represents the total variance explained divided by the sum of cluster variance that is the size of input space X. For example, in case of an input space with $D = 8$ features partitioned in 3 clusters with cluster variances (2.5, 2.1, 1), the total variance explained by the combination of all clusters' latent components is $2.5+2.1+1=5.6$, and the proportion of variance explained by the latent components is $5.6/8=70\%$. In this study, the most suitable number of clusters is obtained using Eq. (4).

$$N^* = Min\ \{k \mid PV_k \ge 0.5\} \tag{4}$$

The Dendrogram of the proposed hierarchical agglomerative clustering

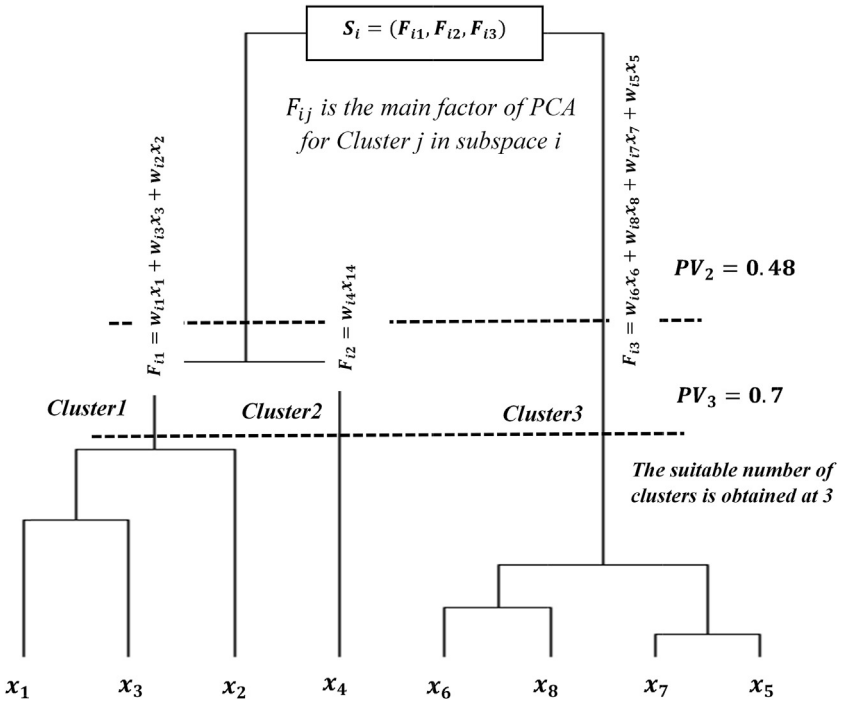

Fig. 2. Dendrogram of the proposed hierarchical agglomerative clustering in CLSP method.

for 8 input features is shown in Fig. 2. The most suitable number of clusters in this example is 3.

After implementing the CLSP method, K clusters are obtained with K latent components $F_1, \ldots, F_K$. Then $S_t = (F_1, \ldots, F_K)$ is set as the input space for training component network t and $P_t(S)$ is prepared as the projected training dataset S into S_t. Thus, all projected training instances are used to train component t.

2.2. The proposed sequential negative correlation learning

Negative correlation learning is an explicitly diverse ensemble development method based on bias–variance–covariance decomposition of the ensemble's generalization error. In order to understand the ensemble's generalization abilities, Theorem 1 is recalled from Ueda and Nakano, (1996) and formulated as follows:

Theorem 1. *The generalization error of an ensemble model averaged over all possible training datasets D_N of size N, in which all components are trained using the same dataset, can be expressed using the following bias–variance–covariance decomposition:*

$$E_{f_{ens}}(D_v) = \frac{1}{N}\sum_{n=1}^{N}\left[\frac{1}{T}\overline{Var}(X_n) + \frac{T-1}{T}\overline{Cov}(X_n) + \overline{Bias}(X_n)^2\right] \tag{5}$$

where D_v is the validation dataset, and

$$\overline{Var}(X_n) = \frac{1}{T}\sum_{t=1}^{T} E_{D_N}\left\{(f_t(X_n) - E_{D_N}\{f_t(X_n)\})^2\right\} \tag{6}$$

$$\overline{Cov}(X_n) = \frac{1}{T(T-1)}\sum_{t=1}^{T}\sum_{k\neq t} E_{D_N}\{(f_t(X_n) - E_{D_N}\{f_t(X_n)\})(f_k(X_n) - E_{D_N}\{f_k(X_n)\})\} \tag{7}$$

$$\overline{Bias}(X_n) = \frac{1}{T}\sum_{t=1}^{T}(E_{D_N}\{f_t(X_n)\} - y(X_n)) \tag{8}$$

where N is the number of training instances, T is the ensemble size, $f_t(X_n)$ and $f_{ens}(X_n) = \frac{1}{T}\sum_{t=1}^{T} f_t(X_n)$ are the output of component t and the ensemble model for X_n, respectively. The proof for Theorem 1 is provided by Geman et al. (1992). Eq. (5) shows that managing the covariance term $\overline{Cov}(X_n)$ explicitly helps control the disagreement among ensemble components' outputs, hence producing a better generalized ensemble model. The cost function of NCL for component t $(1 \le t \le T)$ is given in Eqs. (9) and (10):

$$E_t = \left[\sum_{n=1}^{N}(f_t(X_n) - y(X_n))^2\right] + \lambda P_t \tag{9}$$

$$P_t = \sum_{n=1}^{N}(f_t(X_n) - f_{ens}(X_n))\sum_{L=1, L\neq t}^{T}(f_L(X_n) - f_{ens}(X_n)) = -\sum_{n=1}^{N}(f_t(X_n) - f_{ens}(X_n))^2 \tag{10}$$

where P_t is the error correlation penalty coefficient and λ is a scaling coefficient parameter that controls the trade-off between the objective and penalty function. When $\lambda = 0$, the ensemble components are trained independently with the BP algorithm. McKay and Abbass (2001) revealed that NCL only works to drive the ensemble components away from their mean but not necessarily away from each other. Therefore, low diversity with NCL is not surprising. Thus, the SNCL algorithm is proposed in the current work to train the components of the Boosted-NNE model. The SNCL algorithm in combination with the CLSP method employs a moving average for the correlation penalty term in the error function of component t that measures the error correlation between itself and previous components $(1, 2\ldots, t-1)$ in the boosting approach. It sequentially positions each component in a different place relative to previous components, in the hope that the components of the proposed model trained by SNCL get pushed away from each other, thus increasing diversity among them. The proposed cost function in the SNCL algorithm for training component t is given in Eqs. (11) and (12):

$$E_t = \sum_{k=1}^{P}\sum_{n=1}^{N}(f_{tk}(S_t(X_n)) - y_k(X_n))^2 + \lambda P_t \tag{11}$$

$$P_t = \sum_{k=1}^{P}\left[\sum_{n=1}^{N}(f_{tk}(S_t(X_n)) - y_k(X_n))\sum_{L=1}^{t-1}(f_{Lk}(S_t(X_n)) - y_k(X_n))\right] \tag{12}$$

Algorithm 3. The proposed SNCL algorithm for training component t in the boosting approach

Input:

- ✓ Training dataset $\{(S_t(X_j), Y_j) |\ X_j \epsilon X,\ Y_j \epsilon Y,\ 1 \le j \le N\}$, $S_t(X_j)$ is the projection of X_j into subspace S_t obtained using the CLSP method;
- ✓ η_e and γ_e are the learning rate and momentum rate in the BP algorithm
- ✓ Integer R is the number of iterations in BP.

