Letter: On non-iterative learning algorithms with closed-form solution

Article *in* Applied Soft Computing · July 2018

DOI: 10.1016/j.asoc.2018.07.013

Ponnuthurai N. Suganthan
Nanyang Technological University

Letter: On Non-iterative Learning Algorithms with Closed-form Solution

Ponnuthurai Nagaratnam Suganthan
School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore
email: epnsugan@ntu.edu.sg

July 9, 2018

Abstract

This letter discusses non-iterative learning methods with closed-form solution such as the kernel ridge regression and randomization based single hidden layer feedforward neural networks like random vector functional link (RVFL). Similarities and differences between these methods are also discussed. Irrelevance of kernel-trick for randomized neural networks is explained. The need for dual formulation or constrained optimization formulation for kernel methods and RVFL is distinguished. Finally, the articles in this special issue focusing on non-iterative learning methods with closed-form solution are summarized. A common conclusion in these articles is that the RVFL developed in the early 1990s outperforms the extreme learning machines (ELM). This conclusion is consistent with the earlier findings [1, 2, 3] that the direct links enhance the performance of the RVFL.

Keywords: Kernel Ridge Regression, Random Vector Functional Link, Extreme Learning Machines, Randomized Neural Networks, Non-iterative Learning.

1 Introduction

There are several motivations for writing this letter. During the review process of the special issue[1] some authors exhibited confusions between kernel methods, neural networks, kernel functions, activation functions, kernel tricks and so on. In addition, authors also attempted to propose new names for existing methods, for example, constrained optimization formulation or dual formulation for RVFL in order to develop kernel RVFL. In fact, the final solution of kernel RVFL is identical to the kernel ridge regression (KRR) solution developed in the late 1990s. Hence, it was felt that these issues can be documented for the benefits of the research community. Another motivation is to summarize the special issue articles with closed form solutions as they are closely related and several of these articles also compare methods such as RVFL, ELM, KRR, kernel ELM, and so on.

The major variation between RVFL and ELM is the presence and absence of direct connections between inputs and outputs, respectively. For the first time, the importance of direct connections between inputs and outputs was demonstrated experimentally in [1, 2, 3]. Some of the special issue articles [4, 5, 6, 7] also compared RVFL and ELM and arrived at the same conclusion.

This letter is organized as follows. In the next two sections original KRR and RVFL are introduced. Subsequently, issues such as kernel functions, activation functions, kernel tricks, dual formulation (i.e. constrained optimization formulation) are discussed. Finally the special issue articles dealing with these issues are summarized.

2 Kernel Ridge Regression (KRR)

Ridge regression (RR) is a popular machine learning algorithm based on least squares. A ridge or regularization term is added to the least squares learning objective to mitigate the over-fitting (i.e. the

[1] Applied Soft Computing journal's special issue on 'Non-iterative Approaches in Learning' guest edited by Dr P. N. Suganthan, Prof. Sushmita Mitra and Dr Ivan Tyukin with September 2016 as the paper submission deadline.

model-complexity). The ridge regression is formulated as:

$$\min_{w}\left[\sum_{i}(x_i w - y_i)^2 + \lambda\|w\|^2\right] \tag{1}$$

where $\mathbf{X} = [x_1^T, x_2^T, \ldots, x_n^T]^T$ is an $n \times d$ data matrix, $Y = [y_1, y_2, \ldots, y_n]^T$ is an $n \times 1$ output vector and w is a $d \times 1$ weight vector in the single output configuration. The regularization parameter λ is generally determined through cross-validation. The above learning objective has a closed form solution given by:

$$w = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T Y \tag{2}$$

where $\mathbf{I}$ is an identity matrix. Saunders et al. [8] proposed to enhance the ridge regression to perform non-linear regression by applying the kernel trick. According to the Representer Theorem, one can express the solution of w as a linear combination of the samples in the feature space $\phi(x)$ as $w = \sum_i \alpha_i \phi(x_i)$. The learning objective is then rewritten as:

$$\min_{\alpha} \|Y - \mathbf{K}\alpha\|^2 + \lambda\alpha^T\mathbf{K}\alpha \tag{3}$$

Similarly, its closed form solution is:

$$\alpha = (\mathbf{K} + \lambda\mathbf{I})^{-1}Y \tag{4}$$

where $\mathbf{K}$ is a kernel matrix and $\mathbf{K}_{ij} = k(x_i, x_j) = \langle\phi(x_i), \phi(x_j)\rangle$. Thus, instead of explicitly computing the coordinates of the samples in feature space through $\phi(x)$, one can simply use the kernel trick with kernel functions such as Gaussian kernel, linear kernel, polynomial kernel and so on.

