Machine Learning (2023) 112:3881–3916
https://doi.org/10.1007/s10994-023-06364-3

A hybrid ensemble method with negative correlation learning for regression

Yun Bai[1,2] · Ganglin Tian[3] · Yanfei Kang[1] · Suling Jia[1]

Received: 27 September 2021 / Revised: 22 May 2023 / Accepted: 5 June 2023 /
Published online: 23 August 2023

Abstract

Hybrid ensemble, an essential branch of ensembles, has flourished in the regression field, with studies confirming diversity's importance. However, previous ensembles consider diversity in the sub-model training stage, with limited improvement compared to single models. In contrast, this study automatically selects and weights sub-models from a heterogeneous model pool. It solves an optimization problem using an interior-point filtering linear-search algorithm. The objective function innovatively incorporates negative correlation learning as a penalty term, with which a diverse model subset can be selected. The best sub-models from each model class are selected to build the NCL ensemble, which performance is better than the simple average and other state-of-the-art weighting methods. It is also possible to improve the NCL ensemble with a regularization term in the objective function. In practice, it is difficult to conclude the optimal sub-model for a dataset prior due to the model uncertainty. Regardless, our method would achieve comparable accuracy as the potential optimal sub-models. In conclusion, the value of this study lies in its ease of use and effectiveness, allowing the hybrid ensemble to embrace diversity and accuracy.

Keywords Hybrid ensemble · Diversity · Negative correlation learning · Optimization

Editor: Zhi-Hua Zhou.

✉ Yanfei Kang
yanfeikang@buaa.edu.cn

Yun Bai
baiyun12138@buaa.edu.cn

Ganglin Tian
ganglin.tian@imt-atlantique.net

Suling Jia
jiasuling@buaa.edu.cn

[1] School of Economics and Management, Beihang University, Beijing 100191, China

[2] The Centre for Processes, Renewable Energies and Energy Systems (PERSEE), MINES Paris - PSL University, Sophia Antipolis, France

[3] Faculty of Microwave, Observation, and Perspection of Environment, IMT Atlantique, Plouzané 29280, France

1 Introduction

Ensemble learning has been proven to be theoretically and empirically superior to single models by state-of-the-art literature as a method of combining pre-trained models in a certain way to obtain final predictions (Brown et al., 2005; Chandra and Yao, 2006; Mendes-Moreira et al., 2012). Typically, ensemble models sample the input space of data and features, such as cross-validation or down-sampling of data (LeBlanc and Tibshirani, 1996). Meanwhile, features can be selected by calculating feature importance (Mendes-Moreira et al., 2012). Then, ensembles are followed by combining multiple but homogeneous weak learners to form a strong learner to achieve higher accuracy. The famous examples of ensemble models are bagging (Breiman, 1996), boosting (Freund et al., 1996), and stacking (Wolpert, 1992). In recent years, solutions based on ensemble models often achieve good results in *Kaggle competitions* (Taieb and Hyndman, 2014; Hoch, 2015; Bojer and Meldgaard, 2020).

In some pioneering studies, researchers attempted to train completely heterogeneous models for the same input space and then averaged or weighted the predictions of these models. This approach considered that heterogeneous models were more likely to increase diversity during training and produce more robust results compared to homogeneous models (Zhao et al., 2010; Mendes-Moreira et al., 2012). Training with heterogeneous models also refers to the hybrid ensemble. For load prediction, Salgado chose several support vector machines and neural networks, ranked and filtered the candidates, and finally weighted the predictions of the selected models. Their hybrid ensemble model improved performance by 25% over the best single predictor (Salgado et al., 2006). Ala'raj took five classifiers and combined their predictions. The experimental results demonstrated the ability of the proposed method to improve the accuracy of credit scoring prediction (Ala'raj and Abbod, 2016). Qi constructed a hybrid ensemble model for predicting slope stability in geology, which included six sub-models, such as support vector machines and artificial neural networks. A genetic algorithm was introduced to calculate the classification weights for each model. This hybrid ensemble outperformed any single model, even though the single model already had its optimal parameters (Qi and Tang, 2018). Some researchers constructed ensembles containing both homogeneous and heterogeneous models. For example, Merz chose six multivariate adaptive regression splines and six back-propagation networks to build a model pool, ranked the sub-models by principal components with the variance from the learning process to highlight the contributions of different sub-models (Merz and Pazzani, 1999).

Scholars have identified model diversity as a critical factor to hybrid ensemble success (Brown, 2004; Webb and Zheng, 2004; Chandra and Yao, 2006). In recent years, researchers have put effort into ensemble diversity and generalization. The authors developed a pruning method for classification ensembles utilizing the tradeoff between accuracy and diversity (Bian and Chen, 2021). Several methods to increase the diversity of sub-models within an ensemble are also proposed. For earlier schemes, practitioners trained models with cross-validation or chose different parameter combinations for homogeneous models, followed by majority voting or weighted averaging of the model predictions. Cross-validation yet provided limited improvement for model accuracy, and Stone proved as early as 1974 that estimators generated by cross-validation behaved similarly (Stone, 1974). Hansen and Salomon proposed using neural networks to construct ensembles in the 1990 s. They used neural networks to fit different parts of the training data, which were then majority voted as the result of ensemble (Hansen and

Salamon, 1990). Both Ting and Cano obtained a diversity of sub-models by using different subsets of features (Ting et al., 2011; Cano and Krawczyk, 2020). Ting emphasized that unstable learners could generate sufficient diversity of global models since they were more sensitive to data changes (Ting et al., 2011). Cano suggested dynamically monitoring the model pool to eliminate the oldest and weakest sub-models in time for the streaming data scenario (Cano and Krawczyk, 2020). Sirovetnukul pointed out that a hybrid ensemble could learn negative knowledge from less well-performed models that were easily ignored and removed in previous studies. Such knowledge could help the models converge to better solutions while producing diverse results (Sirovetnukul et al., 2011). Brown considered the negative knowledge across sub-models and provided quantitative methods for the diversity of hybrid ensembles (Brown et al., 2005).

Some empirical evidence demonstrated the ability of Negative Correlation learning (NCL) to increase model diversity and improve ensemble models (Liu and Yao, 1999; Liu et al., 2000; Chandra and Yao, 2006; Sirovetnukul et al., 2011; Alhamdoosh and Wang, 2014; Peng et al., 2020). NCL introduces a correlation penalty term in the objective function of each sub-model to measure the deviation from the current ensemble. All sub-models can be trained simultaneously and interactively on the same training set, and the final experimental results will achieve a bias-variance-covariance balance, as theoretically deduced. Current applications of NCL are focused primarily on the training process of ensemble neural networks to diversify each sub-model (Liu and Yao, 1999; Liu et al., 2000; Tang et al., 2009; Alhamdoosh and Wang, 2014; Hadavandi et al., 2015; Peng et al., 2020). Although the ensemble neural network trains the sub-models with diversity under NCL, they are still structurally homogeneous models, differing only in specific parameters. To our knowledge, only some studies apply NCL to hybrid ensembles. Next, we will discuss the feasibility of using NCL to improve hybrid ensembles.

Generally, ensembles contain two stages: sub-model training and combination (Merz, 1999). Previous ensembles used NCL as a penalty term to train diverse sub-models in the first stage, followed by some basic methods, such as majority voting or simple averaging, to combine the predictions, ignoring the role of diversity in the second stage. In contrast, the hybrid ensemble trains multiple heterogeneous models based on the consensus that heterogeneous models will produce diverse predictions in the first stage (Zhao et al., 2010; Mendes-Moreira et al., 2012). In the second stage, if we apply NCL to the objective function to optimize the weights of each sub-model, it is possible to select a diverse set of sub-models to obtain the final results. We present the methods for obtaining diversity at different stages of the ensemble models in Table 1 for comparison.

To improve hybrid ensembles with NCL, we design a generic scheme in this study for regression problems. Eleven well-established regression prediction methods, including ensemble and generalized linear regression models, are fed to the model pool. Each sub-model is trained and generates a set of predictions. Cross-validation and grid search are applied to the training process to obtain the predictor with the optimal parameters. Subsequently, we view the process of the second stage of hybrid ensembles, sub-model combination, as an optimization problem. This problem can be solved using the interior-point filter line-search algorithm (Wächter and Biegler, 2006), which is a solver in the Gekko optimizer developed by Beal et al. (2018). We add NCL as a penalty term to the objective function of the optimization problem. We designed several experiments to evaluate the proposed method from multiple dimensions. The hybrid ensemble for regression based on NCL achieves excellent results, demonstrating its great potential.

The main contributions of this study are three-fold:

Table 1 Methods for obtaining diversity at different stages of the ensemble model

Ensemble models	Stage 1: sub-models training	Stage 2: sub-models combination
Ensemble neural networks	Homogeneous sub-models are trained simultaneously and interactively to increase the diversity of sub-models during the training process	Majority voting or simple averaging is used to combine the predictions, not considering the diversity within the ensemble
Hybrid ensembles	Heterogeneous models are trained separately to ensure diversity	A diverse subset of predictions is selected and weighted by NCL

1. Initially, this study attempts to migrate the application scenario of NCL from the traditional sub-model training stage to the sub-model combination stage, with good results in a hybrid ensemble consisting of heterogeneous sub-models.
2. The model selection and combination process is treated as an optimization problem. This problem leads to a diverse set of sub-models in the model pool, given by a weight vector.
3. Ultimately, the approach in this study again verifies that diversity is the key to the success of ensemble models, and it is an innovation to ensure model diversity in both stages of the hybrid ensemble.

The rest of this paper is organized as follows. Section 2 introduces the theories and methods involved in the proposed framework. Section 3 presents a hybrid ensemble based on NCL, accounting for model diversity. In Sect. 4, we systematically investigate the application of the proposed method on twenty publicly available datasets and analyze the contribution of NCL to performance improvement. Section 5 reviews the background of our proposed method, illustrates the method's ability to remedy some of the shortcomings of current hybrid ensemble studies, and synthesizes the experimental performance and scope for improvement of our method. Finally, Sect. 6 concludes the paper.

2 Related works

This section first introduces ambiguity and bias-variance-covariance decompositions, which are the theoretical basis for Negative Correlation Learning (NCL) to increase the diversity of hybrid ensembles (Brown et al., 2005). The general form of the NCL is presented in the second part. The third part shows the computational principles and applications of the interior-point filter line-search algorithm.

2.1 Two types of decomposition

In the context of multiple regression, there is a dataset containing n samples with $\{(x_1, y_1), \dots, (x_n, y_n)\}$. The objective of the problem is to find a function f that maps $\mathbb{R}^n$ to $\mathbb{R}^1$ to gain predictive capability for future data. In machine learning, f is a model or an estimator.

$$f(x_i) = y_i, \qquad f : \mathbb{R}^n \rightarrow \mathbb{R}^1, x_i \in \mathbb{R}^n, y_i \in \mathbb{R}^1. \tag{1}$$

2.1.1 Ambiguity decomposition

In a general scenario, m sub-models can form a hybrid ensemble f_h with a weighted average. f_h is a convex combination of all components:

$$f_h = \sum_{j=1}^{m} \omega_j f_j, \tag{2}$$

where $\sum_{j=1}^{m} \omega_j = 1$, and f_j is the predictions of j_{th} sub-model. According to Brown, the Mean Square Error (MSE) ζ_h of f_h can be expressed as the difference between the following two terms (Brown et al., 2005):

$$\zeta_h = \sum_{j=1}^{m} \omega_j \zeta_j - \frac{1}{n} \sum_{j=1}^{m} \sum_{i=1}^{n} \omega_j (f_h(x_i) - f_j(x_i))^2, \tag{3}$$

where $\zeta_j = \frac{1}{n} \sum_{i=1}^{n} \left(f_j(x_i) - y_i\right)^2$. The first term of Eq. (3) is the weighted average of the MSE of each sub-model; the second is the ambiguity term. Equation (3) indicates that ζ_h is less than the weighted average ζ_j of all sub-models, given that the sub-models are not identical and the second ambiguity term is positive. This fact reveals that the more significant the difference between each sub-model and the current hybrid ensemble, the larger the ambiguity term and the smaller the MSE of the hybrid ensemble. Notably, without an established criterion to judge the best model in advance, it is efficient to use the hybrid ensemble directly, even if some member has the lowest error.

2.1.2 Bias-variance-covariance decomposition

The MSE of the sub-models and the hybrid ensemble are employed in the ambiguity decomposition to measure diversity; the higher the second term in Eq. (3), the more diverse the ensemble. However, as the sub-models increase in volume, they are more likely to deviate from the actual value, although they would get more diverse. This situation leads to an increase in the first term of ζ_h when it is not so beneficial to consider increasing the diversity of the hybrid ensemble. Thus, balancing the diversity and accuracy of the sub-models and ensemble is of interest. The bias-variance-covariance decomposition is a well-defined trade-off (Brown et al., 2005).

