Received: 30 December 2023 | Revised: 21 March 2024 | Accepted: 8 April 2024 | Published online: 15 April 2024

RESEARCH ARTICLE

Journal of Data Science and Intelligent Systems
2024, Vol. 00(00) 1–9
DOI: 10.47852/bonviewJDSIS42022395

BON VIEW PUBLISHING

Monte Carlo Simulation-Based Regression Tree Algorithm for Predicting Energy Consumption from Scarce Dataset

Tony Darmanto[1],*, Jimmy Tjen[2] and Genrawan Hoendarto[2]

[1]*Department of Information System, Widya Dharma Pontianak University, Indonesia*

[2]*Department of Informatics, Widya Dharma Pontianak University, Indonesia*

Abstract: Most data-driven techniques rely on the availability of data. Hence, when the data provided are not sufficient, the algorithm might not work as intended. Thus, it is important to be able to predict the dynamics of the data, even when the number of available data is low, or scarce. This study aimed to predict the power consumption of a building given a scarce dataset via a novel Monte Carlo simulation-based Regression Tree (MCRT) algorithm. The main idea is to train Monte Carlo simulation on each leaf generated by the regression tree algorithm. Thus, the prediction no longer depends on the average of the samples contained in the leaf, but now depends on the probability of the samples. The proposed algorithm was validated on 2 datasets obtained from Universitas Widya Dharma Pontianak (UWDP), Indonesia, and Trapeznikov Institute of Control Sciences (TICS), Russia. To show that the MCRT algorithm is better than the regression tree (RT) algorithm, a two-tail hypothesis was proposed. Based on the experiments which were run on Python software with 16 GB RAM, 7th Gen Core i7 machine on 50 datasets randomly generated from the UWDP electrical data, it can be concluded that the MCRT algorithm performs better than the previous RT algorithm used to model scarce datasets with P-value = 0.000319. Furthermore, the proposed algorithm improves the model predictive accuracy of the RT algorithm by up to 2%.

Keywords: Monte Carlo simulation, regression tree, power consumption, scarce dataset

1. Introduction

This paper aims to study the performance of Monte Carlo (MC) simulation-based regression tree (RT) in predicting power consumption inside a building given a scarce dataset, i.e., when there are a lot of missing data points in the dataset. Along with the development of technology, especially in an era where human "tedious" tasks are being handled by Artificial Intelligence, it is not unusual to find a building equipped with various kinds of sensors that can precisely measure the dynamics of the building [1, 2]. This information is useful as a mathematical model can then be derived from it in order to simulate the future dynamics of the structure [3]. The approach based on this information is called the data-driven approach [4–6]. The data-driven approach is a machine-learning technique that focuses on learning mathematical models to represent the dynamics of a dataset. This approach is somehow easier to deploy rather than the model-based approach, which depends on a strong knowledge of the structure's physical properties. Various methods apply this approach. Some of them are MC simulation [7–10] and the RT from Decision Tree Learning [11–13].

1.1. Previous work

The MC simulation is a stochastic model that predicts an outcome of a process based on the probability of random variables [14]. In Hoendarto et al. [8], the authors proposed an idea to predict the power consumption inside a structure with the MC simulation. In their research, they have shown that the MC algorithm can predict the power consumption inside a building with an accuracy of more than 90%. In Mardani Najafabadi and Taki [9], the MC simulation was used to optimize the energy consumption of a cucumber greenhouse located in Golshan, Iran. It was shown that the MC simulation helped to control the gas and improve the energy consumption efficiency inside the greenhouse via precise modeling. In the study by Shen et al. [10], the authors employed a combination of MC simulation and support vector regression model to forecast household power consumption. Their research demonstrated that this approach led to a notable 12% reduction in power consumption. Similarly, Ding et al. [7] utilized MC simulation and probability density analysis to predict nuclear energy consumption in China and America. Their findings suggested that the proposed model showed promise in accurately forecasting nuclear energy consumption from 2019 to 2023.

The RT algorithm, instead, is an algorithm that focuses on splitting the dataset based on the similarities of features presented in the dataset. The idea is to group the whole dataset into smaller

*Corresponding author: Tony Darmanto, Department of Information System, Widya Dharma Pontianak University, Indonesia. Email: tony_d@widyadharma.ac.id

subsets of data based on their variance. Smarra et al. [12] used the RT algorithm together with entropy (referred to as entropy-based subset selection RT or e-ss RT) from information theory to detect the presence of faults inside a building based on seismic dynamic data recorded by accelerometers located within a structure. They have shown that the entropy-based RT can potentially detect damages within a structure while also requiring significantly fewer parameters compared to the classical RT algorithm (around 5) to detect faults. The research paper by Panjaitan et al. [11] showed that the random forest, which is a collection of RTs, can precisely model the energy consumption of a building with an accuracy above 90%. Recently, Tjen et al. [13] modified the algorithm proposed by Smarra et al. [12] to predict power consumption in a university building located in Pontianak, Indonesia. They have shown that the modified RT algorithm can precisely predict power consumption inside the building even when the available data are scarce.

In particular, the papers stated above have shown the capability of RT and MC simulation in predicting the dynamics of data. However, it is well known that data-driven approach performance depends mostly on the availability of the dataset [15]. Hence, when only a few data exist, the approach might not produce optimal results [16–18].

1.2. Contributions

As mentioned earlier, data-driven approaches such as MC simulation and RT algorithms may not yield the desired outcome when dealing with scarce datasets. However, there must be a way to achieve precise predictions even with limited data. Therefore, this paper is organized to highlight three contributions:

1) The enhancement of the RT algorithm proposed in Tjen et al. [13] by applying MC simulation in each leaf of the tree generated by the RT algorithm, thereby improving prediction accuracy.
2) The comparison between the Monte Carlo-based Regression Tree (MCRT) algorithm to the RT algorithm proposed in Tjen et al. [13] and followed by the proof that the MCRT algorithm indeed enhances the predictive accuracy of the previous RT algorithm.
3) The methodology was validated on 2 real datasets: UWDP, Indonesia, and Trapeznikov Institute of Control Sciences (TICS), Russia.