The Kernel Ridge Regression (KRR) can be used to solve classification problems by defining the output vector Y with 0-1 coding [9]:

$$Y_{ij} = \begin{cases} 1 & \text{if } i^{th} \text{ sample belongs to } j^{th}\text{class} \\ 0 & \text{otherwise} \end{cases}$$

Multiple outputs would be required for multi-class classification with 0-1 coding. KRR has demonstrated highly competitive performances in a recent extensive benchmarking studies [10].

3 Random Vector Functional Link Network (RVFL)

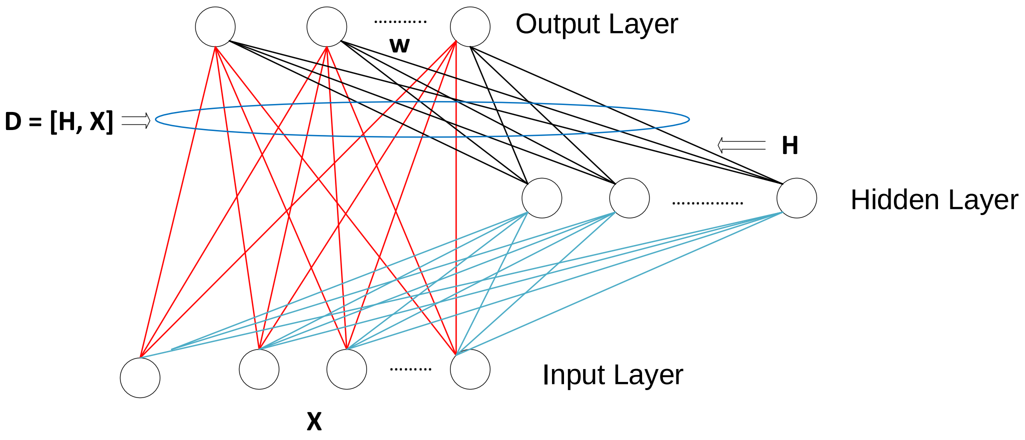

Figure 1: The structure of RVFL

Neural networks are commonly used for function approximation. These networks are usually trained by minimizing a loss function while the errors are back-propagated to determine the suitable network parameters by using gradient-based methods. The problems with back-propagation based optimization methods are slow training process, the solution may not converge to a single global minimum if there exists many local minima and it is also sensitive to learning rate setting. Such issues can be circumvented by randomization based methods, for example, randomly fixing the network configurations such as the connections or some parts of the network parameters, or randomly corrupting the input data or the parameters during the training [11, 12, 13, 14, 15, 16, 17]. Among these, the Random Vector Functional Link Network (RVFL) [11] shown in Fig. 1 performs better [1, 2].

RVFL is a single layer feed-forward neural network in which weights and biases of the hidden neurons are randomly generated within a suitable range. The inputs $\mathbf{D}$ of output neurons in RVFL consist of non-linearly transformed features $\mathbf{H}$ from hidden layer neurons and original input features $\mathbf{X}$ directly from input layer. Suppose that the input data has d features and there are J hidden neurons, there are total $d+J$ inputs for each output node. The direct links greatly improve the performance of RVFL networks [1, 2]. Since the hidden layer parameters are randomly generated and kept fixed, the learning objective is reduced to computing output weights, w, only. The learning objective is as follows:

$$\min_{w} \|\mathbf{D}w - Y\|^2 + \lambda \|w\|^2 \tag{5}$$

A closed form solution can be obtained by using either regularized least squares (i.e. $\lambda \neq 0$) or Moore-Penrose pseudoinverse (i.e. $\lambda = 0$). Using Moore-Penrose pseudoinverse, the solution is given by: $w = \mathbf{D}^{+}Y$ while using the regularized least squares (or ridge regression), the closed form solution is given by:

$$\text{Primal Space:} \quad w = (\mathbf{D}^T\mathbf{D} + \lambda\mathbf{I})^{-1}\mathbf{D}^T Y \tag{6}$$

