SCJ SOFT COMPUTING JOURNAL
ISSN: 2322-3707
University of Kashan

Solving nonlinear inverse heat transfer problems using Trended Regression Tree

Vahid Norouzifard[1,2*], Mohamad Ali Movafaghpour[3]

[1]Department of Mechanical Engineering, Faculty of Mechanical Engineering, Jundi-Shapur University of Technology, Dezful, Iran, vnorouzi@jsu.ac.ir.

[2]Faculty of Mechanical Engineering, Sahand University of Technology, Tabriz, Iran, norouzifard@sut.ac.ir.

[3]Associate professor, Department of Industrial Engineering, Faculty of Mechanical Engineering, Jundi-Shapur University of Technology, Dezful, Iran. movafaghpour@jsu.ac.ir

Abstract. Determining unknown boundary conditions or material properties in heat transfer problems can be performed by solving inverse heat transfer problems. Solving inverse problems is an optimization problem. The inverse problems are ill-posed, and their accuracy is highly sensitive to the number and location of the sensors, as well as the selected optimization technique. Therefore, the development of new methods that are more stable to solve heat transfer inverse problems has always been of interest. In this paper, a new method is introduced to solve the inverse problems based on the Regression Tree (RT) methods. A new class in RT's are introduced in which we try to fit a linear regression function on each leaf node as compared to classic RT's where only a simple average value of data points assigned to each leaf is calculated to improve the accuracy of the fitted models. We call this new class of RT as Trended RT (TRT). Besides TRT, a modified version of TRT, Bounded TRT (BTRT), is also used in the present study which the regressed line is bounded to observed limits on dependent variable. To investigate the method's efficiency in handling inverse heat transfer problems, the estimation of three different types of boundary conditions in the steady-state heat transfer problem of a steel plate, using the temperature of specific points, is considered. The effects of the number and the location of sensors on the performance of the mentioned methods are also investigated.

Keywords: Regression Tree (RT); Trended Regression Tree (TRT);, Bounded TRT (BTRT); Data mining; Heat Transfer; Inverse Problem; Model Fitting.

1. Introduction

Solving thermal inverse problems to calculate boundary conditions or unknown physical properties - using temperature data measured at one or more points of the body - has been widely used in recent decades. The main challenge of solving inverse problems is the ill-posed nature of these problems, which means any error or noise in the input measured data can amplify and cause much larger errors to appear in the problem's solution. Therefore, the search for new solution methods that reduce the sensitivity of inverse problems'

*Department of Mechanical Engineering, Faculty of Mechanical Engineering, Jundi-Shapur University of Technology, Dezful, Iran, vnorouzi@jsu.ac.ir solutions to the noise of input data has started since the emergence of inverse problems and continues.

The inverse heat transfer problem is an optimization problem in which an objective function should be minimized. The classical methods of solving the thermal inverse problems are based on least-squares minimization of the difference between measured and calculated temperatures. New optimization methods, such as the genetic algorithm method, neural networks, and machine learning, have also attracted the attention of researchers in recent years.

For the first time, Rodensky et al. [1] used the genetic algorithm to estimate the unknown convection heat transfer coefficient. In their study, they used first-order Tikhonov regularization to reduce the error resulting from data noise on the inverse solution results. In the early versions, the genetic algorithm solved inverse problems slowly. Thus, the effort to modify the genetic algorithm to reduce the error and increase the speed of the solution process has continued in recent years. Imani et al. [2] used the modified elitist genetic algorithm to solve the problem of estimating the unknown thermal properties depending on the temperature of the material. Mazidi Sharifabadi et al. [3] recently studied the inverse problem of unknown heat flux estimation using a modified genetic algorithm.

The neural network method is another optimization method used in recent years to solve thermal inverse problems. For example, the following reports are highlighted. Deng and Hwang [4] used the neural network method to solve one-dimensional and two-dimensional direct and inverse thermal problems. Famouri et al. [5] studied the combined method of neural network with a modified elitist genetic algorithm to solve the thermal inverse problem to estimate the unknown conductivity coefficient and heat capacity depending on the temperature of the material. Ghadimi et al. [6] used the artificial neural network and function estimation methods to estimate the heat flux generated in the locomotive brake pad.

Machine learning methods are one of the latest ways to solve inverse problems. Tamdan-Jahormi et al [7] studied the potential of deep machine learning in solving linear and non-linear thermal inverse problems.

Recently, Norouzifard & Movafaghpour [8] successfully used the regression tree to predict the transient unknown boundary condition of a single dimension rod and compared its results with the conventional inverse problems solution methods and neural network method results. According to the comparison, the regression method results are more accurate than the other two methods, and the regression method was faster than the others. In regression, the problem consists of j = 1, 2, . . . , n data points, namely $(x_1^j, x_2^j, ..., x_i^j, ..., x_m^j, y^j)$, where $X^j = (x_1^j, x_2^j, ..., x_i^j, ..., x_m^j)$ and y^j are the coordinate of j'th point in the multidimensional measurement space, and the j'th desired unknown boundary condition, respectively. The variable y is typically referred to as the response or dependent variable. The X variable vector is referred to as the independent variables or predictor variables in various references. Here, the term X is used as the vector of temperatures measured by the sensor, and y is the unknown parameter of the inverse problem (heat flux, convection heat transfer coefficient, etc.). In the regression method, relationships between output variables and a set of independent input variables are estimated by automatically learning from a set of related observations [9]. In data mining with regression analysis, having several examples to train the desired algorithm, and each of them is characterized by certain input and output variables, the ultimate goal is to approximate the functional relationship between them. Once the performance relationship is estimated, it can be used to predict the level of the output variable for new experiments and measurements that have yet to be realized. In general, regression analysis can be useful under two conditions:

1) When the process in question is like a black box, there is limited knowledge of the underlying mechanism of the system. In this case, regression analysis can accurately relate the output variables to the input variables without needing to know the details of the complex internal mechanism (Bai et al., 2014 [10]; Venkatesh et al., 2014 [11]; Cortez et al., 2009 [12]; Davis and Ierapetritou, 2008 [13]). Often, the user wishes to gain valuable insights into the actual underlying functional relationship between the independent variables and the response, despite the unknown nature of the system's internal relationships. This is where the interpretability of the regression method becomes important for systems with a black-box nature.

2) When the detailed simulation model links the input variables to the output variables, it usually does so through some complex relationships. In such cases, the intermediate variables may even be known, but it is very complicated and expensive to perform the calculations comprehensively in a short and acceptable time. In this case, regression analysis can approximate the overall behavior of the system with much simpler functions while maintaining a desirable level of accuracy so that the computations can be done more cheaply, [14]–[17]; for example, in cases where the system response should be online using simple hardware. Over the past years, regression analysis has been used as a powerful tool in a wide range of applications, including investigating the process of carbon dioxide absorption [18], [19] and estimating the thermodynamic properties of ionic liquids [20], [21].

Recently, Carrizosa et al. [22], by reviewing continuous linear and integer optimization models for building cluster regression models, suggested that the use of optimization models in building these models leads to models with greater readability and interpretability. Prado et al. [23] also proposed an algorithm for problems where a large number of input variables are involved in the estimation of the output variable so that the accuracy of the results can be improved by using posterior Bayes probabilities.

The above-mentioned literature review shows that the effectiveness of the regression method in solving thermal inverse problems has not been studied sufficiently and merits more research. In this research, the effectiveness of the regression tree method for solving the steady-state heat transfer inverse problems in a two-dimensional plate is investigated; then we will try to find the best sensor configuration to predict the system behavior. In the present inverse problems, unknown boundary conditions of a metal body are estimated by measuring the temperature at some points in the body. The direct solution was used to generate measurement data. Part of the generated data was used to train the regression algorithm, and some data was used to test the accuracy of the inverse solution.

Recent hybrid and ensemble studies—Hoseini et al. [28] (bagging SVM for credit rating), Nasri-Lowshani and Goli-Bidgoli [29] (Gradient Boosting vs. K-means+PCA for link quality), Saberi et al. [26] (ensemble classifiers with Cuckoo Search for stock prediction), and Sheikhan and Abbasi [27] (C4.5+RBF/SVM with Shuffled Frog Leaping for intrusion detection)—demonstrate that feature selection and model aggregation effectively mitigate ill-posedness and data scarcity. While these methods rely on external aggregation (voting, boosting, sequential filtering) to enhance robustness, our TRT/BTRT approach improves predictive accuracy through internal structural refinement—fitting bounded linear regressions at leaf nodes—offering a complementary strategy that reduces computational overhead and minimizes error propagation in inverse heat transfer problems.

2. The problem definition

A carbon steel plate with dimensions of 100×100 mm was considered. As the plate's thickness is very small compared to its height and width, and the top and bottom surfaces are considered insulating, heat transfer through the thickness of the plate can be neglected. Therefore, a two-dimensional thermal model was used for the direct solution. The thermal boundary conditions were applied to the square plate sides, and the temperature of nine points was recorded as temperature measurement data. Figure 1 shows a schematic view of the problem's geometry.

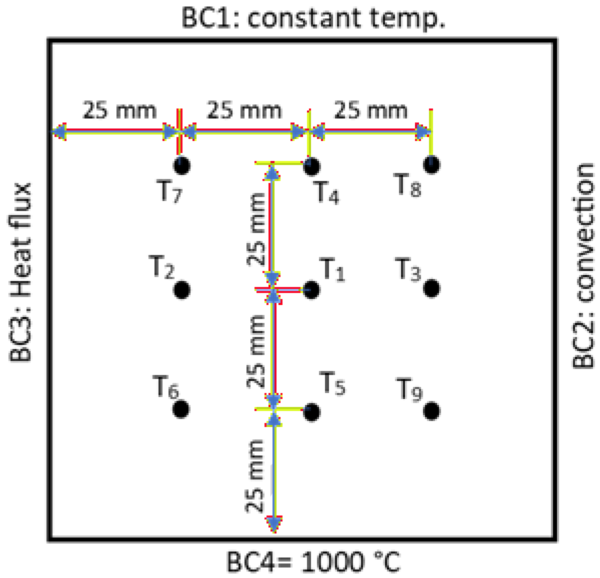

FIGURE 1. Two-dimensional schematic view of the geometry of the problem.