This paper follows the methodology presented by Smarra et al. [12] and Tjen et al. [13]. Specifically, this paper highlights the modification of the RT algorithm by incorporating the MC simulation proposed by Hoendarto et al. [8] into each leaf of the tree generated by the RT algorithm. Thus, instead of deriving the prediction as the mean of the samples contained in the leaves, the MCRT algorithm predicts the outcome of a process using MC simulation. The details of this algorithm will be discussed in the next section.

This paper is organized as follows: the first part is the introduction. Section 2 will discuss the proposed algorithm and how it is constructed to handle scarce datasets. Section 3 will explain the datasets used in this paper. In Section 4, the discussion on the numerical results of the proposed algorithm is presented, and finally, the last section will conclude the research.

2. Proposed Algorithm

This section will mainly discuss the proposed algorithm. Specifically, this section will explain how to modify the RT algorithm to accommodate scarce datasets and then demonstrate how to integrate MC with the RT algorithm. Readers are encouraged to refer to Breiman et al. [19], Loh [20], Lewis [21], and Jain et al. [22] for basic concepts related to the RT algorithm and training models within RT-based model leaves. Additionally, it is recommended to read Betz et al. [23] and Xie [24] for the discussion of MC simulation-based algorithms.

2.1. Monte Carlo-based regression tree approach

The MCRT algorithm consists of 3 main parts: (1) RTs model derivation, (2) assigning the MC model to leaves, and (3) MCRT model prediction. To have a better illustration, let us assume the following case: let $X \in [X_D\ X_E]; X \in \mathbb{R}^{m\times n}$ be an electrical dataset that is related to a structure, where $X_D = [\boldsymbol{y}\ \boldsymbol{m}\ \boldsymbol{d}\ \ \boldsymbol{h}\ \boldsymbol{m_i}\ \boldsymbol{s}\ ];\ X_D \in \mathbb{R}^{m\times n_d}$ is a matrix that contains the time information such as year($\boldsymbol{y}$), month($\boldsymbol{m}$), day($\boldsymbol{d}$), hour($\boldsymbol{h}$), minutes($\boldsymbol{m_i}$)) and second($\boldsymbol{s}$), while $X_e = [x_1\ x_2\ \cdots\ x_{me}]; X_e \in \mathbb{R}^{m\times n_d}$ is a matrix that contains the electrical parameters (e.g., voltage, current, active power, etc.,). Suppose that X is scarce, i.e., for any time instance t, $X(t+1) = [X_D(t+1)\ X_E(t+1)]$ might not be (while it is possible) the direct continuation of $X(t)$. Given X, suppose that the goal of this process is to predict the outcome of a certain electrical parameter, $x_i \in X_E$ at instance k, namely $x_i(k)$. Then, the prediction for $x_i(k)$ which is $\hat{x}_i(k)$ via MCRT algorithm can be found by following these steps:

Step 1. Let $E = \left[x_j\ x_{j+1}\ x_{j+2} \ldots x_{jm}\right]; E \in \mathbb{R}^{m\times p}; E \subset X_E$ be a matrix that contains all electrical parameters that are related to x_i, where $j = 1, 2, 3, \ldots, m_e$. The first step is to generate an RT model for each parameter in E. In particular:

$$\begin{aligned}\hat{x}_j(k) &= f_{RT1}(x_D(k)),\\ \hat{x}_{j+1}(k) &= f_{RT2}(x_D(k)),\\ &\vdots\\ \hat{x}_{jm}(k) &= f_{RTm_e}(x_D(k))\end{aligned} \tag{1}$$

where $x_D(k) = [y(k)m(k)d(k)h(k)m_i(k)s(k)]$ is the time parameter at instance k. Let $\hat{x}_j(k) = \left[\hat{x}_j(k)\ \hat{x}_{j+1}(k) \ldots \hat{x}_{jm}(k)\right]$ be a vector of prediction. Given Equation (1), then it is possible to derive the RT model for $\hat{x}_i$ as:

$$\hat{x}_i(k) = f_{RTi}\left(x_D(k), \hat{x}_j(k)\right) \tag{2}$$

At this point, if E is chosen carefully, e.g., via the entropy analysis as done in Tjen et al. [13], it is already possible to obtain the prediction for $\hat{x}_i(k)$. However, in this case, the prediction power of the model in Equation (2) will be enhanced by assigning an MC model to each leaf of the tree generated by Equation (2).

Step 2. Let T denote the RT model as in Equation (2) and $l_1, l_2 \ldots, l_a$ be leaves corresponding to the tree T. Let $l_f(\tilde{x}(k)) = \{a : \tilde{x}(k) \in l_a\}$ be a function that assigns the value $\tilde{x}(k) = \left[x_D(k), \hat{x}_j(k)\right]$ to the right leaf l_a in T. And finally, let $x_{l_f(\tilde{x}(k))}$ be the vector that contains the value of x_i in a specific leaf due to $l_f(\tilde{x}(k))$ (i.e., x_{l_2} contains all samples of x_i belonging to l_2 etc.,). Then for each leaf, with a slight abuse of notation, it is possible to assign a random variable:

$$A_{l_f(\tilde{x}(k))} = \begin{cases} 1 \text{ if } x_i(k) < q_{l_f(\tilde{x}(k))}(1) \\ 2 \text{ if } q_{l_f(\tilde{x}(k))}(1) \le x_i(t) < q_{l_f(\tilde{x}(k))}(2) \\ 3 \text{ if } q_{l_f(\tilde{x}(k))}(2) \le x_i(t) < q_{l_f(\tilde{x}(k))}(3) \\ 4 \text{ if } x_i(k) \ge q_{l_f(\tilde{x}(t))}(3) \end{cases} \quad (3)$$

where $q_{l_f(\tilde{x}(t))} = \left[q_{l_f(\tilde{x}(t))}(1)\ q_{l_f(\tilde{x}(t))}(2)\ q_{l_f(\tilde{x}(t))}(3)\right]$ is a vector that contains the value of 1st, 2nd (also known as median) and 3rd quartile of $x_{l_f(\tilde{x}(t))}$. Let

$$p_{l_f(\tilde{x}(k))} = \left[p_{l_f(\tilde{x}(k))}(1)\ \ldots\ p_{l_f(\tilde{x}(k))}(4)\right] \quad (4)$$

be a vector that contains the probability of samples taking values from 1 to 4 from each corresponding random variable $A_{l_f(\tilde{x}(k))}$ and

$$b_{l_f(\tilde{x}(k))} = \left[b_{l_f(\tilde{x}(k))}(1)\ \ldots\ b_{l_f(\tilde{x}(k))}(4)\right] \quad (5)$$

be a vector that contains the average of samples x_i corresponding to the value of $A_{l_f(\tilde{x}(k))} = 1$ until $A_{l_f(\tilde{x}(k))} = 4$, i.e., $b_{l_f(\tilde{x}(k))}(n)$ is the average of samples x_i which satisfies the condition $A_{l_f(\tilde{x}(k))} = n$.

Step 3. Given Equations (3–5), it is now possible to estimate the value of $\hat{x}_i(k)$. For a vector of random number $x_r \in [0, 1]^{n_r}$, where $o = 1, 2, \ldots,\ n_r$, let

$$g(o) = \begin{cases} b_{l_f(\tilde{x}(k))}(1) \text{ if } x_r(o) < p_{l_f(\tilde{x}(k))}(1) \\ b_{l_f(\tilde{x}(k))}(2) \text{ if } x_r(o) \in \left[p_{l_f(\tilde{x}(k))}(1), \sum_{i=1}^{2} p_{l_f(\tilde{x}(k))}(i)\right) \\ b_{l_f(\tilde{x}(k))}(3) \text{ if } x_r(o) \in \left[\sum_{i=1}^{2} p_{l_f(\tilde{x}(k))}(i), \sum_{i=1}^{3} p_{l_f(\tilde{x}(k))}(i)\right) \\ b_{l_f(\tilde{x}(k))}(4) \text{ if } x_r(o) \ge \sum_{i=1}^{3} p_{l_f(\tilde{x}(k))}(i) \end{cases}$$

Given g, the MCRT estimate for $x_i(k)$ is defined as

$$\hat{x}_i(k) = \frac{1}{n_r} \sum_{i=1}^{n_r} g(i) \quad (6)$$

Algorithm 1 shows the pseudocode for the MCRT algorithm. Concerning the time complexity, the RT algorithm has a time complexity of $O(m \times n^2)$ [25] where m denotes the number of samples and n is the number of features, while the MC simulation is estimated to run with the time complexity of $O(m^2)$ [26]. In our use case, since the calculation is repeated for a many times of leaves, with each leaf having m_a samples, the overall complexity for our algorithm is in $O(\sum_{i=1}^{a} m_i^2)$ which is less than $O(m^2)$ as $\sum_{i=1}^{a} m_i = m$. Thus, $\sum_{i=1}^{a} m_i^2 \le m^2$ due to the quadratic expansion. Hence, the whole MCRT algorithm runs within the time complexity of $O(m^2 + m \times n^2) = O(m(m + n^2))$. Note that this estimation toward time complexity is the upper bound for the algorithm, as $O(\sum_{i=1}^{a} m_i^2) \le O(m^2)$. With a proper setup in step 1 (e.g., by using the feature selection proposed in Smarra et al. [12], it is possible to select a minimum number of features, resulting in faster execution of the entire algorithm.

Algorithm 1: Monte Carlo-Based Regression Tree

Input : $X = [X_D\ X_E]$, $\tilde{x}$, n_r, E, x_i
Output : $\hat{x}_i$
Process :
Initialization
$q = [\,]$
$p = [\,]$
$b = [\,]$

RTs model derivation
for x *in* E
  $\hat{x}_j = regression\_tree(X_D,\ x_j)$
end for
$\hat{x}_i = regression\_tree([X_D\ \hat{x}_j],\ x_i)$

Assigning MC model to leaves
$l = number\_of\_leaf(\hat{x}_i)$
for $i = 1\ : l$
  $x_l = leaf\_sample(T, x_i, l)$
  $q(i,:) = quartile([0.25\ 0.5\ 0.75], x_l)$
  $n_l = number\_of\_sample(x_l)$
  $p\_temp = zeros(4, 1)$
  $b\_temp = zeros(4, 1)$
  for $j\ =\ 1\ :\ length(x_l)$
    if $x_l(j) < q(l, 1)$
      $p\_temp(1) = p\_temp(1) + 1$
      $b\_temp(1) = b\_temp(1) + _l(j)$
    elseif $q(l, 1) \le x_l(j) < q(l, 2)$
      $p\_temp(2) = p\_temp(2) + 1$
      $b\_temp(2) = b\_temp(2) + x_l(j)$
    elseif $q(l, 2) \le x_l(j) < q(l, 3)$
      $p\_temp(3) = p\_temp(3) + 1$
      $b\_temp(3) = b\_temp(3) + x_l(j)$
    else
      $p\_temp(4) = p\_temp(4) + 1$
      $b\_temp(4) = b\_temp(4) + x_l(j)$
    end if
  end for
  $p(l,:) = p\_temp/length(x_l)$
  $b(l,:) = b\_temp/length(x_l)$
end for

