Received: 18 July 2024 | Revised: 27 August 2024 | Accepted: 18 September 2024 | Published online: 23 September 2024

Journal of Data Science and Intelligent Systems
2024, Vol. 00(00) 1–10
DOI: 10.47852/bonviewJDSIS42023861

RESEARCH ARTICLE

Decision Tree Regression with Residual Outlier Detection

Swee Chuan Tan[1],*

[1]*School of Business, Singapore University of Social Sciences, Singapore*

Abstract: This paper introduces a framework for identifying outliers in predictions made by regression tree models. Existing robust regression approaches tend to focus on the *construction* stage, which builds regression models that are less sensitive to outliers. In contrast, our approach focuses on identifying outliers during the *prediction* stage. The process of our proposed approach begins with building a regression tree using a training dataset. Predictions significantly deviating from the mean within each terminal node are automatically labeled as outliers. We show how the labeled data can be explored to better understand the characteristics of the outliers. We also identify the situations under which the data exploration may not work well. Further, we make use of the outlier labels and training data to construct an anomaly detector. Our results show that the proposed method can effectively detect outliers that may exist within datasets. Such outliers, when removed, result in improved data quality. Insights into its effectiveness and potential caveats are also discussed.

Keywords: regression tree, anomaly detection, outlier, robust regression

1. Introduction

Outlier detection involves identifying data points or values that depart significantly from normal observations. It is useful for identifying hidden anomalies and enhancing data integrity. Traditionally, outlier detection has found applications in areas where identifying rare and unusual cases is important, such as in defect classification [1]. Recently, outlier detection has also been used in modern fields such as sensor data analysis [2], medical image analysis [3], AI of Things [3], and Particle Physics [4]. In terms of learning methods, many studies have also started using deep learning given its promise in better detection performance. All these new developments reflect a growing interest in anomaly detection across diverse fields.

Despite significant advancement in deep learning and its potential in many domains, classical machine learning algorithms like decision trees remain highly relevant. There are two important reasons: (1) machine learning and deep learning are not mutually exclusive, and they can complement one another to complete a learning task more comprehensively; (2) deep learning is not a universal solution, particularly in anomaly detection. A recent benchmarking study using 104 datasets found that tree-based approaches could detect singleton anomaly while deep learning methods fail [5]. In addition, decision trees are more interpretable, which is valuable for explaining the underlying conditions leading to anomalies [6]. Further, tree-based approaches tend to work well in high-dimensional spaces. For example, a recent study reported superior performance of tree-based algorithms for anomaly detection in the presence of irrelevant features [7].

*Corresponding author: Swee Chuan Tan, School of Business, Singapore University of Social Sciences, Singapore. Email: jamestansc@suss.edu.sg

This paper extends an earlier work [8] on the use of decision tree for regression cum anomaly detection. As far as we know, decision trees are normally used for either regression or classification (e.g., to detect anomalies) tasks and seldom or never carry out the two tasks simultaneously for the same dataset. Indeed, performing regression with anomaly detection offers a unique advantage because estimated target values can be further qualified as normal or abnormal. If an estimation is abnormal, then further decision can be taken to improve the overall outcome of an application.

Regression trees are powerful tools for estimating numeric targets, offering fast predictions, ability to handle rough regression surfaces related to discrete data, and providing useful insights into the data using decision rules. However, outliers may exist within the terminal (leaf) nodes of regression tree models due to the need to maintain a minimum leaf node size. As a result, a leaf node may wind up including undesirable instances within its sub-partition space. We name such anomalies as *Terminal-node Anomalies*.

1.1. Terminal-node anomalies

Figure 1 shows a regression stump, where the vertical dotted line splits the data into nodes T1 and T2 using the input x. Here, a circled outlier (o_2) has been included in node T2 to fulfill a minimum leaf node size requirement of four instances. Similarly, node T1 has a dotted-circled outlier (o_1) that occurs *locally* within its subspace. It is commonly recognized that o_2 is an influential outlier because of its negative effect in changing the gradient of a simple linear regression line; while o_1 is less detrimental as compared to o_2, it is still undesirable due to its negative effect on the model quality [9]. Since o_1 and o_2 are within terminal nodes of the regression tree, they are considered as *Terminal-node Anomalies*.

Generally, the impact of outliers on regression models has been a long-standing problem, even in modern contexts [10]. In general,

Figure 1
A regression stump with two nodes and two outliers

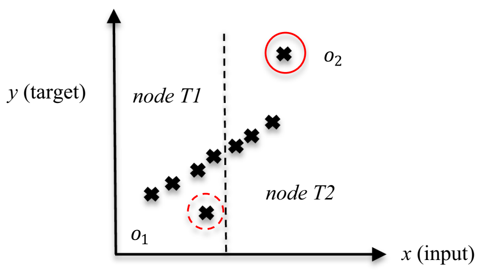
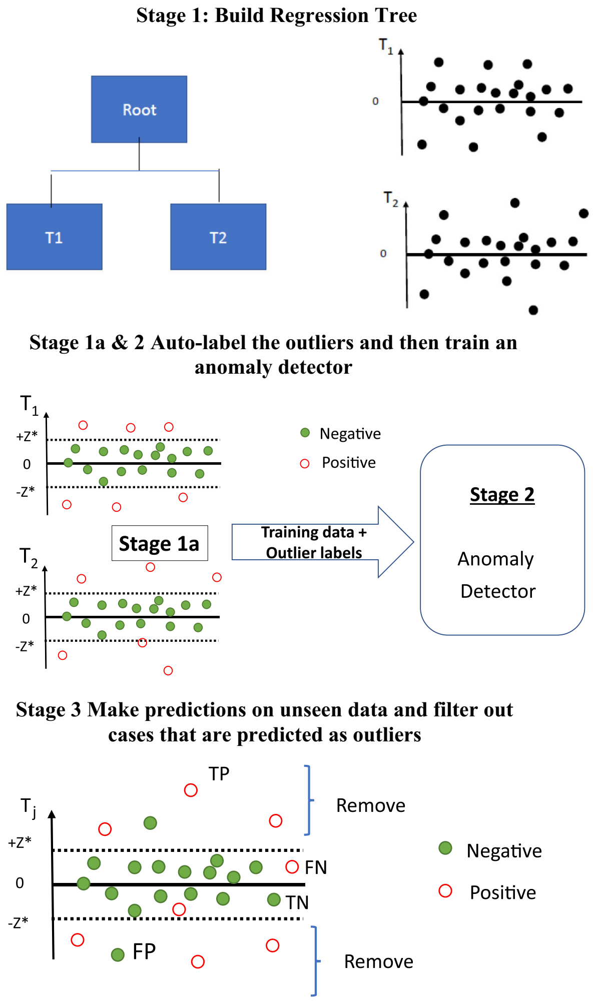


Figure 2
An illustration of the proposed 3-Stage framework


Stage 1: Build Regression Tree
Root
T1
T2
Stage 1a & 2 Auto-label the outliers and then train an anomaly detector
+Z*
0
-Z*
Negative
Positive
Stage 1a
Training data + Outlier labels
Stage 2
Anomaly Detector
Stage 3 Make predictions on unseen data and filter out cases that are predicted as outliers
TP
Remove
FN
TN
FP
Remove
Negative
Positive

influential global outliers like o_2 can distort the model, while local outliers like o_1 increases the prediction error [11].