Two different direct thermal solutions with linear and non-linear thermal conduction conditions were performed. As shown in the problem's schematic (Figure 1), the temperature of the bottom side of the plate (BC1) was 1000 °C and constant in all runs. Three different types of boundary conditions were considered for the other sides (BC2-BC4), consisting of convection (BC2), constant heat flux (BC3), and constant temperature (BC4). The convection heat transfer coefficient, heat flux, and temperature value in each run were randomly selected from 0 to 500 W/m2.°C, 0 to -250 kW/m2, and 0 to 1000 °C, respectively.

3. Direct and inverse analysis

3.1. Direct thermal model. This research investigated the temperature distribution within a steel plate using a 2D finite element model. The ANSYS finite element code was used to solve the direct thermal model. Following the problem described in the previous section, 15,000 runs were conducted, and a subroutine in APDL (Ansys Parametric Design Language) was developed to determine random boundary conditions and solve the model continuously.

Figure 2 shows the present 2D-FE model of the steel plate, which is meshed with Plane 55 thermal elements. In the linear thermal solution, the thermal conductivity was 45 W/m.

°C. In non-linear analysis, temperature-dependent thermal conductivity was considered as listed in Table 1. For each run, the following steps are repeated: first, random numbers are generated and assigned to the mentioned boundary conditions; then, the steady-state thermal solution is performed, and the temperature distribution in the plate is calculated. Finally, the boundary conditions and temperature at the 9 points shown in Figure 1 are written in a text file, and another run is started.

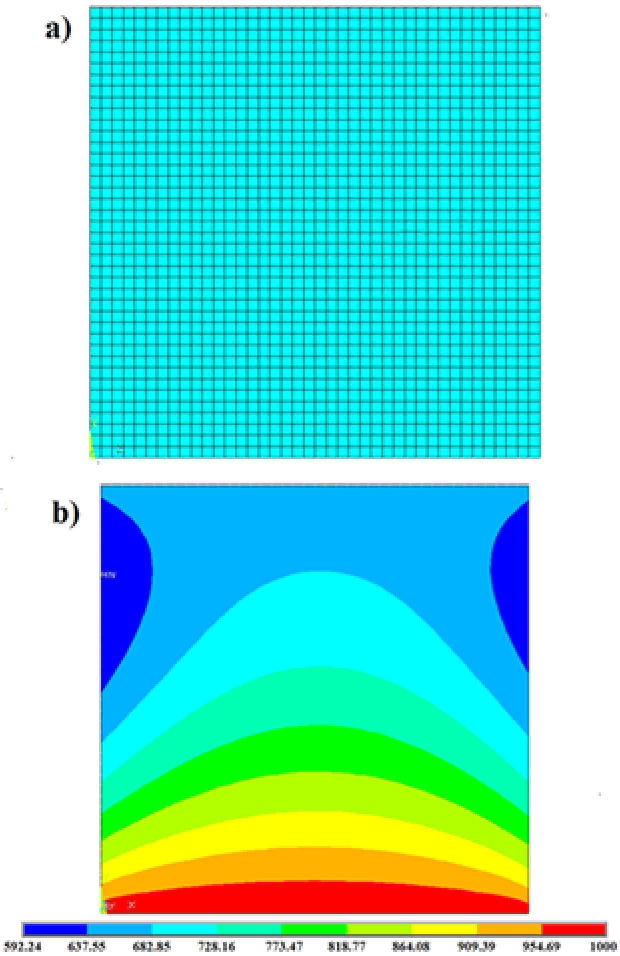

FIGURE 2. a. finite element model of the steel plate meshed by 4-node plane 55 elements, and b. temperature distribution results of one of the linear runs.

3.2. Thermal Inverse Solution: estimation of unknown boundary conditions using the Regression Tree algorithm. A regression tree is a type of machine learning tool that can achieve good prediction accuracy and easy interpretation. It has therefore been the focal point of much research in recent years. A Regression Tree (RT) uses a graph or tree model to express the model fitted to the data. This model is created in an iterative process so that a node containing some observations is partitioned into two child nodes based on certain rules in each iteration. The process of dividing a parent node into child nodes stops when it is a Terminal node. A Terminal node is formed when the number of observations in that node is less than a predetermined threshold called the Partitioning Threshold (PT). PT in each RT of each problem or data set is determined by trial and error. Finally, the average y value of all data points assigned to each node predicts the y value for every future test point X, which will be assigned to that node.

When constructing an RT, the best split point for partitioning data points is determined by exhaustive search; that is, all data points are examined as possible split points. The split point that minimizes an error criterion is selected as the best split point.

TABLE 1. Temperature-dependent thermal conductivity of the steel plate [24]

Temperature (°C)	0	100	200	300	400	600	800	1000
Thermal conductivity (W/m °C)	48.9	47.6	45.7	43.1	39.6	32.9	28.8	27.7

In this paper, we develop an RT algorithm that hierarchically partitions observations into a binary tree-like structure. Then we examine the average value of the response variable over the observations at each end node. Regression Trees are useful tools for decision support and predictive analysis due to their simple structure and ease of use. The resulting RT resembles a hierarchical clustering scheme. Each parent node is divided into two branches based on the PT. Figure 3 shows an example binary tree. Each parent node may be divided into two child nodes, and the nodes containing data points less than PT form terminal nodes (leaves).

At each partitioning step, the RT algorithm chooses an input variable xi and partitions the data points based on their xi values until the RT is constructed. The selection of the best input variable and best split point is based on minimizing an error function such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), or Mean Absolute Error (MAE). Tree construction is terminated using a predefined stopping criterion, such as the depth of the tree or the minimum number of data points assigned to each leaf node.

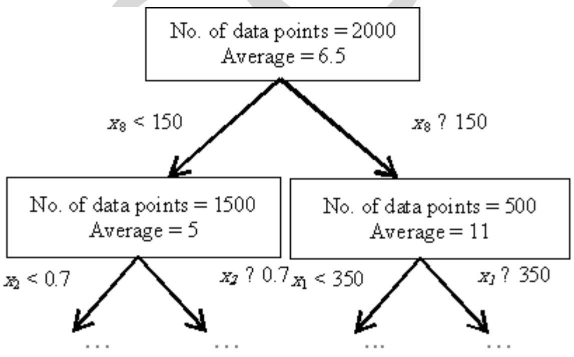

FIGURE 3. A schematic view of a sample binary tree as an RT.

For example, the RT shown in Fig.3 represents a set of 2000 data points with an average y equal to 6.5. The set of these data points is partitioned into two sets: those having their x8 value less than 150 are put in the left branch, and the others are put in the right branch. Each of the branches has a significantly different y value. In this case, the average y values are 5 and 11 for the left and right branches, respectively.

Building a binary regression tree is an iterative process of partitioning the input data points j = 1 , . . . ,n. At each iteration, all data points are sorted according to each of the variable dimensions i = 1, . . . , m. An error function, such as SE, is calculated for the split point x_i^k as follows:

$$SE(i,k) = SE(\text{leftbranch}) + SE(\text{rightbranch})$$

$$SE(i,k) = 1/n_1 \sum_{(j|x_i^j<x_i^k)} (y_j - \overline{y_{left}})^2 + 1/n_2 \sum_{(j|x_i^j\geq x_i^k)} (y_j - \overline{y_{right}})^2$$

Where:

$$\overline{y_{left}} = 1/n_1 \sum_{(j|x_i^j<x_i^k)} y_j,$$
$$\overline{y_{right}} = 1/n_2 \sum_{(j|x_i^j\geq x_i^k)} y_j$$

with regard to each split point is incrementally slid through the values of the subject variable. The best split point is where SE reaches its minimum value. All input variable dimensions i = 1, 2, . . . , m and all possible split points j = 1, 2, . . . , n are evaluated, and the best one is selected.

The pseudo code of the RT construction algorithm used in this research is described as follows:

Step 1. Start with a single node as the root node that includes all the data points.

Step 2. For every i'th variable in the current node:

For every k'th data point in the current node:

Assume x_i^k , as the split point, and calculate its relative SE(i, k) using Eq.1.

Step 3. Find the least SE(i, k) and partition the current parent node into two child nodes, "left child" and "right child".

Step 4. If each one of the recently generated child nodes contains data points less than PT, mark it as a leaf node; else, add it to the "candidate nodes" stack for future splitting.

Step 5. If the "candidate nodes" stack is empty, then go to Step 6; otherwise, pick a candidate node from the "candidate nodes" stack as the current node and go to Step 2.

Step 6. Stop

In the following, assuming that all data points j = 1, 2, . . . , n include the m-dimensional temperature measured in the simulation (as x_i^j 's, i = 1, 2, . . . , m) and the boundary condition equivalent to them (as y^j) are located in the j-th row of the raw data table.

we draw the flowchart of the implementation of the regression tree construction. The column corresponding to $x_1^j, x_2^j, x_3^j, ..., x_9^j$ are the temperatures of the points T1, T2, T3, . . . , T9, respectively as shown in Figure1.

3.3. Thermal Inverse Solution using the Trended Regression Tree (TRT).

3.3.1. *Trended Regression Tree (TRT).* Traditional RT stops with a binary tree that at each of its terminal leaves, c, a value $\overline{y^c}$ represents the average output variable for all the instances assigned to that leaf. Only averaging the output variables, $y^j, (j \in c)$, may ignore some useful information. Therefore, we prefer to develop a linear regression function at each terminal leaf. Specifically, after the full-size RT is produced, we perform a linear independence analysis among i = 1, 2, . . . , m dimensions at each leaf to extract a linearly independent set of dimensions $i' = 1, 2, ..., m', (m \leq m')$. The correlation analysis is done because some leaves may contain P data points and fit a linear regression function of the form:

$$\widehat{y^j} = a_0 + a_1 x_1^j + a_2 x_2^j + ... + a_i x_i^j + a_m x_m^j$$

requires m +1 data points. In such cases, it is crucial to ignore some dimensions i to be able to fit a linear regression function with m'-dimensional inputs. Fig.4 depicts a sample Trended Regression Tree (TRT).The overall pseudo-code of the Trended Regression Tree (TRT) construction algorithm used in this research is described as follows:

Step 1. Start with a single node as the root node that includes all the data points.