MCRT model prediction
$l_t = target\_leaf(T, \tilde{x})$
$x_r = rand([0\ 1], n_r)$
$g = [\,]$
for $i = 1 : length(x_r)$
  if $x_r(i) < p(l, 1)$
    $g(i) = b(l, 1)$
  elseif $p(l, 1) \le\ x_r(i) < sum(p(l, 1 : 2))$
    $g(i) = b(l, 2)$
  elseif $sum(p(l, 1 : 2)) \le\ x_r(i) < sum(p(l, 1 : 3))$
    $g(i) = b(l, 3)$
  else
    $g(i) = b(l, 4)$
  end if
end for
$\hat{x}_i = mean(g)$

3. Research Methodology

In this section, the discussion will commence with an overview of the datasets utilized in this research. Specifically, two datasets, namely UWDP from Indonesia and TICS from Russia, will be

examined. Additionally, an explanation will be provided on how the algorithm discussed in Section 2 was adjusted to accommodate the unique requirements of the chosen datasets. Subsequently, the methodology employed for conducting numerical simulations in this research will be expounded upon. This will include a detailed description of the simulation procedures, parameter configurations, and performance metrics utilized to assess the algorithm's effectiveness. Through rigorous numerical simulations, the goal is to demonstrate the efficacy of the algorithm and its capacity to yield meaningful insights from real-world data.

3.1. 1st case study: Universitas Widya Dharma Pontianak (UWDP)

The first electrical dataset was obtained from a University building located in Pontianak, West Kalimantan Indonesia. The data for this 10-floor building were provided by the Indonesian state electricity company or PLN (Indonesian: *Perusahaan Listrik Negara*), in which the dataset consists of 3,113 samples with 18 features: date, times of the day, frequency, and 3 phases of voltages (V), currents (A), active powers (W), reactive powers (VA), and apparent powers (VAR). Figure 1 shows the picture of the UWDP main campus building.

Due to unknown reasons, the UWDP dataset provided by PLN consists of only 3,113 samples while it was measured for about 3 years, dated from 1st January 2020 up to 27th February 2023. Furthermore, the dataset was recorded completely at a random period and there is a gap from 1 sample to the next consecutive sample (i.e., the next data are recorded on a different date from the previous data).

Figure 1
Universitas Widya Dharma Pontianak (UWDP) main building located in West Kalimantan, Indonesia

3.2. 2nd case study: Trapeznikov Institute of Control Sciences (TICS)

The second electrical dataset was obtained from the TICS which is located in Russia. The dataset is measured from 1st January 2021 until 31st December 2021 and sampled every second. The dataset was collected from the administrative and laboratory buildings. The administrative building has 30 feeders while the laboratory building has 65 feeders. The whole data were separated into monthly datasets, where each dataset consists of around 300,000 samples (varied for each month) with 21 features, where it has the same 18 features as in the UWDP dataset + another 3 phases of total harmonic distortion data. Figure 2 shows the picture of the TICS building.

Figure 2
Trapeznikov Institute of Control Science (TICS) main building located in Trapeznikov, Russia

3.3. Power consumption via MCRT algorithm

The goal of this research is to develop an algorithm capable of predicting the power consumption of a building even with a scarce dataset. In this regard, regarding the proposed algorithm in Section 2, the total active power (P) is chosen to be the x_i for both case studies. In addition, only three phases of currents, namely I_1, I_2, and I_3, were taken into account as parameters related to P. This choice stems from findings presented in Tjen et al. [13], which demonstrate that currents are the most influential parameters in predicting power consumption. In particular, the first step of the MCRT algorithm shown in Section 2 yields the following RT models:

$$\hat{I}_1(k) = f_{RT1}(x_D(k)),$$

$$\hat{I}_2(k) = f_{RT2}(x_D(k)),$$

$$\hat{I}_3(k) = f_{RT3}(x_D(k)),$$

$$\hat{P}(k) = f_{RTP}\left(x_D(k), \hat{I}_1(k), \hat{I}_2(k), \hat{I}_3(k)\right) \tag{7}$$

