A Machine Learning Model Selection considering Tradeoffs between Accuracy and Interpretability

***Abstract*—Using black box machine learning models (e.g., Deep Neural Networks) in high-stakes domains such as healthcare, criminal justice and real-time systems can cause serious problems due to their complexity and poor interpretability. Moreover, model selection with interpretability in addition to accuracy is one of emerging research areas with lack of model agnostic and quantitative interpretability metrics. In this work, we adopt a quantitative interpretability metric, and then introduce a trade-offs methodology between accuracy and interpretability, which can be demonstrated by increasing interpretability of ML models while allowing accuracy to drop up to given thresholds. In our experimental results, interpretability in terms of simulatability operation count (SOC) is improved up to 97% with minimal 2.5% accuracy drop in the MLP estimator on Auto MPG dataset in the regression (up to 64% with minimal 1.5% accuracy drop in the MLP estimator on MNIST dataset in the classification).**

***Index Terms*—Interpretable Machine Learning (IML); Explainable ML; Model Selection techniques;**

I. Introduction

This paper is an extension of work originally presented in ICITEE 2021 with 1)addition of classification problem and 2)more clarified outcomes (i.e. graphs, tables) [1].

Machine learning (ML) models are actively being adopted in various fields including high-stakes fields such as public health and criminology. However, most of them can be described as 'black box' models with lack of transparency and accountability [2]. For example, a CNN model misleads its prediction due to incorrect learning (lack of accountability) when the model has learned to detect a metal token that radiology technicians place on the patient in the corner of the image field of view at the time they capture the image [3]; this is because it is challenging to detect such behavior due to the intrinsic black-box model properties (no transparency).

According to interpretable machine learning (IML) by Molnar et al. [4], there is an increasing interest to consider interpretability in addition to accuracy in model selection. However, due to lack of quantitative evaluation methods, assessing interpretability is a challenging task. Moreover, even if we can develop or adopt a quantitative interpretability metric, another challenging issue is how can we improve accuracy and interpretability together? (Or if it is extremely challenging, with trade-offs concepts, alternatively how can we improve interpretability with minimal accuracy drop?). Generally, accuracy and interpretability are inversely related to each other [19] [20]. Therefore, one realistic alternative can be the practicality of such trade-offs, in other words can we build simpler (high interpretability) and accurate 'enough' (acceptable accuracy drop) models?

To answer these questions, we adopt a simple yet effective quantitative interpretability metric, simulatability operation count (SOC) [5], following the main contributions of this paper.

- Evaluate accuracy and interpretability of popular models for regression and classification problems: tree based models, multi-layer perceptron (MLP) and support vector machine (SVM).
- Apply a trade-offs method between accuracy and interpretability to improve interpretability of the estimators, by allowing accuracy to decrease up to certain thresholds.

II. Motivation and Related Work

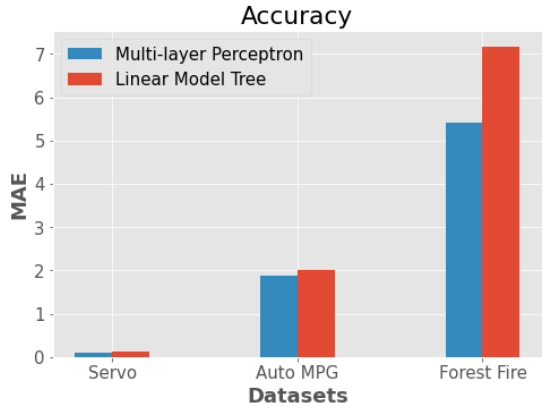
A. Motivation

Although superiority of black-box models lead to their widespread use, they have lower interpretability compared to tree-based models (e.g., linear model tree), which are able to perform competitively on the both regression and classification tasks As shown in Figure 1a of the motivating example, the linear model tree (LMT) compared to multi-layer perceptron regression (MLP), LMT has similar accuracy (MAE) on the two simple datasets (Servo and AutoMPG), and a lower accuracy (higher MAE) on a complex dataset (Forest Fire). Whereas, the interpretability of LMT in terms of SOC is significantly better (lower SOC) than MLP in the Servo and AutoMPG datasets (Figure 1b).

