TECHNICAL REPORT 3336
FEBRUARY 2024

An Approach to Improving Trained Decision Tree Based Models Without Data

Dr. Benjamin Michlin
Joshua Duclos
Dr. Jamal Rorie
NIWC PACIFIC

Naval Information Warfare Center (NIWC) Pacific
San Diego, CA 92152-5001

This page is intentionally blank.

TECHNICAL REPORT 3336
FEBRUARY 2024

An Approach to Improving Trained Decision Tree Based Models Without Data

Dr. Benjamin Michlin
Joshua Duclos
Dr. Jamal Rorie
NIWC PACIFIC

Administrative Note:

This report was approved through the Release of Scientific and Technical Information (RSTI) process in February 2024 and formally published in the Defense Technical Information Center (DTIC) in February 2024.

Naval Information Warfare Center (NIWC) Pacific
San Diego, CA 92152-5001

NIWC Pacific
San Diego, California 92152-5001

P. M. McKenna, CAPT, USN
Commanding Officer

M. J. McMillan
Executive Director

ADMINISTRATIVE INFORMATION

The work described in this report was performed by the Basic and Applied Research Division of the Cyber/Science and Technology Department (717), Naval Information Warfare Center (NIWC) Pacific, San Diego, CA. The NIWC Pacific Naval Innovative Science and Engineering (NISE) Program provided funding for this Applied Research project.

Released by
John deGrassie, Division Head
Basic and Applied Research Division

Under authority of
Carly Jackson, Department Head
Cyber/Science and Technology Department

ACKNOWLEDGMENTS

Editor: MRM

EXECUTIVE SUMMARY

The Dataless Decision Tree Improvement (D2TI) algorithm introduces a method for improving the performance of trained decision tree-based regression models without requiring access to any data. This novel approach leverages information contained within the trained decision tree to identify the decision regions. Data points are sampled from these regions in order to train a neural network that is able to better approximate the function modeled by the original decision tree. This method is demonstrated on several benchmark data sets representing varied characteristics and problem domains. A typical r^2 increase of $11.4 \pm 1.5\%$ is observed over the underlying decision tree with one outlier realizing even greater improvement. Considerations and applicability of the method are explored.

This page is intentionally blank.

CONTENTS

Figures

Tables

This page is intentionally blank.

1. INTRODUCTION

Models based upon the decision tree (DT) architecture have become ubiquitous, at least in part, due to their many benefits. These types of models have fast inference times, are relatively interpretable, and have a low resource footprint. Often, it may be desirable to improve these trained models, however in many scenarios the data (training and otherwise) may not have been retained or the data may be sensitive and not disseminated.

A Dataless Decision Tree Improvement (D2TI) method for improving pre-trained decision tree-based models without requiring access to any training, validation, or test data is proposed. The structure of a trained decision tree model contains a minimal set of information: the prediction values at a leaf and the split thresholds at a node. Some decision tree implementations, such as in scikit-learn, contain additional information such as the number of samples arriving at each leaf and node [1]. However, even if the minimal set of information within a tree is assumed (leaf values and split thresholds), the D2TI methodology may show improvement over the base decision tree.

The D2TI method leverages and improves upon the base structure of a decision tree: the step function. By construction, the leaves and decision thresholds create a piece-wise function that acts as a step function. Figure 1 shows an example of a decision tree fit to a one-dimensional input (a noisy sine wave). Each component of the decision tree output, $f(x)$, defines a "decision region" and represents a single-leaf value. The discontinuities in $f(x)$ may not represent an optimal approximation for all datasets. The motivating concept behind the D2TI method is to create a smooth transition between decision regions to produce a more performant model without requiring access to data.