We can transform the above primal space solution to the dual space solution as follows:

$$(\mathbf{D}^T\mathbf{D} + \lambda\mathbf{I})\mathbf{D}^T = \mathbf{D}^T(\mathbf{D}\mathbf{D}^T + \lambda\mathbf{I}) \tag{7}$$

$$(\mathbf{D}^T\mathbf{D} + \lambda\mathbf{I})^{-1}(\mathbf{D}^T\mathbf{D} + \lambda\mathbf{I})\mathbf{D}^T(\mathbf{D}\mathbf{D}^T + \lambda\mathbf{I})^{-1}Y = (\mathbf{D}^T\mathbf{D} + \lambda\mathbf{I})^{-1}\mathbf{D}^T(\mathbf{D}\mathbf{D}^T + \lambda\mathbf{I})(\mathbf{D}\mathbf{D}^T + \lambda\mathbf{I})^{-1}Y \tag{8}$$

$$\mathbf{D}^T(\mathbf{D}\mathbf{D}^T + \lambda\mathbf{I})^{-1}Y = (\mathbf{D}^T\mathbf{D} + \lambda\mathbf{I})^{-1}\mathbf{D}^T Y \tag{9}$$

$$\mathbf{D}^T(\mathbf{D}\mathbf{D}^T + \lambda\mathbf{I})^{-1}Y = \quad w \tag{10}$$

Hence, the output can be expressed as follows for the case of dual space:

$$\text{Dual Space:} \quad w = \mathbf{D}^T(\mathbf{D}\mathbf{D}^T + \lambda\mathbf{I})^{-1}Y \tag{11}$$

Depending on the number of training samples or total feature dimensions (i.e. input features plus total number of hidden neurons), dual or primal solution can be used to reduce the complexity of the matrix inversion.

4 Kernel Function Versus Activation Function

The function $k(x_i, x_j)$ is known as an inner-product kernel or simply a kernel. It computes the inner product of the images under an embedding ϕ of two data points. One can thus, create a Kernel or Gram matrix $\mathbf{K}$ given a kernel and a training set. A valid $\mathbf{K}$ is positive semi-definite ($v^T\mathbf{K}v \geq 0$) implying all its eigenvalues are positive and it is symmetric $\mathbf{K}_{ij} = \mathbf{K}_{ji}$. Hence, a valid kernel is also called Mercer kernel. The Kernel matrix contains all the information required for the learning procedure except the target or class labels. One can simply transform the input data into a higher dimensional feature space using kernel trick to potentially better discriminate the data samples.

An activation function maps a linearly weighted summation of inputs to an output in a neural network. These bounded functions, $f(\cdot)$, are generally used in the hidden and output layers of neural networks to mimic the firing of biological neurons. Some of the desired properties of activation functions are: non-linearity (without which a multi-layer network can simply be represented by a single layer network), continuity and smoothness throughout the range of their argument (so that it is possible to employ for gradient based optimization methods) and monotonic, if applicable [18].

Both $\mathbf{K}$ and $f(\cdot)$ project the original features into some transformed features which are generally obscure for human interpretation. There is no specific rule for the choice of a suitable kernel function or activation function implying that domain-knowledge and trial-and-error procedures may be used to select and tune these functions. However, kernel functions possess symmetry and positive semi-definiteness properties while activation functions are not required to satisfy these properties. Due to this important difference, it would be inappropriate and confusing to state for example,"kernel function of RVFL neural networks", as in some of the submissions to this special issue.

5 Kernel Trick for Neural Networks

One can compare a classification algorithm using kernels such as Support Vector Machines (SVM) and a single-hidden layer neural network using activation functions. Both these methods transform the original input data into some feature space $\mathcal{Z}$ and then perform classification in that space. We can also explain this by elucidating the tenets of SVM and neural networks. In SVM, if the data is not linearly separable in the input space, non-linear separation in the input space can be obtained by projecting the data to higher dimensional feature space. Based on the formulation of SVM, this can easily be performed by applying the kernel trick (inner product in feature space) without having to explicitly represent each data in feature space thereby avoiding colossal computational requirements. Neural network learns based on the feature representations provided by the hidden layer and it can approximate any function as long as there are sufficiently many hidden units. The hidden layers of the neural networks carry complex representations of the input that eventually helps the output layer. Thus, both the kernel function and hidden layers with nonlinear activation functions assist the classification process by projecting the data into some space to make the classification easier.