Step 2. For every i'th variable in the current node:

For every k'th data point in the current node:

Assume x_i^k , as the split point, and calculate its relative SE(i, k) using Eq.1.

Step 3. Find the least SE(i, k) and partition the current parent node into two child nodes, "left child" and "right child".

Step 4. If each one of the recently generated child nodes contains data points less than PT, mark it as a leaf node; else, add it to the "candidate nodes" stack for future splitting.

Step 5. If the "candidate nodes" stack is empty, then go to Step 6; otherwise, pick a candidate node from the "candidate nodes" stack as the current node and go to Step 2.

Step 6. For every terminal node c with n data points:

Calculate the Pearson Correlation Coefficient $\rho(i, l)$ between each pair of input dimensions i (i = 1, 2, ...,m) and l (l = i+1, i+2, ...,m)

Ignore every input dimension i that:

$$\max_l \{|\rho(i, l)|\} > \operatorname{mode}_l \{|\rho(i, l)|\}$$

If the number of remaining input dimensions (i.e. m') is greater than n, ignore extra (m'-n) input dimensions.

Fit an m'-dimensional linear regression function on data points in the current node c.

Step 7. Stop

At step 6 we try to avoid collinearity by regressing an overfitted model, by ignoring some less effective dependent variables. Less dependent variables are those dimensions of datapoints which present a correlation coefficient less than the mode of the sample. Using the mode of the correlation coefficient of all dependent variables at each node, represents a robust adaptive threshold based on the observed data at each leaf of the RT.

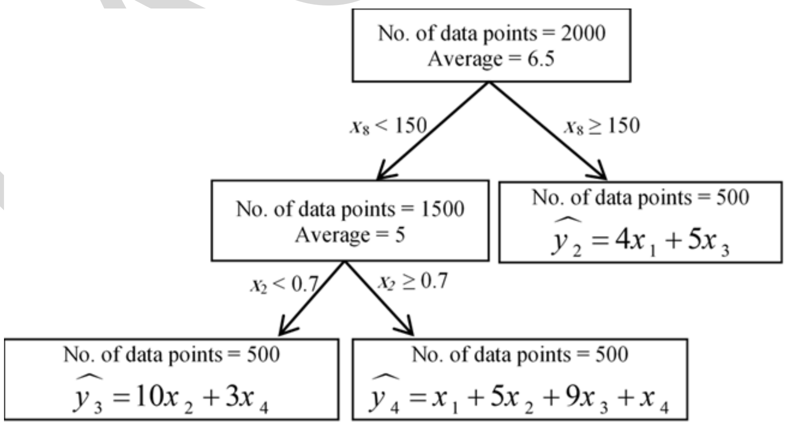

Figure 4. A schematic view of a sample binary tree as a TRT and fitted regression functions on leaf nodes.

3.3.2. *Trended Regression Tree (TRT).* After the linear regression function is fit on each terminal node, we are ready to map newly gathered data points into the output variable y. When mapping newly gathered data points into the output variable y, some data points may generate extreme y values. For example, if the observed range for y-values of the data points at each terminal node c is [ymin, ymax]c, a newly assigned data point X ($X^{n+1} = (x_1^{n+1}, x_2^{n+1}, ..., x_m^{n+1})$) may yield a new $\widehat{y^{n+1}} = a_0 + a_1x_1^{n+1} + a_2x_2^{n+1} + ... + a_ix_i^{n+1} + ... + a_mx_m^{n+1}$ value that is out of the observed range [ymin , ymax]c for the respective terminal node c. In such cases, one can revise the generated $\widehat{y^{n+1}}$, by setting it equal to the nearest bound ymin or ymax of the respective terminal node. Making such revisions on excessive y-value and conveying them to the observed range [ymin , ymax]c of the terminal node yields another version of TRT, which is called "Bounded TRT" here. When dealing with certain problems, TRT yields high-quality results, and in some cases, Bounded TRT may also yield high-quality results.

4. Results

The gathered dataset consists of 15000 data points. Each data point consists of nine temperatures read by nine sensors as independent variables (T1, . . . , T9) and three boundary condition values as dependent variables (y1, y2, and y3). We set the Partitioning Threshold (PT) as 2000. To construct an efficient and strong model, we try to:

1. Find a suitable and small set of sensors whose gathered data is more informative, and

2. Select the proper model among RT, TRT, and Bounded TRT, which yields the least prediction error regarding the selected subset of sensory data.

Besides finding an informative subset of (five to nine) sensors and the strongest model, the results should be reliable. A reliable model means that it does not depend on the data used for the experiment. Therefore, we use 5-fold experimentation in which we partition all available data into five random sets, and every performance measure, such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) in experimental results, is the average of five experiments. In each experiment, we use four partitions for model generation (training the model) and the fifth one for verification (testing the model). After selecting the most informative subset of sensory data greedily, we tried to select the best regression tree model among RT, TRT, and BTRT.

Absolute values of unknown boundary condition, which are considered as the dependent variable, vary drastically within the observed range; perhaps, to reach a reliable performance measure, we use Relative Error (RE) instead of Absolute Error (AE):

$RE_i = \frac{Y_i - E_i}{E_i}$

In which Yi and Ei are the true and estimated values of the unknown boundary condition predicted by the model. The two kinds of performance measures we used are Mean Absolute Error (ME) of the above-mentioned Relative Error (REi):

$MAE = 1/n \sum_{i=1}^{n} |RE_i|$

and Root Mean Squared Error (RMSE) of the above-mentioned Relative Error (REi):

$RSME = \sqrt{1/n \sum_{i=1}^{n} {RE_i}^2}$

4.1. Linear heat transfer analysis. Table 2 presents the performance measures of the developed regression algorithms with 5 to 9 sensors. The respective plots of the above performance measures are also depicted in Figures 5-10. Since the Regression Tree (RT) algorithm has relatively huge errors, we ignore it when plotting performance measures.

Since the error measures given in Table 2 are relative errors (Equation 4), having a zero MAE or RMSE error measure does not mean an absolute zero error (Ei), but it means that a tiny absolute error divided into a true value Yi results in a numeric value which is smaller than the significant digits used in the spreadsheet software used. Besides, the regular behavior of thermal invers solution helps the TRT algorithm in achieving such a small error in predicting Thermal Inverse Solution at boundary condition 1 & 3 (BC1 & BC3).

As one can infer from Figures 5-10, BC2 is the most challenging dependent variable in the linear flux problem, as it causes the most prediction errors. RMSE values are higher than MAE, which means that although the mean value of errors is low, some extremely high errors inflate the square error.

The significantly reliable and precise Trended Regression Tree (TRT) outperforms the bounded Trended Regression Tree (BTRT). It is worth noting that TRT yields a zero MAE and RMSE in most cases (BC1 and BC3 in the present model), even with five sensors. It means that we can use it for stronger models with less sensory data gathered from even fewer sensors!

To investigate the performance of the present algorithm with two sensor data sets, the performance of all possible double sensor combinations is tested to predict unknown boundary conditions. Table 3 presents the two best and the worst performance measures of the developed regression algorithms with two sensors. As shown in Table 3, the TRT and BTRT algorithms exhibit better performance compared to the RT. The gap between the MAE in the worst and the best case is extremely high in the TRT and BTRT algorithms. Additionally, it can be observed that the performance of the BTRT algorithm surpasses that of the TRT in predicting BC2 and BC3, which are more challenging boundary conditions than BC1. Since the above model yields a low prediction error, it is a promising model for using less sensory data for predicting linear heat flux based on sensed temperatures.

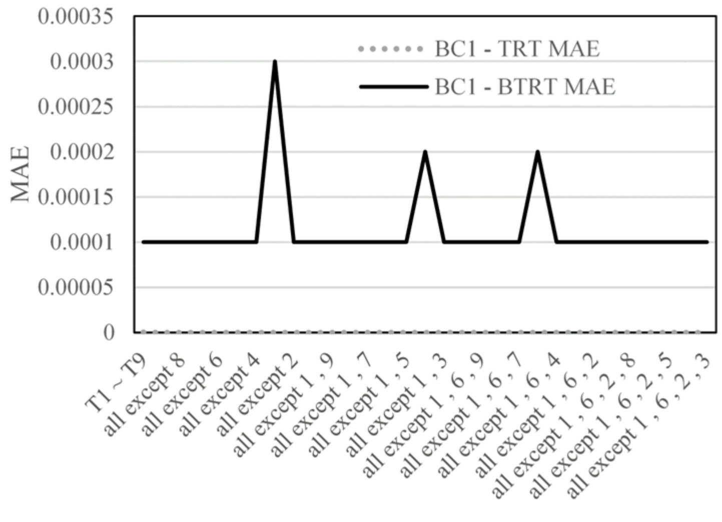

Figure 5. MAE plots versus different thermocouple compositions for BC1.