For simplicity, given the simple average form of the hybrid ensemble $f_h = \frac{1}{m} \sum_{j=1}^{m} f_j$ and the unbiased estimation of the ground truth $\hat{y} = E(y)$, the bias-variance-covariance decomposition is written as the following equation:

$$E\left((f_h - \hat{y})^2\right) = B^2 + \frac{1}{m} V + \left(1 - \frac{1}{m}\right) C, \tag{4}$$

where B, V, and C are the averaged bias, variance, and covariance of each sub-model in the hybrid ensemble. The equations for the three terms are as follows:

$$B = \frac{1}{m} \sum_{j=1}^{m} \left(E(f_j) - \hat{y}\right), \tag{5}$$

$$V = \frac{1}{m} \sum_{j=1}^{m} E\left((f_j - E(f_j))^2\right), \tag{6}$$

$$C = \frac{1}{m(m-1)} \sum_{j=1}^{k} \sum_{k \neq j} E\left[(f_j - E(f_j))(f_k - E(f_k))\right]. \tag{7}$$

Unlike ambiguity decomposition, the bias-variance-covariance decomposition can reduce the error of the hybrid ensemble by decreasing the covariance without increasing the bias and variance. Additionally, the covariance term can be negative, implying that negative correlations between sub-models can contribute to the prediction of the hybrid ensemble.

2.2 Negative correlation learning

Liu has proposed to achieve diversity within an ensemble by NCL (Liu and Yao, 1999). They designed NCL as a training method for neural network ensembles. It adds a penalty term to the objective function of each network and trains all networks simultaneously and interactively before combining them. The purpose of this training pattern is not to obtain multiple accurate and independent neural networks but to capture the correlations and derive sub-networks with negative correlations using penalty terms, which in turn form a robust combination. Brown also used NCL by adding a heuristic penalty term to the mean squared error as an objective function (Brown et al., 2005). They systematically control the bias-variance-covariance trade-off by optimizing this objective function. In addition, they derived a systematic upper bound on the strength of negative correlation, which tended to stabilize as the number of models within the ensemble increased. As mentioned in Table 1 before, there are sub-model training and combination stages in generating an ensemble model. The application of NCL in neural network ensembles belongs to the first stage and the objective function for training the sub-model in a typical ensemble is given below:

$$F_j = \zeta_j + \lambda p_i(n), \tag{8}$$

$$\begin{aligned} p_i(n) &= \frac{1}{n}\sum_{i=1}^{n}(f_j(x_i) - f_h(x_i))\sum_{k\neq j}(f_k(x_i) - f_h(x_i)) \\ &= -\frac{1}{n}\sum_{i=1}^{n}(f_j(x_i) - f_h(x_i))^2. \end{aligned} \tag{9}$$

It is still given that m networks in the ensemble and n samples in the dataset. For the j_{th} network, its objective function F_j during training processing is MSE with an NCL penalty term. In Eqs. (8) and (9), λ is the negative correlation strength. When λ equals 0, F_j is equivalent to MSE ζ_j, and the higher the λ, the stronger the negative correlation strength of the objective function. Previous approaches to increasing model diversity, such as changing the model structure, were mainly implicit. Contrastingly, the NCL controls model diversity explicitly by adding a penalty term to the objective function using only the parameter λ. The effect of NCL is to pull the predictions of the sub-models away from the ensemble while drawing the ensemble closer to the actual values (Reeve and Brown, 2018).

2.3 Interior-point filter line-search algorithm

The interior-point filter linear-search algorithm has mature applications in many fields as a general-purpose method for solving optimization and programming (Simmons et al., 2019; Carpio et al., 2021; Pulsipher et al., 2022). This optimization algorithm has been well integrated as an Interior Point OPTimizer (IPOPT) solver in Gekko for friendly use, which is designed by Beal et al. (2018). As an algebraic modeling language, it excels in solving dynamic optimization problems. Additionally, Gekko is a Python library that integrates model building, analysis tools, and optimization visualization. Following, we will briefly introduce IPOPT (Wächter and Biegler, 2006).

For convenience, researchers are used to writing the objective function and constraints of the optimization problem by adding equation constraints and slack variables in the standard form, as in Eq. (10):

$$\begin{aligned} &arg \min_{x \in \mathbb{R}^n} F(x) \\ &s.t.\ c(x) = 0, \\ &x_i \geq 0. \end{aligned} \tag{10}$$

To solve an optimization problem using the interior point method, one adds an auxiliary barrier to Eq. (10) and, correspondingly, removes the inequality constraint, as in Eq. (11):

$$\begin{aligned} &arg \min_{x \in \mathbb{R}^n} \ \phi_\mu(x) = F(x) - \mu \sum_{i=1}^{m} \ln(x_i) \\ &s.t.\ c(x) = 0. \end{aligned} \tag{11}$$

As introduced by Wächter, μ is a logarithmic barrier term, and $\mu > 0$ (Wächter and Biegler, 2006). As $\mu \rightarrow 0$, the optimization problem (11) is more likely to converge to an optimal solution. The solution of Eq. (11) starts with a relatively small μ, such as 0.1, and then iterates using the Newton method combined with a linear search. IPOPT then determines whether the current feasible solution reduces $\phi_\mu(x)$ compared to the previous feasible solution. In the absence of a feasible solution, IPOPT transforms the problem (11) into a feasibility restoration phase by finding a feasible solution that minimizes the norm of the constraint violation $\|c(x)\|_1$, temporarily ignoring the objective function, and thus solving it flexibly. The above steps are repeated, with μ being reduced each time, until the solution of Eq. (11), or the solution satisfying the first-order optimality condition, is found. All the procedures would be done by Gekko automatically.

3 Hybrid ensemble for regression with negative correlation learning

As one of the most fundamental mathematical problems, regression has many well-established models designed from different perspectives. A hybrid ensemble is a method to solve regression by the weighted average of the predictions of multiple members. In this study, by introducing NCL in the hybrid ensemble, the sub-models with diversity will be selected, combined, and weighted to improve the prediction accuracy. Specifically, this section examines these aspects: model pool construction, sub-model training stage, sub-model combination stage, evaluations, and the proposed hybrid ensemble framework.

3.1 Model pool construction

Many ensemble models adopt cross-validation to train homogeneous models and perform majority voting to select models that work well. In contrast, this study draws on the conclusion of Mendes-Moreira that heterogeneous models control diversity and perform better than homogeneous candidates in the model training stage (Mendes-Moreira et al., 2012). When constructing the model pool, we chose the models from different methods. Eleven regression models are selected in this study, including Simple Linear Regression (SLR) (Zou et al., 2003), Ridge Regression (RR) (Hoerl and Kennard, 1970), Bayesian Regression (BR) (Box and Tiao, 2011), Stochastic Gradient Descent

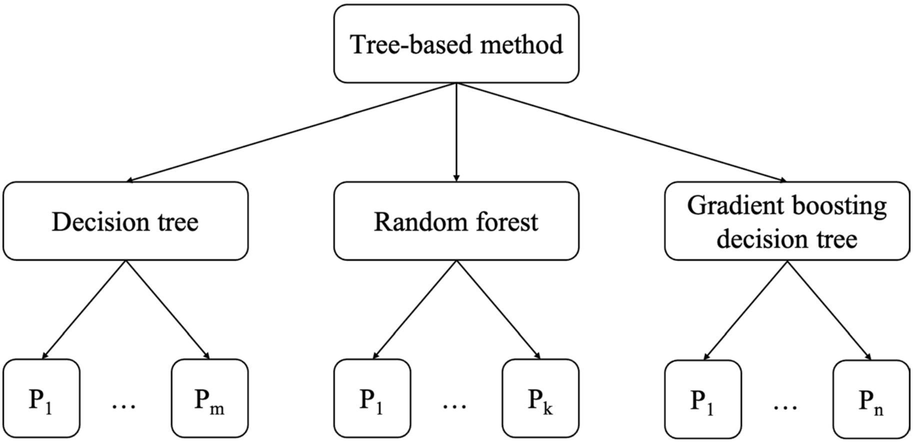

Fig. 1 The relationship between the method, models, and sub-models. The top level is a tree-based *method*, the middle level is different *models*, and the bottom level are *sub-models* written as P_i with different parameter sets. The sub-models serve as the members of the hybrid ensembles in this paper

Regression (SGDR) (Jain et al., 2018), Polynomial Regression (PR) (Stigler, 1974) from *Linear methods*; Decision Tree Regression (DTR) (Wu et al., 2008), Random Forest Regression (RFR) (Ho, 1995), and Gradient Boosting Decision Tree (GBDT) (Friedman, 2001) from *Tree-based methods*; Adaptive Boosting Regression (ABR) (Solomatine and Shrestha, 2004), Support Vector Regression (SVR) (Drucker et al., 1997), and Multilayer Perceptron Regression (MPR) (Rosenblatt, 1961). *Methods*, *models*, and *sub-models* will be mentioned several times in this paper, and we have drawn an example in Fig. 1 to distinguish these three terms.

3.2 Sub-model training stage

In practice, grid search is to find the best parameter set of a model to improve the prediction (Chicco, 2017). Cross-validation is the basis for judging whether a parameter set is good or not (Geisser, 1975). In a typical model fitting task, there will be situations where the training set predicts better than the test set, also known as over-fitting, which can be solved by cross-validation. In this paper, a 5-fold cross-validation is used, whereby the training data is divided into five equal parts, and a model with a particular parameter set is fitted five times. The model takes one copy of the data from the training set as the validation set and the remaining four copies as a new training set. After five fits, the prediction scores on each validation set are averaged as the final score of the current model. Once the grid search has traversed all possible parameter combinations, the highest-scoring parameter set is taken as optimal.

Figure 2 illustrates the process of grid search and cross-validation. The value range for each parameter is first set manually to form a discrete parameter space. The grid search then traverses the space to obtain all parameter sets, calculates the average prediction error on each validation data, and selects the parameter set with the lowest error. Once the grid search and cross-validation are finished, we expect to obtain the best parameter set for a model.

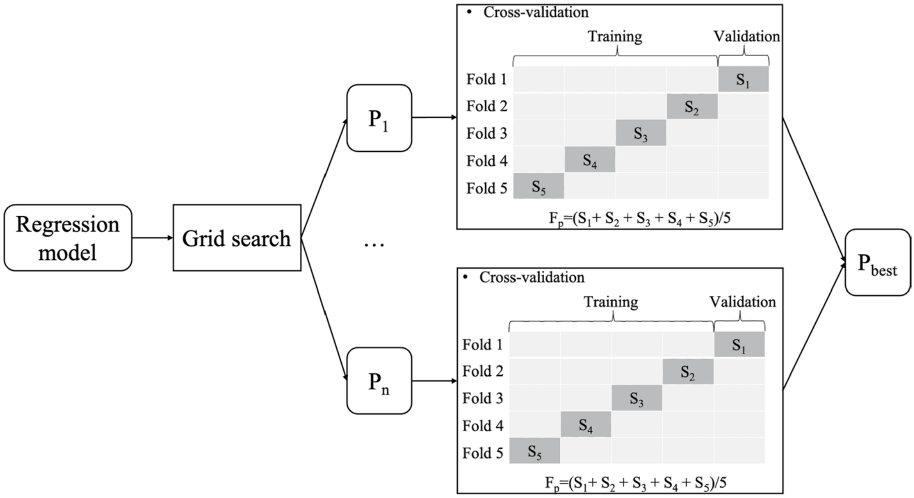

Fig. 2 The process of grid search and cross-validation

3.3 Sub-model combination stage

3.3.1 Objective function for hybrid ensemble

This subsection explains the difference between the proposed NCL-based hybrid ensemble and the neural network ensemble in Liu and Yao (1999) and claims the contributions in detail. We have introduced how NCL is used in neural network ensemble in Sect. 2.2. If combining the Eqs. (8) and (9), we get the error function for each network:

$$F_j = \zeta_j - \frac{\lambda}{n} \sum_{i=1}^{n} \left(f_j(x_i) - f_h(x_i)\right)^2. \tag{12}$$

All network members optimize the error function (12) during training and achieve interaction between members by the penalty term in the function. The ensemble of networks thus is trained on the 'sub-model training' stage in Table 1.

Unlike the neural network ensemble containing homogeneous members, we applied a heterogeneous ensemble to generate the estimators with specialty and accuracy in the different regions of solution space (Brown et al., 2005). Training and interacting sub-models with different architectures in parallel are challenging, so we train each model separately, incorporating diversity in the 'sub-model combination' stage. We consider designing an optimization problem to implement a hybrid ensemble in which candidate sub-models are automatically selected and assigned weights. We still wrapped the error function of each sub-model as a penalty term to encourage the emergence of diversity as Eq. (12). Then the objective function of the hybrid ensemble is obtained with the weighted average of all the error functions and is written as follows:

$$
\begin{aligned}
&arg\min_{\omega}\Phi(\omega) = \sum_{j=1}^{m}\omega_j\left\{\zeta_j - \frac{\lambda}{n}\sum_{i=1}^{n}\left(f_j(x_i) - f_h(x_i)\right)^2\right\},\\
&s.t.\ \sum_{j=1}^{m}\omega_j = 1,\\
&\quad 0 \leq \omega \leq 1.
\end{aligned}
\tag{13}
$$

At this point, we claim the contribution of optimizing Formula (13) to the performance of the final hybrid ensemble. Formula (13) and Formula (3), also the ambiguity decomposition, are similar in form, with the only difference being the λ in the second term in Formula (13). Further, the Formula (3) describes the performance of the hybrid ensemble. The issue then naturally arises on why hybrid ensemble optimizes Formula (13) instead of Formula (3). There are three explanations: (1) it would be overfitted if one only minimising the Formula (3) with focus on the training data; (2) the ensemble diversity cannot be guaranteed if only the second term of Formula (3) is optimized without causing a change in the first term, as both terms contain variance (Brown, 2004). (3) Formula (3) can be decomposed into the three terms in Formula (4) considering the sample distribution. The NCL penalty term in Formula (13) could control the covariance through λ without causing bias and variance terms to change and obtain the trade-off between accuracy and diversity.