In regression tree methods, outliers can impact the decision tree structure due to the use of the noise-sensitive squared error loss function. For example, the variance of Node T2 (c.f., Figure 1) becomes unusually large due to o_2; this can affect split point selection and the final regression tree structure. To mitigate this issue, robust loss functions such as the Huber loss function [12] can be considered. This function employs $L1$ and $L2$ norms for large and small residuals, respectively, thereby reducing the impact of extreme outliers. Similarly, Tukey's bisquare loss function [13] reduces the effect of clear-cut outliers by setting them to a constant while assigning proportional penalties to the remaining residuals.

The robust-loss-function approach is beneficial for creating regression trees that are less sensitive to outliers. However, it primarily focuses on the tree induction process and does not consider another key issue that occurs during the prediction stage. Firstly, the expected value of a node can be skewed by terminal-node anomalies. For example, Node T1 will have a lower expected value due to anomaly o_1, while Node T2 will have a higher expected value due to anomaly o_2. This results in underestimations for normal data points in Node T1 and overestimations for normal data points in Node T2, as illustrated in Figure 1. This limitation stems from the piecewise constant predictions made within each node. Hence, this paper proposes to detect and treat these anomalies during the *prediction* stage. Consequently, when our proposed method encounters new instances such as o_1 or o_2, it will recognize them as outliers, indicating that their predicted target values should be interpreted with caution. For example, the predictions that are predicted to be outliers can be used for noise filtering applications.

1.2. Illustration of the proposed method

The proposed three-stage framework is illustrated as an example in Figure 2. In Stage 1, a regular regression stump with two leaf nodes T1 and T2 is constructed. In Stage 1a, outliers that depart from their expected means within their respective leaf nodes are identified. Specifically, data points that are beyond their critical values (Z^*) are identified and labeled as outliers (i.e., positives) automatically. At this point, the results obtained from Stage 1a may be used to explore the characteristics of the outliers. We will conduct such explorations in Section 4.4.

In Stage 2, the same training input records, in conjunction with their corresponding newly identified outlier labels, are used to build an anomaly detector. The classification output of this anomaly detector can be used to certify the quality of individual predictions made by the regression tree during the actual predictions of unseen data (i.e., Stage 3).

In Stage 3, the trained regression tree and the anomaly detector are deployed for estimation and anomaly detection, respectively. The predicted outliers can be true positives (TPs) or false positives (FPs). Similarly, the predicted normal instances can be true negatives (TNs) or false negatives (FNs) [14].

A good anomaly detector should maximize the number of TPs and TNs and minimize the number of FPs and FNs. Naturally, two key performance measures here would be the TP rate (i.e., sensitivity) and TN rate (i.e., specificity) [15]. These performance measures can be used to tune the anomaly detector in Stage 2.

In Stage 3, the predicted outliers can then be further investigated or treated. For example, Figure 2 Stage 3 shows a simple treatment to remove cases that are predicted as outliers. A good detector should minimize the possible negative effects, which include wrongly removing FPs and incorrectly keeping FNs. In Section 4, we will show how this treatment leads to reduction of prediction errors, which can be useful in situations where the influence of noise or outliers is to be minimized.

With the proposed framework, this paper presents a decision tree approach to tackle the problem of simultaneous regression and outlier detection. In addition, the paper tackles two common issues encountered in anomaly detection.

1) **Verification without Ground Truth:** Unlike many studies that rely on anomaly labels or scores for objective performance evaluation, our approach allows the use of data with no ground truth for anomalies. We introduce a novel verification method based on the reduction of mean absolute error (MAE) to objectively validate whether detected cases are outliers.
2) **Interpreting Outliers:** We demonstrate how our proposed method interprets the meaning of an outlier and extracts insights from the decision tree.

The rest of this paper is organized as follows. Section 2 discusses the main approach of related work and how our proposed method differs from them. Section 3 presents the data used and the experimental setup. Section 4 analyzes the results and discusses the practical utility of our approach. Section 5 offers some reflections and concludes this paper.

2. Related Works

In the literature, outliers are usually defined as data points that deviate significantly from the overall data pattern, often measured using distance functions like Euclidean distance. However, this global approach may overlook outliers that are local anomalies.

One approach to detect local outliers is to define local regions within a dataspace, and any data points deviating from its local regions can be flagged as outliers. For example, the Local Outlier Factor (LOF) method detects data points with lower density than their neighbors as outliers [16]. This method is useful for outlier detections in datasets with varying density regions that may not be spherical. However, it may fail when there are overlapping regions or clusters.

Another approach is the cluster-based outlier detection method, which identifies outliers based on points that deviate from parent clusters [17]. This method is primarily distance-based, and it may not work well for high-dimensional data spaces due to the curse of dimensionality.

Other innovative approaches include autoencoders, which are neural networks trained using normal data [18]. Then in the detection phase, they detect outliers that cause high reconstruction errors.

Unfortunately, autoencoders operate under the assumption that normal data are available for training, which is not the case in our work. In all the four datasets used in this study, they contain no information of the anomaly labels.

One way to mitigate the issue is by using the Robust Variational Autoencoders (RVAE), where the robust loss function helps to attenuate the effects of outliers in the training data, and the probabilistic latent space helps handle uncertainty in the data.

This lack of ground truth in datasets poses a common problem for objective performance evaluation. Fortunately, there is a way to tackle this issue within our proposed framework, and we will provide the explanation in Section 4.2 (experimental setup). Another key issue with autoencoders is that they cannot explain why a specific data point is classified as an outlier, and this can limit their applications in domains where interpretation of outliers is crucial.