Output: Weights vector of component t

1. Initialize the weight vector of the MLP component t $(W = [w_{i,j}])$ between -1 and 1 randomly.
2. For $Iter = 1$ to R Do:
   - 2.1. Compute the error function of component t (E_t) using Eqs. (11–12)
   - 2.2. Compute $\frac{\partial E_t}{\partial w_{i,j}}$:

$$\frac{\partial E_t}{\partial w_{i,j}} = 2\sum_{k=1}^{P}\sum_{n=1}^{N}(f_{tk}(S_t(X_n)) - y_k(X_n))\frac{\partial f_{tk}}{\partial w_{i,j}} + \lambda\sum_{k=1}^{P}\sum_{n=1}^{N}\sum_{L=1}^{t-1}\frac{\partial f_{tk}}{\partial w_{i,j}}(f_{Lk}(S_t(X_n)) - y_k(X_n))$$

   - 2.3. update the weights vector:

$$\Delta w_{i,j}(iter) = -\eta\left(\frac{\partial E_t}{\partial w_{i,j}}\right) + \gamma\Delta w_{i,j}(iter-1) \rightarrow w_{i,j}(iter+1) = w_{i,j}(iter) + \Delta w_{i,j}(iter)$$

3. End

where $S_t(X_n)$ is the projection of X_n into subspace S_t obtained through the CLSP method; $f_{tk}(S_t(X_n))$ and $y_k(X_n)$ are the predicted (by component t) and actual values of target k of instance n, respectively ($1 \le k \le P$), and N is the number of training instances. Algorithm 3 presents the steps of the SNCL algorithm.

2.3. Combination method

After training the MLP components of the Boosted-NNE model, a combination method must be designed to integrate the outputs of the ensemble model components. The goal of the combination phase is to fully benefit from the positive effects of diversity and avoid the negative effects (Yang et al., 2013). As for an ensemble of MLP components, the most commonly utilized combination methods for regression problems are simple averaging and weighted averaging. Apparently the former is just a special case of the latter and cannot perform better than the latter. However, ways of finding appropriate weights for each component in an NNE remain worthy of being explored (Tian et al., 2012; Yang et al., 2013). In our multi-target regression problem, the final prediction of the Boosted-NNE model with T components for target k ($1 \le k \le P$) is computed in Eq. (13):

$$f_{ens,k}(X_n) = \sum_{t=1}^{T} g_{tk} f_{tk}(X_n) \tag{13}$$

where $f_{ens,k}(X_n)$ and $f_{tk}(X_n)$ are the values predicted by the ensemble model and component t for target k of instance n, respectively and g_{tk} is the weight of component t for target k.

There are different methods of obtaining the weight parameters of components in an NNE model. Related works can be divided into two categories. The first comprises methods that find optimal weights based on the constrained least mean square error algorithm (CLMS) with a mathematical framework. By applying an optimization technique, e.g. Lagrange multipliers, the combination weights are found (Ueda, 2000). In the second category, optimal combination weights are found by an evolutionary algorithm based on the results of the individual components in the training dataset (Nabavi-Kerizi et al., 2010). In this paper the CLMS algorithm is applied to obtain the optimal weights of the Boosted-NNE model components. The weights g_{tk} must be computed in such a way as to minimize the expected quadratic deviation of function E for the given training set. In this paper, the problem of computing the components' weight parameters for each target feature k ($1 \le k \le P$) is formulated as follows:

$$Minimize\ E = \sum_{n=1}^{N} \left(\boldsymbol{y_k}(\boldsymbol{X_n}) - \left(\sum_{t=1}^{T} g_{tk} f_{tk}(X_n) \right) \right)^2 \sum_{t=1}^{T} g_{tk} = 1 g_{tk} \ge 0. \tag{14}$$

3. Experimental setup

In this section, the applicability of the Boosted-NNE model is examined using three popular multi-target regression problems and a real dataset of practice–performance relationships. The network components of the proposed ensemble models are MLP networks with one hidden layer. The activation function for all components' neurons is the sigmoid function $G(x) = 1/(1+e^{-x})$. Performance of the Boosted-NNE is compared with some classic ensembles and single models in multi-target regression problems, including

(1) Original NCL: in this model the MLP components are trained with the NCL algorithm.
(2) NNE: in this model the RSM method is used to divide the input features into T subsets with a predefined size of 4 features. In this model, the MLP components are trained using the BP algorithm.
(3) There are two other ensemble models, including the NNE based on a combination of boosting and RSM methods (NNEBR) and NNE based on the bagging method (NNE (bagging)). In the NNEBR model with size T, the MLP components are also trained using the BP algorithm in boosting-like manner by selecting 70% of difficult instances and a random subspace of the original feature space. In the NNE (bagging) model, the MLP components are also trained with the BP algorithm in bagging-like manner by selecting 70% of instances with replacement.

The combination method for NCL, NNE, NNEBR and NNE (bagging) is the average. Four parameters are to be set for the Boosted-NNE model and other ensemble models (number of hidden neurons in component networks L, ensemble size T, learning rate η_e, momentum rate γ_e and number of training epochs). The penalty coefficient λ must also be set for the NCL and Boosted NCL models. In line with Alhamdoosh and Wang, (2014), the exhaustive linear search strategy was adopted to find the optimal values for these parameters. The ensemble size T was searched in the [5, 10] range with step 1 and the number of hidden neurons L in the component networks was searched in the [5, 8] range with step 1. The penalty coefficient λ was searched in the [0, 0.7] range with step 0.1, while the learning rate and momentum rate were searched in the [0.08, 0.1] and [0.84, 0.9] ranges with steps 0.01 and 0.02, respectively. The number of training epochs #E was set in the [600, 1000] range with step 200. All parameter values can be further tuned using other model selection methods, but we are only interested in highlighting the relative performance of the Boosted-NNE in comparison with other ensemble models. All models were assessed using the 10-fold cross-validation procedure. Each dataset was randomly partitioned into ten subsets that initiate 10 runs for each experiment. For each run, one subset was employed for testing and the remaining subsets were used all together for training. The performance measures from all folds were collected and averaged over ten. For the components used in the work, the normalization of data in the $[-1, 1]$ range was applied.

The average of the relative root mean squared error (aRRMSE) over all target features was employed in this work as the performance measure used in different related papers (Aho et al., 2012; Spyromitros-Xioufis et al., 2014). The RRMSE for a target is equal to the root mean squared error (RMSE) for that target divided by the RMSE of predicting that target's average value in the training set. This standardization facilitates performance averaging across non-homogeneous targets. For the kth target feature ($1 \le k \le P$), $RRMSE_k$ is computed with Eq. (15):

$$RRMSE_k = \sqrt{\frac{\sum_{n=1}^{N} (f_{ens,k}(X_n) - y_k(X_n))^2}{\sum_{n=1}^{N} (\overline{y_k} - y_k(X_n))^2}} \tag{15}$$

$$aRRMSE = \frac{\sum_{k=1}^{P} RRMSE_k}{P} \tag{16}$$

where X_n is the input vector of instance n ($n = 1, 2, \ldots N$), P is the number of targets, $f_{ens,k}(X_n)$ and $y_k(X_n)$ are the predicted (by ensemble model) and actual values of output k for instance n, respectively, and $\overline{y_k}$ is the average of target k over the instances. In order to establish the statistical significance, statistical tests are applied to comparatively evaluate the prediction accuracy of the models used in this study.

3.1. Three multi-target regression datasets

Despite the numerous interesting applications of multi-target regression there are few publicly available datasets of this kind, perhaps because most applications are industrial (Spyromitros-Xioufis et al., 2014). In this work, experiments are carried out with three high-dimensional datasets of multi-target regression problems that have a small number of instances with continuous input and output features. A description of each dataset is as follows.

EDM

The Electrical Discharge Machining dataset (Karalič and Bratko, 1997) represents a two-target regression problem. The dataset comprises 154 samples and 16 input features, aimed at predicting two real value targets (D-Gap, D-Flow). The task is to shorten the machining time by reproducing the behavior of a human operator, which controls the values of two features. Each target feature takes 3 distinct numeric values (-1, 0, 1) and there are 16 continuous input features.

Energy Efficiency

This dataset is obtained from an analysis of 12 different building shapes simulated in Ecotect (Tsanas and Xifara, 2012). The buildings differ in terms of glazing area, glazing area distribution, and orientation amongst other parameters. Various settings are simulated as functions of the afore-mentioned characteristics to obtain 768 building shapes. The dataset comprises 768 instances and 8 input features, intended to predict two real value responses (Heating Load (HL) and Cooling Load (CL)).

Concrete Slump Test

Concrete is a highly complex material. The slump flow of concrete is not only determined by the water content, but is also influenced by other concrete ingredients. The dataset contains 103 records with seven components of the input vector (cement (kg/m^3), fly ash (kg/m^3), blast furnace slag (kg/m^3), water (kg/m^3), superplasticizer (kg/m^3), coarse aggregate (kg/m^3), and fine aggregate (kg/m^3)) and three target vector components (slump (cm), flow (cm), 28-day compressive strength (Mpa)). Since in the slump-cone test the slump flow can be deduced by measuring the diameter of the slumped fresh concrete, the minimum slump flow is the bottom diameter (20 cm) of the slump cone (Yeh, 2007).