The differences between the hybrid ensemble and the neural network ensemble can be stated as follows: (1) when training the neural network ensemble, each network has the identical error function as Formula (12) and interacts with other networks. The network optimization and weight updating are simultaneous. (2) The proposed ensemble is post-hoc, consisting of heterogeneous sub-models that are pre-tested for the performance on the validation set before being combined. This operation avoids the homogeneity of the neural network ensemble but preserves the interaction and enhances the generalization of the hybrid ensemble. (3) Formula (13) takes a weighted average of the error functions of all the sub-models instead of optimizing them separately as in the neural network ensemble. Formula (13) focuses more on optimizing weights given the known MSE of sub-models on the validation set. If a sub-model has a higher MSE, Formula (13) puts less emphasis, or weight, on the sub-model. The weight can be zero if $\lambda = 0$. However, if this sub-model has a higher difference from the current hybrid ensemble at the same time, it contributes to the diversity of the ensemble and attracts some attention from the Formula (13). The penalty term achieves the trade-off between diversity and accuracy with this mechanism.

3.3.2 Automatic search algorithm for negative correlation penalty

The λ in Formula (13) controls the strength of the negative correlation penalty. We designed an algorithm to select a suitable λ from a list as Algorithm (1). The basic idea of searching λ is to traverse from 0 to 1 given the step s. We use Gekko to solve Formula (13) to obtain weight vector ω regarding the different sub-models. Algorithm (1) then calculates the error of the generated hybrid ensemble on the validation set under ω. The error here is a simple average of *RMSE*, *MAE*, and *MAPE*, considering that these three metrics are the evaluation criteria in this paper. We attempt to treat these three metrics fairly without preference. After the algorithm targets the optimal λ^* with minimum error, the step s is reduced, and a more refined search is started locally on that λ^*. In this paper, we retain the lambda with three digits, i.e., the search stops when $s < 0.001$.

Algorithm 1 An automatic search algorithm for optimal λ^*

```
Input:
    $\mathcal{F}_v \leftarrow (f_1, f_2, ..., f_m)$    ▷ Predictions on validation set
Initialize:
    $\lambda^* \leftarrow 0.1$    ▷ Optimal $\lambda$
    $s \leftarrow 0.1$    ▷ Search step
    $\mathcal{E}^* \leftarrow \infty$    ▷ Errors on the validation set
while $s >= 0.001$ do
    $\mathcal{L} \leftarrow [\lambda^* + i * s, \lambda^* - i * s] \backslash \lambda^*$, $i = 0, 1, .., 10$
    Remove the $\lambda$ that $\lambda > 1$ or $\lambda < 0$ in $\mathcal{L}$
    for $\lambda_i$ in $\mathcal{L}$ do
        Call Gekko to solve $\Phi(w)$, and obtain the weight $\omega$
        Get hybrid ensemble on the validation set: $f_h = \mathcal{F}_t \omega^T$
        Compute errors : $\mathcal{E} = (\mathcal{E}_{RMSE} + \mathcal{E}_{MAE} + \mathcal{E}_{MAPE})/3$
        if $\mathcal{E} < \mathcal{E}^*$ then
            $\mathcal{E}^* \leftarrow \mathcal{E}$
            $\lambda^* \leftarrow \lambda_i$
        end if
    end for
    $s = s/10$
end while
Output:
    Optimal $\lambda^*$ and weights for each sub-model $\omega$
```

3.4 Model evaluation metrics

Root Mean Squared Error (*RMSE*), Mean Absolute Error (*MAE*), and Mean Absolute Percentage Error (*MAPE*) are three metrics to evaluate the accuracy of regression models. The equations of them are as follows with $\hat{y}_i$ the predicted value, and y_i the true value:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}, \tag{14}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right|, \tag{15}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right|. \tag{16}$$

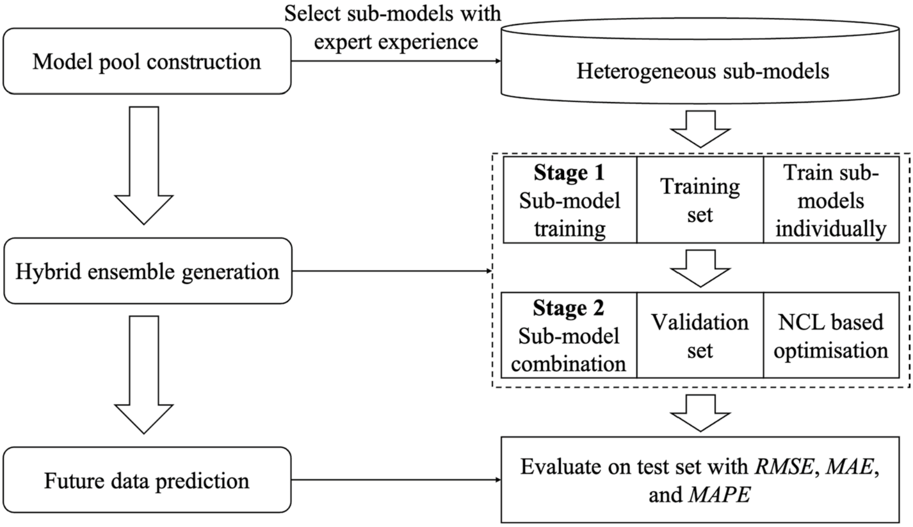
Model pool construction
Select sub-models with expert experience
Heterogeneous sub-models
Stage 1 Sub-model training
Training set
Train sub-models individually
Hybrid ensemble generation
Stage 2 Sub-model combination
Validation set
NCL based optimisation
Future data prediction
Evaluate on test set with *RMSE*, *MAE*, and *MAPE*

Fig. 3 Framework of the hybrid ensemble

3.5 Framework of the hybrid ensemble

Figure 3 demonstrates our proposed hybrid ensemble framework incorporating NCL. This framework includes model pool construction, hybrid ensemble generation, and future data prediction.

Initially, according to expert experience, we select eleven regression sub-models from different aspects like linear models, ensemble models, and neural networks with various structures and parameters to construct the heterogeneous model pool. Additionally, hybrid ensemble generation contains two stages: sub-model training and sub-model combination. In the first stage, the training set from the dataset is trained individually by the heterogeneous sub-models in the model pool. Grid Search and 5-fold Cross-Validation are involved in the training process to find the best parameter set for every model class and avoid overfitting. In the second stage, the NCL-based objective function is designed for model selection and weighting to find sub-models whose predictions have negative correlations, thus enhancing the diversity within the hybrid ensemble. The weights of each sub-model are automatically updated in the process of solving the objective function using the IPOPT solver in the Gekko optimizer. Finally, we treat the test set as future data to evaluate the proposed hybrid ensemble with *RMSE*, *MAE*, and *MAPE*, to see an improvement in contrast to the best-performed sub-model and other benchmarks.

4 Experiments

This section begins with an introduction of the datasets and model configurations in Sect. 4.1. Subsequently, we design several sets of experiments from different perspectives to highlight the strength of the proposed approach. Section 4.2 starts with the simple average of the elements in each ensemble. Section 4.3 applies the NCL method on the potential ensembles and explores whether the prediction accuracy will be improved. Section 4.4

Table 2 Description statistics of twenty datasets

Datasets	# Samples	# Features	Max of Y	Min of Y	Mean of Y	Median of Y	Std. of Y
01-Car	4322	7	8,900,000	20,000	504,785	350,500	578,800
02-House	21,613	20	7,700,000	75,000	540,182	450,000	367,362
03-Insurance	1338	6	63,770	1121	13,270	9382	12,110
04-Life_expectancy	2938	21	89	36	69	72	10
05-Walmart	6435	7	3,818,686	209,986	1,046,965	960,746	564,323
06-Blackfriday	537,577	10	23,961	185	9334	8062	4981
07-PM25	43,824	12	994	0	99	72	92
08-Temperature	7752	30	39	17	30	31	3
09-Power	9568	4	496	420	454	452	17
10-Concret	1030	8	82	2	36	34	17
11-Gas-2011	7410	10	119	28	68	66	11
11-Gas-2012	7628	10	120	12	69	67	10
11-Gas-2013	7152	10	120	43	70	69	12
11-Gas-2014	7158	10	118	27	60	59	10
11-Gas-2015	7384	10	120	26	60	57	11
12-Traffic	48,205	8	7280	0	3260	3380	1987
13-Produce	1198	14	1.12	0.23	0.74	0.77	0.17
14-Election	21,644	27	106	0	1.13	0	6.87
15-Bike	8761	13	3556	0	705	505	645
16-Steel	35,041	10	157	0	27	5	33

analyses the weights assigned by NCL on sub-models. Section 4.5 performs the competitive analysis between the NCL ensemble and the state-of-the-art weighting methods. Section 4.6 compares the prediction effect between the NCL ensembles and the best sub-models in each class. As the final experimental section, Sect. 4.7 provides a sensitivity analysis of the negative correlation penalty parameter λ.

4.1 Datasets and model configurations

In this study, we chose twenty public datasets from Kaggle[1] and UCI machine learning repository[2] to test the proposed NCL-based ensemble. These datasets cover the fields of economy, business, meteorology, and energy. The names and descriptive statistics are listed in Table 2.

Before modeling, data pre-processing is necessary. We first removed samples containing null values for each dataset, then transformed nominal variables into one-hot codes and sequential variables into continuous numeric codes. This paper divided the datasets into training, validation, and test sets. In our experiments, the training set was 50% of the overall. When setting the proportion of the validation set, there were two considerations: (1) the proportion of the validation set cannot be too high. Otherwise, the proportion of

[1] https://www.kaggle.com/

[2] https://archive.ics.uci.edu/ml/index.php

the test set would be too low, and the predictions would face a loss of accuracy. (2) with a high proportion of validation set, the Gekko solver would not produce a feasible solution due to data overload. Hence for most of the datasets in this paper, the validation set proportion was 10% of the total sample. As dataset 06-Blackfriday is sufficiently large and dataset 07-PM25 cannot be solved with a validation set ratio of 10%, the validation set proportions for these two datasets were set to 1% of the total.

Following this, we set the range of values for the critical parameters of each model. The grid search and cross-validation will select the optimal set of parameters from the parameter space for each model. The name, parameter range, and the number of sets for each model are listed in Table 3. All models and their parameters form the model pool for this paper. If the model corresponding to each parameter set is considered a sub-model, the model pool contains 2634 elements.

4.2 Comparison of simple average weighting

Starting with the simple average weighting of sub-models, this section considers the composition of three kinds of ensembles: (1) an ensemble of all sub-models; (2) ensembles of the sub-models within each model class; (3) an ensemble of the best sub-models in each model class.

4.2.1 Diversity of the ensembles

Intuitively, the more types of models in an ensemble, the higher the level of diversity. In practice, however, it is difficult to define the model types and thus to infer whether the ensemble diversity is caused by the variation of parameters or by the model design itself. It has been an opening problem in ensemble learning that needs a consensus diversity measurement. Nevertheless, we measured the diversity in an ensemble with correlation coefficients as introduced in Dutta (2009). For several sub-models in an ensemble, we computed the absolute Pearson correlations pairwise and picked the median value as the diversity measurement. Figure 4 illustrates the diversity values across the ensembles in twenty datasets with stacked bars.

In Fig. 4, the lower values indicate higher diversity since we used the absolute correlations to measure the diversity within an ensemble. The ensemble *SVR* has the lowest correlation and the highest diversity, followed by *DTR*. *Best_models* ranks 3rd and is better than *All_models* that ranks 10th. This fact shows that a diverse ensemble does not expect a large amount sub-models.

4.2.2 Performance of the ensembles

To examine the performance of these ensembles statistically, the Friedman and Nemenyi (FN) tests are used in this section (Demšar, 2006). The FN tests are based on the 13 ensembles in Fig. 4 ranking on the 20 datasets. The original hypothesis $\mathcal{H}_0$ of the Friedman test is that all ensembles do not perform significantly differently on all datasets. If the Friedman test rejects $\mathcal{H}_0$, the Nemenyi test is further used to test whether a significant difference exists between specific ensembles. Suppose the difference between the mean ordinal values of the two ensembles is greater than the threshold range of Nemenyi at a certain confidence level. In that case, the predictions of the two ensembles are significantly different. The results of the FN tests on *RMSE*, *MAE*, and *MAPE* are visualized in Fig. 5.