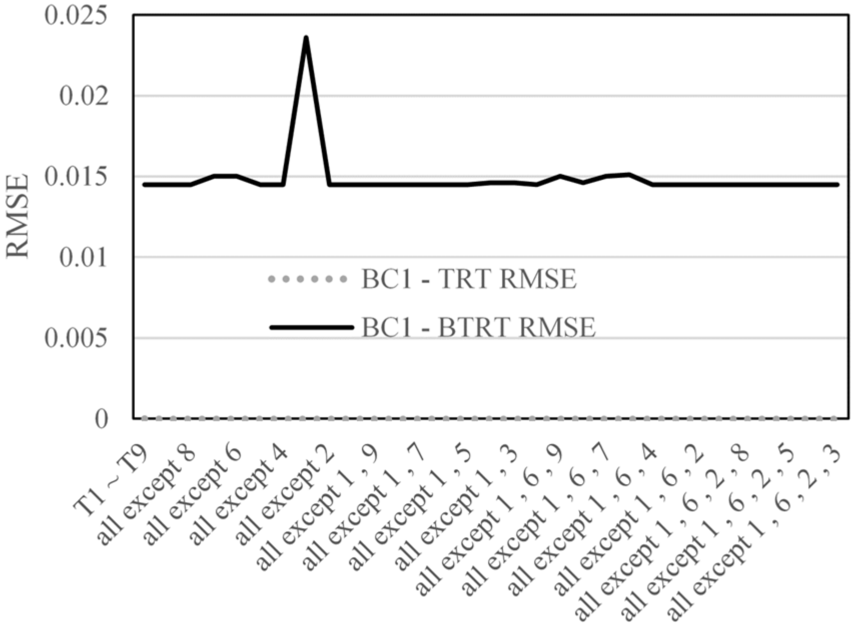

FIGURE 6. RMSE plots versus different thermocouple compositions for BC1.

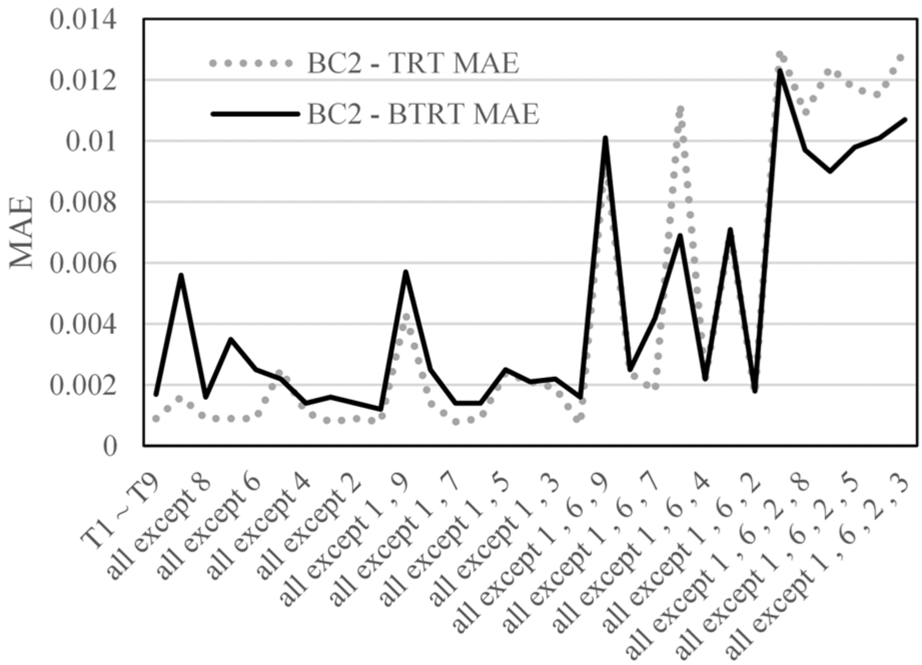

FIGURE 7. MAE plots versus different thermocouple compositions for BC2.

Table 2. Performance measures for linear Flux

	BC1 - RT		BC1 - TRT		BC1 - BTRT		BC2 - RT		BC2 - TRT		BC2 - BTRT		BC3 - RT		BC3 - TRT		BC3 - BTRT	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
$T1 \sim T9$	0.5568	13.097	9E-09	1E-07	0.0001	0.0145	3.3409	74.786	0.0009	0.0276	0.0017	0.0796	2.1998	47.114	4E-08	5E-07	0.0008	0.0575
all except 9	0.5568	13.097	0	0	0.0001	0.0145	3.8618	121.34	0.0016	0.0989	0.0056	0.2906	2.1998	47.114	0	0	0.0008	0.0575
all except 8	0.5577	13.125	0	0	0.0001	0.0145	3.428	86.049	0.0009	0.0387	0.0016	0.0617	2.1951	43.323	0	0	0.0011	0.0614
all except 7	0.5593	13.319	0	0	0.0001	0.015	3.2783	74.916	0.0009	0.0361	0.0035	0.3122	2.1596	42.05	0	0	0.0008	0.0575
all except 6	0.5627	13.398	0	0	0.0001	0.015	3.4195	85.969	0.0009	0.037	0.0025	0.1828	2.2575	44.222	0	0	0.0019	0.1012
all except 5	0.5612	13.323	0	0	0.0001	0.0145	3.4229	87.975	0.0025	0.0994	0.0022	0.0895	2.2118	43.457	0	0	0.001	0.0617
all except 4	0.5805	13.563	0	0	0.0001	0.0145	3.3544	77.079	0.0011	0.0372	0.0014	0.0602	2.1401	42.45	0	0	0.0011	0.0701
all except 3	0.5561	12.926	0	0	0.0003	0.0236	3.4201	84.618	0.0008	0.0184	0.0016	0.0621	2.1514	43.688	0	0	0.001	0.0609
all except 2	0.5645	13.418	0	0	0.0001	0.0145	3.3846	76.291	0.0009	0.0242	0.0014	0.062	2.1474	42.44	0	0	0.0009	0.0584
all except 1	0.5639	13.567	0	0	0.0001	0.0145	3.3317	77.231	0.0008	0.0274	0.0012	0.0594	2.1167	43.401	0	0	0.0011	0.0693
all except 1,9	0.5583	13.334	0	0	0.0001	0.0145	3.4253	76.047	0.0043	0.1093	0.0057	0.1583	2.2101	46.333	0	0	0.0016	0.0981
all except 1,8	0.5663	13.517	0	0	0.0001	0.0145	3.2049	68.621	0.0014	0.0402	0.0025	0.1427	2.2104	42.081	0	0	0.001	0.0749
all except 1,7	0.5582	13.138	0	0	0.0001	0.0145	3.176	73.275	0.0008	0.0227	0.0014	0.0617	2.2028	46.308	0	0	0.0007	0.0573
all except 1,6	0.5555	13.26	0	0	0.0001	0.0145	3.399	82.086	0.0009	0.0324	0.0014	0.0608	2.2567	44.722	0	0	0.0019	0.1004
all except 1,5	0.5568	13.35	0	0	0.0001	0.0145	3.1308	66.241	0.0024	0.0645	0.0025	0.0674	2.1522	42.216	0	0	0.0008	0.0574
all except 1,4	0.5865	14.037	0	0	0.0002	0.0146	3.4935	85.668	0.0021	0.0605	0.0021	0.0642	2.1027	39.581	0	0	0.0008	0.0579
all except 1,3	0.5546	12.897	0	0	0.0001	0.0146	3.4952	84.665	0.0019	0.0711	0.0022	0.0769	2.1247	38.304	0	0	0.0009	0.0578
all except 1,2	0.5603	13.255	0	0	0.0001	0.0145	3.2464	69.867	0.0007	0.0132	0.0016	0.0661	2.1796	44.261	0	0	0.001	0.0671
all except 1,6,9	0.5572	13.063	0	0	0.0001	0.015	3.887	121.4	0.0095	0.1867	0.0101	0.2694	2.2188	42.449	0	0	0.0027	0.1306
all except 1,6,8	0.5679	13.714	0	0	0.0001	0.0146	3.2918	72.843	0.0024	0.0636	0.0025	0.0716	2.1645	43.023	0	0	0.0019	0.1025
all except 1,6,7	0.5681	13.494	0	0	0.0001	0.015	3.3759	82.847	0.0018	0.0499	0.0042	0.3132	2.2928	44.228	0	0	0.0023	0.1114
all except 1,6,5	0.5566	13.098	0	0	0.0002	0.0151	3.2471	69.455	0.0112	0.3136	0.0069	0.086	2.1624	42.797	0	0	0.0025	0.1102
all except 1,6,4	0.5826	13.822	0	0	0.0001	0.0145	3.5534	88.483	0.0022	0.0681	0.0022	0.0684	2.2913	44.02	0	0	0.0033	0.1882
all except 1,6,3	0.5646	13.418	0	0	0.0001	0.0145	3.406	86.668	0.007	0.1917	0.0071	0.1917	2.2064	42.939	0	0	0.0019	0.1008
all except 1,6,2	0.5613	13.415	0	0	0.0001	0.0145	3.2647	68.671	0.0017	0.0408	0.0018	0.0599	2.3335	41.732	0	0	0.0039	0.1401
all except 1,6,2,9	0.5543	13.017	0	0	0.0001	0.0145	3.4688	80.437	0.013	0.3068	0.0123	0.2703	2.329	39.558	0	0.0001	0.0023	0.105
all except 1,6,2,8	0.5573	13.407	0	0	0.0001	0.0145	3.4688	86.498	0.0109	0.2565	0.0097	0.2455	2.5845	40.992	0	0.0001	0.0112	0.4456
all except 1,6,2,7	0.5591	13.455	0	0	0.0001	0.0145	3.2638	72.549	0.0124	0.3018	0.009	0.1344	2.5345	57.701	0	0.0002	0.0023	0.0794
all except 1,6,2,5	0.5649	13.406	0	0	0.0001	0.0145	3.4638	84.872	0.0117	0.2496	0.0098	0.2244	2.7213	42.694	0	0.0001	0.0054	0.1744
all except 1,6,2,4	0.5729	13.305	0	0	0.0001	0.0145	3.3642	76.652	0.0115	0.283	0.0101	0.2681	2.4224	46.065	0	0.0002	0.0058	0.2624

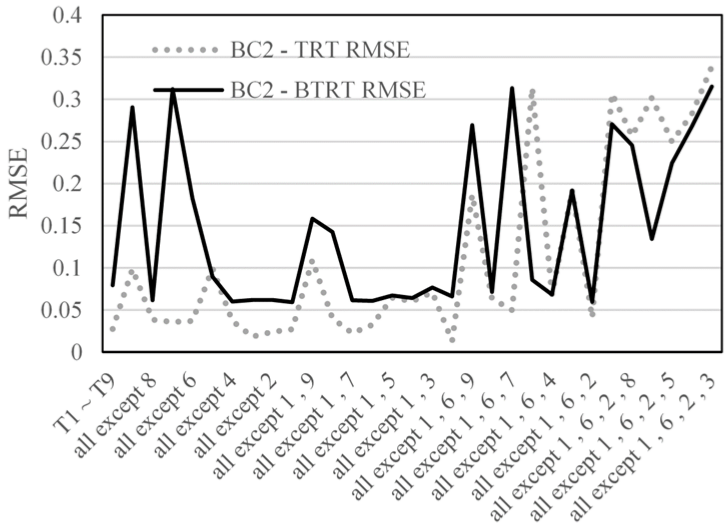

Figure 8. RMSE plots versus different thermocouple compositions for BC2.