Compared to autoencoders, distance-based methods are more intuitive and easier to explain. However, they may not be ideal for detecting *Terminal-node Anomalies* which are specific to regression tree models. Since these are outliers forced into leaf nodes due to the minimum node size requirement, it would be natural to detect them within the tree itself. In addition, decision trees are robust in dealing with issues like missing values and mixed data types. Such issues often require additional data preprocessing for distance-based methods. Finally, most unsupervised distance-based methods assume equal weightages for all features, which is not ideal for high-dimensional problems. In contrast, decision trees typically use a subset of important features to describe a data pattern, and insignificant features are excluded from the tree. We will highlight how this property results in better performance of our method when explaining the visualization results in Section 4.4.

As mentioned earlier, outliers may affect regression tree structure by amplifying the squared error loss function, leading to biased split points and skewed mean values in terminal nodes [19]. To mitigate this issue, robust loss function like the Huber loss function can be used [12]. The Huber loss function is defined as:

$$L(x) = \frac{r^2}{2} \ if \ |r| \ \leq \ \partial; otherwise \ L(x) \ = \ \partial\left(|r| - \frac{\partial}{2}\right).$$

Here, r is the residual and ∂ is a user-specified threshold. The Huber loss function applies squared error ($L2$ norm) for small residuals and absolute loss ($L1$ norm) for large residuals, thereby reducing the influence of outliers.

Another example is the Tukey's bisquare loss function [13]. This function caps the loss for large outliers at a constant of $c^2/6$, when the residuals are greater than the user-defined threshold (c); i.e., $|r| > c$. Otherwise, it assigns a penalty based on the magnitude of each residual, using:

$$\frac{c^2}{6}\left(1 - \left(1 - \frac{r^2}{c^2}\right)^3\right).$$

The above-mentioned loss functions may be used to better estimate the split points and improve prediction estimates, resulting in regression trees that are less affected by outliers [20].

However, it is important to note that these approaches primarily focus on the tree *induction* process only; they do not handle the presence of outliers during the *prediction* stage. When making predictions, a regression tree may come across a new data instance that does not fit into any terminal nodes created during the training phase. In this case, the regression tree proceeds to make a prediction using the nearest terminal node, without knowing that this data instance is indeed an anomaly. This lack of knowledge about anomalies will lead to erroneous predictions and undermines the model's prediction performance.

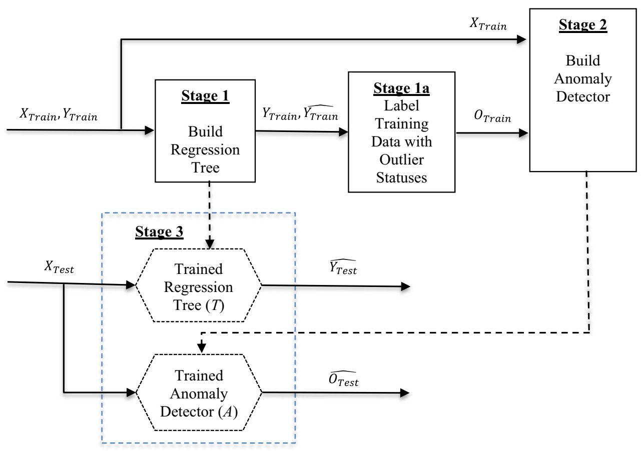
To address this issue, the proposed approach automatically detects anomalies during the *prediction* stage. This is achieved by first creating outlier labels automatically, which are required for training an anomaly detector. Once trained, this anomaly detector operates in tandem with the regression tree, determining whether each prediction should be classified as an outlier. This integrated framework enables simultaneous regression cum anomaly detection capabilities.

3. Proposed Method

This section presents the conceptual framework, which consists of three stages depicted in Figure 3. In the following descriptions of these three stages, we denote $\{X_{Train}, \ Y_{Train}\}$ be a set of training inputs and target, respectively.

- **Stage 1:** Use the $\{X_{Train}, \ Y_{Train}\}$ dataset to construct a regression tree T that consists of m terminal (leaf) nodes. Each terminal node of T is denoted as T_j, where $j = 1, \ 2, \ \ldots, \ m$. At Stage 1a, each training instance i is labeled with a binary outlier status of $o_i = \{yes, \ no\}$. This is achieved by comparing the true target values Y_{Train} and the training results $\widehat{Y_{Train}}$. Note that each training

Figure 3
The proposed conceptual framework



instance i located within a leaf node T_j will have the same prediction value $\widehat{y_{T_j}} = \frac{1}{|T_j|}\sum_{i=1}^{|T_j|} y_i$. For each instance i, if its true target values y_i is significantly different from $\widehat{y_{T_j}}$, it will be declared an outlier. Specifically, we compute $|z_i| = \left|y_i - \widehat{y_{T_j}}\right|/\sigma_{T_j})$, where σ_{T_j} is the standard deviation of y_i in node T_j. We then label o_i using the rule: $o_i = yes\ if\ |z_i| \geq Z^*,\ otherwise\ o_i = no$. In this rule, Z^* is the critical value defined by some level of significance. We will discuss how this value is determined in Section 4.2.

- **Stage 2:** Using the $\{X_{Train},\ O_{Train}\}$ dataset, an anomaly detector denoted as A is constructed. Here, O_{Train} is the binary outlier label automatically generated by Stage 1a. Although not illustrated in Figure 3, it is important to note that Stage 2 requires the target distribution to be balanced. This is crucial because the proportion of "yes" in the outlier target is substantially lower, likely below 5%, requiring up-sampling of these minority cases. For inductive learning methods like a regression tree, replicating the minority cases helps the learner to effectively recognize the classification pattern. A simple approach adopted here is to replicate the minority cases until there is an approximately equal number of instances between "yes" and "no".
- **Stage 3:** Send the independent test set X_{Test} to both T and A for the purpose of estimation and anomaly detection respectively. Any instances detected as outliers by A can have their corresponding estimations marked for further scrutiny, investigations, or necessary interventions. Within the scope of this study, we assume that all the instances identified as outliers will be discarded. This approach is particularly well-suited for scenarios rich in data points, such as data stream environments with sensor/IoT networks [21]. An example is the use of machine learning to improve electricity grid stability [22]. Other examples include e-commerce website traffic and transactions, social media posts, comments, likes, and shares.

4. Data and Experimental Setup

4.1. Data

Table 1 provides details of four datasets obtained from the UCI Machine Learning Repository [23]. We use these datasets in our experiments because they are well-suited for regression tasks, and have different data sizes, ranging from 1503 instances to 10000 instances. Additionally, the number of predictor attributes ranges from four to 11. The proportions of training data labeled outliers are all less than 5% of their respective dataset sizes.