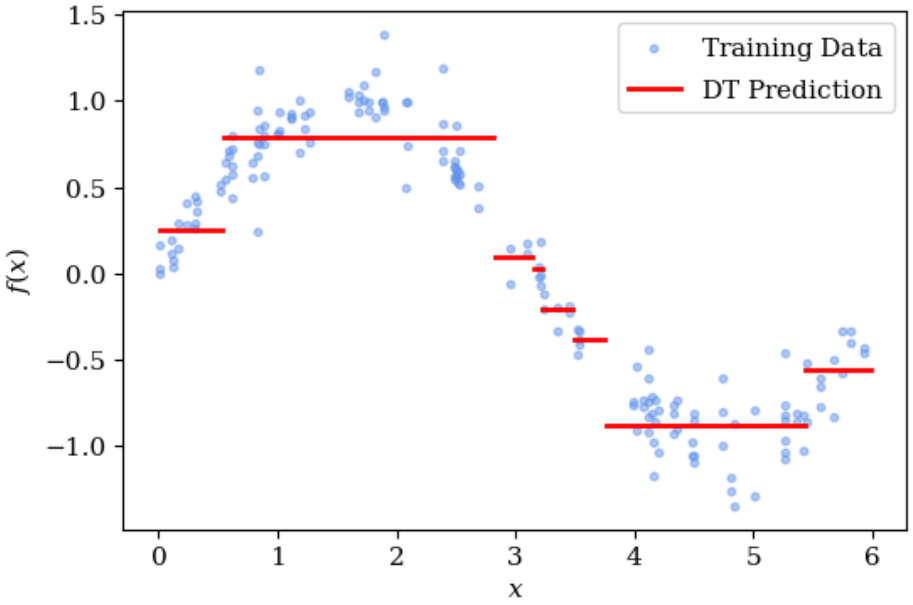


Figure 1. An example decision tree output (red) $f(x)$ vs. input feature x is shown with the training data (blue).

Section 2 presents the D2TI methodology; Section 3 outlines the benchmark datasets and benchmarking details; Section 4 shows the results of the D2TI method; Section 5 discusses considerations and applicability of the D2TI method; and Section 6 proposes future work and reviews the conclusions.

This page is intentionally blank.

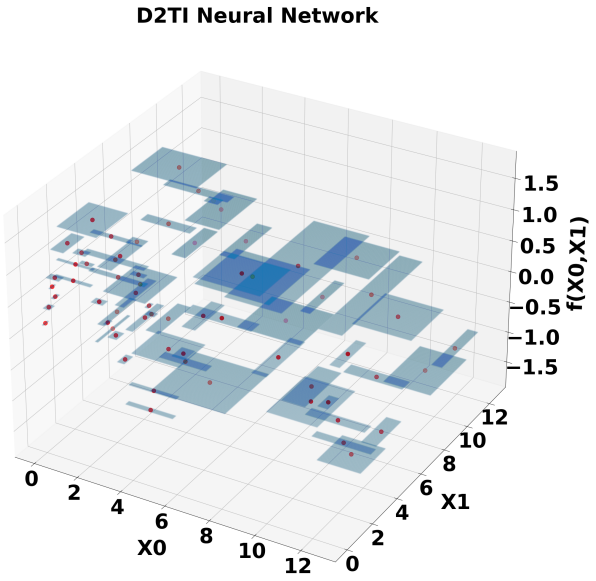
2. METHODOLOGY

Application of the D2TI method begins with the assumption that there is a trained decision tree-based model and its performance must be improved without access to any data. The first step is to analyze the tree structure to find the decision thresholds and to define the decision regions. The decision regions are defined by minimum and maximum values for each feature in the dataset corresponding to a particular prediction value.

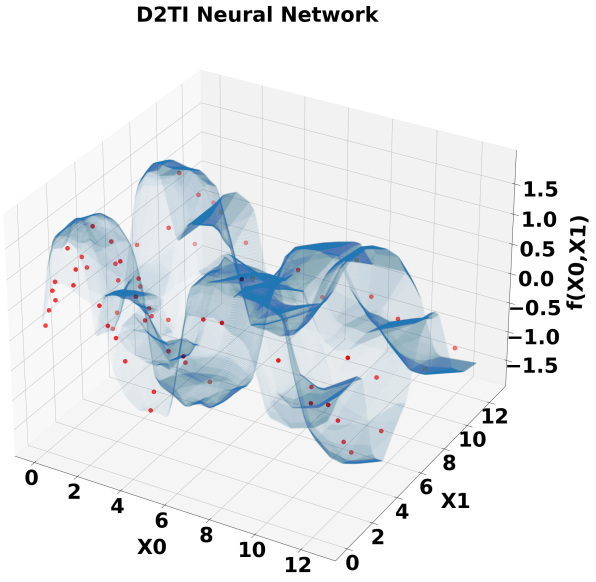
There are cases where a decision region does not have a minimum or maximum value that is well defined. For example, if the decision path to a leaf does not rely on a given feature then that leaf is applicable to all values of that feature. Boundary decision regions are similarly affected as they have a well-defined upper or lower bound for a feature, but not both. For this iteration of the D2TI method, these upper and lower bounds for each region are provided by the true minimum and maximum value of that feature available in the data set. In practice, these values could be supplied by a subject matter expert (SME) or filled with a reasonable assumed value.

