ARTICLE TEMPLATE

A nonparametric optimal neural regression tree model for water quality improvement in a paper manufacturing industry

Tanujit Chakraborty[a] and Zubia Mansoor[b] and Ashis Kumar Chakraborty[a]

[a]Indian Statistical Institute, Kolkata, India
[b]St. Xavier's College, Kolkata, India

ARTICLE HISTORY
Compiled October 22, 2018

ABSTRACT
This work is motivated by a specific problem of a modern paper manufacturing industry, in which boiler inlet water quality is a major concern. Boiler helps to produce power and steam which is used to cook wood chips (along with chemicals) to be supplied to a paper machine. If water treatment plant can't produce water of desired quality as specified by the boiler, then it results in poor health of the boiler water tube and consequently affects the quality of the paper. Variation in inlet water quality of the boiler is due to several crucial process parameters. We formulate this problem into a typical regression problem that involves finding crucial process parameters from the set of available suspected causal variables. Then we build a novel nonparametric hybrid model to solve this regression problem and call it as optimal neural regression tree (NRT) model with the aim to solve the issue of water quality. The major advantage of the proposed optimal NRT model is that it has very less tuning parameters and is easily interpretable as compared to "black-box-like" complex machine learning models. Statistical properties including optimal values of the important model parameters of the proposed model are proved in this paper. We have also experimentally assessed the performance of the proposed model in comparison with the other state-of-the-art models.

KEYWORDS
Water quality; Demineralization process; Regression Tree; Neural networks; Optimal neural regression tree

1. Introduction

This work is motivated by a particular problem in a modern paper industry which produces papers for multiple uses. Paper machines produce papers by using pulp, fibers, fillers, chemical lubricants and a huge amount of water. Boiler produces steam for power generation purposes and also helps to make pulp for paper production. The steam produced in the boiler is used to cook wood chips (along with the cooking chemicals). Steam is also sent to dryer cans to remove water from the sheet produced which the drainage, vacuum, and mechanical pressing sections of the paper machine can't accomplish. The steam from the boilers is also used throughout the mill in heat exchanging, steam-traced piping, stock chests, etc. The boiler stipulates the desired level of water quality to be received from the water treatment plant (WTP). In

Corresponding author: Tanujit Chakraborty. Email: tanujit_r@isical.ac.in

WTP, the process of demineralization (DM Process) is used for removal of dissolved solids by the ion exchange process (IEP). IEP involves two stages of demineralization (Batchelder 1965). In the first stage, it removes cations from water by the cation exchange process and then removes anions from water by the anion exchange process in the second stage (Lhassani et al. 2001). DM process outlet pH happens to be the key performance indicator (KPI) of WTP. It was found that WTP can't produce water of desired quality specified by the boiler. By controlling the water quality, we will be able to address the problem of variation in DM outlet water pH as well as improve boiler water tube health and finally make improvement in the paper manufacturing process. An extensive preliminary analysis was conducted to determine a set of possible causal variables that happen to be the key to water pH level variations. The aim of this paper is to determine the optimal levels of important process parameters to maintain water pH level within the desired standard as specified by the boiler. For this, we would develop a regression model which will also be useful for future prediction of water pH based on the important causal parameters. Controlling water pH will necessarily improve the water quality of the boiler which will be economical for the company as well. The formulation of the above-stated problem along with data collection plan is described in Section 2.

Statistical and machine learning tools like factor analysis, SPC techniques, decision trees, and artificial neural networks, etc have been applied to solve several problems in analyzing water quality of river (Singh et al. 2009; Ouyang et al. 2006; Mahuli, Rhinehart, and Riggs 1993) and surface water planning (Gmar et al. 2017; Bhattacharya and Solomatine 2005). Multivariate statistical tools and pattern recognition methods are found to be effective to capture the temporal and spatial variations in river water quality (Singh et al. 2004; Alberto et al. 2001) and manufacturing process efficiency (Chakraborty, Chakraborty, and Chattopadhyay 2018). It was found that decision trees and neural networks (NN) have the ability to model arbitrary decision boundaries. Regression tree (RT) is found more robust when limited data are available, unlike NN. But decision trees are high variance estimators and greedy in nature (Breiman 2017), whereas neural networks are popular for tackling prediction problems. Advanced neural networks have many free tuning parameters and hence there is a risk of over-fitting in case of small and medium sized data sets. To utilize the positive aspects of these two powerful models, theoretical frameworks for combining both classifiers are often used jointly to make decisions. Mapping tree-based models to neural networks allows exploiting the former to initialize the latter. The idea of mapping tree based models into NN are presented by several papers to solve many supervised learning problems (Sethi 1990; Sirat and Nadal 1990). Similar idea in the interface area of decision trees and neural networks can be found in some recent literature as well (Chen et al. 2005; Balestriero 2017). But the major disadvantages of these algorithms are having many free tuning parameters and thus poor robustness. In spite of the use of neural tree models in practical problems of classification, regression, and forecasting, very little is known about the statistical properties of these models. In this paper, we obtained the upper-bound of neural regression tree model parameters and designed an optimal NRT model to solve the regression task of water quality problem in the paper manufacturing system.

Motivated by the above discussion, we have proposed an optimal NRT model with optimal values of the model parameters. Harnessing the ensemble formulation, we try to exploit the strengths of RT and NN models and overcome their drawbacks. The

proposed model has the advantages of significant accuracy, very less number of tuning parameters and easy interpretability as compared to more "black-box-like" advanced neural networks. Our model is very useful for small or medium sized complex data structures, like in the current water quality problem in paper manufacturing industries. The theoretical bounds for model parameters are proposed. We practically recommend using the highest value of the model parameters as the optimal values when working with small and medium-sized data sets to achieve 'best' performance of the model and this justifies the name of the model as "optimal NRT model". Finally, the robustness of the proposed algorithm is shown through experimental evaluation.

The paper is structured as follows. Section 2 describes the motivating problem and its formulation. In Section 3, we introduce optimal neural regression tree model along with the upper bound of its parameters. Section 4 analyzes the data by the proposed method and compares it with other state-of-the-art. Finally, the concluding remarks are given in Section 5.

2. Motivating example

In this section, we describe the motivating real-life problem and the data collection plan for our study. The data on pH is taken on a regular basis from the Boiler lab. However, when the study was taken up, we considered only the most recent 12 months data. Measurement of water pH level at Boiler lab is done through a digital meter which is calibrated once a month using a standard solution. Hence, the normal gage R&R analysis was not necessary to be carried out to ensure correctness of the values observed using laboratory analysis. DM process outlet pH happens to be the KPI of water treatment plant which gives water of desired quality specified by the boiler. Water treatment plant comprises of two units: fresh water treatment and condensate water treatment. Freshwater treatment plant involves filtration of Freshwater obtained from the tube well followed by treatment of water with chemicals, making the water suitable for Boiler and Turbine. Condensate treatment involves filtration and treatment of condensate water obtained from Paper machines. Quality of condensate coming from Paper machines is the main constraint of the current problem. Once the filtration of water/condensate is done, it is sent to Demineralization plant where water is made suitable to be used by the boiler. Details of the types of equipment used in DM plant processes are given (Vedelago and Millar 2018) along with the process flow diagram in Figure 1.

- Strong acid cation (SAC) exchanger extracts the cations from the water by the cation resin and converts it into mineral acids. A vessel is provided therewith manually operated valves for controlling the flow rates of the inlet and outlet water.
- Strong base anion (SBA) exchanger is an anion exchanger in which mild-steel a vessel is internally lined with rubber to prevent corrosion. An external vessel is also provided with manually operated valves for controlling the flow rates of the inlet and outlet water.
- The mixed bed (MB) unit comprises of a mild steel rubber-lined pressure vessel. Externally, the unit is provided with piping and valves to control the flow of water during service and various regeneration stages.