Table 3 Parameters sets for each model

Models	Parameters	# Parameter sets
SLR	fit_intercept:[True,False]	2
RR	alpha: [0.5,1,2] max_iter:[100,500,1000] solver:[auto, svd, cholesky, lsqr, sparse_cg, sag, saga] tol:[0.0001,0.001,0.01]	189
BR	n_iter:[100,300,500] tol:[0.0001,0.001,0.01] alpha_1:[0.000001,0.0001] alpha_2:[0.000001,0.0001] lambda_1:[0.000001,0.0001] lambda_2:[0.000001,0.0001] compute_score:[True,False] fit_intercept:[True,False]	576
SGDR	loss:[squared_loss,huber,epsilon_insensitive,squared_epsilon_insensitive] penalty:[l1,l2,elasticnet] alpha:[0.00001,0.0001,0.001] max_iter:[500,1000,1500] tol:[0.0001,0.001,0.01] learning_rate:[constant,optimal,invscaling,adaptive]	1296
PR	polynomialfeatures_degree:[2,3] polynomialfeatures_interaction_only:[True,False] polynomialfeatures_include_bias:[True,False] polynomialfeatures_order:[C,F]	16
RFR	n_estimators:[50,100,200] max_depth:[2,3,4] min_samples_split:[2,3,4] min_samples_leaf:[2,3] bootstrap:[True,False]	108
ABR	n_estimators:[10,50,100] learning_rate:[0.01,0.1,1] loss:[linear,square,exponential]	27
GBDT	n_estimators:[50,100,200] learning_rate:[0.01,0.1,0.5] loss:[ls,lad,huber,quantile] min_samples_split:[2,3] max_depth:[2,3,4]	216
SVR	kernel:[linear,poly,rbf,sigmoid] degree:[2,3,4] C:[0.5,1,2] gamma:[scale,auto]	72
DTR	splitter:[best,random] min_samples_split:[2,3] min_samples]_leaf:[2,3] max_features:[auto,sqrt,log2]	24

Table 3 (continued)

Models	Parameters	# Parameter sets
MPR	activation:[identity,logistic,tanh,relu] solver:[lbfgs,sgd,adam] alpha:[0.00001,0.0001,0.001] learning_rate:[constant,invscaling,adaptive]	108

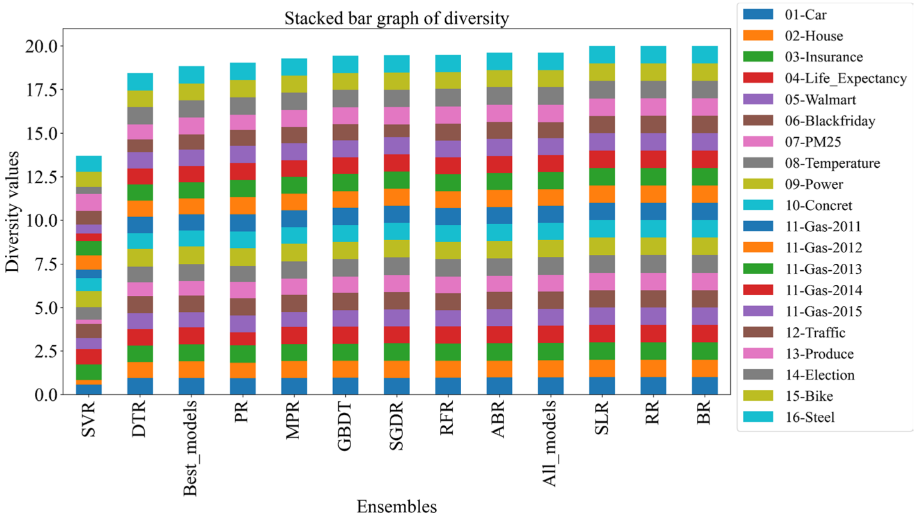

Fig. 4 The diversity values across the ensembles in twenty datasets. X-axis is the ensembles. The name *Best_models* is the ensemble of the best sub-models in each class, *All_models* is the ensemble of all sub-models, and the rest are ensembles of the sub-models in each class with the same name of the models in Table 3. Y-axis is the diversity values of each ensemble for all datasets, with the stacked form

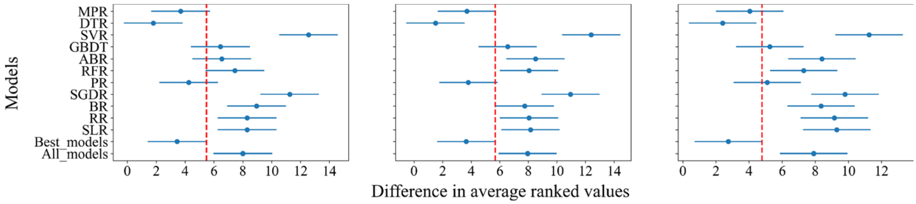

Fig. 5 Friedman and Nemenyi test on *RMSE* (left), *MAE* (middle), and *MAPE* (right). The horizontal axis is the differences in average ranked values of each ensemble, and the vertical axis is the names of the ensembles

In Fig. 5, each ensemble is represented by a line segment running through a point. The points are the average orders of an ensemble over all datasets, and the lower the value on the corresponding horizontal axis, the better the ensemble performs. The intervals of the line segments are the threshold ranges of the Nemenyi test. When comparing two ensembles, they are significantly different if there is no overlapping part of their line segments. We put red dashed lines in the figures to indicate the maximum average ranked values of the *Best_models* ensemble.

As can be seen in Fig. 5, the ensemble *Best_models* is significantly better than *All_models*, in line with that the *Best_models* is more diverse than *All_models* in Fig. 4. Another fact is that the ensemble of linear models, except the *PR*, do not offer either high diversity or good performance. Moreover, we can not tell the significant difference among the ensembles *MPR*, *DTR*, *PR*, and the *Best_models*. In Fig. 4, these good-performing ensembles have similar diversity and rank in the top 5. This phenomenon provides evidence from an experimental perspective that ensemble diversity is associated with performance, whether the ensemble is composed of the same or multiple types of sub-models. There is still an exception in the ensemble *SVR*, which performs unsatisfactorily compared to the others. Although it is far more diverse in Fig. 4 and there are some cases that *SVR* is the best sub-model in Table 12. This mismatch inspires the future search for the balance between diversity and performance, and an ensemble with excessive diversity may be risky.

4.3 Construction of NCL ensemble

This section constructs an NCL ensemble and compares it with other ensembles designed from different aspects. In detail, the NCL-based ensemble was built through the best sub-models in each model class, containing eleven members. The best sub-models were selected by the performance of the model on the validation set. Then the predictions on the validation set were input into the Algorithm (1) to finish the search for an optimal negative correlation strength λ^*. We could also obtain the weights ω for each sub-model through Algorithm (1). Finally, we weighted average the predictions on the test set with ω to generate the final outputs of the NCL-based ensemble.

We would compare the NCL-based ensemble with others considering the sub-model weights 4.3.1, the ensemble members in Sect. 4.3.2, the objective function in Sect. 4.3.3, the training modes in Sect. 4.3.4, and the number of sub-models in Sect. 4.3.5.

4.3.1 NCL-based v.s. simple average ensembles

The difference between NCL-based and simple average ensembles regards the weights of the sub-models. To explore how the weights influence performance, we compared two ensembles: one with the NCL method paying different attention to each sub-models, the other with equal weights. Table 4 demonstrates the improvement of the NCL-based ensemble over the simple average, in which the metrics with a prefix *Imp* are all measured with percentage.

In Table 4, the NCL-based ensemble could improve the simple average in most cases, around 15% in *RMSE*, 17% in *MAE*, and 10% in *MAPE* on the average of the twenty datasets. This fact verifies that the ensemble places varying emphasis on its sub-models to enhance performance further, although the sub-models are already diverse.

4.3.2 Best sub-models v.s. other ensemble members

Different types of ensemble members are considered here, including (1) the best sub-models in each model class, (2) all the sub-models in *DTR*, and (3) the average sub-models in each model class. The sub-models in *DTR* are chosen for the higher diversity and similar

Table 4 Improvement of the NCL-based ensemble over the simple average

Metrics(%)	01-Car	02-House	03-Insurance	04-Life_Expectancy	05-Walmart
ImpRMSE	2.21	18.29	7.00	9.22	59.86
ImpMAE	3.45	19.97	14.16	10.72	71.80
ImpMAPE	−4.48	12.96	15.15	16.25	36.15
Metrics(%)	06-Blackfriday	07-PM25	03-Insurance	09-Power	10-Concret
ImpRMSE	2.57	27.22	15.92	1.74	20.11
ImpMAE	4.13	31.76	17.23	1.87	26.04
ImpMAPE	−19.42	−18.50	16.35	9.74	22.35
Metrics(%)	11-Gas-2011	11-Gas-2012	11-Gas-2013	11-Gas-2014	11-Gas-2015
ImpRMSE	13.68	22.67	28.56	20.41	16.44
ImpMAE	15.48	23.57	31.35	25.93	18.07
ImpMAPE	12.13	17.52	25.20	48.99	13.24
Metrics(%)	12-Traffic	13-Produce	14-Election	15-Bike	16-Steel
ImpRMSE	0.17	−0.17	8.47	20.65	6.82
ImpMAE	0.42	0.17	4.79	26.69	6.85
ImpMAPE	−9.45	−3.02	3.01	6.29	1.56

performance as the *Best-models* in Sect. 4.2. The NCL method is used in all three ensembles with different members to generate the weights of sub-models and obtain the final predictions.

Table 5 presents the prediction errors of the three ensembles. The *Best-NCL* is the ensemble with the best sub-models of each model class. The *DTR-NCL* refers to the ensemble with the sub-models in DTR. The *Mean-NCL* is the ensemble comprising the average predictions generated by each model class. The values with bold font are the minimum values of error metrics.

Table 5 displays that the *Best-NCL* and *DTR-NCL* ensembles take the majority of the minimum errors. In this case, if an ensemble achieves the minimum error on a dataset, we count it as a win. The total competition count for each ensemble is 60, with 20 datasets and 3 metrics. *Best-NCL* wins 31 times out of 60, more than half of them (9, 9, and 13 counts on the three metrics, respectively). *DTR-NCL* wins 25 times, with 8, 10, and 7 counts on the three metrics. The *Mean-NCL* wins just 4 times. The inferior performance of *Mean-NCL* could be explained by the spatial distribution of the predictions from the different models. Taking '01-Car' as an example, we dimensionalized more than 2000 groups of predictions using the t-SNE technique. We visualized them on a two-dimensional plane, as shown in Fig. 6.

The visualization provides an intuitive representation of the prediction distributions. Specifically, predictions generated by one model class with different parameters are distributed in clusters in space and are distinguished from those of other models. We further abstract this distribution as shown in Fig. 7.

In Fig. 7, we suppose there are three model classes, each containing several sub-models. According to Fig. 6, the predictions from the same model class form a cluster in the space. A red star is put in the two-dimensional plane representing the ground truth. The

Table 5 Prediction errors of NCL-based ensembles

Dataset	RMSE			MAE			MAPE		
	Best-NCL	DTR-NCL	Mean-NCL	Best-NCL	DTR-NCL	Mean-NCL	Best-NCL	DTR-NCL	Mean-NCL
01-Car	0.698	**0.684**	0.696	0.332	**0.314**	0.341	**1.989**	1.990	1.999
02-House	**0.381**	0.401	0.428	**0.199**	0.204	0.226	**1.313**	1.403	1.456
03-Insurance	**0.348**	0.388	0.377	**0.191**	0.222	0.260	**0.558**	0.864	0.905
04-Life_Expectancy	0.265	**0.234**	0.319	0.189	**0.157**	0.238	1.075	**0.802**	1.551
05-Walmart	**0.254**	0.284	0.321	**0.141**	0.155	0.200	**1.369**	1.437	1.557
06-Blackfriday	0.718	0.712	**0.711**	0.573	**0.552**	0.561	**7.415**	8.131	7.960
07-PM25	0.549	**0.500**	0.526	0.360	**0.306**	0.351	1.703	**1.647**	1.448
08-Temperature	**0.319**	0.367	0.379	**0.240**	0.273	0.290	**1.117**	1.313	1.304
09-Power	0.230	**0.208**	0.247	0.178	**0.154**	0.195	1.666	**1.102**	1.981
10-Concret	**0.309**	0.323	0.373	**0.226**	0.239	0.297	**0.798**	0.839	0.833
11-Gas-2011	0.346	**0.343**	0.362	**0.198**	0.207	0.226	**1.059**	1.189	1.150
11-Gas-2012	0.344	0.356	**0.338**	0.224	0.238	**0.219**	**1.161**	1.443	1.209
11-Gas-2013	**0.298**	0.353	0.326	**0.208**	0.241	0.227	**1.071**	1.313	1.149
11-Gas-2014	**0.345**	0.358	0.418	**0.211**	0.222	0.268	**1.290**	1.506	2.085
11-Gas-2015	**0.290**	0.294	0.328	0.200	**0.189**	0.238	0.902	**0.834**	0.988
12-Traffic	0.975	0.974	**0.970**	0.848	**0.843**	0.844	**1.459**	1.571	1.519
13-Produce	**0.498**	0.537	0.551	**0.295**	0.327	0.368	**0.568**	0.651	0.730
14-Election	0.034	**0.029**	0.044	0.011	**0.004**	0.017	0.087	**0.027**	0.157
15-Bike	0.405	**0.373**	0.418	0.266	**0.236**	0.284	1.079	**0.924**	1.163
16-Steel	0.075	**0.056**	0.103	0.041	**0.029**	0.056	0.109	**0.100**	0.149