Next, points are sampled from each decision region. A neural network is then trained with the sampled points as input vectors and the leaf values (the decision region values) as targets. The result of this process for a two-dimensional dataset is shown in Figure 2. In this illustrative example, the base decision tree is trained over a two-dimensional sine wave, and the D2TI method shows an improvement over the original decision tree. Concretely, the base decision tree shown in Figure 2a was trained on noisy independent variables $X0$ and $X1$, and $f(X0, X1)$ is the decision tree output showing the decision regions. The D2TI method result shown in Figure 2b was then derived by sampling points from the decision tree regions and modeling them with a neural network. This processes yields a 4.4% improvement over the base decision tree without utilizing the data that was used to train and validate the original decision tree model nor having any knowledge of the underlying process.

It is important to note that as the number of features utilized by a tree increases, the process of region sampling becomes exponentially more complex. The shape of each decision region is an n-dimensional hypervolume, where n is the number of features used by the decision tree thresholds. This can make sampling a thoroughly representative number of points per decision region non-trivial. In order to sample points throughout the entire decision region, a stochastic sampling loop is used to select points from the Cartesian product defining the region with the number of sampled points per region determined empirically.



(a)



(b)

Figure 2. The result of the D2TI method on a two-dimensional dataset. Figure (a) shows the decision regions of the trained decision tree. The red points show the midpoint of each decision region. Figure (b) shows the decision surface output from the D2TI method. The red decision region midpoints are shown for reference.

3. BENCHMARKING

In order to test the D2TI methodology, the process is applied to five standard machine learning datasets: Accelerometer [2], Air Quality [3], CA Housing [4], Diabetes [5], and Red Wine [6]. Information from the datasets is shown in Table 1. These datasets were selected because they cover a wide variety of characteristics: real and categorical data, varying numbers of features, and varying sample sizes.

Table 1. Benchmark datasets.

Dataset	Num. Features	Feature Types	Num. Samples
Accelerometer	5	Real, Categorical	153000
Air Quality	15	Real	9358
CA Housing	10	Real, Categorical	20640
Diabetes	10	Real, Categorical	442
Red Wine	12	Real	4898

A simple preprocessing procedure is performed on the benchmark datasets prior to fitting the base decision tree. For the Accelerometer dataset, categorical features are one-hot encoded. For the Air Quality dataset, a single mostly-missing column is dropped and a standard scaler is applied. For the CA Housing dataset, commonly-engineered features are created (rooms per household, bedrooms percentage, and persons per household), a categorical feature is one-hot encoded, a standard scaler is applied, and rows with mostly-missing features are dropped. For the Diabetes dataset, categorical features are one-hot encoded and a standard scaler is applied. For the Red Wine dataset, a standard scaler is applied.

To train the base decision tree, a stratified train-test split is performed on the datasets with a 20% hold-out set. No additional pre-processing is performed and a default parameter scikit-learn decision tree regressor (version 1.1) is fit to the training set. Ten-fold cross-validation is used to gauge performance metrics – in particular the coefficient of determination r^2 – and to determine their uncertainty bounds.

The decision tree is then improved using the D2TI method. It was determined that 100k points should be stochastically sampled from each decision region and used for modeling. Note that as the depth of the tree increases, so does the number of samples used to train the neural network. Additionally, if the feature space of the application dataset is much larger than the benchmark datasets, then the number of sampled points per region should be increased.