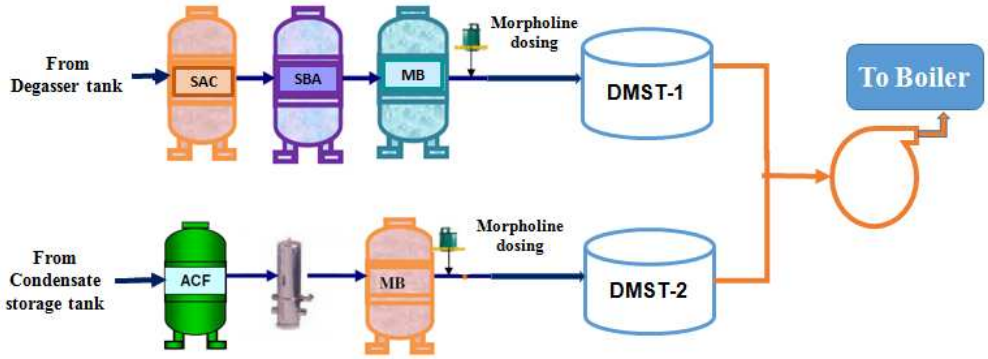

Figure 1. Process Flow Diagram of DM Plant Process

- Activated carbon filters (ACF) are used for the removal of chlorine and organic matter from water. Normally, the activated carbon filters are placed downstream of multi-grade filters and need to be regularly back-washed by reversal of flow so as to keep the surface of the carbon particles clean.

Once the water is treated, it is stored in a tank called DM water storage tank (DMST). The water is pumped from DMST to Boiler via DM Transfer Pump. Chemicals like HCl and Caustic Soda are used for regeneration of the resins in SAC, SBA and MB Exchanger. Morpholine is used for scaling the pH of DM Water. Condensate water is also sent through Mixed Bed exchanger for treatment of water. Transferring DM water of pH in the range of 8.5-9.2 is essential to maintain drum water quality and the health of the water tubes running through Boiler. It was found that the variation in pH of DM water is large and maintaining proper pH level is essential to maintain boiler water quality. The original problem can be considered to be a two-stage problem: 1) developing a prediction model for outlet water pH of DM process; 2) using the model and further actions to reduce variations in pH of DM outlet water and maintain it close to the specified target (i.e., mid value of $8.5 - 9.2$). While developing a prediction model for outlet water pH of DM process, one needs to identify the important parameters of the process which together will be able to explain a significant amount of variation present in the system. This work will also have an impact on the amount of chemical usage, the health of the boiler tube and quality of DM outlet water.

3. Model

In this section, we will describe how a pre-trained RT can be reformulated as a two-layered NN with similar types of predictive behavior. Neural networks and tree-based models are performing well in many real-life engineering problems. The idea of combining decision trees with neural networks was discussed in previous literature but mainly for performing classification tasks (Sethi 1990; Sirat and Nadal 1990; Foresti and Micheloni 2002; Chen, Yang, and Abraham 2007; Chakraborty, Chattopadhyay, and Chakraborty 2018). We extend the approach for regression problems and try to bridge the gap between theory and practice by finding the upper bounds for the tuning parameters of the model. We present an example of a mapping from tree based model to a two-hidden layered neural network in Figure 2.

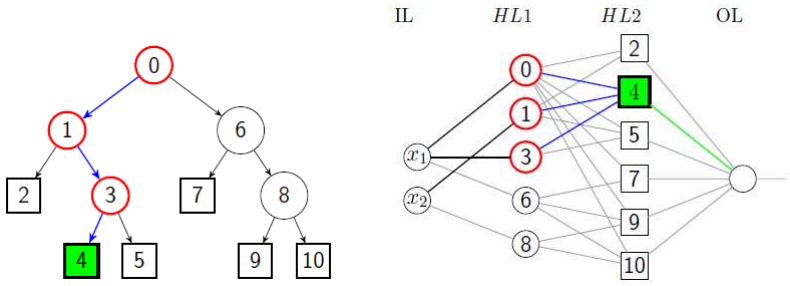

Figure 2. An example of neural regression tree model.

A regression tree (left) and its corresponding NN structure (right). The circle nodes in the tree belong to split nodes and square nodes to leaf nodes. The path to the green shaded leaf (4) consists of all red nodes (0, 1, 3). Numbers in neurons correspond to numbers in tree model nodes. The highlighted connections in the network are those relevant for the activity of the green neuron and its output value.

We are given a training sample $D_n = \{(\underline{X}_1, Y_1), (\underline{X}_2, Y_2), ..., (\underline{X}_n, Y_n)\}$ with n observations on p independent variables. Neural regression tree (NRT) is a nonparametric regression model in which p input features $\underline{X} \in C^p = [0,1]^p$ are observed and we try to predict a square integrable output function $Y \in \mathbb{R}$. A regression tree is a regression function estimate that uses a hierarchical axes-parallel split of the input space, where each tree node corresponds to one of the segmentation subsets in C^p. For simplicity and easy interpretability, let us consider ordinary binary RT where a node has exactly two child nodes or zero child nodes (leaf nodes). A regression tree consists of split nodes (for example, $x^{(i)} \geq \alpha$ for some $i \in \{1, 2, ..., p\}$ and some $\alpha \in C$) and leaf nodes. The feature space C^p is partitioned into axes-parallel hyper-rectangles. The standard splitting criteria MSE (mean squared error) has been used here to create RT. During prediction, the input is first passed into the tree root node. It is then iteratively transmitted to the leaf node that belongs to the subspace in which the input is located; this is repeated until a leaf node is reached. If a leaf represents region R, then the natural regression function estimate takes the simple form $t_n(x) = (\sum_{i=1}^{n} Y_i \mathbb{1}_{X_i \in R})/N_n(R)$, $x \in R$, where $N_n(R)$ is the number of observations in cell R (where, by convention, $0/0 = 0$). In other words, the prediction for a query point x in leaf node R is the average of the Y_i of all training instances that fall into this region R. Let us assume for now that we have at hand a regression tree t_n (whose construction eventually depends upon the data D_n), which takes constant values on each of $k \geq 2$ terminal nodes. It turns out that this estimate may be reinterpreted as two hidden layers neural networks, as summarized here. Let $HL1 = \{H_1, \ldots, H_{k-1}\}$ be the collection of all hyperplanes participating in the construction of t_n. We note that each $H_{k'} \in HL1$ is of the form $H_{k'} = \{x \in C^p : h_{k'}(x) = 0\}$, where $h_{k'}(x) = x^{(i_{k'})} - \alpha_{i_{k'}}$ for some (eventually data-dependent) $i_{k'} \in \{1, 2, ..., p\}$ and $\alpha_{i_{k'}} \in C$. To reach the leaf of the query point $\mathbf{x}$, we try to find, for each hyperplane $H_{k'}$, the side on which x falls ($+1$ for right and -1 for left). Using the above notations, the tree estimate t_n is mapped to the neural network as discussed below. Also, an example is presented in Figure 2 with the description for a clear understanding of the neural regression tree model.

Designing the first hidden layer ($HL1$). The input layer (IL) supplies the features to the first hidden layer of neurons which corresponds to $k-1$ perceptrons, with the threshold activation function defined as $\tau(h_{k'}(x)) = \tau(x^{(i_{k'})} - \alpha_{i_{k'}})$, where $\tau(u) = 2\mathbb{1}_{u\geq 0} - 1$. So, for each split in the tree, there is a neuron in $HL1$ whose activity encodes the relative position of an input x with respect to the concerned split. The output of the first layer are ± 1-vector $(\tau(h_1(x)), \ldots, \tau(h_{k-1}(x)))$, which describes all decisions of the inner tree nodes (it also includes the nodes off the tree path of x). Note that $\tau(h_{k'}(x))$ is $+1$ if x is on one side of the hyperplane $H_{k'}$, -1 if x is on the other side of $H_{k'}$, and by convention $+1$ if $x \in H_{k'}$. It is important to remember that each neuron $k^{'}$ of this layer is connected to one and only one input $x^{(i_{k'})}$, and that the connection has weight 1 and offset $-\alpha_{i_{k'}}$.