4.2. Experimental setup

Recall that Z^* is the critical value used in Stage 1a, which is used to decide whether an instance is an outlier. In this study, we use $Z^* = 1.96$, which corresponds to the 95% confidence level in a two-tailed test, throughout our experiments.

Although we have used $Z^* = 1.96$ based on statistical rule of thumb, one could vary this threshold based on practical domain-specific considerations. For example, if a critical value results in numerous cases being labeled as outliers, then it might be reasonable to increase Z^* so as to reduce the number of outliers. On the other hand, if a reasonable critical value results in an insignificant number of outliers, then it might suggest that the dataset is clean and does not require outlier detection. Additionally, if we have extremely few outliers, the outlier-labeled dataset will become extremely imbalanced, and this can cause the anomaly detector (or any binary classifier) to fail.

Another practical approach is to rank the outliers using the Z scores, and only examine the top N number of cases with the highest Z scores.

All the datasets are randomly partitioned into 70% for training, forming $\{X_{Train},\ Y_{Train}\}$ in Figure 3. The remaining 30% are used for testing, i.e., $\{X_{Test},\ Y_{Test}\}$. For each dataset, the regression tree T and

Table 1
Datasets used for regression and outlier detection

Dataset name	Description
Smart Grid	Local stability analysis of a decentralized smart power grid control [24].
Percent outliers: 2.75%	10000 instances; 11 predictors;
	Target = system stability.
Auction	Verification of simultaneous multi-round auctions for frequency spectra [25].
Percent outliers: 4.15%	2043 instances; 7 predictors;
	Target = verification runtime.
Airfoil Noise	NASA dataset, obtained from a series of aerodynamic and acoustic tests of two- and
Percent outliers: 4.78%	three-dimensional airfoil blade sections conducted in an anechoic wind tunnel [26].
	1503 instances; 5 predictors;
	Target = scaled sound pressure level.
Power Plant	Data points collected from a Combined Cycle Power Plant over 6 years (2006–2011) [27].
Percent outliers: 3.9%	9568 instances; 4 predictor attributes;
	Target = net hourly electrical energy output

the anomaly detector A were constructed using the training data and then evaluated using the testing data.

We have used the Classification and Regression Tree (CART) algorithm available in an IBM product [28]. This is a robust and easily accessible method for academia and industry [29]. Its ability to perform both regression and classification (of anomalies) fits our objective well and helps to simplify implementation of the proposed framework.

The regression tree is set with a minimum allowable terminal node size of 2% of dataset size. This is a reasonable percentage because the smallest dataset is the Airfoil Noise, that contains 1502 instances. This ensures that all terminal nodes (T_j) (except one terminal node of Airfoil Noise model, which contains 21 instances) in the regression tree fulfills $min(|T_j|) \geq 30$. This is akin to using a minimum sample size of 30 for statistical inference. For the anomaly detector, we apply CART with bagging [30] (with its default setting of 10 component models) to improve the detection performance.

To evaluate the robustness and performance of the models, the random train:test partitioning scheme is repeated 30 times. Each time, a unique random-number-generation seed is used. After the 30 iterations, we assess the overall models' performance.

For the purposes of model evaluation, Y_{Test} is used to evaluate T using the MAE of the estimated predictions i.e., $MAE = \frac{1}{n}\sum| y_{Test} - \widehat{y_{Test}}|$, where n is the size of the test dataset.

To evaluate A, we first send $\{X_{Test}, Y_{Test}\}$ through Stages 1 and 1a to generate O_{Test}. It is important to note that O_{Test} is unseen by A. We evaluate the classification performance of A by comparing O_{Test} with $\widehat{O_{Test}}$. This performance provides some indications of its ability to detect outliers. If our purpose is to discard the detected outliers, then sensitivity and specificity are useful measures. The former suggests the proportion of positives (i.e., outliers) that are detected, and the latter suggests the proportion of negatives (i.e., normal instances) that will be sacrificed because they are falsely detected as outliers and will be wrongly discarded.

As mentioned earlier, all the datasets used in our study do not contain any outlier information. Without the ground truth, it will be challenging to perform objective evaluation of results. Fortunately, the negative effects of outliers can be measured using a proxy, which is the *reduction in regression MAE* after the outliers have been detected and removed. Thus, our results report MAEs in three test data scenarios. The first scenario is denoted as "MAE 1" (All test data). This represents the scenario where no outlier detection and filtering process are in place, and the full set of test data is being used for evaluation.

The second scenario is denoted as "MAE 2" (Test data with n' outliers discarded). This is when the anomaly detector has detected and discarded n' number of outliers from the test data. This set of data is used to verify whether there will be a reduction in MAE after the outliers have been removed.

Since the second scenario uses fewer instances, it is important to create a control group to ensure that any improvement in results is not simply due to the use of a smaller sample. Hence, the third scenario, denoted as "MAE 3" (Test data with random sample of n' records discarded), is used to evaluate the model using test data with instances removed randomly. The number of instances removed is n', which is the same as that removed by outlier detection.

Apart from MAE, we also use other evaluation measures for the anomaly detector. First, sensitivity is used to measure the proportion of TP (or outlier) instances detected over the total number of positives. Likewise, specificity measures the proportion of true negative (or normal) instances correctly detected, over the total number of negatives.

4.3. Results

The results of the experiments are presented in Tables 2 and 3. Each result entry in the tables is represented using mean ± standard deviation derived from the performance metrics recorded over 30 runs, where each run uses a unique random split in the training and testing sets (c.f., Section 4.2).

Table 2 shows how well the anomaly detectors can effectively detect true outliers (i.e., sensitivity) and identify normal instances (i.e., specificity). These performance measures allow one to determine whether it is worthwhile to adopt the proposed method for detecting terminal-node outliers. For example, in a data stream environment, one may deal with outliers by simply discarding them. A high sensitivity ensures that a large proportion of true outliers are discarded. Similarly, a high specificity ensures that a large proportion of true negatives are retained.

Table 2 shows that the best-performing model is the one created based on the Auction data, which has testing sensitivity and specificity of 88.2% and 89.7% respectively. This suggests that the dataset contains a significant number of detectable outliers. In particular, the model can detect about 88% of the outliers and keep close to 90% of the normal instances.