Table 3. The best and the worst performance measures for linear Flux with double sensors among nine sensors

	BC1						BC2						BC3					
	RT		TRT		BTRT		RT		TRT		BTRT		RT		TRT		BTRT	
Used sensors	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
2, 6	0.774	17.26	0.004	0.02	0.004	0.02												
4, 9	0.548	12.76	0.105	1.722	0.09	1.242												
6, 9	0.995	21.05	0.683	12.38	0.682	12.38												
3, 4							3.786	121.8	0.256	10.2	0.065	0.443						
4, 9							3.167	71.49	0.273	4.537	0.162	2.366						
4, 6							5.836	138.1	5.716	134.3	5.716	134.3						
2, 4													2.186	43.42	0.121	3.685	0.101	3.606
6, 8													2.098	38.4	0.246	8.62	0.156	2.448
3, 9													3.732	79.46	3.724	85.03	3.724	85.03

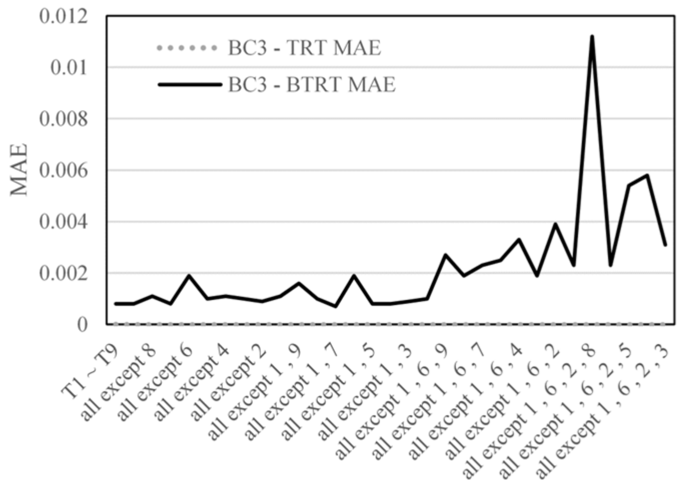

FIGURE 9. MAE plots versus different thermocouple compositions for BC3.

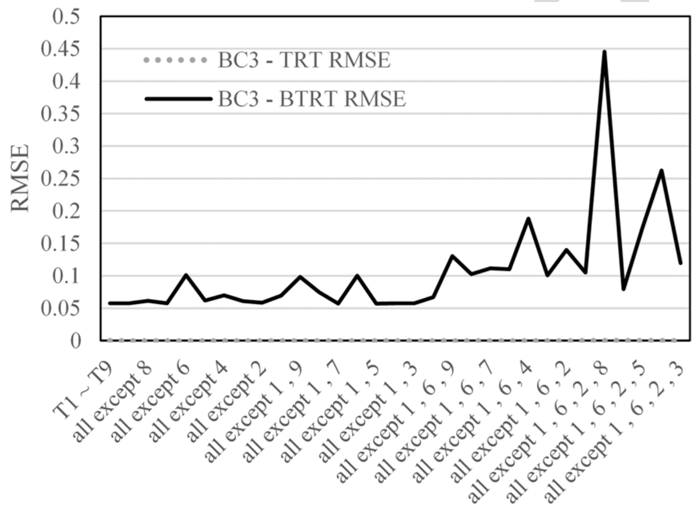

FIGURE 10. RMSE plots versus different thermocouple compositions for BC3.

4.2. Nonlinear heat transfer problem. Table 4 presents the performance measures of the developed regression algorithms with 5 to 9 sensors for the nonlinear problem. The respective plots of the above performance measures are depicted in Figures 11-16. Since the Regression Tree (RT) algorithm has relatively huge errors, we ignore it when plotting performance measures.

As you can infer from Figures 11 and 12, predicting BC1 in the nonlinear heat transfer problem is more complex, but it is still not too complicated, as the maximum values of MAE and RMSE are the same and relatively lower than those of BC2 and BC3. When the number of sensors decreases, the average value of error measures demonstrates a small increasing gradient.

With BC2, a significant difference is exhibited between the mean values of MAE and RMSE. It means there are some extremely large relative errors, which intensify the RMSE. Bounded TRT depicts a significantly better performance in predicting BC2. TRT lacks

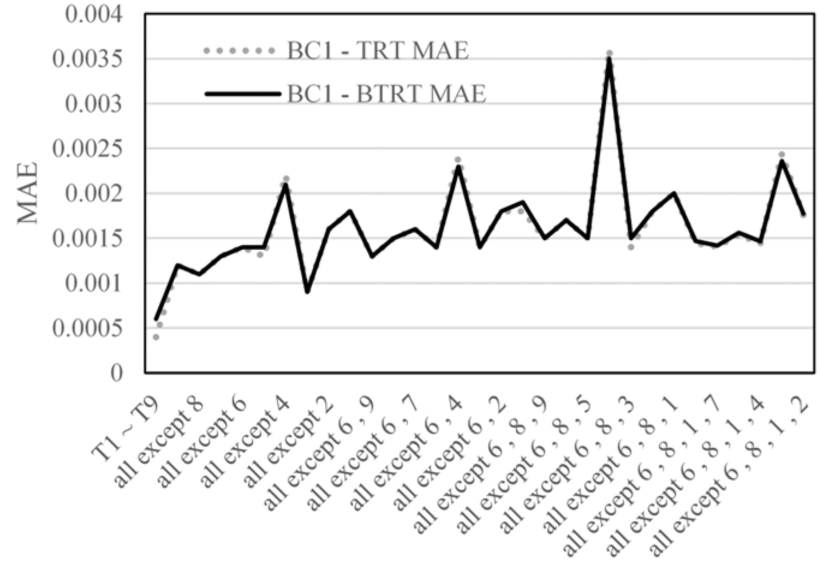

Figure 11. MAE plots versus different thermocouple compositions for BC1.

Table 4. Performance measures for nonlinear Flux

	BC1 - RT		BC1 - TRT		BC1 - BTRT		BC2 - RT		BC2 - TRT		BC2 - BTRT		BC3 - RT		BC3 - TRT		BC3 - BTRT	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
$T1 \sim T9$	0.562	13.36	4E-04	0.006	6E-04	0.015	3.251	79.95	0.039	1.017	0.022	0.467	2.191	42.64	0.005	0.097	0.004	0.07
all except 9	0.556	13.19	0.001	0.03	0.001	0.03	3.518	85.75	0.108	3.128	0.083	3.05	2.187	42.46	0.01	0.126	0.009	0.133
all except 8	0.557	13.11	0.001	0.012	0.001	0.016	3.288	83.04	0.098	2.479	0.04	0.233	2.165	43.02	0.006	0.127	0.005	0.079
all except 7	0.554	13.4	0.001	0.015	0.001	0.018	3.338	81.86	0.046	1.2	0.029	0.977	2.196	42.65	0.007	0.157	0.006	0.08
all except 6	0.568	13.66	0.001	0.028	0.001	0.028	3.349	86.95	0.042	1.027	0.024	0.138	2.211	43.28	0.016	0.189	0.013	0.141
all except 5	0.551	12.76	0.001	0.032	0.001	0.032	3.298	83.3	0.063	1.577	0.036	0.687	2.164	42.65	0.01	0.099	0.01	0.102
all except 4	0.55	12.79	0.002	0.031	0.002	0.028	3.273	83.91	0.085	2.341	0.043	0.977	2.228	45.56	0.008	0.178	0.006	0.081
all except 3	0.556	12.82	9E-04	0.02	9E-04	0.02	3.333	82.3	0.116	2.198	0.046	0.257	2.138	43.05	0.006	0.1	0.006	0.074
all except 2	0.556	13.42	0.002	0.035	0.002	0.035	3.297	82.56	0.05	1.036	0.029	0.674	2.169	42.89	0.01	0.226	0.007	0.082
all except 1	0.548	12.7	0.002	0.047	0.002	0.047	3.264	82.91	0.103	3.298	0.045	0.698	2.207	42.66	0.008	0.143	0.007	0.068
all except 6,9	0.558	13.36	0.001	0.03	0.001	0.03	3.539	82.2	0.119	1.827	0.065	0.834	2.252	45.62	0.019	0.273	0.014	0.125
all except 6,8	0.551	12.77	0.002	0.025	0.002	0.025	3.282	80.91	0.117	3.354	0.047	0.297	2.246	42.97	0.016	0.221	0.013	0.109
all except 6,7	0.543	12.68	0.002	0.029	0.002	0.029	3.259	79.49	0.061	1.826	0.034	0.915	2.216	45.09	0.016	0.188	0.014	0.146
all except 6,5	0.564	13.46	0.001	0.031	0.001	0.032	3.36	84.97	0.089	2.571	0.043	0.796	2.226	45.25	0.019	0.272	0.013	0.126
all except 6,4	0.553	12.94	0.002	0.032	0.002	0.031	3.252	80.85	0.094	2.538	0.045	0.964	2.206	42.76	0.016	0.196	0.014	0.16
all except 6,3	0.552	12.97	0.001	0.029	0.001	0.029	3.243	81.69	0.114	2.164	0.05	0.353	2.198	43.72	0.016	0.196	0.013	0.155
all except 6,2	0.564	13.59	0.002	0.036	0.002	0.036	3.331	83.47	0.05	0.86	0.029	0.538	2.385	46.79	0.008	0.088	0.008	0.114
all except 6,1	0.544	12.7	0.002	0.05	0.002	0.05	3.286	84.02	0.116	3.566	0.046	0.349	2.186	44.52	0.018	0.228	0.015	0.167
all except 6,8,9	0.563	13.51	0.002	0.018	0.002	0.021	3.532	86.35	0.191	4.348	0.085	2.384	2.225	44.8	0.019	0.277	0.015	0.121
all except 6,8,7	0.554	13.33	0.002	0.026	0.002	0.027	3.076	70.3	0.143	2.988	0.057	0.493	2.243	44.63	0.016	0.147	0.015	0.132
all except 6,8,5	0.55	12.71	0.002	0.023	0.002	0.023	3.446	86.87	0.184	4.246	0.061	0.683	2.296	46.3	0.019	0.324	0.016	0.262
all except 6,8,4	0.609	14.78	0.004	0.069	0.004	0.068	3.23	82.49	0.134	3.047	0.061	0.842	2.626	46.8	0.019	0.364	0.018	0.363
all except 6,8,3	0.563	13.76	0.001	0.015	0.002	0.019	3.42	89.14	0.172	4.174	0.056	0.322	2.257	45.41	0.017	0.17	0.014	0.114
all except 6,8,2	0.552	12.95	0.002	0.033	0.002	0.033	3.272	81.72	0.134	3.15	0.056	0.572	2.757	57.22	0.013	0.184	0.011	0.12
all except 6,8,1	0.551	12.82	0.002	0.047	0.002	0.047	3.253	82.8	0.164	4.221	0.05	0.209	2.242	45.44	0.019	0.276	0.016	0.22
all except 6,8,1,9	0.551	12.85	0.001	0.031	0.001	0.031	3.822	120.6	0.369	18.42	0.067	0.251	2.195	42.98	0.013	0.154	0.012	0.127
all except 6,8,1,7	0.559	13.43	0.001	0.019	0.001	0.019	3.275	82.8	0.163	3.719	0.062	0.553	2.152	42.65	0.019	0.357	0.018	0.347
all except 6,8,1,5	0.555	13.07	0.002	0.027	0.002	0.027	3.329	84.95	0.153	3.539	0.057	0.43	2.138	42.15	0.012	0.159	0.011	0.14
all except 6,8,1,4	0.555	12.87	0.001	0.03	0.001	0.03	3.265	84	0.132	2.777	0.057	0.519	2.247	45.26	0.02	0.289	0.015	0.163
all except 6,8,1,3	0.548	12.75	0.002	0.031	0.002	0.026	3.296	83.35	0.156	3.526	0.059	0.469	2.179	42.06	0.013	0.17	0.012	0.154
all except 6,8,1,2	0.56	13.41	0.002	0.033	0.002	0.033	3.256	81.89	0.131	2.762	0.057	0.658	2.173	42.43	0.013	0.212	0.012	0.136