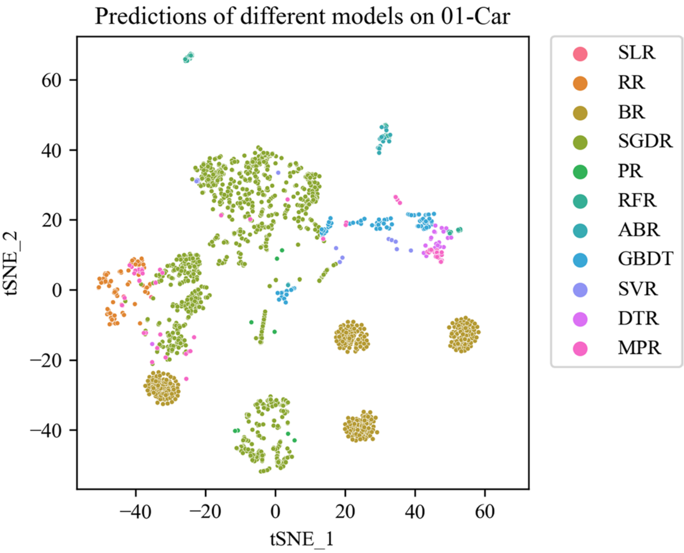

Fig. 6 Predictions distribution in the two-dimensional plane of 01-Car

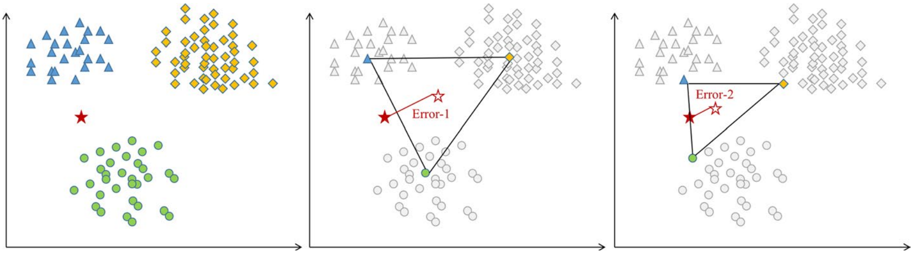

Fig. 7 Illustration of the best and average sub-model predictions for each model class

first sub-plot in Fig. 7 shows the location of the ground truth and predictions. The second sub-plot considers the average of the predictions in each class, which locates in the cluster center. When combining the three cluster centers to form an ensemble, the predictions of the ensemble would be in the center of the triangle region in the sub-plot. In the third sub-plot, we continue to find the best predictions from each class, which is the point that is nearest to the ground truth. Then it is evident that the triangle region shrinks as the points are near to the ground truth than the cluster center. Thus, the ensemble predictions of the best sub-models are closer to the ground truth.

4.3.3 NCL objective function with regularization term

Section 3.3.1 presents the objective function of the NCL ensemble as Formula (13), which includes the MSE and NCL penalty terms. As pointed out by Chen and Yao (2009), the model is easily overfitted when the data has nontrivial noise. The authors suggest adding a regularization term in the objective function to alleviate the overfitting problem. Similar as

Table 6 Prediction errors of NCL ensembles with and without regularization term

Dataset	RMSE		MAE		MAPE	
	Best-NCL	Best-NCLR	Best-NCL	Best-NCLR	Best-NCL	Best-NCLR
01-Car	0.698	0.780	0.332	0.397	1.989	2.358
02-House	0.381	**0.368**	0.199	**0.198**	1.313	**1.276**
03-Insurance	0.348	0.359	0.191	0.194	0.558	0.560
04-Life_Expectancy	0.265	**0.245**	0.189	**0.166**	1.075	**0.941**
05-Walmart	0.254	0.257	0.141	0.142	1.369	**1.295**
06-Blackfriday	0.718	**0.714**	0.573	**0.562**	7.415	8.203
07-PM25	0.549	**0.544**	0.360	**0.356**	1.703	1.720
08-Temperature	0.319	**0.307**	0.240	**0.233**	1.117	1.121
09-Power	0.230	**0.228**	0.178	**0.176**	1.666	**1.647**
10-Concret	0.309	0.312	0.226	**0.223**	0.798	0.817
11-Gas-2011	0.346	**0.345**	0.198	0.198	1.059	**1.053**
11-Gas-2012	0.344	0.344	0.224	0.225	1.161	1.161
11-Gas-2013	0.298	0.305	0.208	0.212	1.071	1.206
11-Gas-2014	0.345	**0.343**	0.211	**0.209**	1.290	**1.269**
11-Gas-2015	0.290	**0.285**	0.200	**0.193**	0.902	0.927
12-Traffic	0.975	**0.970**	0.848	**0.845**	1.459	**1.407**
13-Produce	0.498	0.499	0.295	0.296	0.568	**0.565**
14-Election	0.034	**0.032**	0.011	0.011	0.087	**0.086**
15-Bike	0.405	**0.394**	0.266	**0.255**	1.079	**1.018**
16-Steel	0.075	**0.070**	0.041	**0.038**	0.109	**0.104**

the neural network ensemble in Chen and Yao (2009), we redesigned the Formula (13) as follows:

$$\begin{aligned} & arg \min_{\omega} \Phi(\omega) = \sum_{j=1}^{m} \omega_j \left\{ \zeta_j - \frac{\lambda}{n} \sum_{i=1}^{n} \left(f_j(x_i) - f_h(x_i) \right)^2 \right\} + \sum_{j=1}^{m} \alpha_j \omega_j^T \omega_j, \\ & s.t. \ \sum_{j=1}^{m} \omega_j = 1, \\ & \qquad 0 \leq \omega \leq 1. \end{aligned} \tag{17}$$

where α_j is the strength of the regularization term $\sum_{j=1}^{m} \omega_j^T \omega$. Now we compare the NCL ensemble with and without the regularization term. The α_j for each sub-models is set equal to 0.05 for simplicity. Table 6 illustrates the performance of NCL ensembles on the twenty datasets. The *Best-NCL* is the NCL ensemble without regularization term, and *Best-NCLR* is the NCL ensemble with $\alpha = 0.05$. Table 6 shows that the regularization term marginally improved the NCL ensemble, with more than half of the data sets on each error metric.

4.3.4 Hybrid ensemble v.s. neural network ensemble

As introduced in Sect. 2.2, the NCL was developed in the scenario of the neural network ensemble training period. The hybrid ensemble in this paper transfers the NCL from model

Table 7 Improvement of the hybrid ensemble over the network ensemble

Metrics(%)	01-Car	02-House	03-Insurance	04-Life_Expectancy	05-Walmart
ImpRMSE	−16.38	23.87	4.40	16.69	65.29
ImpMAE	−22.82	27.10	13.98	21.96	73.41
ImpMAPE	−16.57	32.41	37.91	48.46	44.90
Metrics(%)	06-Blackfriday	07-PM25	08-Temperature	09-Power	10-Concret
ImpRMSE	3.07	30.77	26.19	7.07	11.62
ImpMAE	5.02	34.62	27.13	8.13	14.93
ImpMAPE	−35.31	18.43	31.38	3.82	14.15
Metrics(%)	11-Gas-2011	11-Gas-2012	11-Gas-2013	11-Gas-2014	11-Gas-2015
ImpRMSE	8.29	11.59	22.32	20.04	9.30
ImpMAE	9.89	13.49	20.74	23.35	7.17
ImpMAPE	11.02	23.97	38.97	24.54	11.10
Metrics(%)	12-Traffic	13-Produce	14-Election	15-Bike	16-Steel
ImpRMSE	−0.14	8.05	71.78	23.80	42.30
ImpMAE	−0.55	12.89	82.31	27.35	45.23
ImpMAPE	6.86	22.09	81.57	46.58	60.59

training to the combination stage while keeping heterogeneous sub-models as a diverse model pool. It is still worth comparing the ensembles where NCL works in separate stages.

Given the well-predicted multilayer perceptron in Fig. 5, we set up a fully connected forward neural network as the sub-model. After tuning the hyperparameters, we set each sub-network containing two hidden layers, with 16 neurons in each layer. The forward propagation took sigmoid as the activation function, and the backward propagation used gradient descent with a regular term to update the weights and biases with a factor of 0.01. The individual sub-networks were trained in batches to improve robustness and computational speed, with a batch size of 256. To match the number of sub-models in the hybrid ensemble, we also set up 11 sub-networks in the network ensemble. The learning rate of the network ensemble was 0.001, and the negative correlation strength λ of both ensembles was 0.5.

Similar to Tables 4 and 7 illustrates the percentage improvement of the hybrid ensemble consisting of the best sub-models over the network ensemble trained by NCL. In most cases, the hybrid ensemble improves the network ensemble with around 19% on RMSE, 22% on MAE, and 25% on MAPE on the average of all the datasets. Compared to the simple average, the NCL-based hybrid ensemble achieves a higher percentage improvement over the network ensemble. The results indicate that even a simple averaged heterogeneous ensemble outperforms a weight-optimized homogeneous ensemble in our regression case.

4.3.5 The number of sub-models in the NCL ensemble

In this paper, 11 model classes were initially selected empirically according to the model design and architecture, and the corresponding 11 best sub-models were generated based on the prediction results on the validation set, thereby forming the NCL hybrid ensemble.

Table 8 Prediction errors of NCL ensembles with different numbers of sub-models

Dataset	RMSE			MAE			MAPE		
	NCL-All	NCL-T5	NCL-T3	NCL-All	NCL-T5	NCL-T3	NCL-All	NCL-T5	NCL-T3
01-Car	0.698	0.780	0.780	0.332	0.397	0.397	1.989	2.349	2.349
02-House	0.381	0.388	0.388	0.199	0.209	0.209	1.313	1.336	1.336
03-Insurance	0.348	0.374	0.388	0.191	0.212	0.218	0.558	0.558	0.598
04-Life_Expectancy	0.265	0.280	0.306	0.189	**0.184**	0.201	1.075	**1.001**	**1.101**
05-Walmart	0.254	**0.248**	0.599	0.141	0.143	0.424	1.369	**0.969**	2.377
06-Blackfriday	0.718	0.737	0.737	0.573	**0.572**	**0.572**	7.415	7.873	7.873
07-PM25	0.549	0.673	0.736	0.360	0.464	0.468	1.703	2.287	1.723
08-Temperature	0.319	**0.307**	0.332	0.240	**0.233**	0.242	1.117	1.121	1.119
09-Power	0.230	**0.212**	**0.205**	0.178	**0.160**	**0.152**	1.666	**1.290**	**1.096**
10-Concret	0.309	**0.300**	**0.300**	0.226	**0.224**	**0.224**	0.798	**0.785**	**0.784**
11-Gas-2011	0.346	0.367	0.349	0.198	0.221	0.199	1.059	1.202	**0.986**
11-Gas-2012	0.344	0.374	0.351	0.224	0.248	**0.215**	1.161	1.557	1.290
11-Gas-2013	0.298	0.321	0.321	0.208	0.224	0.225	1.071	1.314	1.315
11-Gas-2014	0.345	**0.340**	**0.339**	0.211	**0.206**	**0.208**	1.290	**1.208**	**1.251**
11-Gas-2015	0.290	**0.287**	**0.287**	0.200	**0.196**	**0.196**	0.902	0.941	0.941
12-Traffic	0.975	0.975	0.975	0.848	0.848	0.848	1.459	1.459	1.459
13-Produce	0.498	0.517	0.549	0.295	0.312	0.342	0.568	0.629	0.707
14-Election	0.034	0.037	0.036	0.011	0.013	0.016	0.087	0.100	0.120
15-Bike	0.405	**0.384**	0.410	0.266	**0.243**	0.273	1.079	**0.901**	**1.078**
16-Steel	0.075	**0.069**	**0.055**	0.041	**0.036**	**0.028**	0.109	0.116	**0.099**

Regarding the number of sub-models in the NCL ensemble, we construct the ensemble with the top 5 and top 3 sub-models in each dataset. Table 8 lists the prediction errors of the ensembles with all the sub-models, the top 5 and top 3 sub-models in each dataset.

In Table 8, the column name *NCL-ALL* is the NCL ensemble with all the eleven sub-models, the *NCL-T5* and *NCL-T3* correspond to the ensembles with top 5 and top 3 sub-models. The errors with bolded font are the *NCL-T5* or *NCL-T3* exceeding *NCL-ALL*. It could be observed that *NCL-T5* or *NCL-T3* performs better than *NCL-All* of each metric only on less than half of the datasets. This fact leads to the conclusion that the sub-model that constitutes the ensemble is not necessarily the top performer on the dataset. Some of the less-performing sub-models could still contribute negative knowledge to the ensemble, which coincides with the findings of Sirovetnukul et al. (2011).