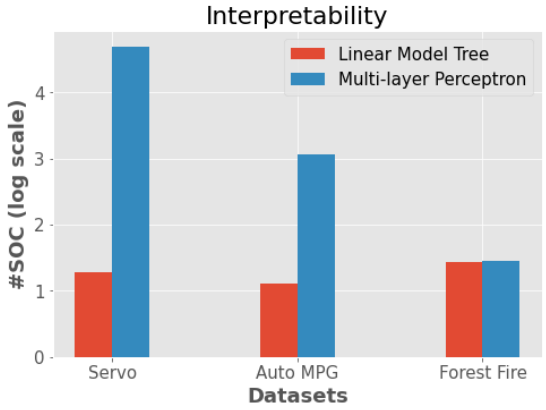
Therefore, for relatively simple datasets (Servo and AutoMPG), we select the LMT model to improve interpretability with negligible accuracy drop. However, if the accuracy drop can not be negligible for complicated datasets like the Forest Fire, we can simplify a complex model by tuning its hyper-parameters to make it more interpretable (e.g. by reducing number of hidden layers or neurons in MLP) within acceptable accuracy range. As a by-product, size of the model may be reduced, training and inference speed may be increased.

B. Related Work

Interpretability/explainability of ML models and explainable AI are now emerging research areas due to wide range use of AI technologies [23]. Rudin [2] argued that using interpretable models is preferred and that they are able to replace complex 'back box' models in terms of transparency and accountability. Farquad et al. [6] extracted if-then based rules from Support Vector Machine with hybrid method: first fit data to SVM and get a reduced set of training data represented by support vectors and train another explainable model. Doshi-Velez and



(a) Accuracy (MAE)



(b) Interpretability (#SOC)

Fig. 1: Motivating Example: Accuracy and interpretability of LMT and MLP Algorithms

Kim [7] introduced a human-grounded proxy metrics that can be built by analyzing human evaluation of interpretability of the model itself or a post-hoc interpretation of a black-box model. Slack et al. [8] performed a study of simulatability and 'what if' local explainability for decision tree, logistic regression and neural network. As a result, the run time operation count was proposed as a global interpretability metric. The authors of [5] derived the SOC formula for several regression models and evaluated their interpretability. The SOC formulas will be adopted to compare interpretability of selected models in our experiments.

III. Methodology

Figure 2 demonstrates the workflow of our experiments. Phase I contains traditional model building steps, which consist of two stages: Data Preprocessing and Model Training. In the preprocessing stage we 1) convert categorical columns of datasets into numerical using OrdinalEncoder [10]; 2) scale using StandardScaler [10] to equate initial effect of each feature on regression coefficients; 3) detect and remove outlier data instances with z-score (greater than 3 [14]); 4) drop highly correlated columns to reduce serious multicollinearity (i.e. if Variance Inflation Factor (VIF) is greater than 10) [15]. In addition, in the model training stage, algorithms are trained

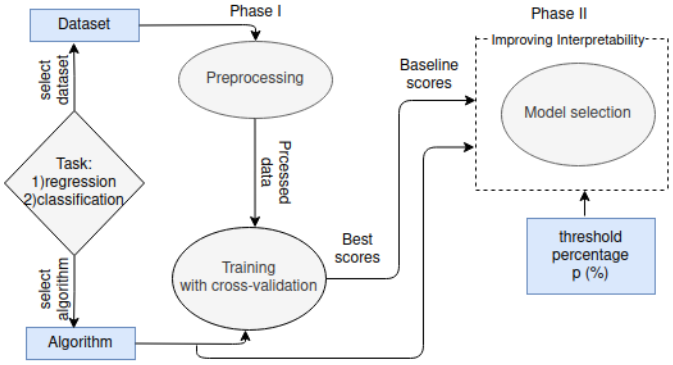


Fig. 2: An Overview of the Trade-offs Methodology

using GridSearchCV from scikit-learn [10] which ensures that we can select the best model among all possible combinations of hyperparameters.