precision for prediction when dealing with a challenging problem, when compared with bounded TRT. Focusing on MAE on BC2, it was realized that when the number of sensors decreases, the average value of MAE follows a small increasing gradient.

Predicting BC3 with TRT and Bounded TRT shows the same behavior as reviewed for predicting BC2. The difference in complexity in predicting the unknown boundary conditions (BC1, BC2, and BC3) by the present algorithms can also be explained by the physics of the heat transfer problem. To keep simplicity, we can consider a one-dimensional

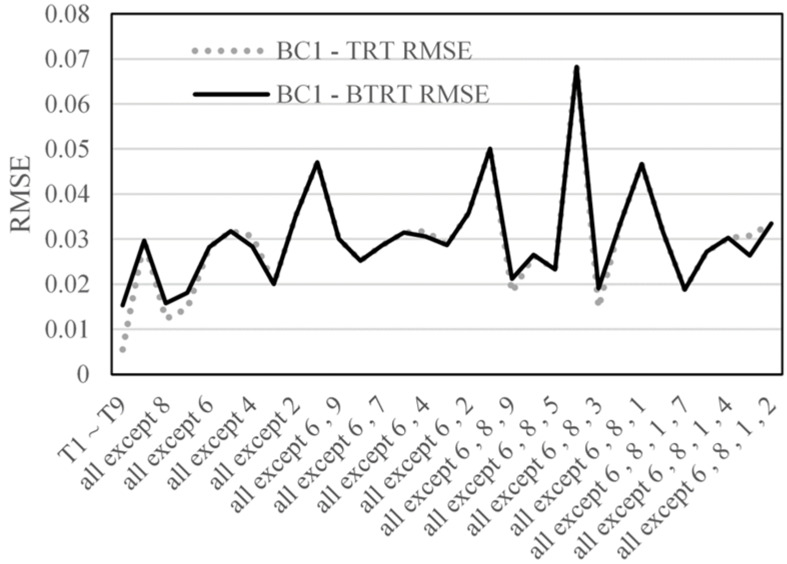

FIGURE 12. RMSE plots versus different thermocouple compositions for BC1.

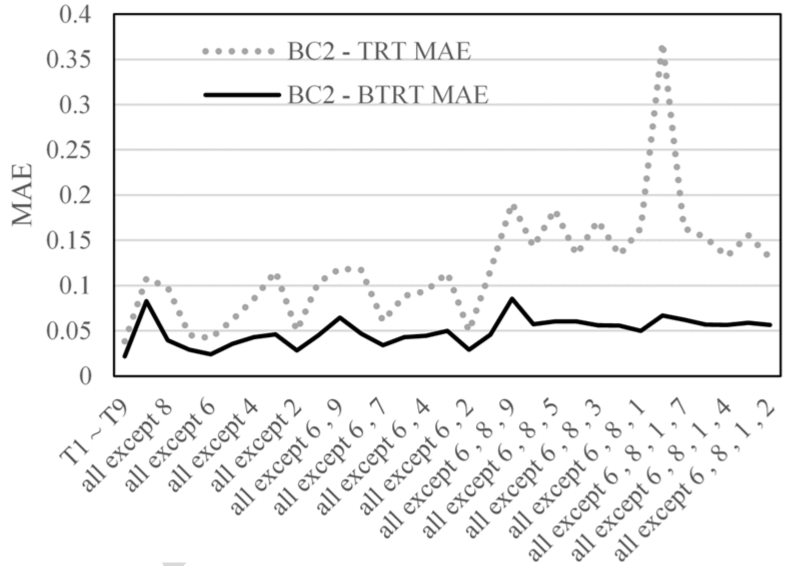

FIGURE 13. MAE plots versus different thermocouple compositions for BC2.

heat transfer problem as shown in Figure 16. On one side of the model, a fixed temperature is considered, and on the other side, three different boundary conditions are considered, respectively, which are fixed temperature, constant heat flux, and convection. To achieve the temperature distribution of the model, the following equations should be solved.

$\frac{d^2T}{dx^2} = 0$

$$\begin{cases} T = T_0, x = x_0 \\ T = cont., x = L \end{cases}$$

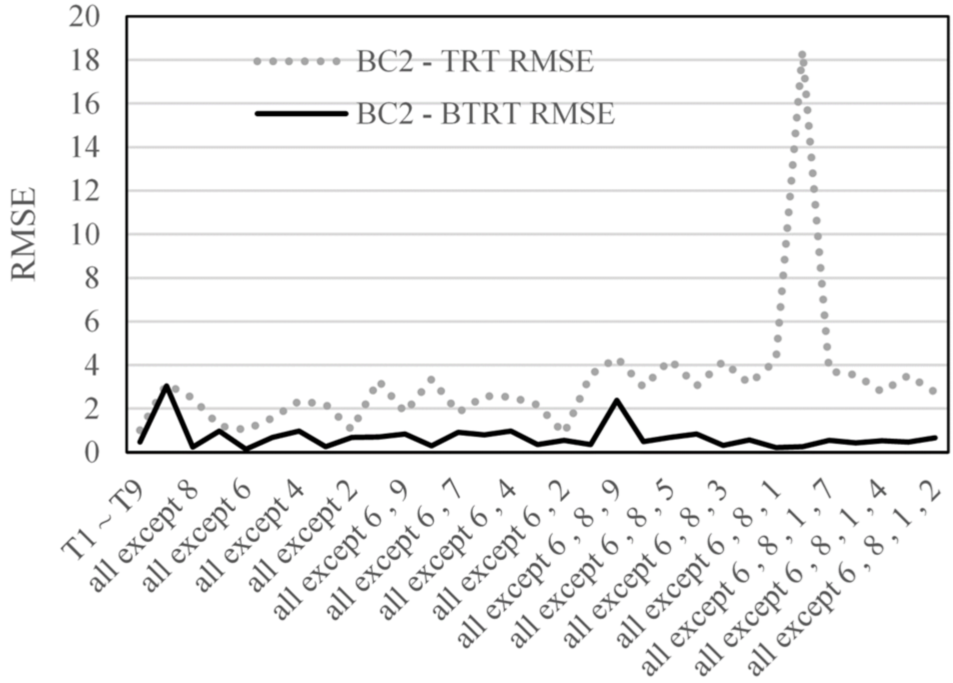

Figure 14. RMSE plots versus different thermocouple compositions for BC2.