In the closed form solution of RVFL (Eq. 11), there exists inner product of the stacked features matrix $\mathbf{D}$. Thus, one may be eager to employ a kernel trick to obtain the inner product in some space. The $\mathbf{D}$ matrix consists of transformed (enhanced/hidden) features together with original features, meaning $\mathbf{D}$ already projects the input data into some space, and a learning via least squares is able to obtain impressive performance in variety of tasks such as classification, time-series forecasting, etc. Thus, it is meaningless to project the already transformed features in some space to another space using the kernel trick. According to Universal Approximation Theorem, simply using sufficient nonlinear hidden neurons facilitate neural networks to learn complex functions under some conditions [19, 20]. This holds true for RVFL with and without direct links. However, replacing $\mathbf{D}\mathbf{D}^T$ in Eq. 11 with a kernel matrix and $\mathbf{D}^T$ with inner products between a test data sample and the training dataset is equivalent to Kernel Ridge Regression (KRR) discussed in Section 2. It is possible for researchers to *propose* an alternative novel derivation for KRR starting from RVFL. However, this approach is meaningless as the structure of RVFL, its hidden neurons, its randomized weights, etc. are all discarded instantly. Importantly, the neat KRR derivation has been known since 1998 [8] and this derivation is similar to widely accepted kernel PCA and kernel LDA derivations [21, 22]. Hence, for the reasons presented in this section, a *novel* kernel RVFL proposal offering an additional name for KRR was rejected during the review process. However, researchers interested in learning relationships between infinitely wide neural networks, Gaussian processes and kernel methods can refer to [23].

6 Constrained Optimization Formulation

In the formulation of RR in Eq. 1, the λ parameter is used to regularize the model complexity. Eq. 1 is also known as $L2$ norm regularized least square problem as λ scales the impact of the squared Euclidean norm $\|w\|^2$ in the overall objective function.

The dual formulation is required for SVM with inequality constraints to apply the kernel trick. The optimization problem is then solved via Lagrangian formulation. Following Vapnik's SVM formulation, Saunders et. al [8] presented the dual formulation for RR with equality constraints to extend the linear regression to non-linear regression in the kernel feature space. Below, we present the formulations of both the SVM and ridge regression (RR) in the dual space:
SVM:

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{subject to} \quad & y_i(w \cdot x_i + b) \geq 1 - \xi_i, \;\; i = 1, 2, \ldots, N \\ & \xi_i \geq 0 \end{aligned} \tag{12}$$

Ridge Regression in Dual Space:

$$\begin{aligned} &\text{Minimize} \quad \lambda\|w\|^2 + \sum_{i=1}^{N} {\xi_i}^2 \\ &\text{subject to} \quad y_i - w \cdot x_i = \xi_i, \;\; i = 1, 2, \ldots, N \end{aligned} \tag{13}$$

The optimization formulation of RVFL (Eq. 5) can also be represented in the dual space as in Eq. 13 as a *novel* constrained optimization based RVFL. Such a reformulation is required to derive a kernel RVFL implementation. But, as we explained in Section 5, a kernel realization for RVFL will be simply renaming the KRR [8] as the kernel RVFL. Due to this reason and the fact that the dual and primal solutions are related according to Eqns (7)-(10), the *novel* constrained optimization formulation (i.e. the dual formulation) for RVFL was rejected during the review process. For researchers interested in investigating randomization strategies in the context of kernel methods, an overview article by Scardapane and Wang [24] summarizes the literature elegantly.

7 An Overview of Special Issue Articles

This section presents overview of special issue articles closely related to RVFL and KRR in two subsections.