For consistency, the deep neural network employed has a static architecture designed to be sufficiently deep and general enough to apply to multiple datasets without fine tuning. It has a hidden layer with 40 neurons, followed by three hidden layers with 64 neurons, a hidden layer of 40 neurons, and a final hidden layer with 8 neurons. The mean squared error is used as the loss function, and the Adam optimizer [7] is used with its default settings. The network is trained for 1000 epochs, and 10-fold cross-validation is used to measure the r^2 performance.

For comparison, a default scikit-learn random forest (RF) regressor is also trained on the benchmark datasets alongside the decision tree. The same stratification and 10-fold cross-validation techniques are used.

This page is intentionally blank.

4. RESULTS

The results from applying the D2TI method on the five benchmark datasets are shown in Table 2. Uncertainties on the r^2 values are derived from applying 10-fold cross-validation. The trained decision tree depth is shown along with the DT model performance, D2TI method performance, and the performance of a random forest (RF) for a baseline. The same results are shown in Figure 3. The nominal performance values for the D2TI method are always better than those for the base decision tree. In most cases, the D2TI method yields a statistically significant improvement upon the default decision tree; only for the Red Wine dataset are the performances statistically equivalent.

Table 2. Results on the five benchmark datasets. The DT and D2TI tree depth and performance of the DT, D2TI method, and RF are shown.

Dataset	Depth	DT (r^2)	D2TI (r^2)	RF (r^2)
Accelerometer	42	0.698 ± 0.007	0.832 ± 0.002	0.812 ± 0.006
Air Quality	25	0.858 ± 0.012	0.900 ± 0.002	0.933 ± 0.008
CA Housing	34	0.608 ± 0.027	0.702 ± 0.017	0.805 ± 0.014
Diabetes	20	-0.145 ± 0.273	0.461 ± 0.024	0.383 ± 0.155
Red Wine	17	-0.130 ± 0.278	0.034 ± 0.057	0.426 ± 0.099

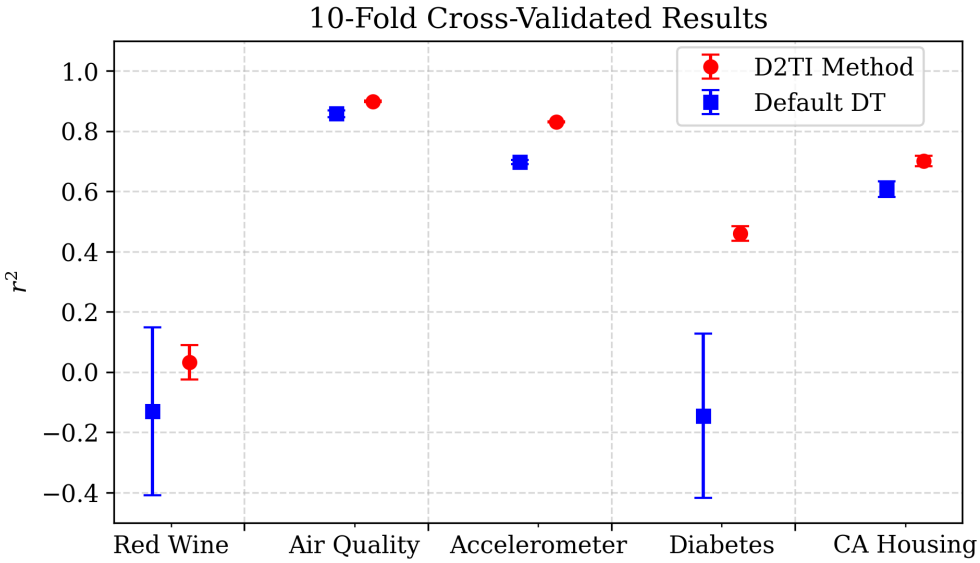


Figure 3. Performance is shown for the base DT model (blue) and the D2TI method (red) for the five benchmark datasets. Uncertainties are derived from 10-fold cross-validation.