The least effective anomaly detection model is the one created based on the Power Plant data, showing a sensitivity of 0.436. This indicates that the detector can only identify 43.6% of the outliers found within terminal nodes. Its specificity is 0.729. If we choose

Table 2
Performance of the anomaly detector

Test performance	Smart Grid	Auction	Airfoil Noise	Power Plant
Sensitivity	0.555 ± 0.156	**0.882** ± **0.094**	0.480 ± 0.141	0.436 ± 0.111
Specificity	0.729 ± 0.087	**0.897** ± **0.028**	0.744 ± 0.088	0.729 ± 0.076

Table 3
MAEs of the regression tree under three scenarios

Test MAE	Smart Grid	Auction	Airfoil Noise	Power Plant
MAE1 (all data)	0.028 ± 0.000	1508 ± 105	3.74 ± 0.22	3.61 ± 0.06
MAE2 (remove outliers)	**0.026** ± **0.001**	**1157** ± **189**	**3.44** ± **0.21**	**3.45** ± **0.09**
MAE3 (remove random data)	0.028 ± 0.001	1489 ± 112	3.73 ± 0.23	3.61 ± 0.07

to discard all the *predicted* outliers, then 43.6% percent of the true outliers will be correctly discarded; but at the same time, 27.1% of the normal cases will be wrongly discarded. In practice, one needs to evaluate whether it is worthwhile to do so.

There are several possible reasons why a dataset may result in creating a better anomaly detector compared to another. First, the outliers should be reasonably separable from the normal instances. In Section 4.4, we will present some visualizations to illustrate this idea.

However, even if we can see separable outliers in a visualization, the number of separable outliers must not be too small (in comparison to the number of normal instances) for building an accurate anomaly detector. We will discuss this when examining the visualization results in Section 4.4. Fortunately, by evaluating sensitivity and specificity, we can make an initial assessment of the suitability of the proposed method for a given dataset. Additionally, observing reductions in MAE after outliers are removed provides confidence that the excluded outliers were indeed detrimental to the model's performance.

In terms of MAE, Table 3 shows the MAEs of the regression tree for each dataset under three scenarios: (i) MAE1 is based on all available test data; (ii) MAE2 is test data excluding the outliers detected; (iii) MAE3 is test data excluding instances randomly discarded.

The corresponding MAE1 and MAE3 results show that the MAEs for these two categories are significantly higher than MAE2 produced by the proposed method. Overall, Table 3 shows that all the models created using our proposed approach can result in statistically significant reductions of *MAE* in the predictions. However, it is important to assess whether these error reductions hold practical significance.

For example, the Power Plant regression tree can reduce the average *MAE*1 of 3.61 to *MAE*2 of 3.45 after all the detected outliers are being discarded. One important consideration here is to assess whether this amount of *MAE* reduction is worth discarding 27.1% of normal cases (i.e., 1 – Specificity) in the Power Plant dataset.

Another example is the model performance for the Smart Grid, where the decrease in MAE from 0.028 to 0.026, though statistically significant, may require further evaluation to determine its practical importance.

4.4. Further comparisons and applications

We will use the outlier labels obtained from Stage 1a for outlier visualization. Additionally, we compare the results with two other methods.

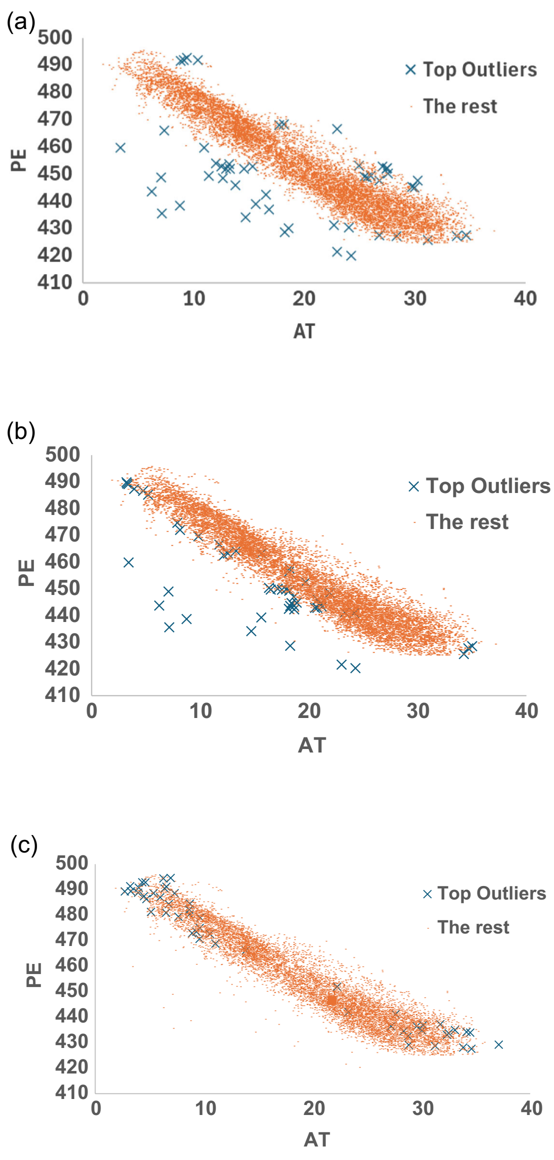
Firstly, we compare the visualizations with a cluster-based anomaly detection method used in an IBM product [31]. This method detects anomalies based on deviations from cluster norms. It uses log-likelihood distance function, and this is suitable for our datasets containing mixed data types. The number of clusters is set to be the same as number of terminal nodes in the corresponding regression tree being compared.

A similar comparison of outlier visualization will also be conducted with the RVAE described in Section 2.

Apart from outlier visualization, we will also demonstrate how the decision rules can be used to better identify the conditions under which outliers occur. Finally, we will show how the method can be used for outlier filtering to improve data quality.

Figure 4(a) shows a scatterplot of the atmospheric temperature (AT) versus the electricity consumption target in the Power Plant dataset. As temperature AT drops, more electricity is consumed, likely due to more intensive use of heating systems. The plot shows all the extreme outliers that are three standard deviations away from the means of their respective terminal nodes. These

Figure 4
(a) Outliers in the Power Plant dataset (proposed method). (b) Outliers in the Power Plant dataset (cluster-based anomaly detection). (c) Outliers in the Power Plant dataset (Robust Variational Autoencoders)



outliers are mostly located at the outer boundaries of the common data points. This suggests the validity of the outlier obtained from Stage 1a and they can facilitate data exploration to better understand outlier characteristics.

Figure 4(b) shows a scatterplot like that of Figure 4(a), but this time with outliers detected by the cluster-based outlier detector. Visually, we can observe that the outliers detected are not as meaningful as compared to the proposed method in Figure 4(a). In particular, the cluster-based method tends to detect only the outliers that are substantially below the normal pattern. Outliers that are above the normal pattern are not detected. In contrast, the proposed method offers a more balanced detections of outliers above and below the normal pattern.