3.2. The practice–performance relationship dataset

A firm's competitive strategy motivates its operations strategy, leading to operations decisions that result in particular desired performances (Amoako-Gyampah and Acquaah, 2008). There is accord in the literature regarding the importance of determining the effects of improvement programs (e.g., quality improvement and supply chain management programs) on manufacturing performances (e.g., cost and manufacturing flexibility) in managers' decision-making processes (Li et al., 2006). Although various structures, such as ideal models, different hypotheses have been benchmarked and tested in practice–performance relationship studies for linking practice to performance, the complexity of attaching mathematical relationships in instances with numerous variables renders it difficult to attain a normative generic model for these linkages (Rusjan, 2005). Modeling the effects of improvement programs on plants' objective performances can be viewed as a multi-target regression problem, where the feature vector describes the plant's strategic capabilities and environmental factors, previously implemented improvement programs, and target features as plants' manufacturing performances.

In this paper, the dataset gathered by Hajirezaie et al. (2010) is utilized. The dataset was collected through a survey questionnaire containing the principal objectives of the practice–performance problem for small and medium-sized plants located in Iran, such as automotive suppliers, oil and gas industry suppliers, electronics, machinery and so on, in mid-2009. The total number of completed questionnaires was 105, so there were 105 instances with 18 input features and 5 target features. These features are described next.

Strategic capabilities (goals)

Six questions considered the importance of the following capabilities: lower selling prices (Var1), superior product design and conformance quality (Var2), more dependable deliveries (Var3), faster deliveries (Var4), superior customer service (Var5) and wider product range (Var6). From the viewpoint of major customers to win orders, the importance of indicators in the last three years was inquired on a scale of 1–5 (in the range of not important to very important).

Past improvement programs

Six main improvement program categories are considered to measure the degree of implementation of these practices over the last three years on a five-point Likert scale, from 1 to 5 (1=no usage, 5=high usage). The category of planning and control (Var7), quality (Var8), product development (Var9), technology (Var10), organization (Var11) and supply chain-related programs (Var12) was added to the model as another independent input.

Environmental/control variables

These variables are environmental factors of a plant (control variables or environmental dynamism) such as the plant size (Var13) measured by the number of employees, plant ownership (Var14) from public to private ownership, and plant age (Var15) as the plant's history. Three variables also describe the external environment of the companies: market dynamics (Var16) on a scale of 1–5 (declining rapidly to growing rapidly), market span (Var17) on a scale of 1–5 (few to many segments) and market concentration (Var18) on a scale of 1 to 5 (few to many competitors).

Manufacturing performance criteria (target features)

The respondents were asked to determine the change in indicators over the last three years on a scale of 1–5 (deteriorated more than 10%, stayed about the same, improved 10–30%, improved 30–50%, improved more than 50%). The indicators are manufacturing conformance and product quality, mixed flexibility, time on the market, customer service and support, delivery speed, delivery dependability, unit manufacturing cost, manufacturing lead time, labor productivity and inventory turnover. These were grouped into 5 main categories: quality conformance (Var19), cost efficiency (Var20), dependability (Var21), speed (Var22) and flexibility (Var23).

4. Experimental results and discussion

In a boosting-like manner, component network t of the Boosted-NNE is trained on the projected subspace S_t obtained using the CLSP method ($t = 2, 3, \ldots T$). A random subspace half of the original input space is selected to obtain subspace S_1 to train component 1 of the Boosted-NNE using the BP algorithm. In this paper, A is set as 50% in the CLSP method. After training the model components, the optimal weights of the MLP components in the combination module of the Boosted-NNE are computed by solving problem (14).

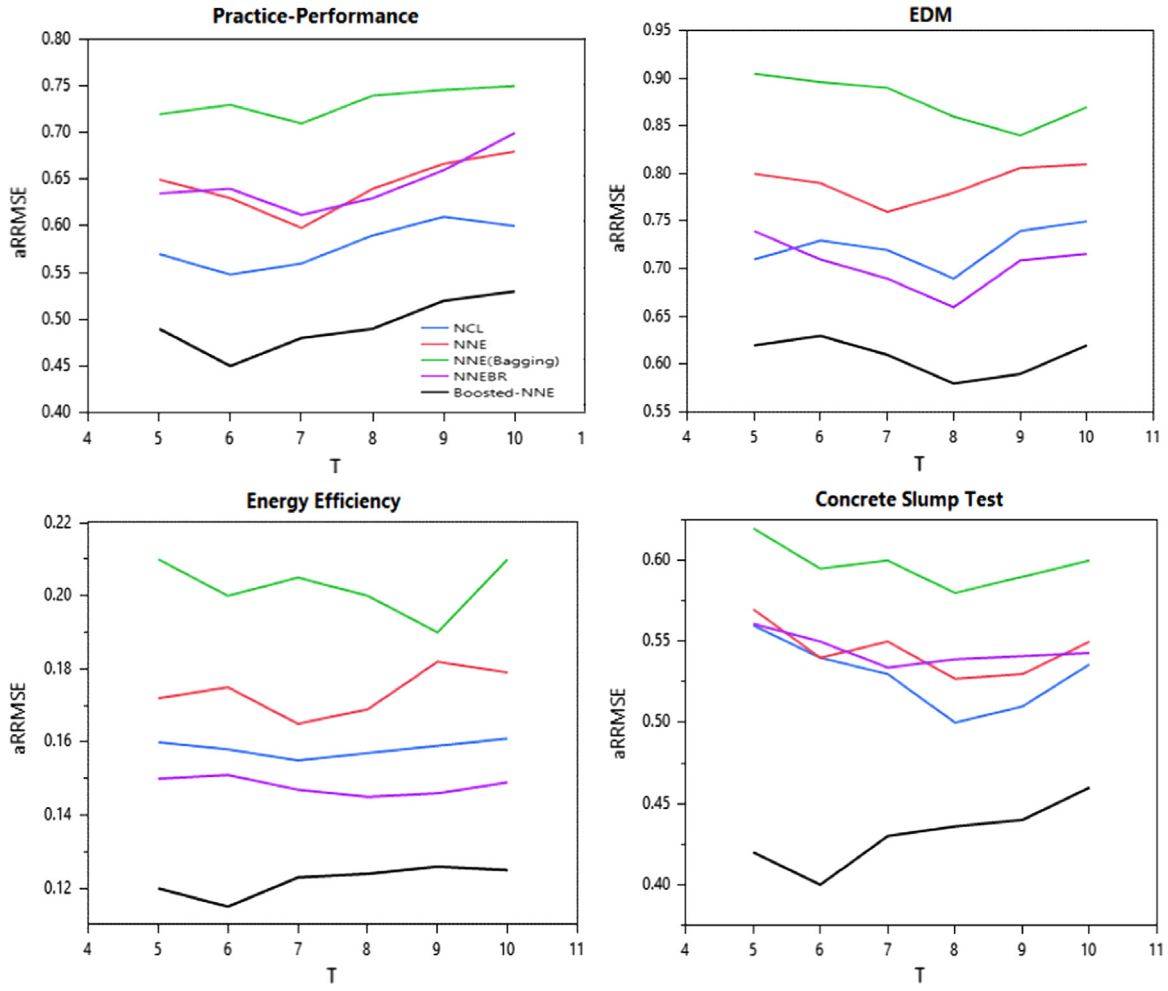

Fig. 3. aRRMSE of the Boosted-NNE and other ensemble models for different ensemble sizes.

4.1. The effects of the parameters on the ensemble models' performance

The performance of the Boosted-NNE model is assessed in different experimental designs. Fig. 4 plots the testing generalization error of the Boosted-NNE and the other ensemble models for different ensemble sizes and for all datasets. For each ensemble size in Fig. 3, the best penalty coefficient and other component network parameters are used.

Fig. 3 indicates that increasing the number of component networks is not always beneficial to ensemble generalization accuracy and the Boosted-NNE's performance is always better than that of other models regardless of ensemble size. Apparently, 6–8 component networks are sufficient to constitute well-generalized ensemble models using the Boosted-NNE. Besides ensemble size, the penalty coefficient of negative correlation learning plays a key role in the performance of the Boosted-NNE and NCL models. Fig. 4 presents the testing aRRMSE of the Boosted-NNE and NCL in terms of the penalty coefficient with the best ensemble size for each dataset obtained in Fig. 3.