Designing the second hidden layer ($HL2$). $HL1$ outputs a $(k-1)$-dimensional vector of ± 1-bits that encodes the precise position of x in the leaves of the tree. The leaf node identity of x can be extracted from the above-mentioned vector using a weighted combination of the bits along with an appropriate threshold function. The second hidden layer has k neurons, one for each leaf, and assigns a terminal cell to x. Let $HL2 = \{L_1, \ldots, L_k\}$ be the collection of all tree leaves, and let $L(x)$ be the leaf containing x. We connect a unit $k^{'}$ from $HL1$ to a unit k'' from $HL2$ iff the hyperplane $H_{k'}$ is involved in the sequence of splits forming the path from the root to the leaf $L_{k''}$. The connection has weight $+1$ if the split by $H_{k'}$ is from a node to a right child in that path, and -1 otherwise. Suppose we have $(u_1(x), \ldots, u_{k-1}(x))$ as the vector of ± 1-bits seen at the output of $HL1$, then the output $v_{k''}(x) \in \{-1, 1\}$ of neuron k'' is $\tau(\sum_{k'\to k''} b_{k',k''} u_{k'}(x) + b^0_{k''})$, where notation $k' \to k''$ means that k' is connected to k'' and $b_{k',k''} = \pm 1$ is the corresponding weight. The offset $b^0_{k''}$ is set to

$$b^0_{k''} = -l(k'') + 1/2, \tag{1}$$

where $l(k'')$ is the length of the path from the root to $L_{k''}$. To understand the intuition behind the choice (1), we can see that there are exactly $l(k'')$ connections starting from the first layer and pointing to k'', and that

$$\begin{cases} \sum_{k'\to k''} b_{k',k''} u_{k'}(x) - l(k'') + 1/2 = 1/2 & \text{if } x \in L_{k''} \\ \sum_{k'\to k''} b_{k',k''} u_{k'}(x) - l(k'') + 1/2 \leq -1/2 & \text{if } x \notin L_{k''} \end{cases} \tag{2}$$

Using (1), the argument of the threshold function is $1/2$ if $x \in L_{k''}$ and is smaller than $-1/2$ otherwise. Hence $v_{k''}(x) = 1$ iff the terminal cell of x is $L_{k''}$. To summarize, $HL2$ outputs a vector of ± 1-bits $(v_1(x), \ldots, v_k(x))$ whose components equal -1 except the one corresponding to the leaf $L(x)$, which is $+1$.

Output layer (OL). Let $(v_1(x), \ldots, v_k(x))$ be the output of the $HL2$. If $v_{k''}(x) = 1$, then the output layer computes the average $\bar{Y}_{k''}$ of the Y_i corresponding to X_i falling in $L_{k''}$. This is equivalent to taking

$$t_n(x) = \sum_{k''=1}^{k} w_{k''} v_{k''}(x) + b_{\text{out}}, \tag{3}$$

where $w_{k''} = \bar{Y}_{k''}/2$ for all $k'' \in \{1, \ldots, k\}$, and $b_{\text{out}} = \frac{1}{2}\sum_{k''=1}^{k} \bar{Y}_{k''}$.

3.1. *Optimal Neural Regression Tree*

We have already described how a given RT can be mapped to a two hidden layered (2HL) NN model. Now we are going to critically summarize the functionality of the neural regression tree (NRT) model (see Figure 2) as follows:

- An RT is built having $(k_n - 1)$ split nodes and k_n leaf nodes which is further mapped into a 2HL NN model having $(k_n - 1)$ and k_n hidden neurons in $HL1$ and $HL2$ respectively. Since the value of k depends on n, so we are going to use k_n instead of k in the rest of the paper.
- In $HL1$, the neurons compute all the tree split decisions and indicate the split directions for the inputs. Further, $HL1$ passes the information to $HL2$.
- The probabilistic interpretation of the network output can be obtained by interpreting the activation functions in the HL.

The tree estimate t_n, depending on D_n, can be seen as a neural network estimate. The architecture of this network is fixed, and so are the weights and offsets of the three layers. A natural idea is then to keep the structure of the network intact and let the parameters vary in a subsequent network training procedure with backpropagation training. In other words, once the connections between the neurons have been designed by the tree-to-network mapping, we can learn network parameters in a better way by minimizing the empirical MSE for this network over the sample D_n.

We propose an optimal NRT model in which we theoretically find the upper bound of these free parameters which are nothing but the optimal values of the parameters for small and medium sample sized data sets. In order to increase the generalization capabilities of the NRT model, we replace the original relay-type activation function $\tau(u)$ with a hyperbolic tangent activation function $\sigma(u) := \tanh(u)$ which has a chosen range from -1 to 1. More precisely, we use in optimal NRT model, $\sigma_1(u) = \sigma(\beta_1 u)$ at every neuron of the first hidden layer and $\sigma_2(u) = \sigma(\beta_2 u)$ at every neuron of the second hidden layer. Here, β_1 and β_2 are positive hyper-parameters that determine the contrast of the hyperbolic tangent activation: larger the parameters β_1 and β_2, the sharper is the transition from -1 to 1. As β_1 and β_2 approach to infinity, the continuous functions σ_1 and σ_2 converge to the threshold function. We call our model optimal NRT in the sense that the optimal values (upper bounds for this case) of the tuning parameters k_n, β_1 and β_2 are derived (see Section 3.2). Besides eventually providing better generalization, the hyperbolic tangent activation functions favor smoother decision boundaries and permit a relaxation of crisp tree node membership. Lastly, they allow operations with a smooth approximation of the discontinuous step activation function. This makes the loss function of the network differentiable with respect to the parameters almost everywhere, and gradients can be backpropagated to train the model.

3.2. *Upper bound of model parameters*

Let us consider the tree in the ensemble and denote by $\mathscr{G}_1 \equiv \mathscr{G}_1(D_n)$, the bipartite graph which models the connections between the input vectors $x = (x^{(1)}, \ldots, x^{(p)})$ and the $k_n - 1$ hidden neurons of $HL1$. Similarly, let $\mathscr{G}_2 \equiv \mathscr{G}_2(D_n)$ be the bipartite graph represents the connections between the first layer and the k_n hidden neurons of $HL2$. Let $M(\mathscr{G}_1)$ be the set of $p \times (k_n - 1)$ matrices $A = (a_{ij})$ such that $a_{ij} = 0$ if $(i, j) \notin \mathscr{G}_1$. Also let $M(\mathscr{G}_2)$ be the $(k_n - 1) \times k_n$ matrices $B = (b_{ij})$ such that $b_{ij} = 0$

if $(i,j) \notin \mathscr{G}_2$. The parameters that specify the first hidden units are contained in a matrix A of $M(\mathscr{G}_1)$ identified by the weights over the edges of $\mathscr{G}_1$ and a column vector of biases b_1 and it is of size $k_n - 1$. Similarly, the parameters of the second hidden units are represented by a matrix B of $M(\mathscr{G}_2)$ of weights over $\mathscr{G}_2$ and by a column vector b_2 of offsets having size k_n. Let us take the output weights and offset to be $W_{\text{out}} = (w_1, \ldots, w_{k_n})^\top \in \mathbb{R}^{k_n}$ and $b_{\text{out}} \in \mathbb{R}$, respectively. And the parameters that specify the NRT model are represented by a "vector" as shown below:

$$\lambda = (A, b_1, B, b_2, W_{\text{out}}, b_{\text{out}}) \in M(\mathscr{G}_1) \times \mathbb{R}^{k_n - 1} \times M(\mathscr{G}_2) \times \mathbb{R}^{k_n} \times \mathbb{R}^{k_n} \times \mathbb{R}.$$

We further assume that there exists a positive constant c_1 such that

$$\|B\|_\infty + \|b_2\|_\infty + \|W_{\text{out}}\|_1 + |b_{\text{out}}| \leq c_1 k_n, \tag{4}$$

where $\|\cdot\|_\infty$ is the supremum norm of a matrix and $\|\cdot\|_1$ is the L_1-norm of a vector. The rationale behind this assumption (4) is that the weights and offsets are taken by the computation units of the second layer and the output layer. We note that this condition is satisfied by the original random tree estimates as soon as Y is assumed to be bounded. Finally, we can assume that the absolute value of $\|Y\|_\infty \leq L < \infty$ almost surely, for some L.