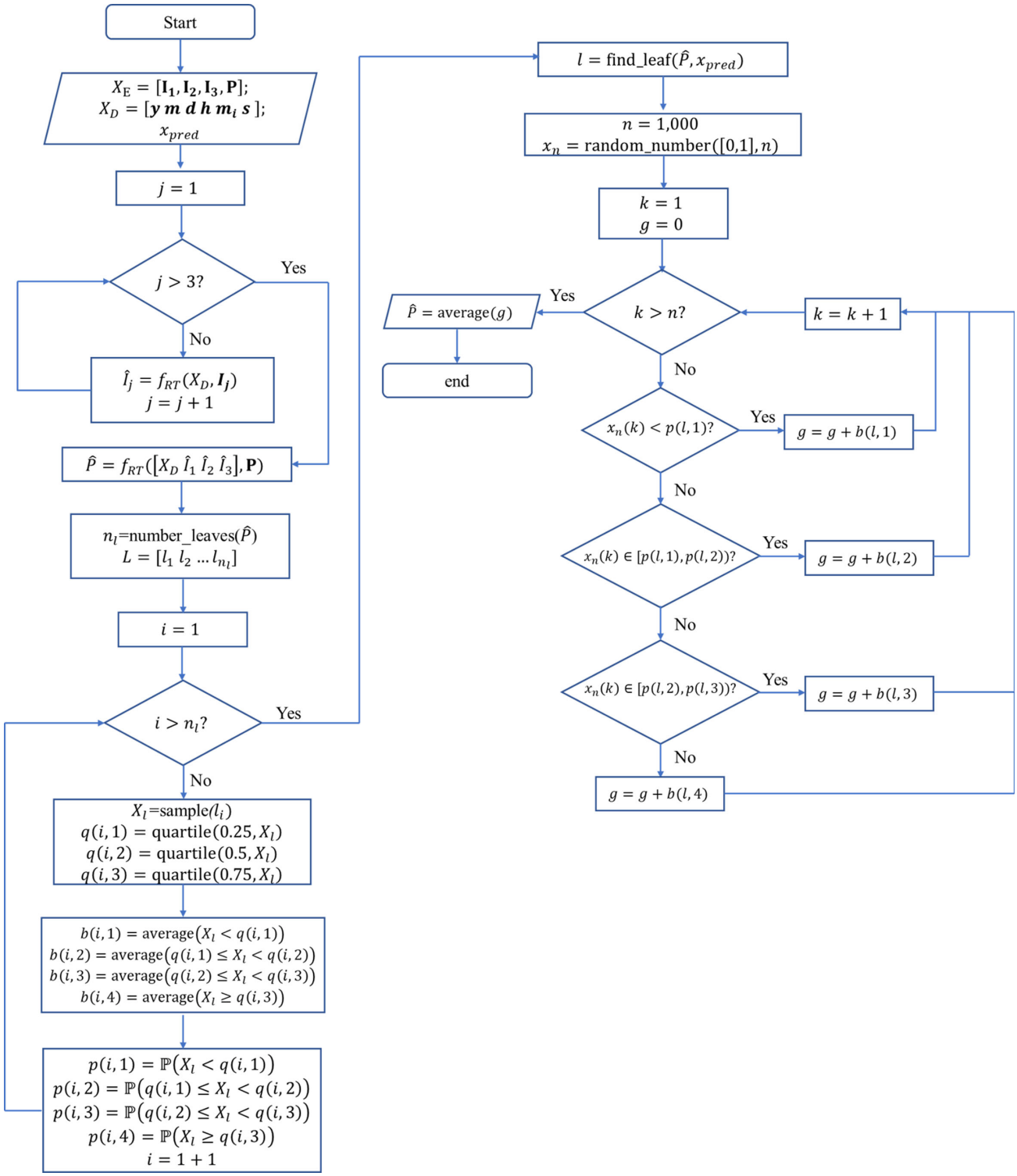
The $n_r = 1000$ was considered for the random numbers in the 3rd step. The choice was made by considering the trade-off between the time complexity and the model predictive accuracy of the algorithm.

In the first use case, the entire dataset was considered, whereas for the second use case, only the dataset from June 2021 was utilized. This choice was informed by the average temperature observed during June, which corresponds to the onset of summer and closely resembles the climate in West Kalimantan. Figure 3 shows the block diagram of the process of predicting the power consumption with the MCRT algorithm.

3.4. Experimental setup

The numerical simulation comprises two main parts: testing the hypothesis and measuring the model's predictive accuracy. It was

Figure 3
Block diagram for predicting the power consumption via the MCRT Algorithm



asserted that the MCRT algorithm enhances the predictive accuracy of the RT algorithm introduced in Tjen et al. [13], which was developed after the e-ss RT proposed in Smarra et al. [12]. To validate the performance of the proposed algorithm, a t-test was conducted on the following hypothesis:

H_0 : The model predictive accuracy of the MCRT algorithm is equal to the RT algorithm.

H_a : The model predictive accuracy of the MCRT algorithm is not equal to the RT algorithm.

The hypothesis testing was done by considering the model predictive accuracy of both algorithms for 50 different datasets generated by taking randomly 50% of the available samples from the 1st dataset as the training dataset and use the rest to validate the model accuracy. The model predictive accuracy in this case is represented by the Normalized Root Mean Square Error (NRMSE) as in Equation (8):

$$A(y,\hat{y}) = (1 - NRMSE(y,\hat{y})) \times 100\%$$
$$NRMSE(y,\hat{y}) = \frac{1}{\sqrt{n}\cdot\bar{y}}\sqrt{\sum_{i=1}^{n}(y(i)-\hat{y}(i))^2} \quad (8)$$

where y is the observed data, $\hat{y}$ is the predicted data, n is the number of samples, and $\bar{y}$ is the mean of y. In this paper, the NRMSE was selected as the evaluation metric to assess the accuracy of the models. The NRMSE measures the average discrepancy between real and estimated values, with lower values indicating better predictive accuracy [27]. This metric is commonly used in regression analysis and provides a standardized measure of model performance.

Both datasets were utilized for the second part, which focused on assessing model predictive accuracy. Specifically, in the first case, 2,800 samples (approximately 90% of the available data) were selected as the training dataset, with the remaining samples used for validation. In the second case, 150,000 samples, equivalent to 50% of the total available data, were used for training the model, while the remainder served as the test dataset. The model predictive accuracy is reported as the accuracy in Equation (8) and also in the root mean square error which is defined as:

$$RMSE(y,\hat{y}) = \frac{1}{\sqrt{n}}\sqrt{\sum_{i=1}^{n}(y(i)-\hat{y}(i))^2} \quad (9)$$

4. Numerical Results and Discussion

In this section, the initial focus will be on presenting the results of the t-test conducted on the proposed hypothesis, demonstrating the superiority of the MCRT algorithm over the RT algorithm proposed in Tjen et al. [13]. Subsequently, the numerical simulations conducted to predict power consumption in both case studies will be showcased.

4.1. Hypothesis testing

Table 1 shows the model predictive accuracy of both MCRT and the RT algorithm proposed by Tjen et al. [13], while Table 2 shows the result of the t-test performed on the data in Table 1.