Fig. 4 demonstrates that the testing generalization error slightly declines as the penalty coefficient value increases. However, the performance of the Boosted-NNE improves when λ is near 0.5. According to this figure, when λ is greater than 0.5, the aRRMSE of NCL and Boosted-NNE increases. In such cases, the training procedure mainly minimizes the error correlation penalty term in the error function rather than the ensemble error.

4.2. Diversity error of the Boosted-NNE model

In this part the characteristics of the Boosted-NNE model are evaluated with 6 components obtained in Fig. 3. The CLSP method results in clustering the original input space of the practice–performance problem and constructing a training space for the component networks in the Boosted-NNE are shown in Table 3. Clearly, the clusters in the subspaces contain different subsets of features. This result increases the difference among subspaces for component training. Moreover, using hierarchical clustering in the CLSP method results in different dimensions of each subspace that can improve the diversity among components. Subspace projection is faster and more stable, as many input features with a small number of instances usually yield to ill-posed problems. The CLSP method, as an implicit diversity creation method, constructs different subspaces for training each component with focus on the difficult instances by manipulating the training dataset distribution. Therefore, it is expected for the Boosted-NNE diversity to be superior to other ensemble models. Furthermore, due to the high-dimensionality and small number of instances in the datasets, the components are hardly able to learn the relationships between inputs and targets. In the case of using the CLSP method, a smaller number of input features in S_t allows component t of the Boosted-NNE to approximate the projection more accurately (the same results are obtained for other datasets).

The scatterplot matrix of the newly obtained subspaces S_2, S_3 and their latent components' equations for the practice–performance problem are shown in Fig. 5. This figure indicates that the

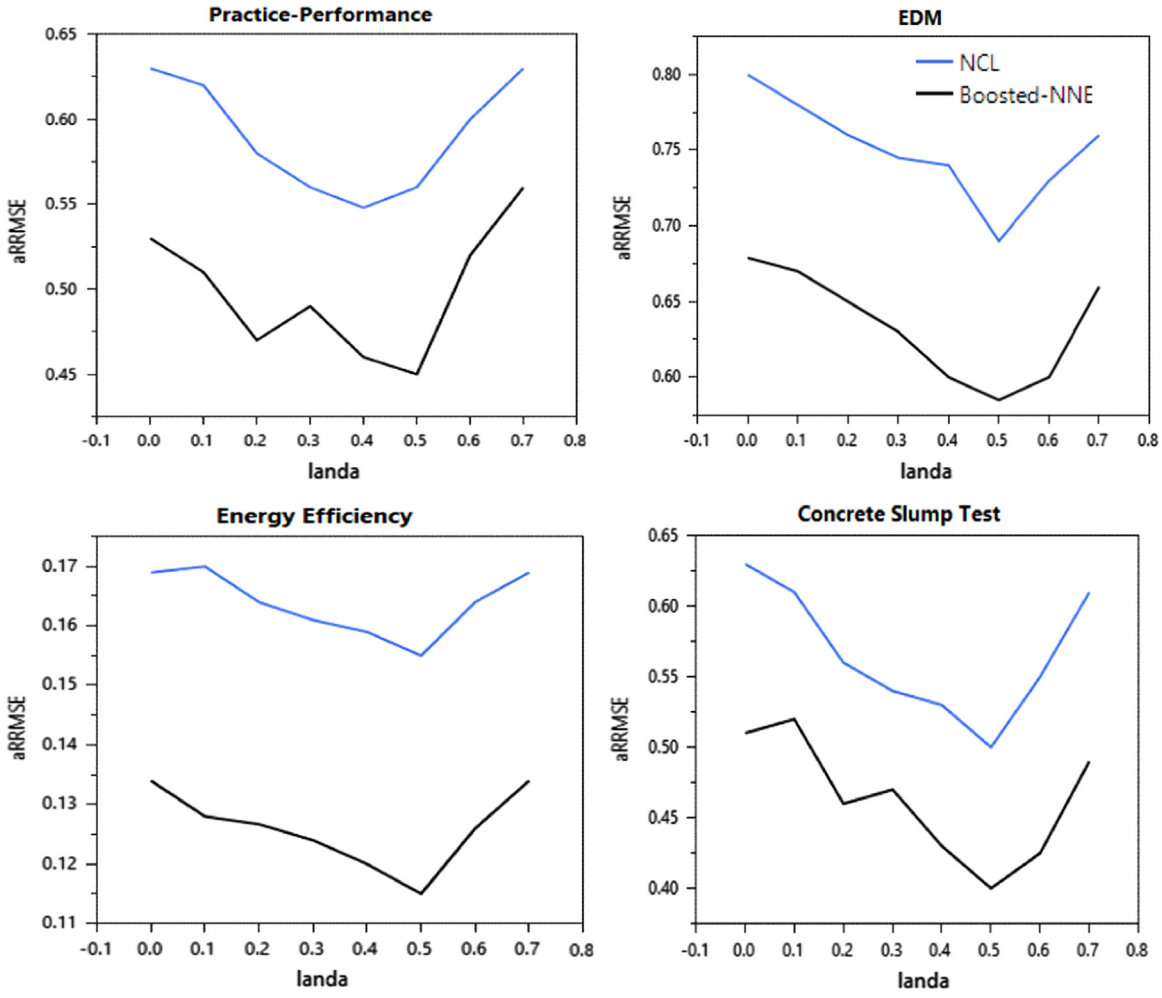

Fig. 4. aRRMSE of the Boosted-NNE and NCL models for different penalty coefficient values and the best ensemble size obtained in Fig. 3.

Table 3
The constructed subspaces using CLSP method in different steps for the practice–performance dataset.

S_2		S_3		S_4		S_5		S_6	
Cluster	**Members**	**Cluster**	**Members**	**Cluster**	**Members**	**Cluster**	**Members**	**Cluster**	**Members**
1	Var3	1	Var9	1	Var5	1	Var4	1	Var2
	Var17		Var10		Var12		Var5		Var13
	Var11		Var17		Var4		Var12	2	Var8
2	Var13	2	Var3	2	Var9		Var14		Var11
	Var15		Var8		Var10	2	Var2	3	Var3
3	Var5	3	Var14		Var6		Var7		Var17
	Var12		Var18		Var16		Var8	4	Var15
	Var4	4	Var6		Var7	3	Var10		Var16
4	Var16		Var11	3	Var1		Var13		Var1
	Var18		Var1		Var18		Var16	5	Var6
5	Var2	5	Var15	4	Var14	4	Var6		Var7
	Var6		Var16		Var15	5	Var15	6	Var5
6	Var9	6	Var4	5	Var3	6	Var3	7	Var12
	Var10		Var13		Var17		Var9		Var4
	Var7		Var2	6	Var2		Var17		Var9
7	Var1	7	Var5	7	Var8	7	Var1	8	Var10
	Var8		Var12		Var13		Var11		Var14
8	Var14		Var7	8	Var11		Var18		Var18

CLSP constructs the interpretable sparse latent components from clusters of features. The motivation is that important features constituting a certain latent component are correlated to each other more so than to other features. If the features are optimally clustered, the nonzero-coefficient of an interpretable latent component can be obtained from the feature correlation matrix in the corresponding cluster. Consequently, the remaining feature coefficients become zero. These results corroborate the findings of Enki et al. (2013). Also, there is low correlation among latent components in each subspace, which decreases redundancy