Therefore, letting $\Lambda_n = \big\{\lambda = (A, b_1, B, b_2, W_{\text{out}}, b_{\text{out}}) : (4) \text{ is satisfied}\big\}$, we see that the neural network implements functions of this particular form

$$f_\lambda(x) = W_{\text{out}}^\top \sigma_2\Big(B^\top \sigma_1(A^\top x + b_1) + b_2\Big) + b_{\text{out}}, \quad x \in \mathbb{R}^p,$$

where $\lambda \in \Lambda_n$. Our aim is to tune the parameters λ using the data D_n such that the function realized by the obtained network becomes a 'good' estimate that can minimize the empirical error. Let $\mathscr{F}_n = \big\{f_\lambda : \lambda \in \Lambda_n\big\}$, where $\mathscr{F}_n$ be the class of neural networks and m_n be the network that minimizes the empirical L_2 error which is defined as

$$J_n(f) = \frac{1}{n} \sum_{i=1}^{n} \big|Y_i - f(X_i)\big|^2$$

over all functions $f \in \mathscr{F}_n$, i.e., $J_n(m_n) \leq J_n(f)$, where $\mathscr{F}_n$ is a rich class of functions, including additive functions, polynomial functions having coefficients of the same sign, products of continuous functions and etc.
Thus, we will try to find out an estimate $m_n : C^p \to \mathbb{R}$ of the regression function $m(\underline{X}) = E[Y|X = x]$. We say m_n is consistent if $E[m_n(X) - m(X)]^2$ tends to 0 as $n \to \infty$ (the expectation is evaluated over $\underline{X}$ and the training sample D_n). We can write using (Györfi et al. 2006, Lemma 10.1)

$$\begin{aligned}\mathbb{E}\big|m_n(X) - m(X)\big|^2 \leq 2\mathbb{E}\Bigg[\sup_{f \in \mathscr{F}_n)} \Bigg|\frac{1}{n}\sum_{i=1}^{n} \big|Y_i - f(\mathbf{X}_i)\big|^2 - \mathbb{E}\big|Y - f(\mathbf{X})\big|^2\Bigg|\Bigg] \\ + \mathbb{E}\Bigg[\inf_{f \in \mathscr{F}_n} \int_{C^d} \big|f(x) - m(x)\big|^2 \mu(\mathrm{d}\mathbf{x})\Bigg],\end{aligned} \tag{5}$$

where μ denotes the distribution of $\underline{X}$. To find the upper-bound of the model parameters, we need to show the **estimation error** (first term in the R.H.S. of Eqn. 5) and **approximation error** (second term in the R.H.S. of Eqn. 5) tend to 0. The former can be proved using non-asymptotic uniform deviation inequalities and covering numbers corresponding to $\mathscr{F}_n$, as shown in Theorem 3.1. Approximation error can be handled using a pseudo-estimate similar to RT generated t_n and application of Lipschitz property on the activation function of the NRT model, as shown in Theorem 3.2. Throughout the paper, we assumed X is uniformly distributed in C^p and $\|Y\|_\infty \leq L < \infty$ almost surely, for some L.

The next two theorems state that with certain restrictions imposed on the number k_n of terminal nodes and with the parameters β_1, β_2 being properly regulated as functions of n, the empirical L_2 risk-minimization provides upper bounds for the NRT model parameters. Larger the value of k_n, β_1 and β_2, the better is the model. That is why, we call the upper bound of the model parameters as the optimal values of NRT model. It is important to note that the depth of the tree and β_1, β_2 are controlled for risk-minimization.

Theorem 3.1. *If k_n, β_2 satisfy $k_n, \beta_2 \to \infty$ such that $k_n = o(n^{1/6})$ and $\beta_2 = o(\log(n))$, then estimation error converges to 0 as $n \to \infty$.*

Proof. For proof, refer to Appendix 1. □

Theorem 3.2. *If k_n, β_1 satisfy $k_n, \beta_1 and \beta_2 \to \infty$ such that $k_n = o(e^{\beta_2}), k_n = o(\beta_1^{1/4})$ and $\beta_1 = o(n^{2/3})$, then approximation error converges to 0 as $n \to \infty$.*

Proof. For proof, refer to Appendix 2. □

Hence, the optimal choice of parameters obtained using empirical risk minimization for optimal NRT model are: (a) $k_n = o(n^{1/6})$ (b) $\beta_1 = o(n^{2/3})$ and (c) $\beta_2 = o(\log(n))$ for practical use in regression problems having small and medium-sized datasets.

4. Application

4.1. *Data*

To identify the causal parameters causing the variations in the DM water outlet pH, at first a brainstorming session was held with the process experts. Since not all the parameters are controllable by the users of the processes, a list of controllable parameters is prepared and accordingly data are collected for the two processes (as shown in Figure 1). The data set collected from the process over a year consists of the observations from the following causal variables: Inlet Flow, Water Pressure (water inlet pressure to the exchanger), Air Pressure, MB stroke and Amount of Morpholine/Chemical dosing (Liter per hr). The values of the response variable (DM water pH) varies between 7 to 10 (refer to Figure 3 for a graphical summary). Sample datasets for DMST-1 and DMST-2 are given in Tables 1 and 2. These data sets will be used for finding a prediction model that can help the company to forecast future water pH level and take further steps for the reduction in water pH variation.

Table 1. Sample data set for DMST-1

Sl. No.	Inlet Flow	Water Pressure	Air Pressure	MB stroke	Amount of chemical	DM water outlet pH
1	1980	5.8	5.0	70	9.96	9.276
2	2150	7.0	7.0	60	8.69	9.094
3	1780	6.0	5.0	45	7.73	8.594
4	2808	5.2	6.4	50	6.54	8.738
5	1590	6.2	5.7	40	5.56	8.592
6	2995	6.0	6.0	50	7.23	9.099
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

Table 2. Sample data set for DMST-2

Sl. No.	Inlet Flow	Water Pressure	Air Pressure	MB stroke	Amount of chemical	DM water outlet pH
1	1489	5.4	6.0	80	10.04	9.246
2	1139	5.8	5.0	65	10.66	9.033
3	1448	5.4	5.0	45	7.69	8.483
4	1703	6.2	5.8	40	7.54	8.594
5	1258	6.2	5.7	35	5.99	7.589
6	1139	6.0	5.2	50	6.97	9.021
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

4.2. *Performance metrics*

The metrics used in this study to evaluate the performance of different regression models (including the proposed model) are Root mean square error (RMSE), Mean absolute percentage error (MAPE), Co-efficient of multiple determination (R^2) and Adjusted R^2 ($AdjR^2$):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2};\ MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \widehat{y}_i}{y_i}\right|;$$

$$R^2 = 1 - \left[\frac{\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}\right];\ AdjR^2 = 1 - \left[\frac{(1-R^2)(n-1)}{n-k-1}\right];$$

where, $y_i, \overline{y}, \widehat{y}_i$ denote the actual value, average value and predicted value of the dependent variable, respectively for the i^{th} instant. Here n and k denote the number of data points and independent variables used for performance evaluation, respectively. The lower the value of RMSE and MAPE and the higher the value of R^2 and Adj R^2, the better is the model is.

4.3. *Preliminary data analysis*

We started the analysis of the data with Anderson-Darling normality test on DM water outlet pH and it confirms that the dependent variable doesn't follow normal distribution (see Figure 3). An absence of normality in the DM water outlet pH data actually removed the possibility of application of conventional parametric regression methods. This leads us to think about the nonparametric regression approaches.

The stability of the process was checked using $\overline{X} - R$ control chart (see Figure 4). From $\overline{X} - R$ chart, it can be easily concluded that DM water outlet pH has high variation and it also indicates the presence of some assignable causes in the process.