From Table 1, it is evident that, on average, the MCRT algorithm performs slightly better than the previous RT algorithm. Notable differences in accuracy are observed in datasets No. 8, 9, 10, 20, 21, 23, and 49, where the MCRT algorithm improves the model predictive accuracy of the RT algorithm by more than 1% (up to 2%). Conversely, in dataset no. 1, 3, 4, 18, 24, 32, 34, 36, 40, 41, 42. 44, 46, and 47, the RT algorithm outperforms the MCRT, albeit with a very subtle difference (less than 0.1%). It is reasonable to assume that the accuracy of the RT algorithm in these datasets is comparable to that of the MCRT algorithm. Overall, the MCRT algorithm outperforms the RT algorithm in 38 out of 50 datasets.

This result demonstrates that integrating the MC algorithm into the leaf of the RT algorithm significantly enhances the predictive capability of the RT-based model. This outcome aligns with expectations, as the classical RT model calculates the output by averaging the samples within the leaf. In contrast, the MCRT algorithm considers the probability of samples, resulting in more accurate predictions compared to the classical RT algorithm.

Table 1
Model predictive accuracy comparison between MCRT and RT

	Accuracy (%)		
N. dataset	RTMC	RT	RTMC-RT
1	86.198	86.209	−0.010
2	86.041	85.994	0.047
3	86.672	86.677	−0.006
4	86.988	86.990	−0.002
5	86.428	86.137	0.291
6	85.365	85.362	0.003
7	86.682	86.358	0.324
8*	86.437	83.953	2.484
9*	87.301	86.274	1.028
10*	87.214	84.592	2.623
11	87.462	87.458	0.004
12	85.622	85.621	0.001
13	87.353	86.660	0.693
14	86.570	85.953	0.617
15	86.376	86.372	0.004
16	86.392	85.827	0.565
17	86.261	86.259	0.002
18	85.484	85.484	0.000
19	85.923	85.921	0.001
20*	87.521	86.222	1.299
21*	86.786	85.775	1.011
22	85.427	85.426	0.000
23*	87.152	85.318	1.834
24	86.564	86.571	−0.007
25	84.974	84.971	0.003
26	84.895	84.893	0.002
27	86.724	86.238	0.487
28	86.751	86.624	0.127
29	86.452	86.450	0.003
30	87.318	87.317	0.001
31	85.554	85.551	0.003
32	86.542	86.544	−0.002
33	87.102	87.101	0.001
34	87.173	87.184	−0.011
35	87.121	87.120	0.001
36	86.251	86.254	−0.002
37	86.535	85.610	0.924
38	85.591	85.588	0.003
39	86.612	86.611	0.001
40	86.415	86.418	−0.002
41	86.219	86.225	−0.006
42	86.329	86.330	0.000
43	87.115	86.870	0.244
44	85.120	85.123	−0.003
45	86.324	86.320	0.005
46	86.812	86.813	−0.001
47	85.890	85.892	−0.002
48	86.986	86.312	0.674
49*	86.401	83.846	2.555
50	85.701	85.701	0.001
Average	86.423	86.066	0.356

*improvement over 1%

Table 2
T-test result of Table 1

Parameter	Value
N. observations	50
Degree of freedom	49
t Stat	3.648556
P-value (one tail)	0.000319
t Critical (one tail)	1.676551
P-value (two-tails)	0.000639*
t Critical (two-tails)	2.009575

*significant at $\alpha = 0.05$

From Table 2 it can be seen that the P-value for 2 tail test is less than 0.05. This means the H_0 is rejected, and thus, H_a is accepted. This outcome is somewhat unsurprising when considering that the MCRT algorithm demonstrates superior performance compared to the RT algorithm, with improvements of up to 2%, as depicted in Table 1. This result indicates that the MCRT does improve the model predictive accuracy of the RT algorithm and thus justifies the claim that the MCRT algorithm outperforms the RT algorithm proposed by Tjen et al. [13].

4.2. Model predictive accuracy

Table 3 shows the model predictive accuracy for MCRT and RT algorithms for both case studies. From Table 3, it is observable that the MCRT algorithm outperforms the RT algorithm for both UWDP and TICS cases. However, a more notable difference comes from the TICS dataset, where the MCRT algorithm accuracy is around 2% higher than the RT algorithm. Instead for the UWDP case, the MCRT still outperforms the RT algorithm, however with a smaller margin which is less than 1%. However, if both cases are considered, it can be seen that the UWDP model's predictive accuracy is much lower than the TICS.

Table 3
Model predictive accuracy for both case studies

Universitas Widya Dharma Pontianak (UWDP)		
Parameter	MCRT	RT
RMSE	292.27	306.25
NRMSE	0.17	0.18
Accuracy	82.76%	81.93%
Trapeznikov Institute of Control Sciences (TICS)		
Parameter	MCRT	RT
RMSE	49.19	72.34
NRMSE	0.06	0.08
Accuracy	94.33%	91.66%

Figures 4 and 5 show the plot for the observed power consumption and its prediction with the MCRT algorithm for both UWDP and TICS datasets, respectively. As shown in Figure 4, the MCRT model failed to predict the peak dynamics of the power consumption, especially when there is a "spike" in the power consumption which occurred due to some electrical appliances being turned on. For this case study, due to the limitation of the available dataset, the model cannot properly learn the dynamics of power consumption. Hence, predicted the dynamic is much worse than in Figure 4.