In Figure 4(c), the outliers detected by the robust variation encoder tend to cluster around the two extreme ends of the normal pattern. This may be due to the lack of normal instances available during autoencoder training, resulting in ineffective discrimination of reconstruction loss when outliers are present.

Though the visualization is not shown here, we wish to report that Local Outlier Factor has the worst performance, with most "outliers" identified within the normal pattern. This could in part be due to the presence of overlapping regions in the data, which violates its density assumption.

Figure 5(a) gives another example of how outliers in the Smart Grid dataset can be visualized using a scatterplot of input feature Tau1 (i.e., reaction time) variable versus the target variable Stab (i.e., stability). Like Figure 4(a), the outliers are mainly located around the outer edge of the normal data points.

Figure 5(b) shows the corresponding results generated using the cluster-based outlier detector, which is quite poor. This poor result is due to the use of all 12 equally weighted features in the dataset, resulting in less effective distance computations because not all features are useful. In contrast, the outliers in Figure 5(a) were found based on only a small subset of the features used in the regression tree. If we use this same subset of features (as the proposed method) for cluster-based outlier detection, the results can be improved significantly, as shown in Figure 5(c). This implies that additional feature selection must be performed before applying the cluster-based outlier detector.

In Figure 5(d), the quality of outliers detected by the robust variation encoder is between that of cluster-based outlier detector with (c.f., Figure 5(b)) and without (c.f., Figure 5(c)) feature selection. The results of autoencoders and LOF are worst and not shown here for comparison.

Figure 6 shows a scatter plot depicted using the Frequency (Hz) and Suction (meter) variables of the Airfoil Noise dataset. The outliers are points that exhibit higher or lower than expected sound pressure. Here, a simple scatter plot is not as effective for identifying the outliers, compared to Figures 4 and 5. This is due to the relatively discrete data properties, and the lack of clear boundary between normal instances and anomalies. The visualization results obtained from other anomaly detection methods are similar, and the corresponding figures have been omitted for brevity.

However, decision rules of the anomaly detector (constructed in Stage 2) may help identify the conditions under which outliers occur. For example, one of the decision rules states that IF FREQUENCY > 1425 Hz and SUCTION > 0.041 m, THEN 93% of the data points in this subregion are outliers. The green boundary in Figure 6 shows the data region defined by this rule. The mean sound pressure in this subregion is 111 dB, which is significantly lower than the sound pressure outside this region of 125 dB, suggesting an area that might be worth investigating.

Another situation where outlier visualization may not work is when the features contain very few discrete values. Using two of the important discrete-valued features of the Auction dataset, namely price (i.e., price currently being verified) and Product (i.e., Product currently being verified), we can see that most outliers seem to occur when the price is very low. However, we cannot tell what the percentage is in the low-price region, as shown in Figure 7.

Although Figure 7 is hard to visualize, we can use the anomaly detector's decision structure to understand the conditions under which anomalies occur. Figure 8 shows a tree map generated from the CART decision tree. It suggests that no anomalies are expected when prices are within a moderate range between 60 to 80, and when bidder's capacity is at least one. For higher prices, anomalies are likely to occur when the product is 4 and below, or

Figure 5
(a) Outliers in the Smart Grid dataset (Proposed Method). (b) Outliers in the Smart Grid dataset (Cluster-based anomaly detection (all features)). (c) Outliers in the Smart Grid dataset (Cluster-based anomaly detection (selected features)). (d) Outliers in the Smart Grid dataset (Robust Variational Autoencoders)

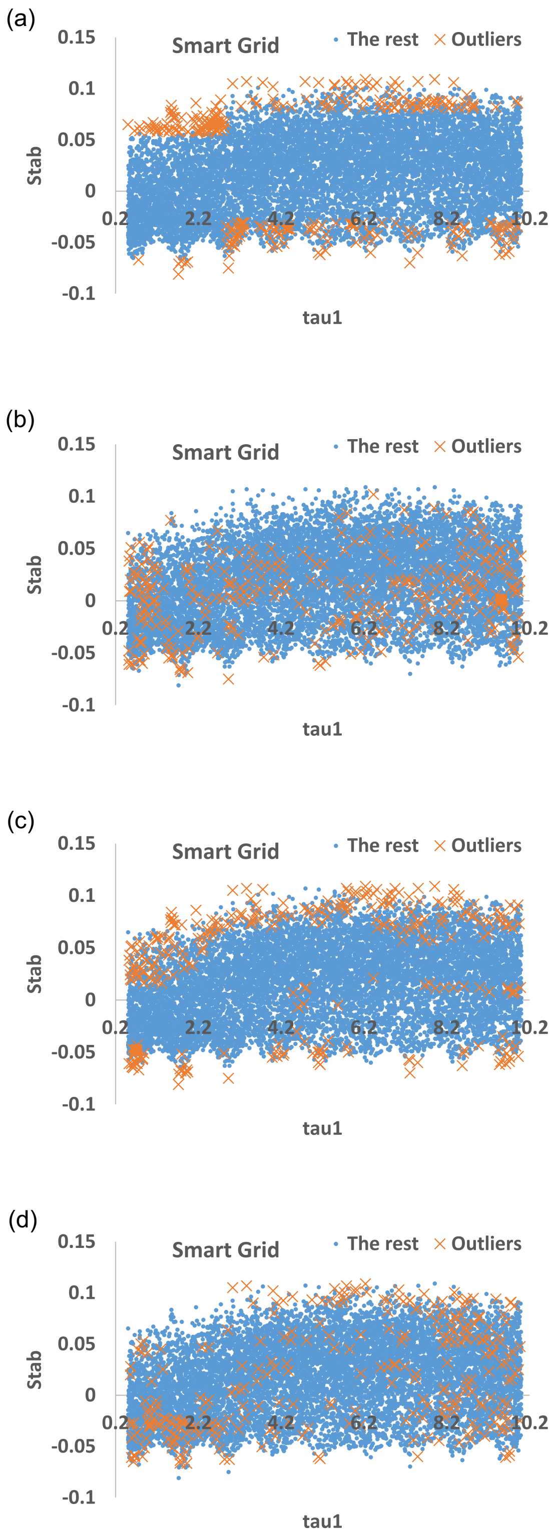


Figure 6
Outliers in the Airfoil Self-noise dataset (Proposed Method)