4.4 Analysis of sub-model weights

As stated in Sect. 4.3.5, the components of the ensemble are not necessarily the good-performing sub-models. In other words, good sub-models may not always contribute to the hybrid ensemble. In the hybrid ensemble, each sub-model is assigned a weight obtained by the NCL penalty term. The weights are regarded as the proportion of the sub-models in the ensemble. This subsection explores whether the weight values of sub-models are related to their ability to contribute to the ensemble. Concretely, for each dataset, we computed the

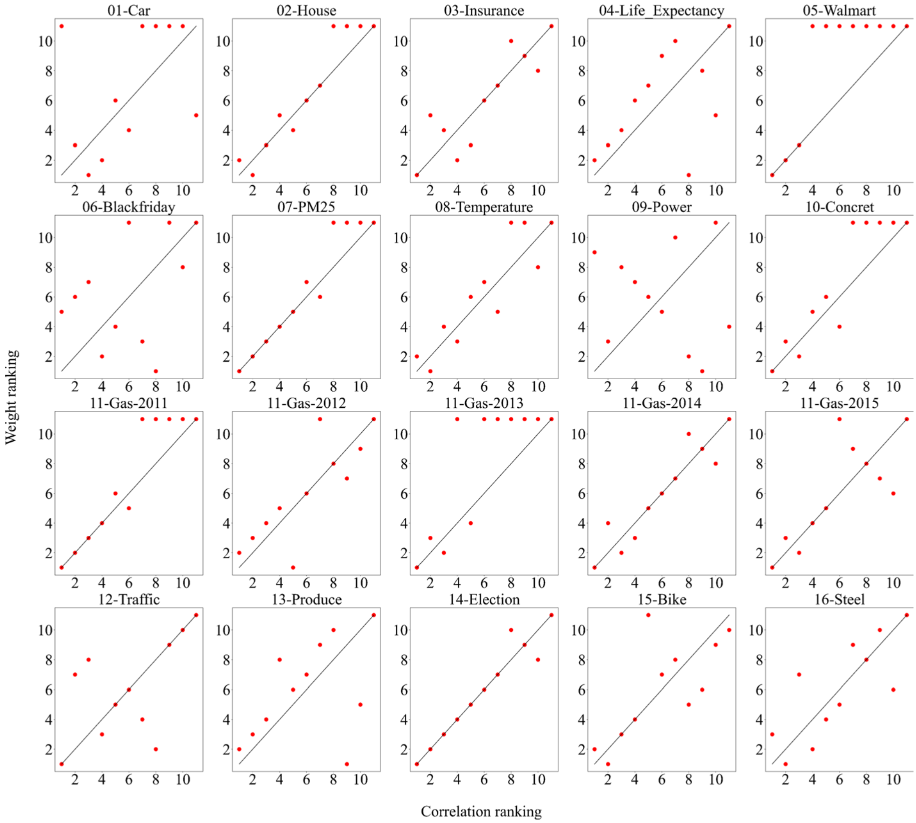

Fig. 8 Scatter plot between the rankings of prediction correlation and sub-model weights for 20 datasets. The x-axis is the correlation rankings of the 11 sub-models, and the y-axis is the weight rankings. Each subplot is titled by the name of the dataset and contains the scatters as sub-models. The identity line in each subplot indicates that a sub-model has the same ranking in correlation and weight

Pearson correlation coefficients between the predictions of each sub-model and the hybrid ensemble. The sub-models were ranked according to their correlation with the ensemble from highest to lowest. Then followed by another ranking list of the sub-model weights from maximum to minimum. We make scatter plots with the two ranking lists as in Fig. 8.

Figure 8 illustrates the relationship between the prediction correlation and weight for each sub-model and dataset. There are some scatters on the top row of several subplots, such as ‘01-Car’ with five and ‘02-House’ with four. These scatters correspond to sub-models with zero weights that are filtered out by the NCL ensemble automatically. Besides that, the other scatters surround the identity line, exhibiting obvious positive correlations. These subplots reveal that the higher the weight, the more the sub-model correlates with the ensemble predictions and the more significant its contribution to the final performance.

4.5 Comparison with state-of-the-art weighting methods

The target of a hybrid ensemble is to assign weights to the sub-model with the supervised or unsupervised method. Besides the NCL-based hybrid ensemble proposed in this study, state-of-the-art methods also assign weights to sub-models. According to the summary of Mendes-Moreira, there are constant and not-constant weighting methods for building an ensemble (Mendes-Moreira et al., 2012). As the name implies, the constant methods assign constant weights to each sub-model. On the other hand, the weights generated by the non-constant methods vary depending on the input data.

4.5.1 Constant weighting methods

The most typical constant weighting method is simple averaging, also called the Basic Ensemble Method (BEM) in Mendes-Moreira et al. (2012). It does not regard the importance of any sub-model nor depend on data attributes and assigns the same weight to all sub-models. In addition to the simple averaging, the sub-models selected by the NCL-based ensemble are considered here for simple averaging, denoted as BEM-NCL, which has a filtering effect compared to the simple averaging of all sub-models.

Another constant method is Generalized Ensemble Method (GEM) (Perrone and Cooper, 1992). GEM generates weights according to the sub-model errors between the actual values and predictions. In contrast to the BEM, there is no need to assume that these errors are mutually independent and zero-mean. Let $e_j(x_i) = y_i - f_j(x_i)$ is the error between true value y_i and prediction $f_j(x_i)$ from the j_{th} sub-model. Then let w_j be the weight assigned on this sub-model, and it is calculated as:

$$w_j = \frac{\sum_{j=1}^{m} C_{ij}^{-1}}{\sum_{i=1}^{m} \sum_{j=1}^{m} C_{ij}^{-1}}, \tag{18}$$

in which $C_{ij} = E[e_i(x), e_j(x)]$ is a symmetric correlation matrix of order M.

Linear Regression (LR) is also a constant weighting method, with the predictions of the individual sub-models as the independent variables and the true values as the dependent variables. After the linear regression has fitted the data, the coefficients are taken as the weights for each sub-model. Unlike GEM, the sum of the linear regression weights does not need to be equal to 1.

4.5.2 Non-constant weighting methods

Meta Decision Trees (MDT) method was proposed by Todorovski and Džeroski (2003) to solve the classification problem, then introduced by Mendes-Moreira et al. (2012) as a method of non-constant weighting. MDT is trained on the predictions of the individual sub-models to target true values. However, it produces a decision tree model rather than a set of coefficients, as in linear regression. This decision tree model is fitted over the new data to produce the final predicted values, and its potential weights are a decision tree.

Mendes-Moreira classified dynamic weighting, based on the local performance of different sub-models, as a non-constant weighting method (Mendes-Moreira et al., 2012). Two intuitive examples are Error Inverse Weighting (EIW) and Error Exponential Weighting (EEW) from Armstrong's design (Armstrong, 2001). These two weighting methods connect weights to errors, assuming that the higher the error, the less the proportion of the sub-model in the overall ensemble. The formulas for EIW and EEW are

Table 9 Comparison with constant and non-constant methods on RMSE

Dataset	BEM	BEM-NCL	GEM	LR	MDT	EIW	EEW	Best-NCL
01-Car	0.696	0.683	1.432	1.527	1.050	0.694	0.694	**0.678**
02-House	0.463	0.423	0.454	0.466	0.521	0.446	0.454	**0.377**
03-Insurance	0.383	0.375	0.428	0.400	0.526	0.376	0.380	**0.355**
04-Life_Expectancy	0.294	0.294	0.535	0.278	0.596	0.281	0.290	**0.266**
05-Walmart	0.636	0.321	**0.256**	**0.256**	0.404	0.492	0.563	**0.256**
06-Blackfriday	0.736	0.718	0.717	**0.716**	0.978	0.732	0.733	0.718
07-PM25	0.752	0.685	0.534	**0.533**	0.758	0.725	0.732	0.549
08-Temperature	0.376	0.371	0.340	0.346	0.448	0.366	0.372	**0.319**
09-Power	0.234	0.234	**0.217**	**0.217**	0.278	0.231	0.233	0.230
10-Concret	0.380	0.329	0.396	0.363	1.001	0.357	0.369	**0.309**
11-Gas-2011	0.405	0.348	0.385	0.382	0.730	0.384	0.395	**0.346**
11-Gas-2012	0.443	0.423	0.474	0.471	0.522	0.406	0.422	**0.344**
11-Gas-2013	0.419	0.319	0.369	0.395	0.484	0.381	0.400	**0.298**
11-Gas-2014	0.436	0.423	0.373	0.377	0.606	0.407	0.421	**0.345**
11-Gas-2015	0.342	0.328	0.329	0.333	0.459	0.321	0.333	**0.290**
12-Traffic	**0.975**	**0.975**	1.030	1.047	1.324	**0.975**	**0.975**	**0.975**
13-Produce	**0.490**	**0.490**	0.536	0.500	0.831	**0.490**	**0.490**	0.498
14-Election	0.036	0.036	**0.009**	0.014	0.079	0.016	0.035	0.034
15-Bike	0.511	0.513	0.414	0.414	0.597	0.483	0.496	**0.405**
16-Steel	0.080	0.080	**0.031**	**0.031**	0.064	0.051	0.077	0.075
Average ranking	5.8	3.7	3.9	3.7	7.55	3.35	4.65	**1.85**

$$EIW_j = \frac{1/Error_j}{\sum_{j=1}^{m} 1/Error_j}, \tag{19}$$

$$EEW_j = \frac{e^{-Error_j}}{\sum_{j=1}^{m} e^{-Error_j}}, \tag{20}$$

in which $Error_j$ can be any of the metrics from *RMSE*, *MAE*, and *MAPE*.

The NCL-based ensemble proposed in this paper is a dynamic weighting method that integrates model selection with model weighting and belongs to the category of non-constant weighting.

4.5.3 Comparison with constant and non-constant methods

After an overview of the classical constant and non-constant weighting methods, this subsection compares the proposed NCL-based ensemble with these weighting methods. The comparisons between our proposed NCL method (noted as Best-NCL) and the state-of-the-art weighting methods are listed in Tables 9, 10, and 11. These three tables contain the RMSE, MAE, and MAPE results on all twenty datasets. Besides, we add another row to describe the average ranking of the methods on all datasets in each table.

Table 10 Comparison with constant and non-constant methods on MAE

Dataset	BEM	BEM-NCL	GEM	LR	MDT	EIW	EEW	Best-NCL
01-Car	0.339	**0.325**	0.677	0.688	0.468	0.337	0.338	**0.325**
02-House	0.248	0.226	0.265	0.275	0.288	0.234	0.244	**0.198**
03-Insurance	0.229	0.225	0.266	0.199	0.298	0.213	0.225	**0.196**
04-Life_Expectancy	0.213	0.213	0.383	0.197	0.448	0.198	0.209	**0.190**
05-Walmart	0.499	0.208	0.151	0.151	0.204	0.329	0.438	**0.142**
06-Blackfriday	0.597	0.574	**0.564**	**0.564**	0.739	0.591	0.593	0.573
07-PM25	0.527	0.476	**0.347**	**0.346**	0.478	0.495	0.512	0.360
08-Temperature	0.288	0.281	0.262	0.268	0.330	0.278	0.285	**0.240**
09-Power	0.182	0.182	**0.160**	**0.160**	0.207	0.179	0.182	0.178
10-Concret	0.298	0.253	0.300	0.267	0.819	0.276	0.290	**0.226**
11-Gas-2011	0.235	0.201	0.236	0.240	0.333	0.219	0.230	**0.198**
11-Gas-2012	0.293	0.277	0.302	0.302	0.355	0.267	0.284	**0.224**
11-Gas-2013	0.303	0.236	0.268	0.290	0.331	0.274	0.294	**0.208**
11-Gas-2014	0.285	0.276	0.235	0.234	0.380	0.258	0.276	**0.211**
11-Gas-2015	0.242	0.235	0.234	0.236	0.332	0.224	0.236	**0.200**
12-Traffic	0.851	0.851	0.855	0.857	1.056	0.850	0.850	**0.848**
13-Produce	**0.290**	**0.290**	0.347	0.296	0.504	0.289	0.289	0.295
14-Election	0.011	0.011	**0.001**	**0.001**	0.014	0.002	0.011	0.011
15-Bike	0.363	0.354	0.287	0.287	0.443	0.332	0.351	**0.266**
16-Steel	0.044	0.044	**0.016**	**0.016**	0.028	0.024	0.043	0.041
Average Ranking	5.9	3.85	4	3.6	7.35	3.45	4.7	**1.85**

As illustrated in Tables 9, 10, and 11, some remarks can be summarised: (1) considering the BEM constant weighting method, NCL is a choice to improve the predictions; (2) Best-NCL performs better than all the methods regarding the number of datasets; (3) the average ranking of Best-NCL is higher than all the methods on RMSE and MAE metrics; (4) on the MAPE metric, the average ranking of Best-NCL is close to that of EEW, although Best-NCL is better than EEW on more datasets. This is caused by the extreme errors on MAPE (06-Blackfriday), while EEW is affected less.