Figure 4
The observed and predicted power consumption for the UWDP building dataset

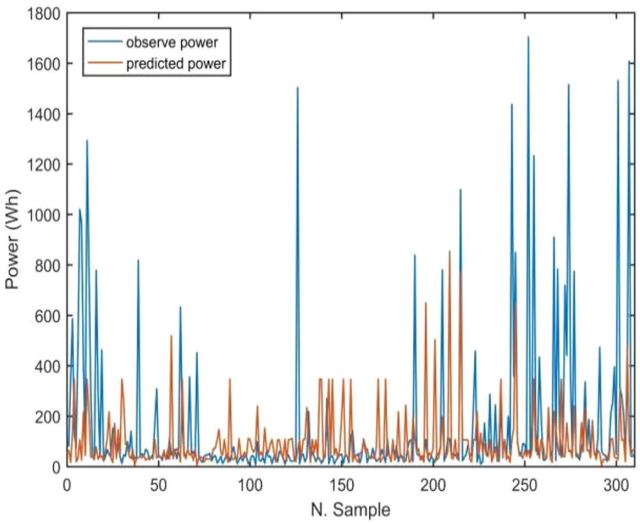
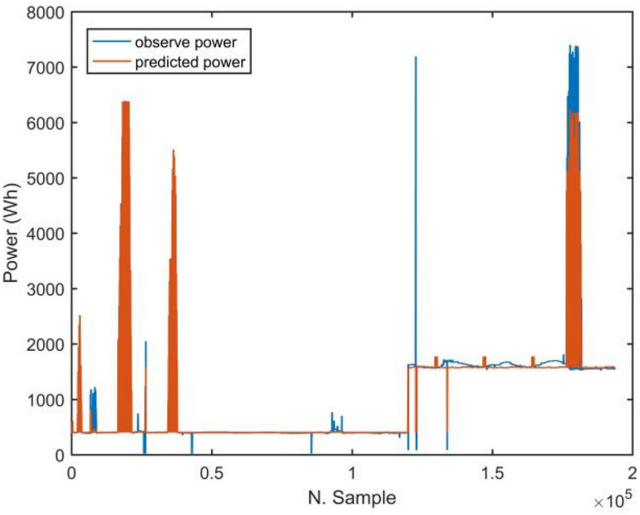


Figure 5
The observed and predicted power consumption for the TICS building dataset



For the TICS instead, it can seen from Figure 5 that the MCRT algorithm followed the pattern almost precisely, which resulted in higher accuracy than in the UWDP case. Even though there are also "spikes" for the TICS dataset (which is unavoidable, as any electrical equipment can be used anytime during the day), it is observable that the MCRT algorithm provided a better estimation of it rather than in the UWDP case, due to the model being exposed to a sufficient amount of data. It is worth noting that the model is only trained with 50% of the available data and validated on the remaining half data, yet able to provide a precise

prediction. This result shows the potency of the MCRT algorithm in predicting the power consumption inside a building.

5. Conclusion

In this paper, the challenge of predicting power consumption in buildings using a scarce dataset was addressed. A novel algorithm based on the RT algorithm was proposed, wherein an MC simulation is assigned to each leaf in the tree generated from the RT. Results from numerical simulations demonstrate that the MCRT algorithm outperforms the previous RT algorithm introduced by Tjen et al. [13]. In a test comprising 50 random datasets, the MCRT algorithm demonstrated superior performance over the RT algorithm in the majority of datasets (38 out of 50), with an average increase in model predictive accuracy of 0.36%. Moreover, the algorithm improved the model predictive accuracy for both case studies by up to 2%. Significance tests also confirmed the superiority of the MCRT algorithm over the RT algorithm.

Given the capabilities of the proposed algorithm, this research holds potential for regulating the usage of electronic appliances in buildings. Firstly, the algorithm can establish a baseline for building power consumption, which can be integrated into Internet of Things devices. Specifically, the algorithm can function as a detector to identify instances where electrical equipment is active when it should be inactive, thereby enabling more efficient energy management.

For further study, it is suggested to switch the MC simulation with another MC-based method, such as Markov Chain Monte Carlo, and find which MC algorithm is the best to be paired with the RT algorithm in order to provide a better estimation of power consumption.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in [Google Drive] at https://docs.google.com/spreadsheets/d/1o8sawOaOcX1kEm-dIdkcCUZhKoBduTAz/edit?usp=drive_link&ouid=115962907255429746256&rtpof=true&sd=true

Author Contribution Statement

Tony Darmanto: Conceptualization, Validation, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration. **Jimmy Tjen:** Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Genrawan Hoendarto:** Validation, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing.

References

[1] Gao, J., Yang, D., Wang, S., Li, Z., Wang, L., & Wang, K. (2023). State of health estimation of lithium-ion batteries based on mixers-bidirectional temporal convolutional neural network. *Journal of Energy Storage*, *73*, 109248. https://doi.org/10.1016/j.est.2023.109248C

[2] Yu, X., Shang, Y., Zheng, L., & Wang, K. (2023). Application of nanogenerators in the field of acoustics. *ACS Applied Electronic Materials*, *5*(9), 5240–5248. https://doi.org/10.1021/acsaelm.3c00996

[3] Saravanan, R., & Sujatha, P. (2018). A state of art techniques on machine learning algorithms: A perspective of supervised learning approaches in data classification. In *2018 Second International Conference on Intelligent Computing and Control Systems*, 945–949. https://doi.org/10.1109/ICCONS.2018.8663155

[4] Arridge, S., Maass, P., Öktem, O., & Schönlieb, C. B. (2019). Solving inverse problems using data-driven models. *Acta Numerica*, *28*, 1–174. https://doi.org/10.1017/S0962492919000059

[5] Fan, C., Yan, D., Xiao, F., Li, A., An, J., & Kang, X. (2021). Advanced data analytics for enhancing building performances: From data-driven to big data-driven approaches. *Building Simulation*, *14*, 3–24. https://doi.org/10.1007/s12273-020-0723-1