7.1 RVFL Related Articles

Vuković et. al [4] presents the Orthogonal Polynomial Expanded Random Vector Functional Link Neural Network (OPE-RVFLNN) that utilizes advantages from nonlinear expansion of the input vector and randomization of the input weights. Through comprehensive experimental evaluation by using 30 UCI regression datasets and nonparametric statistical hypothesis testing, the paper shows results for four orthogonal polynomials (Chebyshev, Hermite, Laguerre and Legendre) and three activation functions (tansig, logsig and tribas). The research reveals that tansig and Chebyshev expansion outperform their counterparts. The research results demonstrate that direct links between the input and the output layer are crucial for improved performance of the network, and ridge regression is better than Moore-Penrose solution. These observations agrees with the conclusions in [1, 2, 3]. There is no significant difference in accuracy between iterative or batch learning approaches. Although computational complexity of this network is expected to be higher than RVFL in learning phase, it shows impressive on-line data processing ability.

Tang et. al [5] propose an ensemble empirical mode decomposition (EEMD) based ensemble model without an iterative training process. This work uses the efficient, fast non-iterative algorithm called random vector functional link (RVFL) network with randomly fixed weights and direct input-output links, as the individual predictor. With crude oil price as the test case, the proposed EEMD-based RVFL network performs significantly better in terms of prediction accuracy than not only prevailing single algorithms (such as RVFL network, extreme learning machine, kernel ridge regression, back propagation neural network, least square support vector regression, and autoregressive integrated moving average), but also their respective EEMD-based ensemble variants. As for speed, RVFL network runs the fastest, faster than ELM [25] too. The proposed EEMD-based RVFL network outperforms all the ensemble methods and even most of the single methods in computation time.

Henriquez et. al [6] contribute to one of the main challenges of single layer feedforward neural networks, which is the selection of the optimal number of neurons in the hidden layer. In particular, an efficient and fast non-iterative method for pruning hidden nodes in neural networks with random weights is presented. Exhaustive experimental evaluations in regression and classification problems show that the combination of the pruning techniques with these types of neural networks improve their predictive performance in terms of mean square error and accuracy without increasing significantly the training time. The pruning technique is also tested with neural networks trained under sequential learning algorithms, where Random Vector Functional Link obtained, in general, the best predictive performance compared to online sequential versions of extreme learning machines and single hidden layer neural network with random weights.

Dash et. al [26] investigates the accuracy in prediction of Indian Summer Monsoon Rainfall (ISMR). ISMR is a major scientific challenge and in this context there is a need to explore applicability of different soft computing techniques. This study investigates the application of Single Layer Feed-forward Neural network (SLFN) in ISMR forecast. This study shows improved performance of SLFN with Levenberg-Marquardt (LM) algorithm and radial basis (radbas) activation function. Random Vector Functional Link neural network (RVFL) and Regularized Online Sequential-RVFL (ROS-RVFL) are also applied for ISMR forecasts and their relative performance with respect to SLFN is compared. It is observed that ROS-RVFL is more accurate and computationally more efficient than SLFN and RVFL. For more accurate ISMR forecasts, length of the training period is found to be an influential parameter. ROS-RVFL with ensemble mean of 8- and 9-year training periods is found to be appropriate.

Mesquita et. al [7] propose a selective ensemble of Randomization based Neural Networks (RNNs) using the Successive Projections Algorithm (SPA) for regression problems. The proposed method, named Selective Ensemble of RNNs using the successive projections algorithm (SERS), employs the SPA for three distinct tasks: feature selection, pruning and ensemble selection. The framework was used to develop three selective ensemble models based on the three RNNs: Extreme Learning Machines (ELM), Feedforward Neural Network with Random Weights (FNNRW) and Random Vector Functional Link (RVFL). The performances of SERS-ELM, SERS-FNNRW and SERS-RVFL were assessed in terms of model accuracy and model complexity on several real world benchmark problems. Comparisons to related methods showed that SERS variants achieved similar accuracies with significant model complexity reduction. Among the proposed models, SERS-RVFL had the best accuracies and all variants resulted in models with similar complexity.

Katuwal et. al [27] present a new ensemble of classifiers that consists of decision trees and a fast-neural network, namely random vector functional link network (RVFL) for multi-class classification. The RVFL partitions the original training dataset into K distinct subsets, where K is the number of classes in a data set, and a decision tree is induced for each subset. The proposed method provides a rich insight into the data by grouping the confusing or hard to classify samples for each class and thereby providing an opportunity to employ fine-grained classification rule over the data. This method is particularly suitable for multi-core or distributed environment where after the partitioning by RVFL, each partition can be run in parallel or distributed across different cores. Both univariate and multivariate (oblique) decision trees are used with RVFL. The performance of the proposed method is evaluated on 65 multi-class UCI datasets. The results demonstrate that the classification accuracy of the proposed ensemble method is significantly better than other state-of-the-art classifiers for data sets with over 500 training samples.