The dataset contains only numerical features and no missing entries, so no data cleaning task was performed. We only transformed the feature values into standardized form to achieve consistent results while using NN and NRT models. The standardization is done using the following equation:

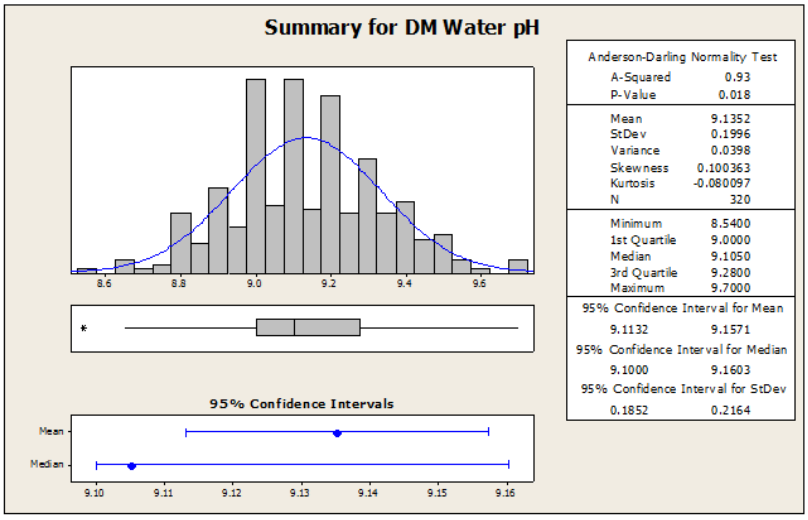

Figure 3. Summary statistics for DM water outlet pH

$X_i^{new} = \frac{X_i - min(X_i)}{max(X_i) - min(X_i)}$; where X_i^{new} is the standardized transformed value of the feature vector and lies within 0 and 1.

As the output of the networks will also lie between 0 and 1, "logsig" is used as output transfer function to bring it back in the original form at the end of the modeling by using,

$Y_i^{pred} = Z_i\{max(X_i) - min(X_i)\} + min(X_i)$; here Z_i is the standardized output within 0 and 1, and Y_i^{pred} is the predicted output of the model.

4.4. *Results*

Since the data sets don't satisfy the basic assumptions of parametric regression models, we decided to use nonparametric regression models. We further shuffled the observations of the water pH dataset randomly and split it into training and testing data sets in a ratio of 70 : 30. Each experiment is repeated 5 times with different randomly assigned training and test sets and we will finally report the averages of the performance metrics observed over 5 times validations in Table 3. A few popular nonparametric regression models such as k-Nearest neighbor (kNN), support vector regression (SVR), Regression splines (B-splines), multiple adaptive regression spline (MARS), RT and NN with 2 hidden layers (NN with 2HL) were applied to the data sets and the results were recorded. Then we started experimenting with our proposed model. The training procedure for the optimal NRT model is as follows. RT is first built using the *scikit-learn* implementation (Pedregosa et al. 2011) for tree designing. From the tree, we extracted the set of all split directions and split positions and used them to build neural network initialization parameters. The NRT models are then trained using the *TensorFlow* framework (Abadi et al. 2016). The optimization with the network model is done by minimizing the MSE on the training set. It is achieved by employing an iterative gradient-descent optimization technique. We have used the default functions available in *TensorFlow* for this. Each neural network (including NRT) model was trained for 100 epochs. From the theoretical results on the upper bounds of the model parameters, we can easily find out the values of the initial contrast parameters of the activation functions of the first and second HL of the model. The value of β_2 will be

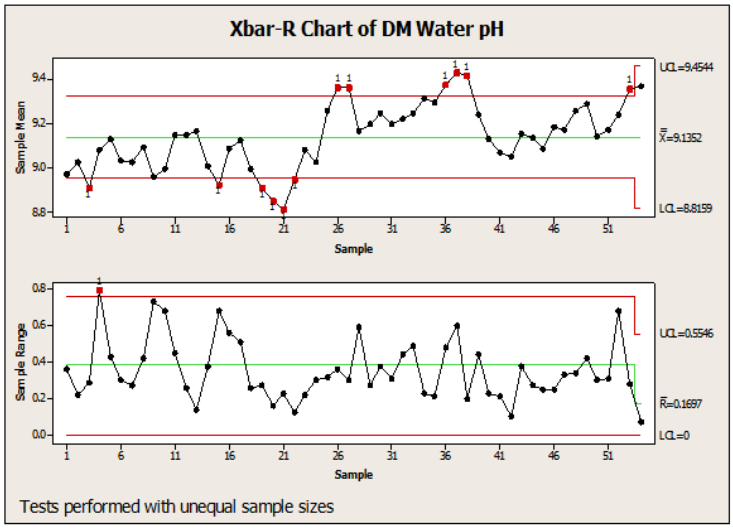

Figure 4. Control chart for DM water outlet pH

lower than β_1 which is confirmed from the theoretical bounds presented in Section 3.2. This is practically very much significant, since for a relatively small β_2, the transition in the activation function from -1 to $+1$ is smoother and a stronger gradient signal reaches the first hidden layer in backpropagation training. Similarly a converse explanation can be given for β_1. In all experiments, we have used $\beta_1 = o(n^{2/3})$ and $\beta_2 = o(logn)$, where n is the number of training samples. Another property of the data set is that it is small sized and here tree based models perform better as compared to NN, since the latter typically need plentiful training data to perform well. For two data sets given in Section 4.1 standard NN (with 2HL) do not achieve comparable performance as compared to proposed optimal NRT model. In a similar way neural decision trees (NDT) were also implemented to asses its performance on the water pH data sets. Since it has many free tuning parameters, the training time and memory requirement are quite high as compared to the proposed model. Our proposed optimal NRT model is faster, especially when trained on a GPU. In Table 3, we present the results of different regression model on the water pH data sets and best results are displayed in bold fonts. From the experimental evaluation of different regression models, it can be concluded that our proposed optimal NRT model outperforms the state-of-the-arts.

4.5. *Recommendation*

The proposed optimal NRT model will be useful for forecasting the future values of water pH given the values of the process variables. This will be helpful for the management and engineers to take preventive actions in the future. However, our objective of the work is not only to develop a competitive prediction model for water pH level in DM process but also to find out optimal level of process parameters (in other words, causal variables) using statistical modeling. To keep DM water pH in the range of $8.5 - 9.2$ as specified by the boiler manual, the recommendations of our model is to maintain MB strokes, water pressure and chemical consumption within a specified range as shown in Table 4. Inlet flow can't be controlled by the user of the process and Air pressure need not be controlled as far as the recommendation of the

Table 3. Quantitative measures of performance for different regression models on test data set (average values of the metrics after 5-fold cross validations)

Regression Models	Data set	RMSE	MAPE	R^2	Adj R^2
kNN	DMST 1	5.56	6.67	60.65	56.21
	DMST 2	6.44	7.95	56.70	50.83
SVR	DMST 1	4.56	5.50	68.56	62.18
	DMST 2	4.98	6.20	63.70	57.83
B-splines	DMST 1	4.32	5.40	69.85	63.30
	DMST 2	6.94	7.21	56.70	49.78
MARS	DMST 1	4.29	5.26	65.95	58.90
	DMST 2	6.74	7.93	57.53	47.05
RT	DMST 1	3.44	4.12	80.52	75.56
	DMST 2	3.89	4.78	76.56	71.23
NN with 2HL	DMST 1	3.86	4.80	76.95	70.03
	DMST 2	4.12	5.91	70.10	64.73
NDT	DMST 1	3.18	3.57	84.70	80.96
	DMST 2	3.40	3.92	82.18	78.14
Optimal NRT	DMST 1	**2.90**	**3.25**	**86.80**	**83.55**
	DMST 2	**3.15**	**3.65**	**84.56**	**82.10**

proposed model goes. Table 4 gives the optimum range of controllable parameters for DMST 1 and DMST 2 based on the recommendation of regression analysis performed by optimal NRT model.

Table 4. Optimal range of causal variables for achieving desired pH level

Process	Range for Water Pressure	Range for MB stroke	Range for chemical consumption	Expected range for DM water outlet pH
DMST 1	5.0-6.0	45-60	6.5-7.5	8.5-9.0
DMST 2	5.0-6.0	40-55	7.5-9.5	8.5-9.1

Table 4 depicts a range for all the important process variables for which the expected DM water outlet pH will be within the required specification, i.e., $8.5-9.2$. However, in order to find out the exact values of the process parameters within the range prescribed in Table 4, a design of experiment would be necessary, which was subsequently carried out to solve the problem. Though we have discussed only the proposed model and its accuracy level, our model also helped the manufacturing process industry to improve its water quality level and to gain monetary benefits due to a reduction in the chemical consumption.