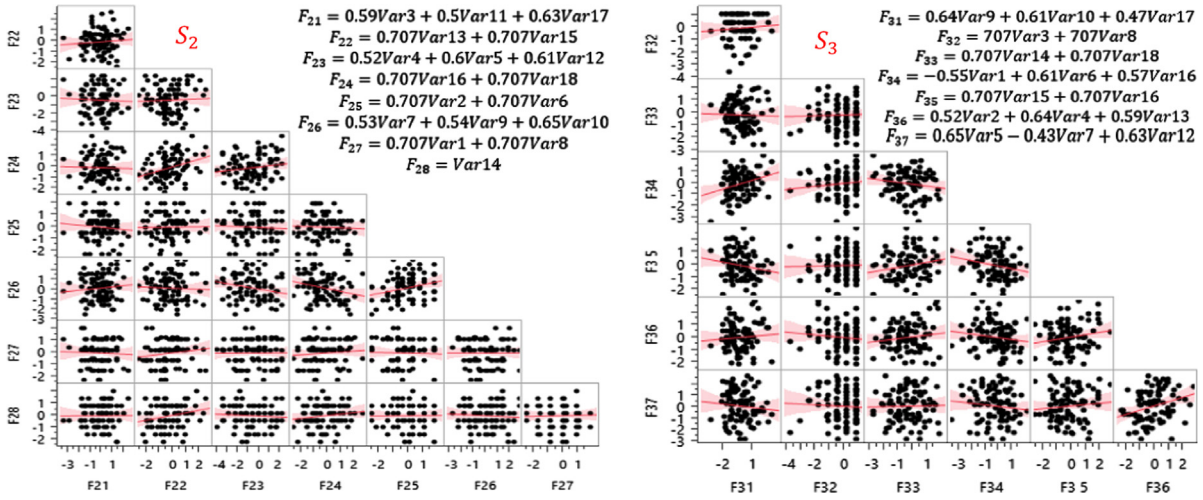

Fig. 5. The scatterplot matrix of the projected subspaces and equations of their latent components for subspace S_2 (for training component 2) and S_3 (for training component 3) obtained using CLSP method for Boosted-NNE in the practice–performance problem.

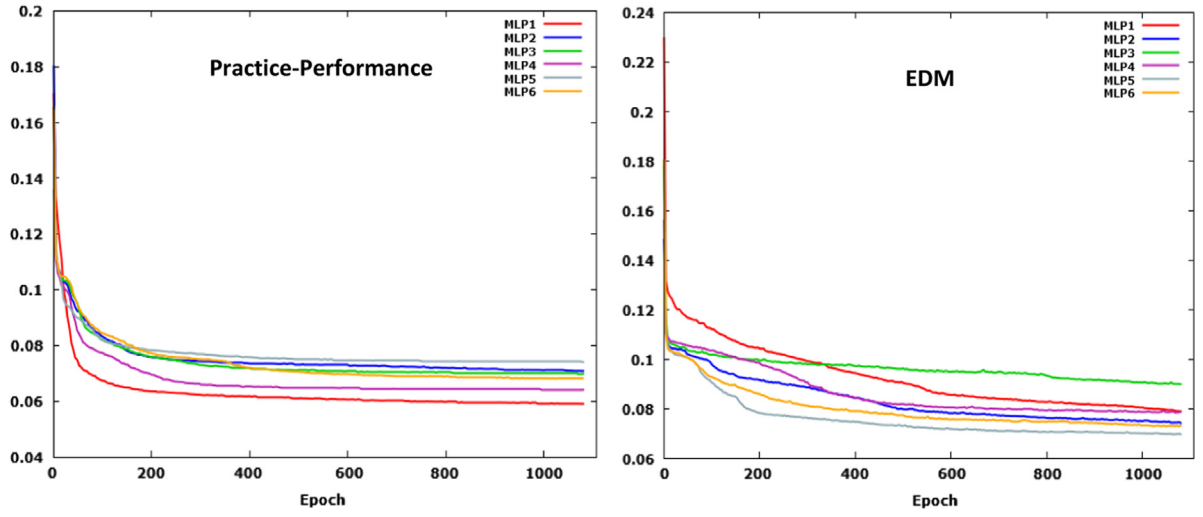

Fig. 6. Convergence trend of prediction error for MLP components of the Boosted-NNE trained with SNCL in the practice–performance and EDM problems.

among them. Now, because of the small number of training instances and high-dimensionality in our cases, feature reduction can improve the components' training accuracy. The same results are obtained for other subspaces.

Fig. 6 displays the convergence trend of each component's prediction error in the training process with the SNCL algorithm for the practice–performance and EDM problems. As shown in Fig. 6, predication error trend of components is decreasing, so the components in the Boosted-NNE model are localized in the training process.

An efficient ensemble model is constructed using components with low errors but that are diverse. These two objectives are in conflict with each other, because if the errors of two components are small, they cannot be very different (García-Pedrajas et al., 2012). The CLSP method is an implicit diversity creation method that constructs new subspaces for training each component with focus on difficult instances by manipulating the training dataset distribution. Additionally, as an explicit diversity creation method, the SNCL uses the objective function proposed in Eq. (11) to train diverse and accurate components. Several diversity measures have been proposed for analyzing the behavior of ensemble models. One of the techniques is diversity-error diagrams (Pardo et al., 2013). These are scatter plots with a point for each pair of components in the ensemble model. The horizontal axis represents the diversity between two components and the vertical axis represents the average error of the two components. In this study, we use the RMSE value for error rate and error correlation value between two components to measure diversity. A smaller error correlation value indicates greater diversity. Fig. 7 shows a diversity-error diagram for the Boosted-NNE model and other ensemble models with 6 components for "cost" target in the practice–performance problem (thus there are 15 points for each model in the scatter plot).

The diversity-error diagram shows that the Boosted-NNE achieved excellent results by reducing individual component's errors and enhancing diversity among them. According to the results, the Boosted-NNE components trained by SNCL tend to be negatively correlated (low correlation prediction error results) because each individual component learns different parts or aspects of the training data, so that the problem of correlated errors can be removed or alleviated. Moreover, a model such as NNE or NNEBR reduces the components' errors at the cost of less diverse ensembles. Fig. 7 shows that it is possible to improve individual components' accuracy at the cost of less diversity while the overall ensemble performance benefits.

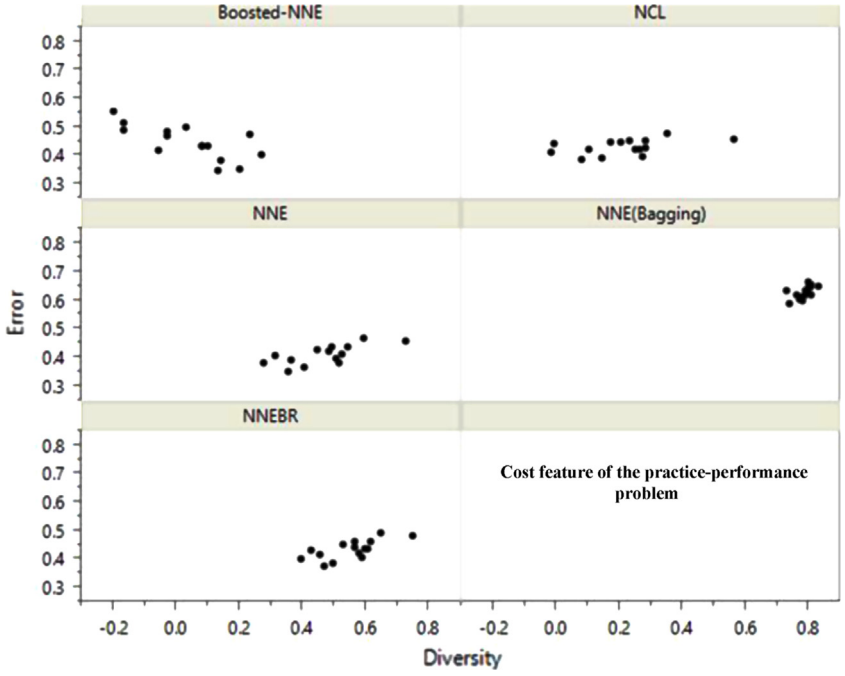

Fig. 7. Diversity-error diagram for the Boosted-NNE in the cost feature of the practice–performance problem.

Table 4
Comparison of RRMSE for multi-target regression datasets. For each dataset, the RRMSE for each target and average of all targets are presented.