7.2 KRR Related Articles

Zhang et. al [28][2] address the challenges of integrating the multi-modal contents of social images simultaneously for classification, because the textual content and visual content are represented in two heterogeneous feature spaces. To address this problem, authors propose a classifier with multi-modal kernel ridge regression (MMKRR) to capture the nonlinear structure of social image features and the correlation between different types of features. Two kernel ridge regression classifiers are learned for the visual features and text features, and a joint learning model is used to reinforce the learning of the two classifiers. An optimization method is proposed to solve the objective function of MMKRR. Then, two combination methods are proposed to obtain the final classification result by combining the classification results based on two kinds of features. A set of experiments are conducted on several social image datasets to demonstrate the superiority of the approach. Experiment results indicate that the proposed approach consistently outperforms the early fusion approaches and late fusion approaches.

Naik et. al [29] presents a hybrid method based on empirical mode decomposition (EMD) with non-iterative kernel ridge regression for accurate short-term predictions of wind speed and wind power using real world data sets over time intervals varying from 10 min to 3 hours ahead. Both wind speed and wind power historical time series data are decomposed using Empirical mode decomposition (EMD) into several intrinsic mode functions (IMFs) and residues which are subsequently used as input vectors to KRR, and its variants. Among the different KRR variants, the wavelet kernel based KRR (EMD-WKRR) exhibits superior wind speed and wind power forecasting performance with the least error metrics and the highest correlation coefficient. Besides its forecasting accuracy is higher than the other non-iterative prediction models like the EMD based Random Vector Functional Link Network (EMD-RVFL), and Extreme Learning Machine (EMD-ELM) prediction models, which are presented in this paper for comparison. From numerical experimentation with several wind farms data, it is also observed that EMD-RVFL exhibits a better short-term wind speed and wind power prediction in comparison to EMD-ELM, and other KRR variants, except the EMD-WKRR prediction model. However, the prediction performance of EMD-RVFL can be improved to an extent closer to that of EMD-WKRR by optimizing the

[2] This article was published in Volume 67. https://www.sciencedirect.com/science/article/pii/S1568494618300899

random weights between the input and hidden layers using a hybrid Firefly-Harmony search algorithm. Further, to reduce the relatively higher execution time of EMD-WKRR, dimensional reduction of kernel matrix is achieved by using a limited set of random support vectors from the training data with a slight loss of prediction accuracy.

8 Conclusions

This letter clarifies confusions encountered during the review process among kernel methods, neural networks, kernel functions, activation functions, kernel tricks, dual formulation/constrained formulation and so on. Further, a novel kernel RVFL derivation is shown to be identical to the KRR derived in the late 1990s. Finally, this letter summarizes the special issue articles with closed form solutions as they are closely related and several of them compare methods such as RVFL, ELM, KRR, kernel ELM, and so on. A common conclusion in these articles is that the RVFL developed in the early 1990s outperforms the ELM developed in mid 2004 by removing the direct links and bias in the RVFL.

References

[1] Y. Ren, P. N. Suganthan, N. Srikanth, and G. Amaratunga, "Random vector functional link network for short-term electricity load demand forecasting," *Information Sciences*, vol. 367, pp. 1078–1093, 2016.

[2] L. Zhang and P. Suganthan, "A comprehensive evaluation of random vector functional link networks," *Information Sciences*, vol. 367, pp. 1094 – 1105, 2016.

[3] L. Zhang and P. N. Suganthan, "Visual tracking with convolutional random vector functional link network," *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3243–3253, 2017.

[4] N. Vuković, M. Petrović, and Z. Miljković, "A comprehensive experimental evaluation of orthogonal polynomial expanded random vector functional link neural networks for regression," *Applied Soft Computing*, 2018.

[5] L. Tang, Y. Wu, and L. Yu, "A non-iterative decomposition-ensemble learning paradigm using rvfl network for crude oil price forecasting," *Applied Soft Computing*, 2018.

[6] P. Henriquez and G. Ruz, "A non-iterative method for pruning hidden neurons in neural networks with random weights," *Applied Soft Computing*, 2018.