5. Conclusions

This paper proposes a novel machine learning paradigm to solve a process efficiency problem in a paper manufacturing industry. The purpose of this article is to develop a model for reducing the variation of water pH level in the DM process in a manufacturing industry. Our study proposed an optimal NRT model that maps RT into a 2-HL NN model with the theoretically obtained upper bound of the parameters of the model. Consequently, the proposed ensemble model successfully demonstrated the best performance and offered a practical solution to the problem of finding optimal level for

the tuning parameters to improve water quality in the DM process. The major advantage of the proposed optimal NRT model is that it has very less tuning parameters, easily interpretable as compared to "black-box-like" advanced NN models and theoretically sound as compared to many similar types of algorithms. It was theoretically shown that there is some gain in considering 2HL in NN, but it is not really necessary to go beyond 2HL in NN (Devroye, Györfi, and Lugosi 2013). The proposed ensemble model will have an edge while working with limited data sets where the benefits of neural optimization can usually not be exploited, but with a specific inductive bias of the optimal NRT, it becomes possible. This also answers the question one may ask about the need for a two-step pipeline (like optimal NRT model) over advanced NN models.

Appendix 1. Proof of Theorem 3.1

The idea of the proof is based on the Theorem 16.1 of (Györfi et al. 2006). The set $\mathscr{F}_n$ contains all neural networks constrained by Eqn. (4) of Section 3.2 with inputs in $\mathbb{R}^p$, two hidden layers of respective size $k_n - 1$ and k_n, and one output unit. We note that $\mathscr{F}_n$ is deterministic, in the sense that it does not depend on D_n, but only on the size n of the original data set. We can derive a useful exponential inequality to prove the Theorem 3.1. using uniformly bounded classes of functions. We have assumed that for each $f \in \mathscr{F}_n$, f satisfies $\|f\|_\infty \leq c_1 k_n$ and Y is also bounded ($\|Y\|_\infty \leq L < \infty$).

Let $z_1^n = (z_1, \ldots, z_n)$ be a vector of n fixed points in $\mathbb{R}^p$ and let $\mathscr{H}$ be a set of functions from $\mathbb{R}^p \to \mathbb{R}$. For every $\varepsilon > 0$, we let $\mathscr{N}_1(\varepsilon, \mathscr{H}, z_1^n)$ be the L_1 ε-covering number of $\mathscr{H}$ with respect to $z_1, \ldots, z_n$. $\mathscr{N}_1(\varepsilon, \mathscr{H}, z_1^n)$ is defined as the smallest integer N such that there exist functions $h_1, \ldots, h_N : \mathbb{R}^p \to \mathbb{R}$ with the property that for every $h \in \mathscr{H}$, there is a $j = j(h) \in \{1, \ldots, N\}$ such that

$$\frac{1}{n} \sum_{i=1}^{n} \left| h(z_i) - h_j(z_i) \right| < \varepsilon.$$

Note that if $Z_1^n = (Z_1, \ldots, Z_n)$ is a sequence of i.i.d. random variables, then $\mathscr{N}_1(\varepsilon, \mathscr{H}, Z_1^n)$ is a random variable as well.

Now, let $Z = (X, Y)$, $Z_1 = (X_1, Y_1), \ldots, Z_n = (X_n, Y_n)$, and $C^p = [0, 1]^p$, we write

$$\mathscr{H}_n = \left\{ h(\mathbf{x}, y) := |y - f(x)|^2 : (x, y) \in C^p \times [-L, L] \text{ and } f \in \mathscr{F}_n \right\}.$$

The functions in $\mathscr{H}_n$ will satisfy the following: $0 \leq h(x, y) \leq 2c_1^2 k_n^2 + 2L^2 \leq 4c_1^2 k_n^2$. This assumption holds for large n such that $c_1 k_n \geq L$ is satisfied. Using Pollard's inequality

(Györfi et al. 2006), we have, for arbitrary $\varepsilon > 0$,

$$\begin{aligned}
&\mathbb{P}\Big\{ \sup_{f\in\mathscr{F}_n} \Big|\frac{1}{n}\sum_{i=1}^{n} \big|Y_i - f(X_i)\big|^2 - \mathbb{E}\big|Y - f(X)\big|^2\Big| > \varepsilon \Big\} \\
&\quad = \mathbb{P}\Big\{ \sup_{h\in\mathscr{H}_n} \Big|\frac{1}{n}\sum_{i=1}^{n} h(Z_i) - \mathbb{E}(h(Z))\Big| > \varepsilon \Big\} \\
&\quad \le 8\mathbb{E}\Big[\mathscr{N}_1\big(\frac{\varepsilon}{8}, \mathscr{H}_n, Z_1^n\big)\Big] \exp\Big(-\frac{n\varepsilon^2}{128(4c_1^2k_n^2)^2}\Big). \qquad (1)
\end{aligned}$$

Next we try to bound the covering number $\big(\mathscr{N}_1(\frac{\varepsilon}{8}, \mathscr{H}_n, Z_1^n)\big)$. Let us consider two functions $h_i(x,y) = |y - f_i(x)|^2$ of $\mathscr{H}_n$ for some $f_i \in \mathscr{F}_n$ and $i = 1, 2$. We get

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} \big|h_1(Z_i) - h_2(Z_i)\big| &= \frac{1}{n}\sum_{i=1}^{n} \Big|\big|Y_i - f_1(X_i)\big|^2 - \big|Y_i - f_2(X_i)\big|^2\Big| \\
&= \frac{1}{n}\sum_{i=1}^{n} \big|f_1(X_i) - f_2(X_i)\big| \times \big|f_1(X_i) - Y_i + f_2(X_i) - Y_i\big| \\
&\le \frac{4c_1k_n}{n}\sum_{i=1}^{n} \big|f_1(X_i) - f_2(X_i)\big|.
\end{aligned}$$

$$\text{Thus,} \qquad \mathscr{N}_1\Big(\frac{\varepsilon}{8}, \mathscr{H}_n, Z_1^n\Big) \le \mathscr{N}_1\Big(\frac{\varepsilon}{64c_1k_n}, \mathscr{F}_n, X_1^n\Big). \qquad (2)$$

The covering number $\mathscr{N}_1(\frac{\varepsilon}{64c_1k_n}, \mathscr{F}_n, X_1^n)$ can be upper bounded independently of X_1^n by extending the arguments of (Lugosi and Zeger 1995) from a network with one hidden layer to a network with two hidden layers. We will now apply Theorem 9.4, Lemma 16.4, and Lemma 16.5 repeatedly (Györfi et al. 2006). Let the neurons of the first hidden layer output belong to the class

$$G_1 = \{\sigma_1(a^\top x + a_0) : a \in \mathbb{R}^p, a_0 \in \mathbb{R}\},$$

and for $0 < \varepsilon < 1/4$,

$$\mathscr{N}_1(\varepsilon, G_1, X_1^n) \le 3^2\Big(\frac{4e}{\varepsilon}\log\frac{2\times 3e}{\varepsilon}\Big)^{p+2} = 9\Big(\frac{6e}{\varepsilon}\Big)^{4p+8}.$$

Next, letting $G_2 = \{bg : g \in G_1, b \in [-c_1k_n, c_1k_n]\}$ we get

$$\mathscr{N}_1(\varepsilon, G_2, X_1^n) \le \frac{4c_1k_n}{\varepsilon}\mathscr{N}_1\Big(\frac{\varepsilon}{2c_1k_n}, G_1, X_1^n\Big) \le \Big(\frac{36ec_1k_n}{\varepsilon}\Big)^{4p+9}.$$

The second units compute the functions of the collection

$$G_3 = \Big\{\sigma_2\Big(\sum_{i=1}^{k_n-1} g_i + b_0\Big) : g_i \in G_2, b_0 \in [-c_1k_n, c_1k_n]\Big\}.$$

Note that σ_2 satisfies the Lipschitz property $|\sigma_2(u)-\sigma_2(v)| \leq \beta_2|u-v|$ for all $(u,v) \in \mathbb{R}^2$. Thus,

$$\mathscr{N}_1(\varepsilon, G_3, X_1^n) \leq \frac{2c_1\beta_2 k_n^2}{\varepsilon}\mathscr{N}_1\Big(\frac{\varepsilon}{2\beta_2 k_n}, G_2, X_1^n\Big)^{k_n-1} \leq \Big(\frac{72ec_1\beta_2 k_n^2}{\varepsilon}\Big)^{(4p+9)k_n+1}.$$