Problem	Target attribute	NCL	NNE	NNE (Bagging)	NNEBR	Boosted-NNE	Single MLP
Practice–performance	Quality conformance	0.52	0.59	0.66	0.61	0.46	0.75
	Cost efficiency	0.61	0.7	0.72	0.66	0.52	0.73
	Dependability	0.45	0.52	0.62	0.62	0.37	0.69
	Speed	0.48	0.58	0.64	0.58	0.39	0.78
	Flexibility	0.68	0.6	0.82	0.59	0.51	0.89
	aRRMSE	0.548	0.598	0.692	0.612	0.45	0.768
EDM	D-Gap	0.74	0.78	0.87	0.69	0.61	0.92
	D-Flow	0.64	0.74	0.81	0.63	0.56	0.87
	aRRMSE	0.69	0.76	0.84	0.66	0.585	0.895
Energy Efficiency	HL	0.16	0.17	0.18	0.16	0.13	0.19
	CL	0.15	0.16	0.2	0.13	0.1	0.21
	aRRMSE	0.155	0.165	0.19	0.145	0.115	0.2
Concrete Slump Test	Slump	0.56	0.48	0.59	0.56	0.41	0.71
	Flow	0.51	0.61	0.63	0.51	0.44	0.87
	Compressive strength	0.43	0.5	0.51	0.48	0.36	0.38
	aRRMSE	0.50	0.53	0.58	0.52	0.40	0.65

Table 5
Optimal parameter values of each dataset for the results in Table 4.

Dataset	Boosted-NNE						NCL						NNE					NNE (Bagging)					NNEBR				
	T	λ	L	η_e	γ_e	$\#E$	T	λ	L	η_e	γ_e	$\#E$	T	L	η_e	γ_e	$\#E$	T	L	η_e	γ_e	$\#E$	T	L	η_e	γ_e	$\#E$
Practice–performance	6	0.5	6	0.1	0.9	1000	6	0.4	7	0.09	0.9	800	7	8	0.09	0.88	1000	7	8	0.1	0.88	1000	7	7	0.09	0.86	600
EDM	8	0.5	6	0.1	0.9	1000	8	0.5	6	0.09	0.88	1000	7	7	0.09	0.88	800	9	8	0.1	0.9	1000	8	7	0.09	0.88	600
Energy Efficiency	6	0.5	7	0.09	0.88	1000	7	0.5	7	0.09	0.88	1000	7	6	0.1	0.9	800	9	8	0.1	0.9	1000	8	6	0.1	0.88	800
Concrete Slump Test	6	0.5	6	0.1	0.9	1000	8	0.5	7	0.1	0.88	1000	8	6	0.1	0.88	800	8	7	0.1	0.9	1000	7	7	0.1	0.9	600

Furthermore, the results signify that diversity might be more useful when the components are mostly accurate. For instance, in the cost target of the practice–performance problem, the Boosted-NNE achieves the best accuracy by improving the components' diversity, but with the particularity that most individual components are still highly accurate. This finding suggests that promoting diversity may be advisable, provided accurate components are available.

4.3. Comparison with other ensemble models

Table 4 displays a comparison of the performance of the Boosted-NNE and five popular ensemble and single models: NNE, NCL, NNE (Bagging), NNEBR and single-MLP neural networks. The reported results were collected based on the best parameter values for each model and dataset (see Table 5) and were selected using the search criteria explained in Section 3.

From Table 4, it is evident that the Boosted-NNE model always performs better than other ensemble and single models. Moreover, the Boosted-NNE model is less complex than other ensemble models in four cases (compare the T and L columns in Table 5).

In order to establish the statistical significance, the Friedman (Sheskin, 2003) test is applied to comparatively evaluate the prediction accuracy between the best obtained ensemble models in Table 4 for each case. The experiment is designed such that the statistical significance between the models' RMSE will be examined. For this purpose, the tests are carried out on the prediction errors of four datasets obtained from 5 ensemble models. In Table 6, the results of applying the Friedman test are shown in order to detect whether there are any differences in the results. This test is applied with a level of confidence of $\alpha = 0.05$.

Friedman's tests indicate significant differences in the prediction results. Regarding these results, a post-hoc statistical analysis is required. Thus, the best performing model associated with the lowest rank (the Boosted-NNE model) will be chosen as the

Table 6
Average ranks obtained by each model in Friedman test.

Model	Ranking			
	Practice–performance	EDM	Energy Efficiency	Concrete Slump Test
NNE	3.14	3.23	3.58	2.68
NCL	2.48	2.87	2.96	3.25
NNE (Bagging)	4.16	4.61	4.61	4.23
NNEBR	3.56	2.5	2.39	2.98
Boosted-NNE	1.66	1.79	1.46	1.86
Friedman statistic (P-value)	28.8(0)	32.74 (0)	37.21(0)	25.51(0)

Table 7
Post-hoc statistical analysis results.

Model	Practice–performance		EDM		Energy Efficiency		Concrete Slump Test	
	P_{Holm}	$P_{Hochberg}$	P_{Holm}	$P_{Hochberg}$	P_{Holm}	$P_{Hochberg}$	P_{Holm}	$P_{Hochberg}$
NNE	0.0038	0.0038	0.0054	0.0055	0.00089	0.0008	0.0061	0.0062
NCL	0.0092	0.0093	0.0065	0.0065	0.0025	0.0024	0.00032	0.00032
NNE (Bagging)	0	0	0	0	0	0	0	0
NNEBR	0.0024	0.00241	0.008	0.0084	0.0041	0.0041	0.005	0.0052

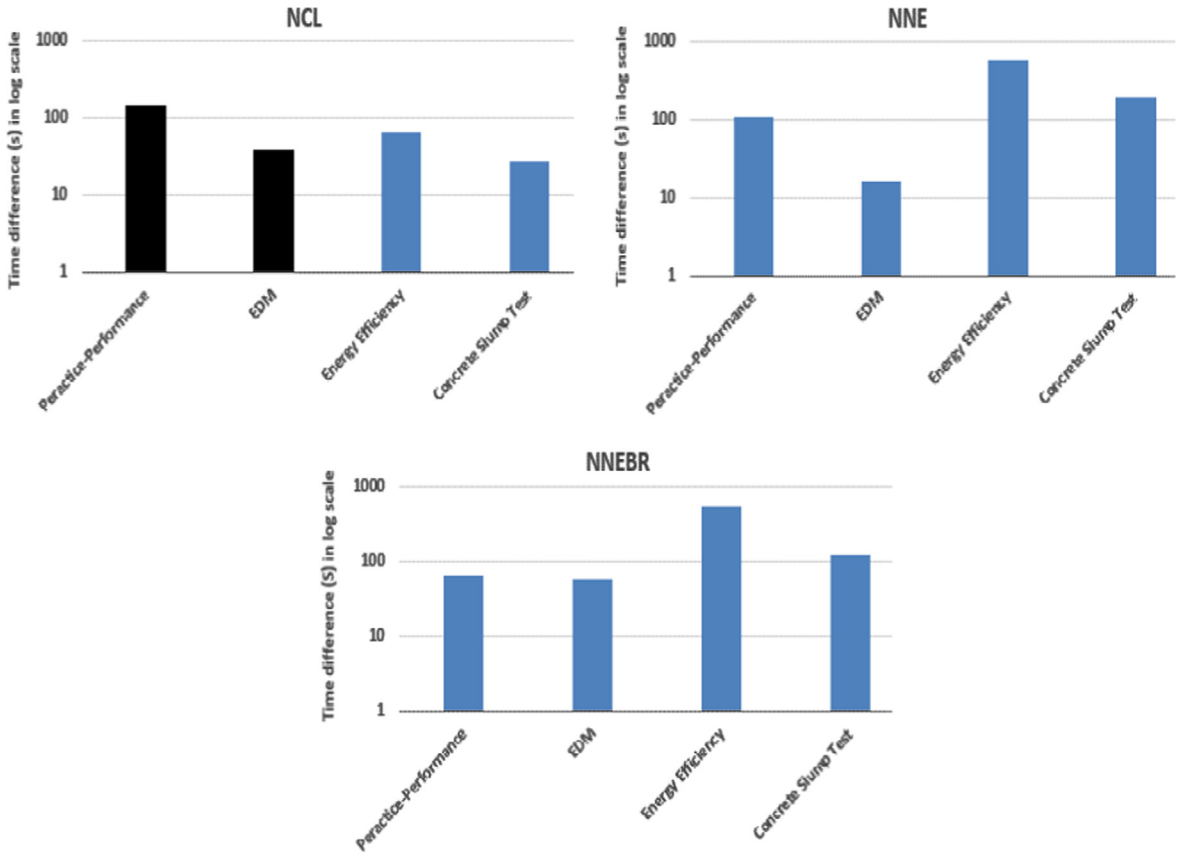

Fig. 8. The differences in execution time (s) on a logarithmic scale between the proposed Boosted-NNE and the NCL, NNE and NNEBR models. Black bars indicate the proposed model is faster, and blue bars signify it is slower. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