To significantly show the comparison between the Best-NCL and other methods, the FN tests were performed on the prediction results of the eight weighting methods on the 20 datasets. Figure 9 presents the results of the statistical analysis of the FN test.

From Fig. 9, our proposed Best-NCL performs better than the other methods on *RMSE* and *MAE* significantly. In *MAPE*, Best-NCL performs comparably to the two non-constant methods, EIW and EEW, and outperforms the other weighted methods. These results are in line with what is observed from Tables 9, 10, and 11. All the constant weighting methods listed in this section perform unsatisfactorily, although BEM-NCL with the same sub-models as Best-NCL. The experiments confirm the superiority of the NCL ensemble and illustrate that fusion of sub-model selection and weighting is necessary when building the ensemble.

Table 11 Comparison with constant and non-constant methods on MAPE

Dataset	BEM	BEM-NCL	GEM	LR	MDT	EIW	EEW	Best-NCL
01-Car	1.842	1.817	3.628	3.705	2.380	1.782	**1.731**	1.923
02-House	1.449	1.476	1.757	1.818	2.050	1.347	1.285	**1.269**
03-Insurance	0.656	0.678	0.820	**0.515**	0.651	0.580	0.597	0.555
04-Life_Expectancy	1.211	1.211	2.956	1.295	6.873	1.074	1.024	**1.013**
05-Walmart	2.021	1.590	0.968	0.855	2.900	1.309	**0.920**	1.277
06-Blackfriday	6.202	7.601	7.909	7.814	14.491	4.350	**1.966**	7.415
07-PM25	1.434	1.649	1.926	1.916	3.112	1.360	**1.317**	1.703
08-Temperature	1.322	1.284	1.267	1.337	2.109	1.275	1.255	**1.117**
09-Power	1.724	1.724	**1.303**	1.311	2.452	1.583	1.499	1.666
10-Concret	0.999	0.820	1.040	0.954	2.409	0.927	0.913	**0.798**
11-Gas-2011	1.163	1.079	1.332	1.230	2.228	1.104	1.087	**1.059**
11-Gas-2012	1.371	1.374	2.037	2.015	2.292	1.300	1.249	**1.161**
11-Gas-2013	1.452	**1.057**	1.439	1.403	2.065	1.229	1.116	1.071
11-Gas-2014	2.382	2.303	1.911	1.550	1.929	1.812	1.350	**1.290**
11-Gas-2015	1.006	1.001	1.160	1.148	1.348	0.960	0.950	**0.902**
12-Traffic	1.332	1.332	1.849	1.847	4.023	1.308	**1.301**	1.459
13-Produce	0.557	0.557	0.783	0.558	1.217	**0.545**	0.551	0.568
14-Election	0.089	0.089	0.011	**0.010**	0.019	0.021	0.085	0.087
15-Bike	1.355	1.346	1.112	1.110	1.360	1.196	1.125	**1.079**
16-Steel	0.115	0.115	**0.077**	**0.077**	0.128	0.092	0.111	0.109
Average Ranking	5.2	4.55	5.4	4.5	7.4	3.25	**2.5**	2.85

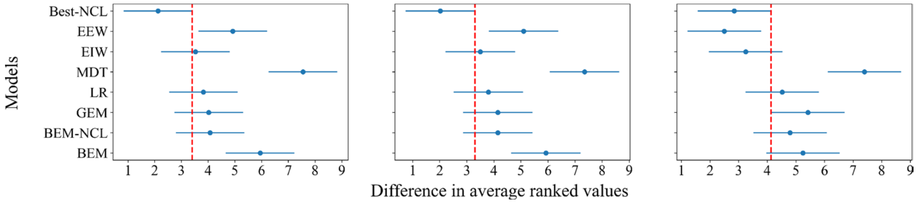

Fig. 9 Friedman and Nemenyi test on *RMSE* (left), *MAE* (middle), and *MAPE* (right). The horizontal axis is the differences in average ranked values of each method with the vertical axis the names of them

4.6 Comparison with best sub-model in each group

The previous subsections compared and analyzed NCL-based ensemble with other ensemble methods. This subsection aims to continue the exploration of the NCL-based ensemble concerning the best sub-models in each model class. Table 12 lists the model class that the best sub-model belongs to for each dataset on the validation and test set under the three metrics. There are columns named '$\lambda = 0$' also in Table 12, given that the hybrid ensemble only selects one sub-model when there is no NCL. Bolded fonts in Table 12 are the sub-models that perform consistently on the validation and test sets. If the NCL ensemble outperforms the best sub-model on the final test set, that sub-model is marked with a star.

Table 12 Comparison with best sub-models

Dataset	RMSE			MAE			MAPE		
	Validation set	Test set	$\lambda = 0$	Validation set	Test set	$\lambda = 0$	Validation set	Test set	$\lambda = 0$
01-Car	BR	DTR	BR	RFR	SVR	BR	PR	BR	BR
02-House	GBDT	SVR	GBDT	**DTR**	**DTR**	GBDT	ABR	SVR	GBDT
03-Insurance	GBDT	MPR*	GBDT	**MPR**	**MPR**	GBDT	RR	RFR	GBDT
04-Life_Expectancy	MPR	SVR	DTR	SVR	DTR	DTR	SVR	MPR	DTR
05-Walmart	**MPR**	**MPR**	GBDT	**MPR**	**MPR***	GBDT	DTR	GBDT	GBDT
06-Blackfriday	ABR	PR*	PR	SVR	ABR	PR	RFR	DTR	PR
07-PM25	SVR	GBDT*	DTR	**SVR**	**SVR**	DTR	**RFR**	**RFR**	DTR
08-Temperature	MPR	DTR*	MPR	DTR	SVR*	MPR	RFR	DTR*	MPR
09-Power	ABR	DTR	GBDT	SVR	DTR	GBDT	PR	DTR	GBDT
10-Concret	**MPR**	**MPR***	MPR	**MPR**	**MPR***	MPR	SVR	MPR*	MPR
11-Gas-2011	**MPR**	**MPR**	SVR	**MPR**	**MPR**	SVR	SVR	GBDT	SVR
11-Gas-2012	**MPR**	**MPR***	DTR	DTR	MPR	DTR	GBDT	MPR	DTR
11-Gas-2013	**DTR**	**DTR***	MPR	DTR	MPR*	MPR	SGDR	MPR	MPR
11-Gas-2014	SVR	MPR*	GBDT	DTR	MPR*	GBDT	RFR	DTR	GBDT
11-Gas-2015	SVR	MPR*	SVR	DTR	MPR	SVR	RFR	MPR*	SVR
12-Traffic	PR	GBDT	RFR	SGDR	ABR	RFR	SVR	DTR	RFR
13-Produce	RFR	SDGR*	GBDT	RFR	PR	GBDT	SVR	MPR	GBDT
14-Election	**MPR**	**MPR**	DTR	**DTR**	**DTR**	DTR	GBDT	DTR	DTR
15-Bike	**DTR**	**DTR**	GBDT	**DTR**	**DTR**	GBDT	DTR	SVR	GBDT
16-Steel	DTR	MPR	GBDT	DTR	MPR	GBDT	GBDT	DTR	GBDT

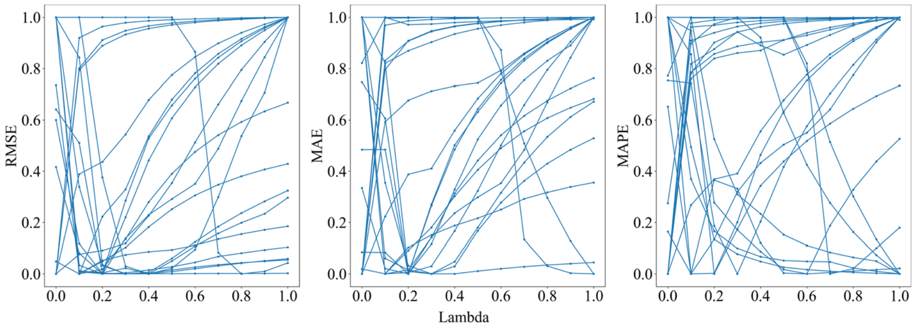

Fig. 10 Relationship between ensemble performance and λ. The horizontal axis is λ values from 0 to 1 with step 0.1, and the vertical axis is the corresponding prediction metric *RMSE*, *MAE*, and *MAPE*

From Table 12, the single sub-model selected by the NCL ensemble when $\lambda = 0$ might differ from those in the columns ‘Validation set’ since the objective function takes MSE error. Table 12 is an ideal example of model instability. In the 20 datasets, only a few sub-models perform both best on the validation and test sets. Model instability occurs when, despite promising results for the current local model, the original optimal model is hard to maintain once new data are available and the data distribution changes. In some cases, our proposed NCL ensemble is even better than the best-performing sub-models, according to the star marks in Table 12, and is a relatively robust ensemble under the RMSE metric.

Building NCL ensembles is a challenging task. Not only do we have to compare and select sub-models on the validation set thoroughly, but we also have to manipulate the approach to solve the optimization problem, which will undoubtedly consume some time. However, if the proposed NCL ensemble eventually achieves results comparable to the best-performing sub-model, it means that the ensemble can overcome model instability to some extent. This is quite important. In practice, testing a best-performing sub-model involves picking from a large model pool, which is as time-consuming as building an NCL-based ensemble. Once the data distribution changes in the future, this sub-model may not continue to predict well, as no perfect single model is suitable for all data. In this case, the NCL-based ensemble performs more robustly and is a better choice.

4.7 The sensitivity analysis of negative correlation strength

The parameter λ in the NCL objective function controls the strength of the negative correlation. If λ is close to 0, the NCL objective function can only pick the sub-model with the lowest MSE error on the validation set, which is no different in methodology from selecting a sub-model based on other metrics. If λ is close to 1, The optimization task will search for the most diverse sub-models for the ensemble. We plot Fig. 10 to present how the prediction errors change with λ.

From Fig. 10, we observe that the *RMSE* and *MAE* errors first decrease at λ within 0.1 and 0.2. This phenomenon shows in more than half of the datasets. There are also datasets with higher data volume that decreases at higher λ, such as 06-BlackFriday, and some datasets take lower λ and will increase when λ is higher than 0.1. The *MAPE* error is more sensitive than the other two metrics on the change of λ. Our findings fit that of Brown et al. (2005). The authors found an upper bound of λ, and λ stabilized when the number of sub-models in the ensemble increased to a certain level. There is a similar pattern in our sensitivity analysis of λ. According to Fig. 10, it is necessary to try different λ for different datasets. Thus, the Algorithm 1 designed in this paper to automatically search λ is beneficial.

5 Discussion

As one of the essential branches of ensemble models, the hybrid ensemble has made significant progress in research and practice. However, the hybrid ensemble still faces the problem of choosing the appropriate model subset and assigning weights to the sub-models. Simply averaging the predictions of all sub-models does not achieve the expected results; even the corrections using weighted averaging methods are limited. This study proposes a novel method for a hybrid ensemble that automatically selects models and generates appropriate weights, yielding comparable performance with the optimal sub-models in regression problems.

A body of studies has experimentally demonstrated that diversity is a critical factor in the success of hybrid ensembles. Most studies investigated the sub-model training stage, working on sampling the data and modifying the parameters of homogeneous models, but the diversity generated in this way could be improved. This study proposes a regression prediction framework incorporating NCL, considering the diversity in both the sub-model training and combination stages. Eleven regression models with different structures and parameters are chosen in the sub-model training stage to build a model pool and fit the training set separately. Second, the study extends the use of NCL from the previous model training to the model combination. Using the interior-point filter linear-search algorithm in the Gekko solver, we solve the optimization problem of model selection and combination to select the negatively correlated model directly sets from the model pool and generate weighted predictions simultaneously. Furthermore, the solution to the optimization problem depends on the negative correlation strength λ. Based on this, an algorithm is designed to automatically search for the optimal λ, avoiding the time wastage of manual search and testing.

The experimental results support that using NCL in the hybrid ensemble is a beneficial initiative, and the importance of diversity is demonstrated in both stages of the ensemble. In the sub-model training stage, if all the sub-model predictions are projected into a two-dimensional plane, it is evident that those from the same model class will gather into a cluster. Spatially, the best sub-model in each model class is closer to the true value than the average center of each class. The range of geometries formed by the best sub-models is thus more minor, and the ensemble falling in this range has a higher probability of exceeding the average of each model class. This paper also demonstrates that the ensemble performance is related to the sub-model diversity and that it is statistically better to construct the ensemble with the best sub-models from different model classes.

In the sub-model combination stage, we innovatively considered diversity and solved the optimization problem by incorporating NCL using an interior-point filtering linear-search algorithm. The experimental results show some inspiring points: (1) the NCL ensemble improves the simple average that lacks selecting and weighting procedures; (2) the NCL penalty is beneficial in the sub-models with higher diversity, such as the best sub-models and DTR members; (3) the NCL ensemble can be improved further by adding a regularization term in the objective function; (4) the NCL ensemble performs better than the network ensemble with training the homogeneous sub-models; (5) it is necessary to keep some not-satisfying sub-models in the ensemble due to the negative knowledge they may offer; (6) the weights of the sub-models are in line with the ensemble performance; (7) as a non-constant weighting method, NCL ensemble is superior to other constant weighting methods; (8) the NCL ensemble can overcome the model instability and performs close to the best sub-model; (9) the auto-searching algorithm is helpful in finding an optimal λ.