Also, letting

$$G_4 = \{wg : g \in G_3, w \in [-c_1k_n, c_1k_n]\},$$

By assuming without loss of generality $c_1, \beta_2 \geq 1$, we get

$$\mathscr{N}_1(\varepsilon, G_4, X_1^n) \leq \frac{4c_1k_n}{\varepsilon}\mathscr{N}_1\Big(\frac{\varepsilon}{2c_1k_n}, G_3, X_1^n\Big) \leq \Big(\frac{144ec_1^2\beta_2 k_n^3}{\varepsilon}\Big)^{(4p+9)k_n+2}.$$

Finally, we can write

$$\mathscr{F}_n = \Big\{\sum_{i=1}^{k_n} g_i + b_{\text{out}} : g_i \in G_4, b_{\text{out}} \in [-c_1k_n, -c_1k_n]\Big\},$$

We conclude

$$\begin{aligned}\mathscr{N}_1(\varepsilon, \mathscr{F}_n, X_1^n) &\leq \frac{2c_1k_n(k_n+1)}{\varepsilon}\Big[\mathscr{N}_1\Big(\frac{\varepsilon}{k_n+1}, G_4, X_1^n\Big)\Big]^{k_n} \\ &\leq \Big(\frac{144ec_1^2\beta_2(k_n+1)^4}{\varepsilon}\Big)^{(4p+9)k_n^2+2k_n+1}. \qquad (3)\end{aligned}$$

Combining (1)-(3) together, we obtain

$$\begin{aligned}&\mathbb{P}\Big\{\sup_{f\in\mathscr{F}_n}\Big|\frac{1}{n}\sum_{i=1}^n\big|Y_i-f(X_i)\big|^2 - \mathbb{E}\big|Y-f(X)\big|^2\Big| > \varepsilon\Big\} \\ &\leq 8\Big(\frac{9216ec_1^3\beta_2(k_n+1)^5}{\varepsilon}\Big)^{(4p+9)k_n^2+2k_n+1}\exp\Big(-\frac{n\varepsilon^2}{2048c_1^4k_n^4}\Big).\end{aligned}$$

Now let $\widetilde{Z}$ be a non-negative random variable. Using this for any $0 < \varepsilon < 1/4$ and large n, we can write

$$\mathbb{E}\big[\widetilde{Z}\big] = \int_0^\infty \mathbb{P}[\widetilde{Z} > t]dt \leq \varepsilon + \int_\varepsilon^\infty \mathbb{P}[\widetilde{Z} > t]dt. \qquad (4)$$

Using (4) for any $0 < \varepsilon < 1/4$ and large n, we can write

$$\mathbb{E}\Big[\sup_{f\in\mathscr{F}_n}\Big|\frac{1}{n}\sum_{i=1}^{n}\big|Y_i - f(X_i)\big|^2 - \mathbb{E}\big|Y - f(X)\big|^2\Big|\Big]$$

$$\leq \varepsilon + 8\int_{\varepsilon}^{\infty}\Big(\frac{9216ec_1^3\beta_2(k_n+1)^5}{t}\Big)^{(4p+9)k_n^2+2k_n+1}\exp\Big(-\frac{nt^2}{2048c_1^4k_n^4}\Big)\,\mathrm{d}t$$

$$\leq \varepsilon + 8\Big(\frac{2048c_1^4k_n^4}{n\varepsilon}\Big)\exp\Big[\big((4p+9)k_n^2+2k_n+1\big)\log\Big(\frac{9216ec_1^3\beta_2(k_n+1)^5}{\varepsilon}\Big) - \frac{n\varepsilon^2}{2048c_1^4k_n^4}\Big]$$

$\to 2\varepsilon \quad (n\to\infty), \quad$ if the conditions of Theorem 3.1 holds.

As $\varepsilon \to 0$, we get the estimation error converges to 0 and the proof is complete.

Appendix 2. Proof of Theorem 3.2

To prove Theorem 3.2, let us consider a piecewise constant function (pseudo-estimate) similar to the RT-tree t_n, with only one difference: the function computes the true conditional expectation $\mathbb{E}[Y|X \in L_{k''}]$ in each leaf $L_{k''}$, but not the empirical one, viz. $\bar{Y}_{k''}$. In another way, we can write $(W^{\star}_{\text{out}})_{k''} = \mathbb{E}[Y|X \in L_{k''}]/2$ and $b^{\star}_{\text{out}} = \sum_{k''=1}^{k_n}\mathbb{E}[Y|X\in L_{k''}]/2$ in Eqn. (3) of Section 3. This tree-type pseudo-estimate has the form

$$t_{\lambda^\star}(x) = W^{\star\top}_{\text{out}}\tau\Big(B^{\star\top}\tau(A^{\star\top}x + b_1^{\star}) + b_2^{\star}\Big) + b^{\star}_{\text{out}}, \quad x\in\mathbb{R}^p,$$

for some $\lambda^\star = (A^\star, b_1^\star, B^\star, b_2^\star, W^\star_{\text{out}}, b^\star_{\text{out}})$. We have

$$\begin{aligned}
\mathbb{E}\Big[\inf_{f\in\mathscr{F}_n}\int_{C^p}\big|f(x)-m(x)\big|^2\mu(dx)\Big] &= \mathbb{E}\Big[\inf_{f\in\mathscr{F}_n}\int_{C^p}\big|f(x)-m(x)\big|^2\mu(dx)\Big] \\
&\quad -2\mathbb{E}\Big[\int_{C^p}\big|t_{\lambda^\star}(x)-m(x)\big|^2\mu(dx)\Big] + 2\mathbb{E}\Big[\int_{C^p}\big|t_{\lambda^\star}(x)-m(x)\big|^2\mu(dx)\Big] \\
&\leq 2\mathbb{E}\Big[\int_{C^p}\big|f_{\lambda^\star}(x)-t_{\lambda^\star}(x)\big|^2\mu(dx)\Big] + 2\mathbb{E}\Big[\int_{C^p}\big|t_{\lambda^\star}(x)-m(x)\big|^2\mu(dx)\Big]. \qquad (5)
\end{aligned}$$

We are going to show that the two expectation terms in the R.H.S. of Eqn. (5) tends to zero under certain restrictions on k_n, β_1 and β_2. The second term is easy to understand since $m\in\mathscr{F}_n$ and for $n\to\infty$ it will converge to 0. But to deal with the first term we take recourse to the following steps.

By definition, $t_{\lambda^\star}(x) = W^{\star\top}_{\text{out}}\Big[\tau\Big(B^{\star\top}\tau(A^{\star\top}x+b_1^\star)+b_2^\star\Big)\Big] + b^\star_{\text{out}}$

and $\qquad f_{\lambda^\star}(x) = W^{\star\top}_{\text{out}}\Big[\sigma_2\Big(B^{\star\top}\sigma_1(A^{\star\top}x+b_1^\star)+b_2^\star\Big)\Big] + b^\star_{\text{out}}.$

Using the fact $|a-b|^2 \leq |a|^2 - |b|^2, \|Y\|_\infty \leq L < \infty$ and $\|W^\star_{\text{out}}\| \leq \frac{k_nL^2}{4}$; we can

write for all $x \in \mathbb{R}^p$

$$
\begin{aligned}
&\left|f_{\lambda^\star}(x) - t_{\lambda^\star}(x)\right|^2 \\
&\quad \le \frac{k_n L^2}{4} \times \left\|\sigma_2\Big(B^{\star\top}\sigma_1(A^{\star\top}x + b_1^\star) + b_2^\star\Big) - \tau\Big(B^{\star\top}\tau(A^{\star\top}x + b_1^\star) + b_2^\star\Big)\right\|^2 \\
&\quad \le \frac{2k_n L^2}{4}\Bigg[\left\|\sigma_2\Big(B^{\star\top}\tau(A^{\star\top}x + b_1^\star) + b_2^\star\Big) - \tau\Big(B^{\star\top}\tau(A^{\star\top}x + b_1^\star) + b_2^\star\Big)\right\|^2 \\
&\qquad + \left\|\sigma_2\Big(B^{\star\top}\sigma_1(A^{\star\top}x + b_1^\star) + b_2^\star\Big) - \sigma_2\Big(B^{\star\top}\tau(A^{\star\top}x + b_1^\star) + b_2^\star\Big)\right\|^2\Bigg] \\
&\quad = \frac{k_n L^2}{2}\Big[I_1 + I_2\Big]. \qquad (6)
\end{aligned}
$$

To find the upper bounds of I_1 and I_2, we will use the following properties of functions:

- Recall that σ_i is tan hyperbolic activation function and τ is a threshold activation function, then we can write for all $u \in \mathbb{R}$, $|\sigma_i(u) - \tau(u)| \le 2e^{-2\beta_i|u|}$ for all $i = 1, 2$.
- Also, σ_i satisfies the Lipschitz property for continuous functions $|\sigma_i(u) - \sigma_i(v)| \le \beta_i|u - v|$ for all $(u, v) \in \mathbb{R}^2$ and $i = 1, 2$.