In the cases of the Red Wine and Diabetes datasets, when the base learner performs poorly (negative r^2), the D2TI method is still able to improve upon the underlying decision tree model and produce a positive r^2. Further, the minimum improvement of the D2TI method over the default DT in terms of percentage difference in r^2 is $4.6 \pm 1.3\%$ on the Air Quality dataset and the maximum improvement of $488 \pm 1063\%$ on the Red Wine dataset. The mean improvement, discounting the poor-performing Red Wine and Diabetes datasets, is $11.4 \pm 1.5\%$. Additionally, the nominal D2TI method performance is also better than a default random forest for the Accelerometer and Diabetes datasets. For further validation, Figure 4 shows the results on the hold-out test set – all values are consistent with the performance expected from the 10-fold cross-validation.

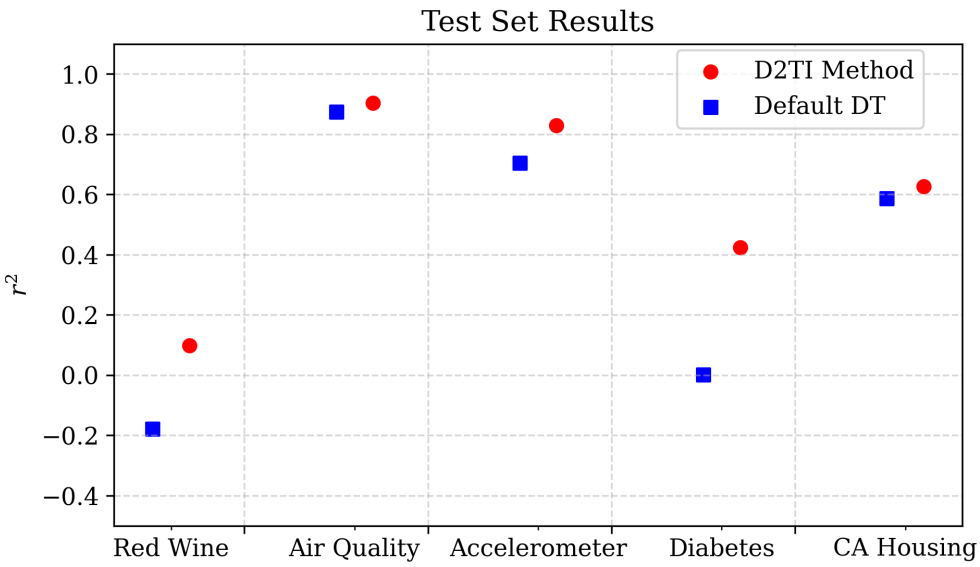


Figure 4. Test set performance is shown for the base DT model (blue) and the D2TI method (red) for the five benchmark datasets.

The depth of the base DT has a significant impact on how well the D2TI method performs. Performance and improvement over the default DT is generally better when the methodology is applied to deeper trees. Figure 5 shows the performance, r^2 vs. the base decision tree depth, for the D2TI method and the default DT for the Accelerometer dataset. For a base tree depth below approximately 20, the default DT outperforms the D2TI method, and for larger tree depths the D2TI method outperforms the default DT. When the base decision tree is trained with default scikit-learn parameters, the tree is expanded to a depth of 42.