A limitation of this study is that the eleven sub-models in the model pool need to cover more established models in the regression field, which also provides researchers with the freedom to replace candidate models. This study also needs a more in-depth exploration of how the data features influence the ensemble effect.

6 Conclusion

We developed a hybrid ensemble approach incorporating negative correlation learning, considering model diversity in the sub-model training and combination stages. NCL acts as a penalty term for the objective function to be optimized, assisting in the model selection process to find subsets with diversity. Experiments on twenty publicly available regression datasets confirm the effectiveness of this approach.

First, the proposed method is user-friendly and easy to understand. Practitioners no longer need to evaluate the effectiveness of individual models using various accuracy indexes to select the best one, nor do they need to blindly weight the candidate models, since the hybrid ensemble with the addition of NCL can fully demonstrate prediction accuracy that approximates or exceeds that of the best sub-model with appropriate penalty strength. Additionally, the predictions from any model can be added to the model pool as an element for the calculation. Even if the model does not work well, this method will discard it automatically. Therefore, our proposed method has practical implications.

Acknowledgements We thank the reviewers for their time and efforts in reviewing this manuscript. We are also grateful for all the valuable comments that help to improve the paper. Yanfei Kang is supported by the National Natural Science Foundation of China (No. 72171011).

Author Contributions YB: algorithm design, data experiment, and paper writing GT: data experiment YK: main theory, paper revision, and submission SJ: paper revision and suggestions

Funding Yanfei Kang is supported by the National Natural Science Foundation of China (No. 72171011).

Data availibility The datasets used in this paper are all from the Kaggle open platform.

Code availability We make our codes publicly on Github(https://github.com/BaiyunBuaa/Hybrid-ensemble-based-on-Negative-Correlation-Learning), please feel free to try!

Declarations

Conflict of interest (check journal-specific guidelines for which heading to use) We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, and there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled “A hybrid ensemble method with negative correlation learning for regression’.

Ethical approval This paper does not contain any studies with human participants or animals performed by any of the authors.

Consent to participate Informed consent was obtained from all individual participants included in the study.

Consent for publication The author confirms that this publication has been approved by all co-authors.

References

Ala'raj, M., & Abbod, M. F. (2016). A new hybrid ensemble credit scoring model based on classifiers consensus system approach. *Expert Systems with Applications, 64*, 36–55. https://doi.org/10.1016/j.eswa.2016.07.017

Alhamdoosh, M., & Wang, D. (2014). Fast decorrelated neural network ensembles with random weights. *Information Sciences, 264*, 104–117. https://doi.org/10.1016/j.ins.2013.12.016

Armstrong, J. S. (2001). *Principles of Forecasting: a Handbook for Researchers and Practitioners*. Springer.

Beal, L. D., Hill, D. C., Martin, R. A., & Hedengren, J. D. (2018). Gekko optimization suite. *Processes, 6*(8), 106. https://doi.org/10.3390/pr6080106

Bian, Y., & Chen, H. (2021). When does diversity help generalization in classification ensembles? *IEEE Transactions on Cybernetics, 52*(9), 9059–9075. https://doi.org/10.1109/TCYB.2021.3053165

Bojer, C. S., & Meldgaard, J. P. (2020). Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting*. https://doi.org/10.1016/j.ijforecast.2020.07.007

Box, G. E., & Tiao, G. C. (2011). *Bayesian Inference in Statistical Analysis*. Wiley.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140. https://doi.org/10.1007/BF00058655

Brown, G. (2004). Diversity in neural network ensembles. PhD thesis, Citeseer.

Brown, G., Wyatt, J., Harris, R., & Yao, X. (2005). Diversity creation methods: A survey and categorisation. *Information Fusion, 6*(1), 5–20. https://doi.org/10.1016/j.inffus.2004.04.004

Brown, G., Wyatt, J. L., Tino, P., & Bengio, Y. (2005). Managing diversity in regression ensembles. *Journal of machine learning research, 6*(9), 1621–1950.

Cano, A., & Krawczyk, B. (2020). Kappa updated ensemble for drifting data stream mining. *Machine Learning, 109*(1), 175–218. https://doi.org/10.1007/s10994-019-05840-z

Carpio, R. R., Taira, D. P., Ribeiro, L. D., Viera, B. F., Teixeira, A. F., Campos, M. M., Secchi, A. R., et al. (2021). Short-term oil production global optimization with operational constraints: A comparative study of nonlinear and piecewise linear formulations. *Journal of Petroleum Science and Engineering, 198*, 108141. https://doi.org/10.1016/j.petrol.2020.108141

Chandra, A., & Yao, X. (2006). Evolving hybrid ensembles of learning machines for better generalisation. *Neurocomputing, 69*(7–9), 686–700. https://doi.org/10.1016/j.neucom.2005.12.014

Chen, H., & Yao, X. (2009). Regularized negative correlation learning for neural network ensembles. *IEEE Transactions on Neural Networks, 20*(12), 1962–1979. https://doi.org/10.1109/TNN.2009.2034144

Chicco, D. (2017). Ten quick tips for machine learning in computational biology. *BioData Mining, 10*(1), 1–17. https://doi.org/10.1186/s13040-017-0155-3

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research, 7*, 1–30.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A., Vapnik, V., et al. (1997). Support vector regression machines. *Advances in Neural Information Processing Systems, 9*, 155–161.

Dutta, H. (2009). Measuring diversity in regression ensembles. *In IICAI, 9*, 17.

Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. *In Icml, 96*, 148–156.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics, 29*, 1189–1232.

Geisser, S. (1975). The predictive sample reuse method with applications. *Journal of the American Statistical Association, 70*(350), 320–328. https://doi.org/10.1080/01621459.1975.10479865

Hadavandi, E., Shahrabi, J., & Shamshirband, S. (2015). A novel boosted-neural network ensemble for modeling multi-target regression problems. *Engineering Applications of Artificial Intelligence, 45*, 204–219. https://doi.org/10.1016/j.engappai.2015.06.022

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence, 12*(10), 993–1001. https://doi.org/10.1109/34.58871

Ho, T.K. (1995). Random decision forests. In: Proceedings of 3rd International Conference on Document Analysis and Recognition, vol. 1, pp. 278–282. https://doi.org/10.1109/ICDAR.1995.598994. IEEE.

Hoch, T. (2015). An ensemble learning approach for the kaggle taxi travel time prediction challenge. In: DC@ PKDD/ECML.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics, 12*(1), 55–67. https://doi.org/10.2307/1267351

Jain, P., Kakade, S.M., Kidambi, R., Netrapalli, P., & Sidford, A. (2018). Accelerating stochastic gradient descent for least squares regression. In: Conference On Learning Theory, pp. 545–604. PMLR

LeBlanc, M., & Tibshirani, R. (1996). Combining estimates in regression and classification. *Journal of the American Statistical Association, 91*(436), 1641–1650. https://doi.org/10.1080/01621459.1996.10476733

Liu, Y., & Yao, X. (1999). Ensemble learning via negative correlation. *Neural Networks, 12*(10), 1399–1404. https://doi.org/10.1016/S0893-6080(99)00073-8

Liu, Y., Yao, X., & Higuchi, T. (2000). Evolutionary ensembles with negative correlation learning. *IEEE Transactions on Evolutionary Computation, 4*(4), 380–387. https://doi.org/10.1109/4235.887237

Mendes-Moreira, J., Soares, C., Jorge, A. M., & Sousa, J. F. D. (2012). Ensemble approaches for regression: A survey. *ACM Computing Surveys (CSUR), 45*(1), 1–40. https://doi.org/10.1145/2379776.2379786

Merz, C. J. (1999). Using correspondence analysis to combine classifiers. *Machine Learning, 36*(1), 33–58. https://doi.org/10.1023/A:1007559205422

Merz, C. J., & Pazzani, M. J. (1999). A principal components approach to combining regression estimates. *Machine Learning, 36*(1), 9–32. https://doi.org/10.1023/A:1007507221352

Peng, T., Zhang, C., Zhou, J., & Nazir, M. S. (2020). Negative correlation learning-based relm ensemble model integrated with ovmd for multi-step ahead wind speed forecasting. *Renewable Energy*. https://doi.org/10.1016/j.renene.2020.03.168

Perrone, M.P., & Cooper, L.N. (1992). When networks disagree: Ensemble methods for hybrid neural networks. Technical report, Brown Univ Providence Ri Inst for Brain and Neural Systems. https://doi.org/10.1142/9789812795885_0025

Pulsipher, J. L., Zhang, W., Hongisto, T. J., & Zavala, V. M. (2022). A unifying modeling abstraction for infinite-dimensional optimization. *Computers & Chemical Engineering, 156*, 107567.

Qi, C., & Tang, X. (2018). A hybrid ensemble method for improved prediction of slope stability. *International Journal for Numerical and Analytical Methods in Geomechanics, 42*(15), 1823–1839. https://doi.org/10.1002/nag.2834

Reeve, H. W., & Brown, G. (2018). Diversity and degrees of freedom in regression ensembles. *Neurocomputing, 298*, 55–68. https://doi.org/10.1016/j.neucom.2017.12.066

Rosenblatt, F. (1961). Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Lab Inc Buffalo NY. https://doi.org/10.1007/978-3-642-70911-1_20

Salgado, R.M., Pereira, J.J., Ohishi, T., Ballini, R., Lima, C., & Von Zuben, F.J. (2006). A hybrid ensemble model applied to the short-term load forecasting problem. In: The 2006 IEEE International Joint Conference on Neural Network Proceedings, pp. 2627–2634. https://doi.org/10.1109/IJCNN.2006.247141. IEEE

Simmons, C. R., Arment, J. R., Powell, K. M., & Hedengren, J. D. (2019). Proactive energy optimization in residential buildings with weather and market forecasts. *Processes, 7*(12), 929. https://doi.org/10.3390/pr7120929

Sirovetnukul, R., Chutima, P., Wattanapornprom, W., & Chongstitvatana, P. (2011). The effectiveness of hybrid negative correlation learning in evolutionary algorithm for combinatorial optimization problems. In: 2011 IEEE International Conference on Industrial Engineering and Engineering Management, pp. 476–481. https://doi.org/10.1109/IEEM.2011.6117963. IEEE.

Solomatine, D.P., & Shrestha, D.L. (2004). Adaboost. rt: a boosting algorithm for regression problems. In: 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541), vol. 2, pp. 1163–1168. https://doi.org/10.1109/IJCNN.2004.1380102. IEEE

Stigler, S. M. (1974). Gergonne's 1815 paper on the design and analysis of polynomial regression experiments. *Historia Mathematica, 1*(4), 431–439. https://doi.org/10.1016/0315-0860(74)90033-0

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological), 36*(2), 111–133. https://doi.org/10.1111/j.2517-6161.1974.tb00994.x

Taieb, S. B., & Hyndman, R. J. (2014). A gradient boosting approach to the kaggle load forecasting competition. *International Journal of Forecasting, 30*(2), 382–394. https://doi.org/10.1016/j.ijforecast.2013.07.005

Tang, K., Lin, M., Minku, F. L., & Yao, X. (2009). Selective negative correlation learning approach to incremental learning. *Neurocomputing, 72*(13–15), 2796–2805. https://doi.org/10.1016/j.neucom.2008.09.022

Ting, K. M., Wells, J. R., Tan, S. C., Teng, S. W., & Webb, G. I. (2011). Feature-subspace aggregating: Ensembles for stable and unstable learners. *Machine Learning, 82*(3), 375–397. https://doi.org/10.1007/s10994-010-5224-5

Todorovski, L., & Džeroski, S. (2003). Combining classifiers with meta decision trees. *Machine learning, 50*(3), 223–249. https://doi.org/10.1023/A:1021709817809

Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming, 106*(1), 25–57. https://doi.org/10.1007/s10107-004-0559-y

Webb, I., & Zheng, Z. (2004). Multistrategy ensemble learning: Reducing error by combining ensemble learning techniques. *IEEE Transactions on Knowledge and Data Engineering, 16*(8), 980–991. https://doi.org/10.1109/TKDE.2004.29

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks, 5*(2), 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Philip, S. Y., et al. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems, 14*(1), 1–37. https://doi.org/10.1007/s10115-007-0114-2

Zhao, Q.L., Jiang, Y.H., & Xu, M. (2010). Incremental learning by heterogeneous bagging ensemble. In: International Conference on Advanced Data Mining and Applications, pp. 1–12 . https://doi.org/10.1007/978-3-642-17313-4_1. Springer

Zou, K. H., Tuncali, K., & Silverman, S. G. (2003). Correlation and simple linear regression. *Radiology, 227*(3), 617–628. https://doi.org/10.1148/radiol.2273011499

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.