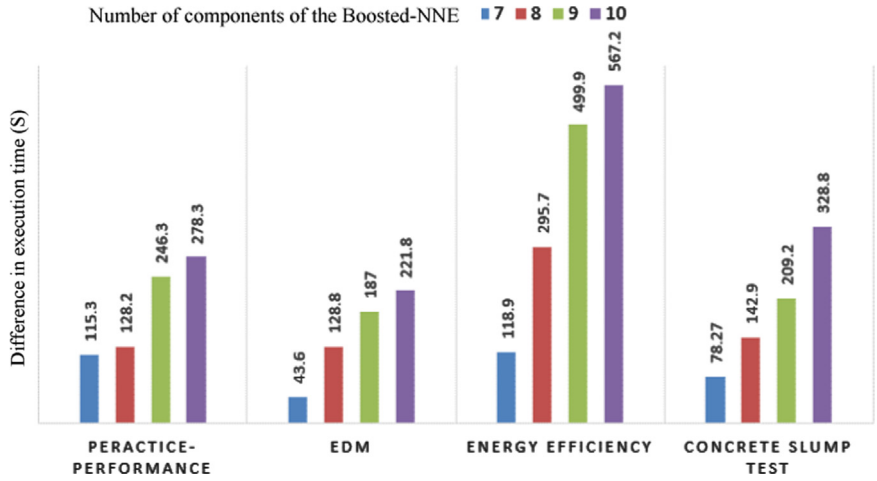

Fig. 9. The differences in training time (s) between the Boosted-NNE model with 7–10 components and 6 components in 4 cases.

Table 8
Sensitivity analysis in modeling practice–performance relationships problem.

Change percentage	Planning and control	Quality	Product development	Technology	Organization	Supply chain related programs
+2%	2.16	2.53	2.06	2.03	1.74	3.04
+5%	10.48	11.23	9.35	9.89	7.00	14.77
+10%	26.98	29.39	25.31	25.54	19.95	35.88
Average change percentage	13.21	14.38	12.24	12.49	9.57	17.89

control model for comparison with the remaining ensembles. Table 7 presents all *P*-values for each comparison that involves the control model.

Table 7 indicates that in terms of the Holm and Hochberg tests (Hochberg, 1988), the Boosted-NNE model is statistically better than the others. Based on the results, the Boosted-NNE model outperforms the other ensemble models and can be considered a promising alternative for high-dimensional multi-target regression problems with a small number of instances.

Speed considerations are challenging to measure, because not only is an algorithm being evaluated but also a specific implementation. However, since execution time is relevant in any machine learning task, the models are compared by measuring their execution time efficiency in a 10-fold cross validation. For an impartial comparison, all experiments were done in the same machine and with the same parameters (the number of components for the ensemble models and hidden neurons was set as 6). The difference in the training times (s) of the Boosted-NNE model, NNE, NNEBR and NCL is shown in Fig. 8. For the majority of problems, the Boosted-NNE model is slower as expected, but even in the worst case the time bounds are not dramatically high. With the practice–performance and EDM datasets, which have more dimensions in the input space than the Energy Efficiency and Concrete Slump Test datasets, the Boosted-NNE model is faster than NCL (which uses all input features in training each component); thus, our proposed model is efficient for high-dimensional input spaces. Moreover, because random subspaces with size 4 were used to train every component of NNE and NNEBR, it was anticipated that the Boosted-NNE would be slower than them. This is not a very serious drawback of our model, as training is usually done off-line. It is worth noting that although the proposed approach is slower than NNE and NNEBR in terms of training time, the differences in testing time are negligible, with the only difference being the projection of a single instance into different subspaces.

Finally, the training time efficiency of the Boosted-NNE model is evaluated in terms of the ensemble size as an important parameter of the model, which is related to an ensemble model's complexity. The differences in training time (s) between the Boosted-NNE model with 7–10 components and 6 components in 4 cases are shown in Fig. 9. This figure indicates that by adding more components in ensemble model, the training time does not increase dramatically.

4.4. The managerial perspective of the proposed model

Diverse surveys over the last two decades show that 60–80% of companies are not capable of meeting their strategic goals, because they cannot implement their planned strategies in practice. Studies related to the historical performance of several companies provide evidence that strategic goals remain only optimistic ideas of top management that are never translated into operational initiatives (Kaplan and Norton., 2005). In fact, there seems to be a deep gap between strategy and operations, which can be thought of as an obstacle to implementing strategy in companies. Correspondingly, the majority of strategy and management experts strongly emphasize developing approaches to create an integrated linkage between strategy and operations. According to Kaplan and Norton (2005(, linking strategy to operations in a company requires performing key activities.

Using the proposed Boosted-NNE model introduced in this paper, it is feasible to create a quantitative relationship between improvement programs and strategic goals of a plant. This model also explicitly addresses the portfolio of improvement programs with the most impact on implementing strategies and meeting strategic goals. In the absence of such model, performing any improvement programs separately and without spotting their interactive impact on strategy would mean sporadic efforts with lack of integration, while a successful plant requires performing these programs and actions using a systematic and integrated approach.

For example, the proposed Boosted-NNE model can be utilized for sensitivity analysis. Sensitivity analysis enables using the variation of one or more input features within a pre-defined range to observe the effect of varying values on a plant's objective performance criteria.

Sensitivity analysis must be used to estimate how sensible (or robust) the results are for the given input features. It also allows understanding the particular trends in the process or fine-tuning values for certain key features. The degree of improvement program implementation in a plant besides the MSE rate percentage for the new dataset can be changed. The percentages of MSE rate for all changes are shown in Table 8. The average percentage change can be indicative of the importance of a particular variable.

The average percentage value change in Table 8 indicates that the model's performance is mostly associated with supply chain-related programs. Thus, supply chain program implementation supports companies with achieving significant performance improvements. As mentioned before, one explanation may refer to the primary operation of the considered plants in this research, which involves supplying various parts for the automotive industry. The main practices underpinning the supply chain programs are supply/distribution strategy in order to change the level of intermediation; rethinking and restructuring the organization and management of supplier portfolio; and increasing the level of coordination of planning decisions and flow of goods to customers.

We can design a heuristic search module to analyze the effects of implementing different bundles of improvement programs using the Boosted-NNE model and to identify the best bundle that achieves fewer gaps between gained performance and managerial objectives.

5. Conclusions and future works

In this paper, we developed a novel Boosted-NNE model for modeling multi-target regression problems. The proposed model has the following characteristics:

(1) MLP components of the Boosted-NNE model are trained in a boosting-like manner by combining ideas from subspace projection methods and a new NCL algorithm.
(2) The proposed model employs a new cluster-based subspace projection (CLSP) method. It is based on the notion of clustering input features to construct an input space for training each component in the Boosted-NNE model with focus on difficult instances in each step of the boosting approach.
(3) The MLP components are pushed away from each other using the proposed SNCL algorithm in combination with the CLSP method to train the NNE model.
(4) The constrained least mean square error (CLMS) algorithm is used to obtain the optimal weights of the components in the Boosted-NNE model's combination module.

The proposed Boosted-NNE model was compared with other ensemble and single models using EDM, Energy Efficiency, and Concrete Slump Test datasets as three popular multi-target regression problems and a real case of the practice–performance relationships problem. The experimental results showed that the prediction accuracy of the proposed Boosted-NNE model was statistically better than other ensemble models and can be considered a promising alternative for multi-target regression problems. By applying the proposed model to the practice–performance problem, it is possible to explicitly address the portfolio of improvement programs that has the most impact on strategy implementation and meeting strategic goals.

Future research efforts should be devoted to (1) using other types of neural networks or fuzzy systems equipped with the negative-correlation learning as components of the proposed model, (2) focus on developing efficient feature clustering algorithms in CLSP, and estimating the latent component for each cluster using non-linear approaches, (3) evolutionary algorithms equipped with negative correlation learning in SNCL could be used to train the Boosted-NNE model components.

Acknowledgments

The authors express their deep gratitude to Prof. Moattar Husseini and Dr. Hajirezaei for their support and many constructive suggestions.