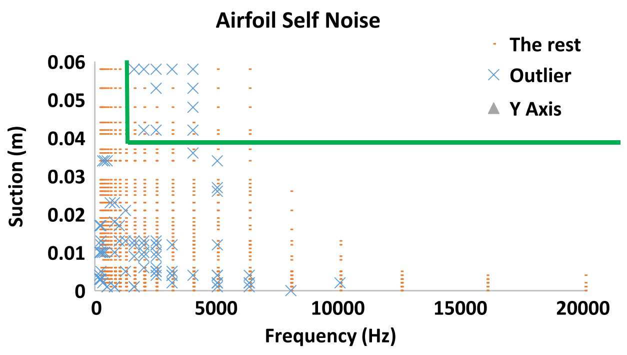


Figure 9
Outliers in the Auction dataset

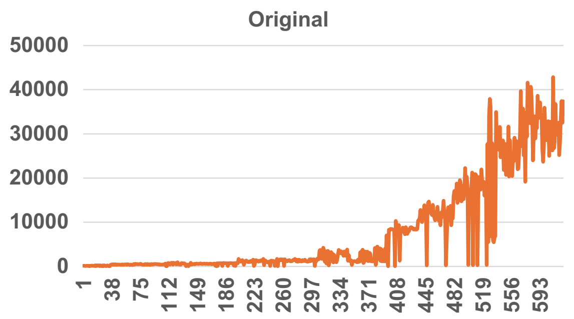


Figure 7
Outliers in the Auction dataset (Proposed Method)

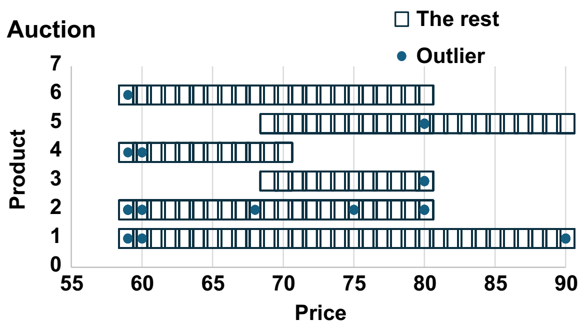


Figure 10
Outliers removed in the Auction dataset

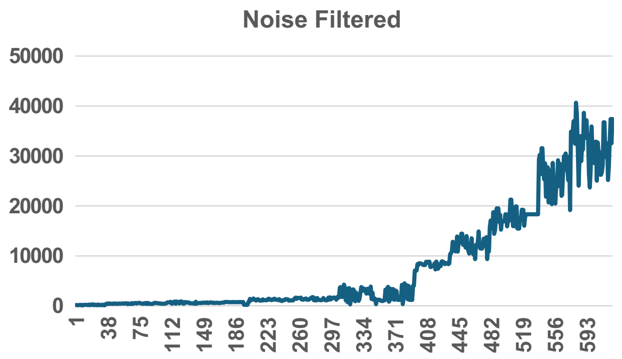
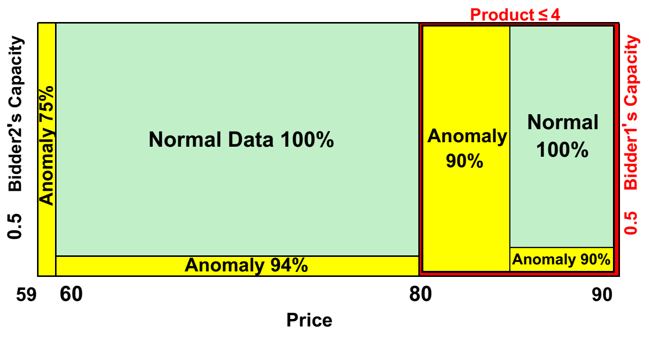


Figure 8
Decision tree map shows the conditions leading to anomalies in the Auction dataset



when the bidder's capacity is less than one. Thus, this is more informative as compared to Figure 7.

Figure 8 is also aligned with the test results (c.f., Table 2) based on the Auction dataset. Recall that there is a significant MAE reduction and high sensitivity and specificity in the models created from this dataset. This suggests that outlier filtering can lead to significant reduction of errors. To demonstrate this, we sort the data points based on their predicted target values in ascending order. We then plot the original target values, as shown in Figure 9.

By filtering out the outliers predicted by the anomaly detector, we observe that the data points become significantly cleaner, as shown in Figure 10.

5. Conclusion

Finally, we discuss several key insights derived from this study. Firstly, the notion of Terminal-node Anomalies is a subtype of residual outliers in regression. This type of anomalies is different from the common definition of outliers in the literature. This explains why most of the common anomaly detection methods do not perform well, since they were not originally designed for the same purpose.

Secondly, we have shown how residual anomalies can be used to construct an anomaly detector, which in turn helps qualify the validity of an estimation. As far as we know, we are not aware of a similar approach reported in the existing literature.

Thirdly, most distance-based methods assign equal weight to all features, making their use problematic in high-dimensional spaces. In contrast, the proposed approach uses only a subset of features identified by the regression tree (see Figure 5), resulting in better performance.

Fourthly, automatically labeled outliers can facilitate outlier visualization in certain datasets. However, this approach may be less effective in complex data spaces or with features having limited discrete values. In such cases, the map representation derived from the decision rules can offer deeper insights into the conditions that give rise to outliers.

Note that the proposed method is not without limitations. Firstly, the method requires sufficiently large terminal nodes, typically with a size of 30 instances or more. For small datasets (e.g., with 100 instances), a minimum leaf node size of 30 is impractical. In practice, we advise exercising caution when working with datasets containing fewer than 1,000 instances.

Another limitation is the need to artificially increase the number of outliers in order to use CART as an anomaly detector, as CART requires sufficient instances for effective tree induction [29].

Finally, we offer some possible research directions. Since our approach fundamentally differs from the existing methods that use robust loss functions, a direct comparison is not provided in this paper. However, we believe integrating the proposed method with a robust loss function could further enhance performance. This is because the resulting leaf node estimates will become more resistant against outliers, thereby making anomalies even more detectable during the prediction stage.

Ethical Statement

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

Conflicts of Interest

The author declares that he has no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in Kaggle at https://doi.org/10.34740/KAGGLE/DSV/9355696.

Author Contribution Statement

Swee Chuan Tan: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration.