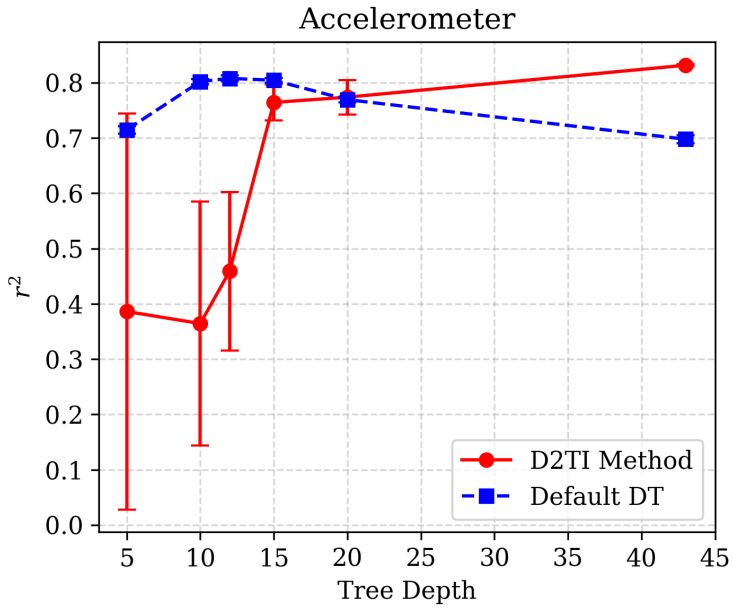
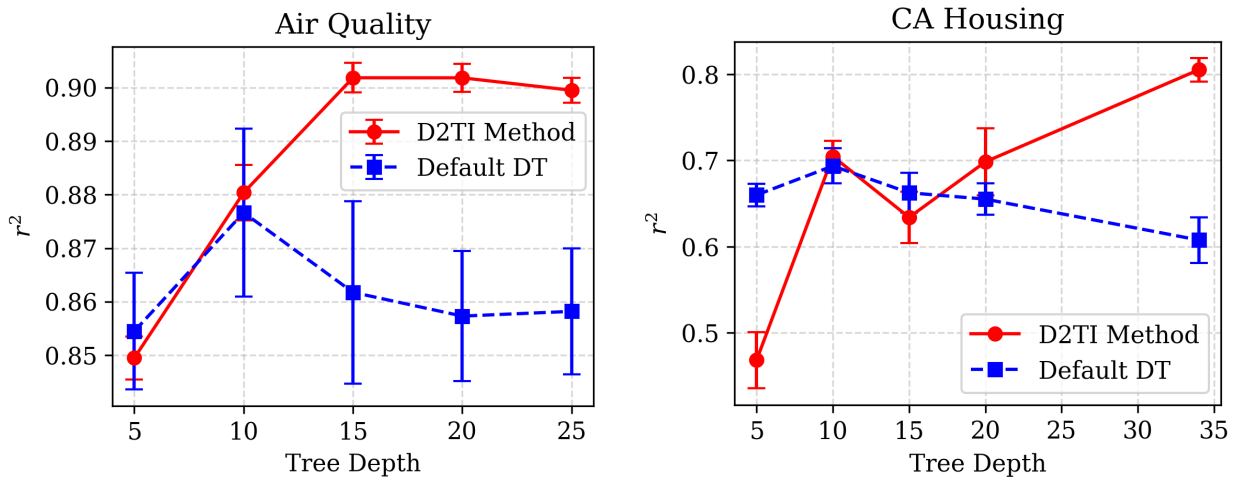


Figure 5. Performance of the D2TI (red) and the default DT (blue) methods are shown vs. the base decision tree depth for the Accelerometer dataset. The default tree has a depth of 42.

Similar performance vs. tree depth behavior is seen for the Air Quality and CA Housing datasets shown in Figure 6. As in Figure 5, Figure 6a shows the same behavior where the performance of the D2TI model improves with tree depth and then plateaus. Figure 6b has the same general behavior but does not seem to plateau – although, a moderately deep base decision tree is required for the D2TI method to improve over a default decision tree. The default tree has a depth of 25 for the Air Quality dataset and 34 for the CA Housing dataset.



(a) (b)

Figure 6. Performance of the D2TI method (red) and the default DT (blue) is shown vs. the base decision tree depth for the Air Quality dataset (a), and the CA Housing dataset (b). The default tree has a depth of 25 for the Air Quality dataset and 34 for the CA Housing dataset.

The performance vs. tree depth for the Diabetes dataset, Figure 7a, shows a different behavior. D2TI method improvement over the default decision tree performance is seen for all base decision tree depths, except for possibly a depth of 5 where the D2TI method and default DT are consistent within uncertainty. The default decision tree trained on the Diabetes dataset has a depth of 20.

The performance vs. tree depth for the Red Wine dataset, Figure 7b, shows a different behavior from the other four benchmark datasets. Here, the performance of both the D2TI method and the default DT decrease with tree depth. However, although they are consistent within uncertainty, the nominal performance values of the D2TI method are consistently better than those for the default decision tree. The default decision tree trained on the Red Wine dataset has a depth of 17.