Phase II is the main part of our methodology for model selection. First, the SOC of the selected models from the training stage will be calculated. Then, we try to reduce SOC values by allowing the accuracy to decrease by up to a threshold percentage ($p\%$) from the baseline accuracy scores obtained in the training stage (trade-offs between accuracy and interpretability). It can be achieved by reducing parameters that affect SOC from Table I below.

Esti.	K_t or A_t	SOC formula
LMT	N/A	$2D + 2P + 1$
DT	N/A	$2D + 1$
MLP	A_t	$2 \times N_{H+1} + \sum_{h=1}^{H}(2 \times N_h + A_t) \times N_{h+1}$
	Relu	$A_t = 1$
	Sigmoid	$A_t = 4$
	Tanh	$A_t = 9$
SVM	K_t	$SV \times (K_t + 2)$
	Linear	$K_t = (2P - 1)$
	Polynomial	$K_t = (2P + 1 + d)$
	Sigmoid	$K_t = (2P + 10)$
	RBF	$K_t = (3P + 1)$

TABLE I: SOC formula of Estimators [5]

Since the number of features (P) will be fixed after feature selection, the depth parameter (D) can be reduced to minimize SOC in LMT and Decision Tree (DT). For the MLP, the number of neurons (N), number of hidden layers (H) and type of activation functions (A_t) can be changed. Lastly, for SVM, reducing the number of support vectors (SV) (by tuning hyperparameters, using NuSVR and NuSVC [10]) and choosing a simpler type of kernel function (K_t, e.g. Linear) reduces SOC.

IV. Experimental Results

A. Experimental Setup

Datasets for Regression: To conduct the experiment, three benchmark datasets (from simple to complex) are used: Servo (simple) [11], Auto MPG (medium) [12] and Forest Fire (complex) [13]. The sizes of the chosen datasets are small

because Phase II of our methodology would take enormous time otherwise.

1) Servo: The dataset has 5 features and 167 instances including a bold-highlighted target variable (class). All values of the dataset are discrete; and the target variable is continuous in $[0.13, 7.1]$ and means a raise time of the servo-mechanism. Feature are given in TABLE II

Feature	Description
motor	A,B,C,D,E
screw	A,B,C,D,E
pgain	3,4,5,6
vgain	1,2,3,4,5
class	**0.13 to 7.10**

TABLE II: Servo Features

2) Auto MPG: This dataset has 9 features and 398 instances, predicting attribute 'mpg' (miles per gallon) using 3 multi-valued discrete and 5 continuous attributes. Detailed feature descriptions are given in TABLE III

Feature	Description
mpg	**miles per gallon, continuous output variable**
model year	version of a car
cylinders	power unit of engine
displacement	measure of the cylinder volume
horsepower	power of engine produces
weight	weight of car
acceleration	amount of time taken for car to reach a velocity of 60 miles per hour
origin	multi-valued discrete
name	name of the car

TABLE III: Auto MPG features

3) Forest Fire: This dataset can be represented as a complex dataset having 13 features and 517 instances with a target variable 'area' described in TABLE IV.

Feature	Description
area	**in ha, 0 means less than** 1ha/100 (=100m^2)
X,Y	coordinates of place of fire
month, date	categorical value from jan. to dec. and mon. to sun. correspondingly
temp, wind, rain	meteorological data
RH	relative humidity
FFMC,DMC,DC,ISI	components of Fire Weather Index (FWI) of the Canadian system

TABLE IV: Forest Fire features

Datasets for Classification: The classification datasets include Iris (simple) [24], MNIST (medium) [25] and Pima Indian Diabetes (complex) [26].

4) Iris: The dataset contains 4 features and 150 instances, each of which belongs to one of three classes: Iris Setosa, Iris Versicolour, and Iris Virginica. Features of Iris dataset are real values and described in TABLE V.