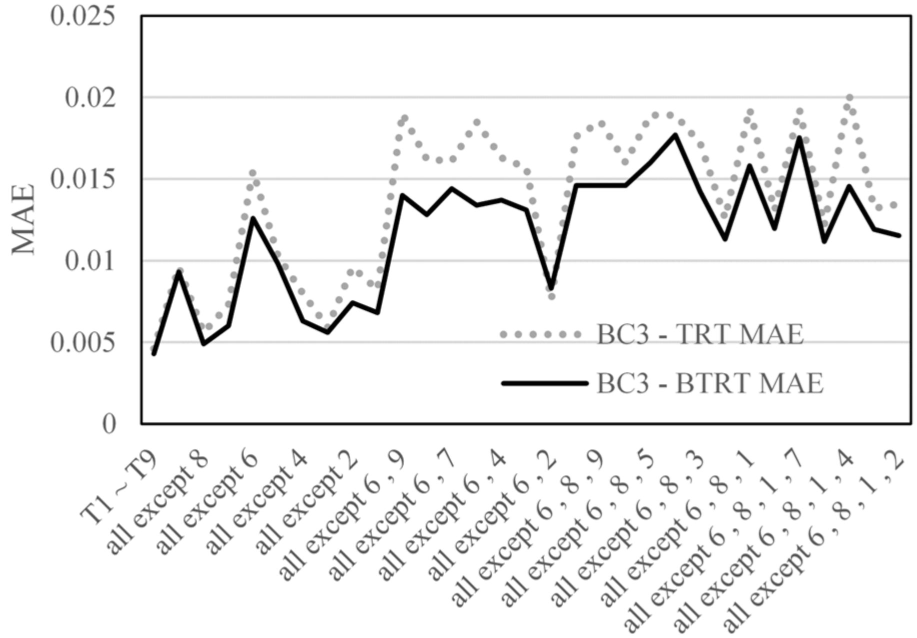

Figure 15. MAE plots versus different thermocouple compositions for BC3.

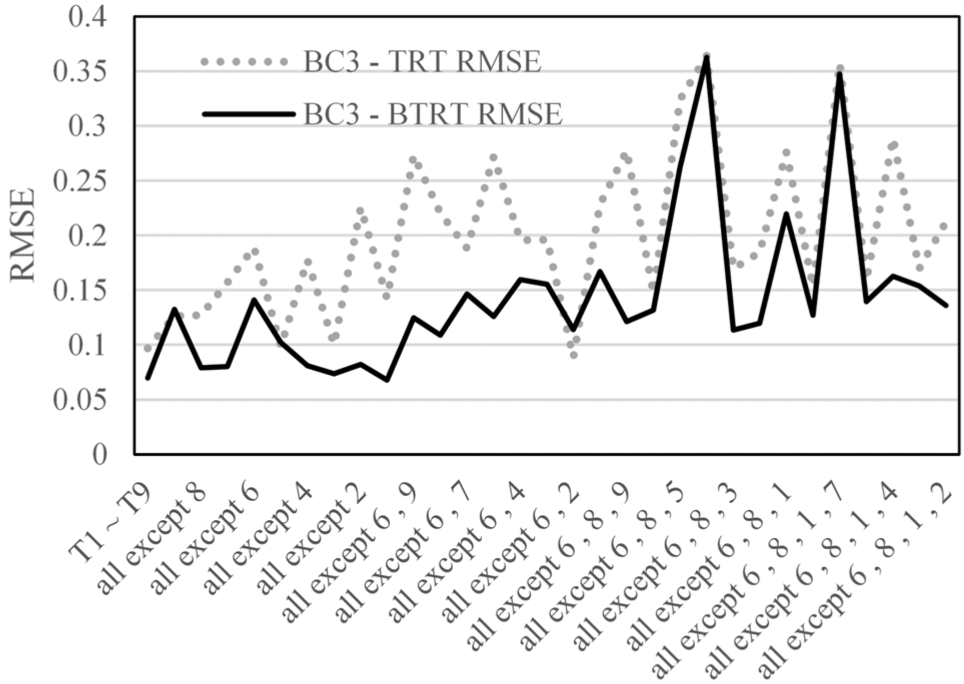

Figure 16. RMSE plots versus different thermocouple compositions for BC3.

$$\begin{cases} T = T_0, x = x_0 \\ h = cont., x = L \end{cases}$$

$$\begin{cases} T = T_0, x = x_0 \\ \dot{q} = cont., x = L \end{cases}$$

A combination of equation 7 with equations 8a, 8b, or 8c can be equivalent in complexity to the boundary conditions of BC1, BC2, and BC3, respectively. The solution to the mentioned heat transfer problems is available in the textbooks [25]. The first problem with two constant-temperature boundary conditions (8a) yields a linear temperature distribution throughout the model. The second one (8b) is nonlinear, even if the body's thermal conductivity is defined as constant (linear problem in the present paper) or temperature-dependent (nonlinear problem). The last problem (8c) can cause a linear temperature distribution in the linear thermal conductivity and a nonlinear temperature distribution in the nonlinear problem. Therefore, the prediction of BC1 and BC3 in the linear problem is easier with fewer sensors than BC2. But defining a temperature-dependent thermal conductivity for the body, BC3, can also cause a nonlinear temperature distribution, which increases the difficulty of BC3 prediction by the inverse method. Finally, BC2 affects both the heat flux and the temperature of the surface on which BC2 is applied. Therefore, the prediction of BC2 is more complicated than the other two BCs in both linear and nonlinear problems.

For the nonlinear problem, all possible three-sensor combinations are tested to predict the unknown boundary conditions to investigate the present algorithm's performance with three sensor data. Table 4 presents the two best and the worst performance measures of the developed regression algorithms with three sensors. As shown in Table 4, the TRT

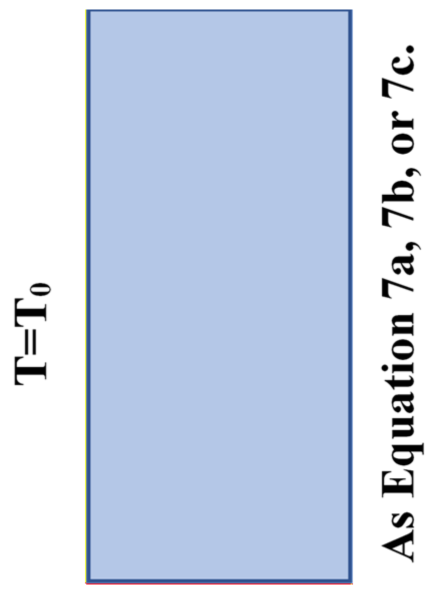

Figure 17. RMSE plots versus different thermocouple compositions for BC3.

and BTRT algorithms exhibit better performance compared to the RT, as seen for the linear problem in Table 2. The gap between the MAE in the worst and the best case is extremely high in the TRT and BTRT algorithms. Additionally, it can be observed that the performance of the BTRT algorithm surpasses that of the TRT in predicting BC2 and BC3, which are more challenging boundary conditions than BC1.

Table 5. The best and the worst performance measures for linear Flux with three sensors among nine sensors

	BC1						BC2						BC3					
	RT		TRT		BTRT		RT		TRT		BTRT		RT		TRT		BTRT	
Used sensors	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
1, 4, 9	0.555	13.02	0.002	0.023	0.002	0.022												
4, 6, 7	0.543	12.66	0.004	0.08	0.004	0.075												
3, 5, 6	0.676	15.17	0.083	1.892	0.065	1.552												
4, 7, 8	0.555	13.4	0.004	0.124	0.004	0.122												
1, 8, 9							3.036	71.01	0.124	1.72	0.055	0.541						
1, 6, 9							3.117	74.63	0.148	3.027	0.055	0.728						
2, 6, 7							5.73	136.2	2.154	74.62	1.844	73.48						
2, 5, 6													2.24	42.72	0.012	0.122	0.015	0.302
2, 5, 7													2.285	45.61	0.015	0.123	0.014	0.136
3, 8, 9													3.648	77.1	1.104	57.58	1.046	57.53

Figure 18 shows the maximum and minimum RMSE of BC2 predicted by the TRT and BTRT methods versus the number of sensors. As seen in Figure 18, the maximum error decreases dramatically as the number of sensors increases. But the minimum error does not show a considerable improvement by increasing the number of sensors. Therefore, it justifies that the appropriate locating of the sensors can be more effective than increasing the number of sensors.

Figure 19 graphically shows the sensor's location which have the best performance among the three sensor combinations as presented in Table 5. From the first observation on the model schematic (Figure 1), it is expected that the combination of sensors that are closer to the unknown boundary condition should have better performance than the other combinations. But Figure 18 shows a different pattern, especially for BC1. For more

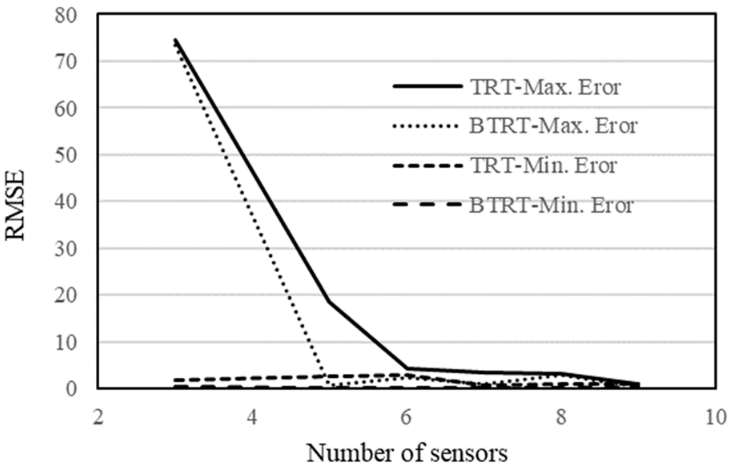

FIGURE 18. RMSE versus the number of sensors to predict BC2

comparison, the results of the three sensors just near BC1 (4,7 & 8) are also listed in Table 5. As seen, the performance of sensors 1, 4, and 9 by the TRT and BTRT algorithms, especially in RMSE errors, is considerably better. Therefore, it emphasizes the importance of the present investigations to determine the optimized location and number of sensors before any experimental measurements to achieve better inverse results with a minimum number of sensors.

5. Conclusion

In this research, for the first time, the regression tree (RT), trended regression tree (TRT), and bounded trended regression tree (BTRT) methods were used to solve the linear and non-linear heat transfer problems. To investigate the methods' efficiency, the estimation of three different types of boundary conditions (BC) was considered, and the following results were obtained:

- The RT's algorithm results in relatively huge errors, especially when predicting BCs with non-linearities compared to the TRT and BTRT methods.

- In the linear heat transfer problem, the TRT algorithm yields zero MAE and RMSE for the linear BCs in most cases, even with less sensory data.

- For the nonlinear BC prediction, MAE and RMSE results for TRT and BTRT methods are closer together, but, in some cases, with lower sensory data, the performance of the BTRT is better than the TRT.