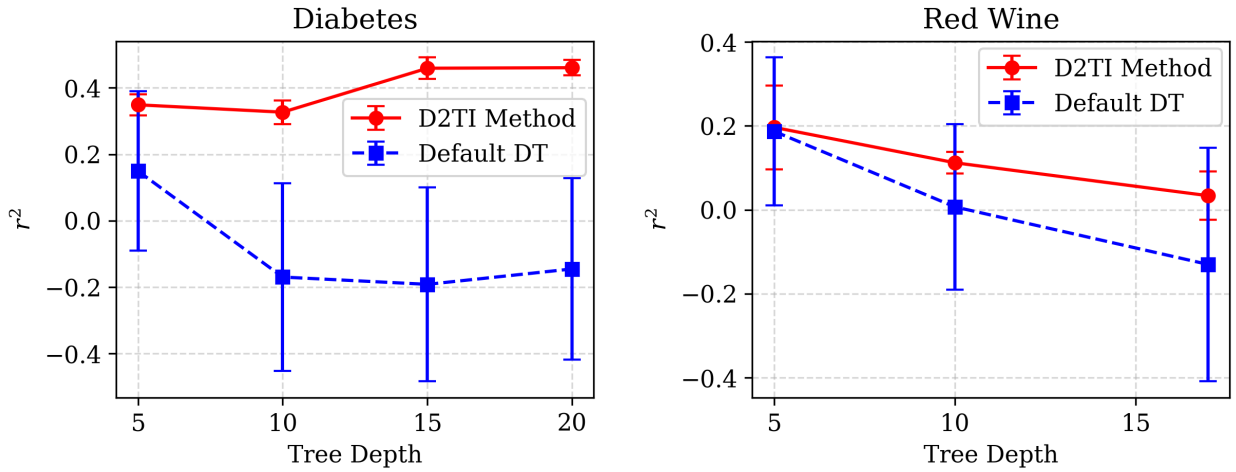


Figure 7. Performance of the D2TI (red) and the default DT (blue) methods are shown vs. the base decision tree depth for the Diabetes dataset (a), and the Red Wine dataset (b). The default tree has a depth of 20 for the Diabetes dataset, and 17 for the Red Wine dataset.

5. CONSIDERATIONS AND APPLICABILITY

When reviewing the results in Section 4, it can be seen that the performance improvement observed from the D2TI method is directly related to the tree depth of the base decision tree. In the case of the five benchmark datasets, when a default decision tree is used as the base model, a significant performance improvement is seen from the D2TI method. However, if the base tree depth were to be tuned to the best cross-validated score, there would not be an improvement from the D2TI method on the Accelerometer dataset. For this dataset, a practitioner may have tuned the tree depth to be 12. For the other four benchmark datasets, if the base decision tree models were to be tuned, the performance would likely be equivalent within uncertainty.

Additional considerations must be made regarding length of training (number of epochs) and overfitting of the D2TI neural network. Here, an "overfit" neural network will converge to the decision tree and provide little of the desirable smoothing characteristics. This has practical utility since the neural network can not be overfit to the point of providing worse performance than the base decision tree – in the worst case scenario it will match the base decision tree performance exactly. However, it is often unclear when training should be stopped to ensure the neural network is appropriately "underfit". The more underfit the network is the more pronounced the smoothing effect will be between the decision regions. For severely underfit networks, not all decision regions may be captured by the neural network. For this paper, the number of epochs was set at 1000 and is proposed as a sufficient default.

Finally, while this paper focuses strictly on regression, there are no conceptual issues barring application of the same D2TI methodology to probabilistic classification problems.

This page is intentionally blank.

6. FUTURE WORK AND CONCLUSIONS

Future work includes expanding the methodology and applying it to ensemble models such as random forests and boosted models using a decision tree as a base estimator. This is expected to be fruitful; if a single tree is improved by the D2TI method, then an entire ensemble of improved trees is expected to be more performant than the traditional implementation.