[6] Zhao, L., & You, F. (2019). A data-driven approach for industrial utility systems optimization under uncertainty. *Energy*, *182*, 559–569. https://doi.org/10.1016/j.energy.2019.06.086

[7] Ding, S., Li, R., Wu, S., & Zhou, W. (2021). Application of a novel structure-adaptative grey model with adjustable time power item for nuclear energy consumption forecasting. *Applied Energy*, *298*, 117114. https://doi.org/10.1016/j.apenergy.2021.117114

[8] Hoendarto, G., Saikhu, A., & Ginardi, R. V. (2023). Electricity power consumption prediction with the Monte Carlo simulation: Case study Universitas Widya Dharma Pontianak. In *2023 14th International Conference on Information & Communication Technology and System*, 176–181. https://doi.org/10.1109/ICTS58770.2023.10330846

[9] Mardani Najafabadi, M., & Taki, M. (2020). Robust data envelopment analysis with Monte Carlo simulation model for optimization the energy consumption in agriculture. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, 1–15. https://doi.org/10.1080/15567036.2020.1777221

[10] Shen, M., Lu, Y., Wei, K. H., & Cui, Q. (2020). Prediction of household electricity consumption and effectiveness of concerted intervention strategies based on occupant behaviour and personality traits. *Renewable and Sustainable Energy Reviews*, *127*, 109839. https://doi.org/10.1016/j.rser.2020.109839

[11] Panjaitan, S. D., Tjen, J., Sanjaya, B. W., Wigyarianto, F. T. P., & Khouw, S. (2023). A forecasting approach for IoT-based energy and power quality monitoring in buildings. *IEEE Transactions on Automation Science and Engineering*, *20*(2), 892–900. https://doi.org/10.1109/TASE.2022.3171561

[12] Smarra, F., Tjen, J., & D'Innocenzo, A. (2022). Learning methods for structural damage detection via entropy-based sensors selection. *International Journal of Robust and Nonlinear Control*, *32*(10), 6035–6067. https://doi.org/10.1002/rnc.6124

[13] Tjen, J., Iskandar, R. J., Willay, T., & Darmanto, T. (2023). Electric power consumption prediction from scarce dataset with entropy-based subset selection regression tree (e-ss RT). In *2023 14th International Conference on Information & Communication Technology and System,* 182–187. https://doi.org/10.1109/ICTS58770.2023.10330843

[14] Binder, K., & Heermann, D. (2014). *Monte Carlo simulation in statistical physics: An introduction*. USA: Springer.

[15] Nandy, A., Duan, C., & Kulik, H. J. (2022). Audacity of huge: Overcoming challenges of data scarcity and data quality for machine learning in computational materials discovery. *Current Opinion in Chemical Engineering*, *36*, 100778. https://doi.org/10.1016/j.coche.2021.100778

[16] Li, F., Shirahama, K., Nisar, M. A., Huang, X., & Grzegorzek, M. (2020). Deep transfer learning for time series data based on sensor modality classification. *Sensors*, *20*(15), 4271. https://doi.org/10.3390/s20154271

[17] Li, H., Wang, P., Hu, H., Su, Z., Li, L., & Yue, Z. (2023). Data-driven reliability assessment with scarce samples considering multidimensional dependence. *Probabilistic Engineering Mechanics*, *72*, 103440. https://doi.org/10.1016/j.probengmech.2023.103440

[18] Zhang, R., Liu, Y., & Sun, H. (2020). Physics-guided convolutional neural network (PhyCNN) for data-driven seismic response modeling. *Engineering Structures*, *215*, 110704. https://doi.org/10.1016/j.engstruct.2020.110704

[19] Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). *Classification and regression trees*. USA: Routledge.

[20] Loh, W. Y. (2011). Classification and regression trees. *WIREs Data Mining and Knowledge Discovery*, *1*(1), 14–23. https://doi.org/10.1002/widm.8

[21] Lewis, R. J. (2000). An introduction to classification and regression tree (CART) analysis. In *Annual Meeting of the Society for Academic Emergency Medicine*, 1–14.

[22] Jain, A., Smarra, F., Behl, M., & Mangharam, R. (2018). Data-driven model predictive control with regression trees—An application to building energy management. *ACM Transactions on Cyber-Physical Systems*, *2*(1), 4. https://doi.org/10.1145/3127023

[23] Betz, W., Papaioannou, I., & Straub, D. (2022). Bayesian post-processing of Monte Carlo simulation in reliability analysis. *Reliability Engineering & System Safety*, *227*, 108731. https://doi.org/10.1016/j.ress.2022.108731

[24] Xie, G. (2020). A novel Monte Carlo simulation procedure for modelling COVID-19 spread over time. *Scientific Reports*, *10*(1), 13120. https://doi.org/10.1038/s41598-020-70091-1

[25] Sani, H. M., Lei, C., & Neagu, D. (2018). Computational complexity analysis of decision tree algorithms. In *Artificial Intelligence XXXV: 38th SGAI International Conference on Artificial Intelligence*, 191–197. https://doi.org/10.1007/978-3-030-04191-5_17

[26] Del Moral, P., Doucet, A., & Jasra, A. (2012). An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistics and Computing*, *22*, 1009–1020. https://doi.org/10.1007/s11222-011-9271-y

[27] Hodson, T. O. (2022). Root-mean-square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development*, *15*, 5481–5487. https://doi.org/10.5194/gmd-15-5481-2022

How to Cite: Darmanto, T., Tjen, J., & Hoendarto, G. (2024). Monte Carlo Simulation-Based Regression Tree Algorithm for Predicting Energy Consumption from Scarce Dataset. *Journal of Data Science and Intelligent Systems*. https://doi.org/10.47852/bonviewJDSIS42022395