[7] D. P. P. Mesquita, J. P. P. Gomes, L. R. Rodrigues, S. A. F. Oliveira, and R. K. H. Galvao, "Building selective ensembles of randomization based neural networks with the successive projections algorithm," *Applied Soft Computing*, 2018.

[8] C. Saunders, A. Gammerman, and V. Vovk, "Ridge regression learning algorithm in dual variables," in *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, (San Francisco, CA, USA), pp. 515–521, Morgan Kaufmann Publishers Inc., 1998.

[9] S. An, W. Liu, and S. Venkatesh, "Face recognition using kernel ridge regression," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–7, June 2007.

[10] L. Zhang and P. Suganthan, "Benchmarking ensemble classifiers with novel co-trained kernal ridge regression and random vector functional link ensembles," *IEEE Computational Intelligence Magazine*, vol. 12(4), pp. 61–72, 2017.

[11] Y. H. Pao and Y. Takefuji, "Functional-link net computing: theory, system architecture, and functionalities," *IEEE Computer*, vol. 25, pp. 76–79, May 1992.

[12] W. F. Schmidt, M. A. Kraaijveld, and R. P. Duin, "Feedforward neural networks with random weights," in *Pattern Recognition, 1992. Vol. II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on*, pp. 1–4, IEEE, 1992.

[13] V. S. G. Te Braake, Hubert A.B. Te, "Random activation weight neural net (rawn) for fast non-iterative training," *Engineering Applications of Artificial Intelligence*, vol. 8, no. 1, pp. 71–80, 1995.

[14] P. Guo, C. Chen, and Y. Sun, "An exact supervised learning for a three-layer supervised neural network," in *Proceedings of the International Conference on neural Information Processing (ICONIP'95)*, pp. 1041–1044, 1995.

[15] P. Guo, “A vest of the pseudoinverse learning algorithm,” in *arXiv, https://arxiv.org/pdf/1805.07828*, pp. 1–5, 2018.

[16] H. Berry and M. Quoy, “Structure and dynamics of random recurrent neural networks,” *Adaptive Behavior*, vol. 14, no. 2, pp. 129–137, 2006.

[17] L. Zhang and P. Suganthan, “A survey of randomized algorithms for training neural networks,” *Information Sciences*, vol. 364, pp. 146–155, Oct, 2016.

[18] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification.* (Second Edition), Wiley-Interscience, 2000.

[19] G. Cybenko, “Approximations by superpositions of sigmoidal functions,” *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[20] K. Hornik, “Approximation capabilities of multilayer feedforward networks,” *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[21] B. Schölkopf, A. Smola, and K.-R. Müller, “Kernel principal component analysis,” in *Artificial Neural Networks — ICANN’97* (W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, eds.), (Berlin, Heidelberg), pp. 583–588, Springer Berlin Heidelberg, 1997.

[22] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers, “Fisher discriminant analysis with kernels,” in *Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (Cat. No.98TH8468)*, pp. 41–48, Aug 1999.

[23] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, “Deep neural networks as gaussian processes,” in *Sixth Int. Conf. on Learning Representations*, pp. 1–10, 2018.

[24] S. Scardapane and D. Wang, “Randomness in neural networks: an overview,” *WIREs Data Mining Knowledge Discovery*, vol. 7, 2017.

[25] L. Wang and C. Wan, “Comments on “the extreme learning machine”,” *IEEE Transactions on Neural Networks*, vol. 19, no. 8, pp. 1494–1495, 2007.

[26] Y. Dash, S. K. Mishra, S. Sahany, and B. K. Panigrahi, “Indian summer monsoon rainfall prediction: A comparison of iterative and non-iterative approaches,” *Applied Soft Computing*, 2018.

[27] R. Katuwal, P. N. Suganthan, and L. Zhang, “An ensemble of decision trees with random vector functional link networks for multi-class classification,” *Applied Soft Computing*, 2018.

[28] X. Zhang, W. Chao, C. Liu, Z. Li, and R. Li, “Multi-modal kernel ridge regression for social image classification,” *Applied Soft Computing*, 2018.

[29] J. Naik, P. Satapathy, and P. K. Dash, “Short-term wind speed and wind power prediction using hybrid empirical mode decomposition and kernel ridge regression,” *Applied Soft Computing*, 2018.