References

[1] Klarák, J., Andok, R., Malík, P., Kuric, I., Ritomský, M., Klačková, I., & Tsai, H. Y. (2024). From anomaly detection to defect classification. *Sensors*, *24*(2), 429. https://doi.org/10.3390/s24020429

[2] Huang, Y., Liu, W., Li, S., Guo, Y., & Chen, W. (2024). MGAD: Mutual information and graph embedding based anomaly detection in multivariate time series. *Electronics*, *13*(7), 1326. https://doi.org/10.3390/electronics13071326

[3] Zingman, I., Stierstorfer, B., Lempp, C., & Heinemann, F. (2024). Learning image representations for anomaly detection: Application to discovery of histological alterations in drug development. *Medical Image Analysis*, *92,* 103067. https://doi.org/10.1016/j.media.2023.103067

[4] Belis, V., Odagiu, P., & Aarrestad, T. K. (2024). Machine learning for anomaly detection in particle physics. *Reviews in Physics*, *12,* 100091. https://doi.org/10.1016/j.revip.2024.100091

[5] Sarkar, S., Mehta, S., Fernandes, N., Sarkar, J., & Saha, S. (2024). Can tree-based approaches surpass deep learning in anomaly detection? A benchmarking study. *arXiv Preprint: 2402.07281*.

[6] Crupi, R., Regoli, D., Sabatino, A. D., Marano, I., Brinis, M., Albertazzi, L., . . . , & Cosentini, A. C. (2024). DTOR: Decision tree outlier regressor to explain anomalies. *arXiv Preprint: 2403.10903.*

[7] Freytsis, M., Perelstein, M., & San, Y. C. (2024). Anomaly detection in the presence of irrelevant features. *Journal of High Energy Physics*, *2024*(2), 1–22. https://doi.org/10.1007/JHEP02(2024)220

[8] Tan, S. C. (2024). Enhancing regression tree predictions with terminal-node anomaly detection. In *Proceedings of the 2023 6th Artificial Intelligence and Cloud Computing Conference*, 21–26. https://doi.org/10.1145/3639592.3639596

[9] Duari, G., & Kumar, R. (2023). Clustering for global and local outliers. In *Machine Intelligence Techniques for Data Analysis and Signal Processing: Proceedings of the 4th International Conference MISP 2022, 1*, 601–610.

[10] Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). *Introduction to linear regression analysis*. USA: John Wiley & Sons.

[11] Jabbar, A. M. (2021). Local and global outlier detection algorithms in unsupervised approach: A review. *Iraqi Journal of Electrical and Electronic Engineering*, *17*(1), 76–87.

[12] Huber, P. J. (1992). Robust estimation of a location parameter. In S. Kotz & N. L. Johnson (Eds.), *Breakthroughs in statistics* (pp. 492–518). Springer. https://doi.org/10.1214/aoms/1177703732

[13] Beaton, A. E., & Tukey, J. W. (1974). The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, *16*(2), 147–185. https://doi.org/10.1080/00401706.1974.10489171

[14] Amin, F., & Mahmoud, M. (2022). Confusion matrix in binary classification problems: A step-by-step tutorial. *Journal of Engineering Research*, *6*(5), 1–12.

[15] Swift, A., Heale, R., & Twycross, A. (2020). What are sensitivity and specificity? *Evidence-Based Nursing*, *23*(2), 2–4.

[16] Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data,* 93–104. https://doi.org/10.1145/342009.335388

[17] Smiti, A. (2020). A critical overview of outlier detection methods. *Computer Science Review*, *38,* 100306. https://doi.org/10.1016/j.cosrev.2020.100306

[18] Saeedi, J., & Giusti, A. (2023). Semi-supervised visual anomaly detection based on convolutional autoencoder and transfer learning. *Machine Learning with Applications*, *11,* 100451. https://doi.org/10.1016/j.mlwa.2023.100451

[19] Daniya, T., Geetha, M., & Kumar, K. S. (2020). Classification and regression trees with gini index. *Advances in Mathematics: Scientific Journal*, *9*(10), 8237–8247.

[20] Galimberti, G., Pillati, M., & Soffritti, G. (2007). Robust regression trees based on M-estimators. *Statistica*, *67*(2), 173–190. https://doi.org/10.6092/issn.1973-2201/3503

[21] Tan, S. C., Yip, S. H., & Rahman, A. (2014). One pass outlier detection for streaming categorical data. In *Proceedings of the 3rd International Workshop on Intelligent Data Analysis and Management*, 35–42.

[22] Aziz, F., & Lawi, A. (2022). Increasing electrical grid stability classification performance using ensemble bagging of C4.5 and classification and regression trees. *International Journal of Electrical and Computer Engineering*, *12*(3), 2955–2962. https://doi.org/10.11591/ijece.v12i3.pp2955-2962

[23] Kelly, M., Longjohn, R., & Nottingham, K. (2024). *The UCI machine learning repository*. Retrieved from: https://archive.ics.uci.edu

[24] Arzamasov, A. (2018). Electrical grid stability simulated data. *UCI Machine Learning Repository*. https://doi.org/10.24432/C5PG66

[25] Ordoni, E., Bach, J., & Fleck, A. K. (2022). Analyzing and predicting verification of data-aware process models: A case study with spectrum auctions. *IEEE Access*, *10,* 31699–31713. https://doi.org/10.1109/ACCESS.2022.3154445

[26] Brooks, T., Pope, D., & Marcolini, M. (2014). Airfoil self-noise. *UCI Machine Learning Repository*. https://doi.org/10.24432/C5VW2C

[27] Tufekci, P., & Kaya, H. (2014). Combined cycle power plant. *UCI Machine Learning Repository*. https://doi.org/10.24432/C5002N

[28] IBM. (2022). *IBM SPSS modeler* (Version 18.2.2) [Software]. IBM Corporation.

[29] Hernández, V. A. S., Monroy, R., Medina-Pérez, M. A., Loyola-González, O., & Herrera, F. (2021). A practical tutorial for decision tree induction: Evaluation measures for candidate splits and opportunities. *ACM Computing Surveys*, *54*(1), 1–38. https://doi.org/10.1145/3429739

[30] González, S., García, S., Del Ser, J., Rokach, L., & Herrera, F. (2020). A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. *Information Fusion*, *64,* 205–237. https://doi.org/10.1016/j.inffus.2020.07.007

[31] IBM. (2022). *IBM SPSS modeler 18.3: Algorithms guide*. Retrieved from: https://www.ibm.com/docs/en/SS3RA7_18.3.0/pdf/AlgorithmsGuide.pdf

How to Cite: Tan, S. C. (2024). Decision Tree Regression with Residual Outlier Detection. *Journal of Data Science and Intelligent Systems*. https://doi.org/10.47852/bonviewJDSIS42023861