Additional work is being performed to tune the number of training epochs for the neural network. Several methods are being explored including checking the D2TI neural network performance on sample points central to a decision region and comparing it to the performance of samples at the decision region's edge.

Also, as mentioned in Section 2, there are cases where the minimum and maximum values of a feature must be estimated. In the current iteration of the D2TI method, these values are taken directly from the data; in practice, these values must be supplied by an SME. Estimation methods for these values, not requiring access to the data or subject matter expertise, are currently being explored. This includes using additional information contained in a decision tree such as the number of samples in each leaf. Finally, the relationship between the tree depth and D2TI method performance requires further study.

In conclusion, a method of improving trained models that are based on decision trees without access to data is proposed. The initial results of the D2TI methodology for regression analysis are extremely promising and a statistically significant benefit is seen in the coefficient of determination over a variety of benchmark datasets.

This page is intentionally blank.

REFERENCES

1. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. 2011. "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830.

2. Scalabrini Sampaio, G., Vallim Filho, A. R. d. A., Santos da Silva, L., and Augusto da Silva, L. 2019. "Prediction of Motor Failure Time Using An Artificial Neural Network," *Sensors*, vol. 19, no. 19, URL `https://www.mdpi.com/1424-8220/19/19/4342`.

3. De Vito, S., Massera, E., Piga, M., Martinotto, L., and Di Francia, G. 2008. "On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario," *Sensors and Actuators B: Chemical*, vol. 129, no. 2, pp. 750–757, URL `http://dx.doi.org/10.1016/j.snb.2007.09.060`.

4. Pace, R. K. and Barry, R. 1997. "Sparse spatial autoregressions," *Statistics & Probability Letters*, vol. 33, no. 3, pp. 291–297.

5. Dua, D. and Graff, C. 2017. "UCI Machine Learning Repository," URL `https://archive.ics.uci.edu/ml/datasets/Diabetes`.

6. Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. 2009. "Modeling wine preferences by data mining from physicochemical properties," *Decision Support Systems*, vol. 47, no. 4, pp. 547–553, URL `http://dx.doi.org/10.1016/j.dss.2009.05.016`.

7. Kingma, D. P. and Ba, J. 2014. "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*.

INITIAL DISTRIBUTION

84310	Technical Library/Archives	(1)
53629	Dr. Benjamin Michlin	(1)
71740	Joshua Duclos	(1)
53629	Dr. Jamal Rorie	(1)

Defense Technical Information Center
Fort Belvoir, VA 22060–6218 (1)

Naval Innovative Science and Engineering (1)

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
15-02-2024	Final	

4. TITLE AND SUBTITLE

An Approach to Improving Trained Decision Tree Based Models Without Data

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Dr. Benjamin Michlin
Joshua Duclos
Dr. Jamal Rorie
NIWC Pacific

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NIWC Pacific
53560 Hull Street
San Diego, CA 92152–5001

8. PERFORMING ORGANIZATION REPORT NUMBER

TR-3336

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

The NIWC Pacific Naval Innovative Science and Engineering (NISE) Program
53560 Hull Street
San Diego, CA 92152–5001

10. SPONSOR/MONITOR'S ACRONYM(S)

NISE

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

This work was funded under a NISE Applied Research project.

14. ABSTRACT

The Dataless Decision Tree Improvement (D2TI) algorithm introduces a method for improving the performance of trained decision tree-based regression models without requiring access to any data. This novel approach leverages information contained within the trained decision tree to identify the decision regions. Data points are sampled from these regions in order to train a neural network that is able to better approximate the function modeled by the original decision tree. This method is demonstrated on several benchmark data sets representing varied characteristics and problem domains. A typical r2 increase of 11.4 ± 1.5% is observed over the underlying decision tree with one outlier realizing even greater improvement. Considerations and applicability of the method are explored.

15. SUBJECT TERMS

decision tree regression, decision region generalization, dataless training

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Josh Duclos
U	U	U	SAR	26	19b. TELEPHONE NUMBER *(Include area code)* 619-767-4989

Naval Information Warfare Center (NIWC) Pacific
San Diego, CA 92152-5001