Using the above properties of functions, we can write

$$
I_1 \le 4\sum_{j=1}^{k_n} \exp\Big[-4\beta_2\big|\big(B^{\star\top}\tau(A^{\star\top}x + b_1^\star) + b_2^\star\big)_j\big|\Big] \le 4k_n e^{-2\beta_2},
$$

since for every j, $\big|\big(B^{\star\top}\tau(A^{\star\top}x + b_1^\star) + b_2^\star\big)_j\big| \ge 1/2$ from the definition of $(A^\star, b_1^\star, B^\star, b_2^\star)$ (see Section 3 of the paper).

Using Lipschitz property, Cauchy-Schwarz inequality and the definition of $B^\star$, we can find the upper-bound for I_2 as follows:

$$
\begin{aligned}
I_2 &\le \sum_{j=1}^{k_n} \beta_2^2\Big|\Big(B^{\star\top}\big(\sigma_1(A^{\star\top}x + b_1^\star) - \tau(A^{\star\top}x + b_1^\star)\big)\Big)_j\Big|^2 \\
&\le \beta_2^2 k_n^2\big\|\sigma_1(A^{\star\top}x + b_1^\star) - \tau(A^{\star\top}x + b_1^\star)\big\|^2 \\
&\le 4\beta_2^2 k_n^2 \sum_{j=1}^{k_n - 1} \exp\Big(-4\beta_1|(A^{\star\top}x + b_1^\star)_j|\Big) \\
&\le 4\beta_2^2 k_n^3 \exp\big(-4\beta_1\varepsilon\big) \quad \text{(for some fixed } j \text{ and arbitrary } \varepsilon > 0)
\end{aligned}
$$

For all n large enough, choosing $\varepsilon = \frac{\log(\beta_1)}{4\beta_1}$, to get $I_2 \le \frac{4\beta_2^2 k_n^3}{\beta_1}$.
Putting these upper bounds of I_1 and I_2 together and using Eqn. (6) we get the following:

$$
\mathbb{E}\int_{C^p} \left|f_{\lambda^\star}(x) - t_{\lambda^\star}(x)\right|^2 \mu(\mathrm{d}x) \le \left[2L^2 k_n^2 e^{-2\beta_2} + \frac{2\beta_2^2 k_n^4 L^2}{\beta_1}\right]. \qquad (7)
$$

R.H.S. of (7) tends to 0 if the conditions of Theorem 3.2 holds and hence the proof.

References

Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. "Tensorflow: a system for large-scale machine learning." In *OSDI*, Vol. 16, 265–283.

Alberto, Wunderlin Daniel, Dıaz Marıa del Pilar, Amé Marıa Valeria, Pesce Silvia Fabiana, Hued Andrea Cecilia, and Bistoni Marıa de los Ángeles. 2001. "Pattern Recognition Techniques for the Evaluation of Spatial and Temporal Variations in Water Quality. A Case Study:: Suquıa River Basin (Córdoba–Argentina)." *Water research* 35 (12): 2881–2894.

Balestriero, Randall. 2017. "Neural Decision Trees." *arXiv preprint arXiv:1702.07360* .

Batchelder, George W. 1965. "Process for the demineralization of water." Mar. 2. US Patent 3,171,799.

Bhattacharya, Biswanath, and Dimitri P Solomatine. 2005. "Neural networks and M5 model trees in modelling water level–discharge relationship." *Neurocomputing* 63: 381–396.

Breiman, Leo. 2017. *Classification and regression trees*. Routledge.

Chakraborty, Tanujit, Ashis Kumar Chakraborty, and Swarup Chattopadhyay. 2018. "A novel distribution-free hybrid regression model for manufacturing process efficiency improvement." *arXiv preprint arXiv:1804.08698* .

Chakraborty, Tanujit, Swarup Chattopadhyay, and Ashis Kumar Chakraborty. 2018. "A novel hybridization of classification trees and artificial neural networks for selection of students in a business school." *OPSEARCH* 55 (2): 434–446.

Chen, Yuehui, Bo Yang, and Ajith Abraham. 2007. "Flexible neural trees ensemble for stock index modeling." *Neurocomputing* 70 (4-6): 697–703.

Chen, Yuehui, Bo Yang, Jiwen Dong, and Ajith Abraham. 2005. "Time-series forecasting using flexible neural tree model." *Information sciences* 174 (3-4): 219–235.

Devroye, Luc, László Györfi, and Gábor Lugosi. 2013. *A probabilistic theory of pattern recognition*. Vol. 31. Springer Science and Business Media.

Foresti, Gian Luca, and Christian Micheloni. 2002. "Generalized neural trees for pattern classification." *IEEE Transactions on Neural Networks* 13 (6): 1540–1547.

Gmar, Soumaya, Nawel Helali, Ali Boubakri, Ilhem Ben Salah Sayadi, Mohamed Tlili, and Mohamed Ben Amor. 2017. "Electrodialytic desalination of brackish water: determination of optimal experimental parameters using full factorial design." *Applied Water Science* 7 (8): 4563–4572.

Györfi, László, Michael Kohler, Adam Krzyzak, and Harro Walk. 2006. *A distribution-free theory of nonparametric regression*. Springer Science and Business Media.

Lhassani, A, M Rumeau, D Benjelloun, and M Pontie. 2001. "Selective demineralization of water by nanofiltration application to the defluorination of brackish water." *Water research* 35 (13): 3260–3264.

Lugosi, Gábor, and Kenneth Zeger. 1995. "Nonparametric estimation via empirical risk minimization." *IEEE Transactions on information theory* 41 (3): 677–687.

Mahuli, SK, RR Rhinehart, and JB Riggs. 1993. "pH control using a statistical technique for continuous on-line model adaptation." *Computers & chemical engineering* 17 (4): 309–317.

Ouyang, Y, P Nkedi-Kizza, QT Wu, D Shinde, and CH Huang. 2006. "Assessment of seasonal variations in surface water quality." *Water research* 40 (20): 3800–3810.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, et al. 2011. "Scikit-learn: Machine learning in Python." *Journal of machine learning research* 12 (Oct): 2825–2830.

Sethi, Ishwar Krishnan. 1990. "Entropy nets: from decision trees to neural networks." *Proceedings of the IEEE* 78 (10): 1605–1613.

Singh, Kunwar P, Ankita Basant, Amrita Malik, and Gunja Jain. 2009. "Artificial neural network modeling of the river water qualitya case study." *Ecological Modelling* 220 (6): 888–895.

Singh, Kunwar P, Amrita Malik, Dinesh Mohan, and Sarita Sinha. 2004. "Multivariate statistical techniques for the evaluation of spatial and temporal variations in water quality of

Gomti River (India)a case study." *Water research* 38 (18): 3980–3992.

Sirat, JA, and JP Nadal. 1990. "Neural trees: a new tool for classification." *Network: Computation in Neural Systems* 1 (4): 423–438.

Vedelago, Rowan, and Graeme J Millar. 2018. "Process evaluation of treatment options for high alkalinity coal seam gas associated water." *Journal of Water Process Engineering* 23: 195–206.