- Comparing three sensor composition results reveals that it is unlike expectations; using the data from the sensors closer to the unknown BC does not always guarantee the accuracy of inverse results.

According to the present research results, it can be claimed that the RT algorithms have high potential for solving inverse heat transfer problems with acceptable errors. For linear problems, the RT and TRT methods had acceptable and excellent performance, respectively. But, while trying to predict a nonlinear BC in a nonlinear heat transfer problem with lower sensory data, the BTRT algorithm can be a better choice.

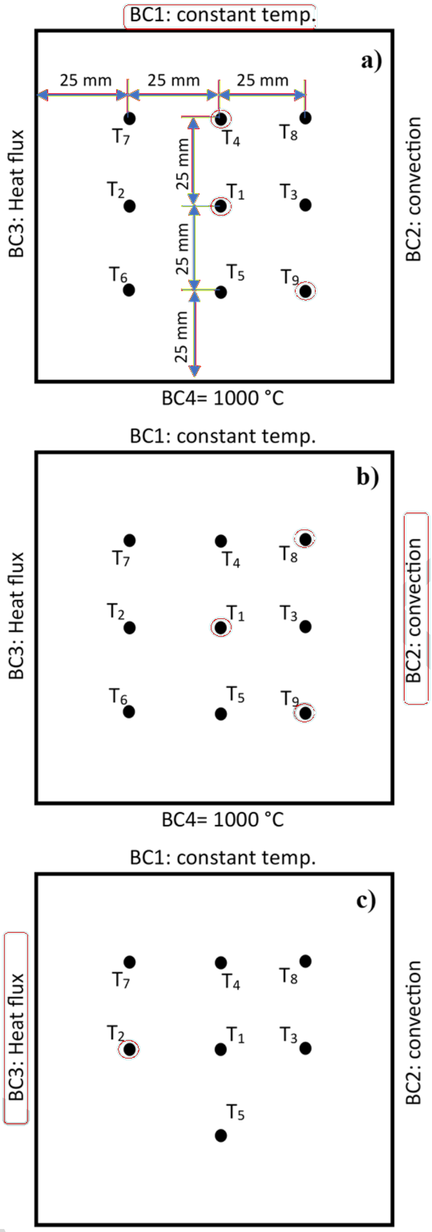

FIGURE 19. location of the sensors with the best performance in prediction of (a) BC1, (b) BC2, and (c) BC3.

References

1. M. Raudensky, K. A. Woodbury, J. Kral, and T. Brezina, GENETIC ALGORITHM IN SOLUTION OF INVERSE HEAT CONDUCTION PROBLEMS, Numer. Heat Transf. , Part B Fundam. An Int. J. Comput. Methodol., vol. 28, no. 3, pp. 293–306, 1995.
2. A. Imani, A. A. Ranjbar, and M. Esmkhani, Simultaneous estimation of temperature-dependent thermal conductivity and heat capacity based on modified genetic algorithm, Inverse Probl. Sci. Eng., vol. 14, no. 7, pp. 767–783, 2006, doi: 10.1080/17415970600844242.

3. M. Mazidi Sharfabadi, M. Alizadeh, and L. Nourpour, The Modification of the Conventional Genetic Algorithm for Solving Inverse Heat Transfer Problems, Modares Mech. Eng., vol. 17, no. 11, pp. 408-418 (in Persian), 2018.
4. S. Deng and Y. Hwang, Applying neural networks to the solution of forward and inverse heat conduction problems, Int. J. Heat Mass Transf., vol. 49, no. 25–26, pp. 4732–4750, 2006, doi: 10.1016/j.ijheatmasstransfer.2006.06.009.
5. M. Famouri, M. Jannatabadi, and H. T. F. Ardakani, Simultaneous estimations of temperature-dependent thermal conductivity and heat capacity using a time efficient novel strategy based on MEGA-NN, Appl. Soft Comput. J., vol. 13, no. 1, pp. 201–210, 2013, doi: 10.1016/j.asoc.2012.08.023.
6. B. Ghadimi, F. Kowsary, and M. Khorami, Heat flux on-line estimation in a locomotive brake disc using artificial neural networks, Int. J. Therm. Sci., vol. 90, pp. 203–213, 2015, doi: 10.1016/j.ijthermalsci.2014.12.012.
7. H. R. Tamaddon-Jahromi, N. K. Chakshu, I. Sazonov, L. M. Evans, H. Thomas, and P. Nithiarasu, Data-driven inverse modelling through neural network (deep learning) and computational heat transfer, Comput. Methods Appl. Mech. Eng., vol. 369, 2020, doi: 10.1016/j.cma.2020.113217.
8. V. Norouzifard and M. A. Movafaghpour, A new method to solve non-linear inverse heat transfer problems using regression tree method, J. Mech. Eng. (University Tabriz), vol. 52, no. 3, pp. 347–356, 2022.
9. A. K. Sen and M. S. Srivastava, Regression Analysis: Theory, Methods, and Applications. New York: Springer, 2012.
10. Y. Bai, P. Wang, C. Li, J. Xie, and Y. Wang, A multi-scale relevance vector regression approach for daily urban water demand forecasting, J. Hydrol., vol. 517, pp. 236–245, Sep. 2014, doi: 10.1016/j.jhydrol.2014.05.033.
11. K. Venkatesh, V. Ravi, A. Prinzie, and D. Van Den Poel, Cash demand forecasting in ATMs by clustering and neural networks, Eur. J. Oper. Res., vol. 232, no. 2, pp. 383–392, 2014, doi: 10.1016/j.ejor.2013.07.027.
12. P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, Modeling wine preferences by data mining from physicochemical properties, Decis. Support Syst., vol. 47, no. 4, pp. 547–553, Nov. 2009, doi: 10.1016/j.dss.2009.05.016.
13. E. Davis and M. Ierapetritou, A kriging-based approach to MINLP containing black-box models and noise, Ind. Eng. Chem. Res., vol. 47, no. 16, pp. 6101–6125, Aug. 2008, doi: 10.1021/ie800028a.
14. J. A. Caballero and I. E. Grossmann, An algorithm for the use of surrogate models in modular flowsheet optimization, AIChE J., vol. 54, no. 10, pp. 2633–2650, Oct. 2008, doi: 10.1002/aic.11579.
15. J. Beck, D. Friedrich, S. Brandani, S. Guillas, and E. S. Fraga, Surrogate based Optimisation for Design of Pressure Swing Adsorption Systems, in Computer Aided Chemical Engineering, vol. 30, Elsevier B.V., 2012, pp. 1217–1221.
16. F. A. C. Viana, T. W. Simpson, V. Balabanov, and V. Toropov, Metamodeling in multidisciplinary design optimization: How far have we really come?, in AIAA Journal, 2014, vol. 52, no. 4, pp. 670–690, doi: 10.2514/1.J052375.
17. C. A. Henao and C. T. Maravelias, Surrogate-based superstructure optimization framework, AIChE J., vol. 57, no. 5, pp. 1216–1232, May 2011, doi: 10.1002/aic.12341.
18. Y. Zhang and N. V. Sahinidis, Uncertainty quantification in CO2 sequestration using surrogate models from polynomial chaos expansion, Ind. Eng. Chem. Res., vol. 52, no. 9, pp. 3121–3132, Mar. 2013, doi: 10.1021/ie300856p.
19. A. Nuchitprasittichai and S. Cremaschi, An algorithm to determine sample sizes for optimization with artificial neural networks, AIChE J., vol. 59, no. 3, pp. 805–812, Mar. 2013, doi: 10.1002/aic.13871.
20. Q. L. Chen, K. J. Wu, and C. H. He, Thermal conductivity of ionic liquids at atmospheric pressure: Database, analysis, and prediction using a topological index method, Ind. Eng. Chem. Res., vol. 53, no. 17, pp. 7224–7232, Apr. 2014, doi: 10.1021/ie403500w.
21. K. J. Wu, Q. L. Chen, and C. H. He, Speed of sound of ionic liquids: Database, estimation, and its application for thermal conductivity prediction, AIChE J., vol. 60, no. 3, pp. 1120–1131, Mar. 2014, doi: 10.1002/aic.14346.
22. E. Carrizosa, C. Molero-Río, and D. Romero Morales, Mathematical optimization in classification and regression trees, TOP, vol. 29, no. 1, pp. 5–33, Apr. 2021, doi: 10.1007/s11750-021-00594-1.
23. E. B. Prado, R. A. Moral, and A. C. Parnell, Bayesian additive regression trees with model trees, Stat. Comput., vol. 31, no. 3, Apr. 2021, doi: 10.1007/s11222-021-09997-3.

24. M. J. Peet, H. S. Hasan, and H. K. D. H. Bhadeshia, Prediction of thermal conductivity of steel, Int. J. Heat Mass Transf., vol. 54, no. 11–12, pp. 2602–2608, 2011, doi: 10.1016/j.ijheatmasstransfer.2011.01.025.
25. TL Bergman, A. Lavine, and F. Incropera, Introduction to heat transfer. John Wiley & Sons, 2011.
26. A. Saberi, E. , Radmand, E. , Pirgazi, J. and Kermani, Buying and selling strategy in the Iranian stock market using machine learning models, with feature selection using the Cuckoo Search algorithm, soft Comput. J., vol. 12, no. 2, pp. 130–145, 2024, doi: 10.22052/scj.2023.252793.1144.
27. M. Abbasi and S. Atiyeh, A Novel Hybrid Method for Network Intrusion Detection System Using Computational Intelligence Algorithms, soft Comput. J., vol. 6, no. 1, pp. 48–65, 2021.
28. R. Hoseini, S. M. , Ebtia, M. and Khochiani, An ensemble method based on bagging SVM for credit rating problem, soft Comput. J., 2024, doi: 10.22052/scj.2024.246330.1063.
29. M. J. Nasri-lowshani and S. Goli-bidgoli, Investigating the Performance of Machine Learning Methods for Link Quality Estimation in Wireless Networks, soft Comput. J., 2025, doi: 10.22052/scj.2025.255691.1280.