References

Aho, T., Zenko, B., Dzeroski, S., Elomaa, T., 2012. Multi-target regression with rule ensembles. J. Mach. Learn. Res. 13, 1–48.
Alhamdoosh, M., Wang, D., 2014. Fast decorrelated neural network ensembles with random weights. Inf. Sci. 264, 104–117.
Amoako-Gyampah, K., Acquaah, M., 2008. Manufacturing strategy, competitive strategy and firm performance: an empirical study in a developing economy environment. Int. J. Prod. Econ. 111, 575–592.
Ayerdi, B., Graña, M., 2014. Hybrid extreme rotation forest. Neural Netw. 52, 33–42.
Breiman, L., 1996. Bagging predictors. Mach. Learn. 24, 123–140.
Brown, G., Wyatt, J., Harris, R., Yao, X., 2005. Diversity creation methods: a survey and categorisation. J. Inf. Fusion 6, 5–20.
Chao, M., Xin, S.Z., Min, L.S., 2014. Neural network ensembles based on copula methods and distributed multiobjective central force optimization algorithm. Eng. Appl. Artif. Intell. 32, 203–212.
Chen, H., Yao, X., 2009. Regularized negative correlation learning for neural network ensembles. IEEE Trans. Neural Netw. 20, 1962–1979.
Enki, D.G., Trendafilov, N.T., Jolliffe, I.T., 2013. A clustering approach to interpretable principal components. J. Appl. Stat. 40, 583–599.
Erdal, H.I., Karakurt, O., Namli, E., 2013. High performance concrete compressive strength forecasting using ensemble models based on discrete wavelet transform. Eng. Appl. Artif. Intell. 26, 1246–1254.
Freund, Y., Schapire, R.E., 1996. Experiments with a new boosting algorithm. In: Proceedings of the Thirteenth International Conference on Machine Learning – ICML, Morgan Kaufman, pp. 148–156.
Garc´ıa-Pedrajas, N.A., Garc´ıa-Osorio, C.E., Fyfe, C., 2007. Nonlinear boosting projections for ensemble construction. J. Mach. Learn. Res. 8, 1–33.
Garc´ıa-Pedrajas, N.A., Ortiz-Boyer, D., 2008. Boosting random subspace method. Neural Netw. 21, 1344–1362.
García-Pedrajas, N., Maudes-Raedo, J., García-Osorio, C., Rodríguez-Díez, J.J., 2012. Supervised subspace projections for constructing ensembles of classifiers. Inf. Sci. 193, 1–21.
Geman, S., Bienenstock, E., Doursat, R., 1992. Neural networks and the bias/variance dilemma. Neural Comput. 4, 1–58.
Hadavandi, E., Shahrabi, J., Hayashi, Y., 2015. SPMoE: a novel subspace-projected mixture of experts model for multi-target regression problems. Soft Comput., 1–19.
Hajirezaie, M., Moattar Husseini, S.M., Abdollahzadeh Barfourosh, A., Karimi, B., 2010. Modeling and evaluating the strategic effects of improvement programs on the manufacturing performance using neural networks. Afr. J. Bus. Manag. 4, 414–424.
Hansen, L.K., Salamon, P., 1990. Neural network ensembles. IEEE Trans. Pattern Anal. Mach. Intell. 12, 993–1001.
Ho, T.K., 1998. The random subspace method for constructing decision forests. IEEE Trans. Pattern Anal. Mach. Intell. 20, 832–844.
Hochberg, Y., 1988. A sharper Bonferroni procedure for multiple tests of significance. Biometrika 75, 800–803.
Kaplan, R.S., Norton, D.P., 2005. Creating the Office of Strategy Management. Division of Research, Harvard Business School.
Karalič, A., Bratko, I., 1997. First order regression. Mach. Learn. 26, 147–176.
Kocev, D., Vens, C., Struyf, J., Džeroski, S., 2013. Tree ensembles for predicting structured outputs. Pattern Recognit. 46, 817–833.
Kotsiantis, S., 2011. Combining bagging, boosting, rotation forest and random subspace methods. Artif. Intell. Rev. 35, 1–18.
Lee, H., Kim, E., Pedrycz, W., 2012. A new selective neural network ensemble with negative correlation. Appl. Intell. 37, 488–498.
Li, S., Ragu-Nathan, B., Ragu-Nathan, T.S., Rao, S.S., 2006. The impact of supplychain management practices on competitive advantage and organizational performance. Omega 34, 107–124.
Liu, Y., Yao, X., 1999. Ensemble learning via negative correlation. Neural Netw. 12, 1399–1404.
Mas'ud, A.A., Stewart, B.G., McMeekin, S.G., 2014. Application of an ensemble neural network for classifying partial discharge patterns. Electr. Power Syst. Res. 110, 154–162.
Masnadi-Shirazi, H., Vasconcelos, N., 2007. Asymmetric boosting. In: Proceedings of the 24th International Conference on Machine Learning. ACM, pp. 609–619.
Masoudnia, S., Ebrahimpour, R., Arani, S.A.A.A., 2012. Combining features of negative correlation learning with mixture of experts in proposed ensemble methods. Appl. Soft Comput. 12, 3539–3551.
McKay, R., Abbass, H.A., 2001. Anti-correlation: a diversity promoting mechanism in ensemble learning. Aust. J. Intell. Inf. Process. Syst. 7, 139–149.
Nabavi-Kerizi, S.H., Abadi, M., Kabir, E., 2010. A PSO-based weighting method for linear combination of neural networks. Comput. Electr. Eng. 36, 886–894.
Pardo, C., Diez-Pastor, J.F., García-Osorio, C., Rodríguez, J.J., 2013. Rotation forests for regression. Appl. Math. Comput. 219, 9914–9924.

Rodriguez, J.J., Kuncheva, L.I., Alonso, C.J., 2006. Rotation forest: a new classifier ensemble method. IEEE Trans. Pattern Anal. Mach. Intell. 28, 1619–1630.
Rumelhart, D.E., McClelland, J.L., 1986. Parallel Distributed Processing: Explorations in the Micro Structure of Cognition, Foundations. MIT Press, Cambridge, MA.
Rusjan, B., 2005. Model for manufacturing strategic decision making. Int. J. Oper. Prod. Manag. 25, 740–761.
Shahrabi, J., Hadavandi, E., Asadi, S., 2013. Developing a hybrid intelligent model for forecasting problems: case study of tourism demand time series. Knowl.-Based Syst. 43, 112–122.
Sheskin, D.J., 2003. Handbook of Parametric and Non-parametric Statistical Procedures. CRC Press.
Solomatine, D.P., Shrestha, D.L., 2004. AdaBoost. RT: a boosting algorithm for regression problems. In: Proceedings of the 2004 IEEE International Joint Conference on Neural Networks, pp. 1163–1168.
Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., Vlahavas, I., 2014. Multi-Label Classification Methods for Multi-Target Regression. ⟨arXiv:1211.6581⟩.
Tang, K., Lin, M., Minku, L., Yao, X., 2009. Selective negative correlation learning approach to incremental learning. Neurocomputing 72, 2796–2805.
Tian, J., Li, M., Chen, F., Kou, J., 2012. Coevolutionary learning of neural network ensemble for complex classification tasks. Pattern Recognit. 45, 1373–1385.
Tsanas, A., Xifara, A., 2012. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. Energy Build. 49, 560–567.
Ueda, N., 2000. Optimal linear combination of neural networks for improving classification performance. IEEE Trans. Pattern Anal. Mach. Intell. 22, 207–215.
Ueda, N., Nakano, R., 1996. Generalization error of ensemble estimators. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 90–95.
Vigneau, E., Qannari, E., 2003. Clustering of variables around latent components. Simul. Comput. 32, 1131–1150.
Xiong, L., Mao, S., Jiao, L., 2015. Selective ensemble based on transformation of classifiers used SPCA. Int. J. Pattern Recognit. Artif. Intell. 29.
Yang, J., Zeng, X., Zhong, S., Wu, S., 2013. Effective neural network ensemble approach for improving generalization performance. IEEE Trans. Neural Netw. Learn. Syst. 24, 878–887.
Yeh, I., 2007. Modeling slump flow of concrete using second-order regressions and artificial neural networks. Cem. Concr. Compos. 29, 474–480.
Zhai, J.-H., Xu, H.-Y., Wang, X.-Z., 2012. Dynamic ensemble extreme learning machine based on sample entropy. Soft Comput. 16, 1493–1502.