5) MNIST: The dataset has 784 features and 70000 (60k training and 10k test images) instances, predicting one of 10 digits. Features of MNIST consists of a 28x28 array of real values of all pixels in the picture.

Feature	Description
sepal length	1.0 - 6.9 cm
sepal width	0.1 - 2.5 cm
petal length	4.3 - 7.9 cm
petal width	2.0 - 4.4 cm
class	**Setosa, Versicolour, Virginica**

TABLE V: Iris features

6) Diabetes: There are 768 instances of 9 features of real values, which are described in TABLE VI. The outcome is positive or negative for diabetes test.

Feature	Description
Pregnancies	0 - 17
Glucose	0 - 199
Blood Pressure	0 - 112
Skin Thickness	0 - 99
Insulin	0 -846
BMI	0.0 - 67.1
Diabetes Pedigree Function	0.078 - 2.42
Age	21 - 81
class	**0, 1**

TABLE VI: Diabetes features

Algorithms for Regression: LMT uses an estimator [22] based on Quinlan's M5 design [9]; and MLP Regressor and SVR are taken from scikit-learn library [10]. Mean Absolute Error (MAE) is used as an accuracy metric.

Algorithms for Classification: DT, MLP Classifier and SVM implementations of scikit-learn library [10] were used to deal with classification task. The percentage of correct predictions (Accuracy) is used as an accuracy metric.

B. Results and Analysis

Preprocessing for Regression: After the preprocessing stage, Servo is reduced to 152 data instances; and Auto MPG is reduced to 367 data instances with 6 features ('displacement' and 'horsepower' are dropped due to collinearity issue, i.e. VIF $>$10; and 'name' is not used); and Forest Fire is reduced to 468 data samples.

Model Training for Regression: As mentioned previously, models are trained using GridSearchCV which allows testing a wide range of hyper-parameter combinations. As illustrated in Figure 3, the lowest error corresponds to $gamma = 0.05$ and $C = 1000$ and concave down shape ensures that we select the suboptimal points.

	Servo	Auto MPG	Forest Fire
LMT	0.133	1.889	6.847
MLP	**0.096**	1.890	5.376
SVR	0.183	**1.830**	**5.212**
Lin. Reg.	0.863	2.304	6.723
Other Ref.	0.220 [16]	2.020 [17]	6.334 [18]

TABLE VII: Accuracy Performance (MAE) of Trained Models with references. (Lowest error values in bold).

Overall results using GridSearchCV in the model training stage are provided in Table VII. MLP outperforms other estimators on Servo dataset, SVR on Auto MPG and Forest Fires datasets. For comparison purposes, results of the Scikit

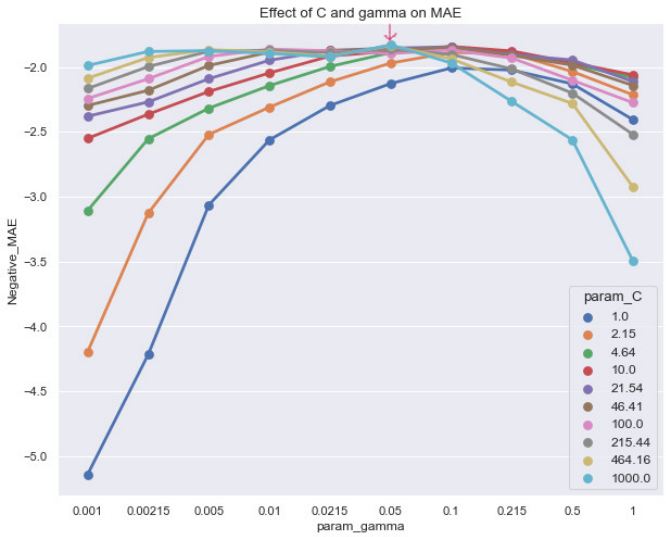


Fig. 3: Training SVR on Auto MPG dataset.

Learn's default Linear Regression and other references are provided.

Model Training for Classification: Like the regression training phase, GridSearchCV is used to find best combinations of hyperparameters for the models. One of the examples

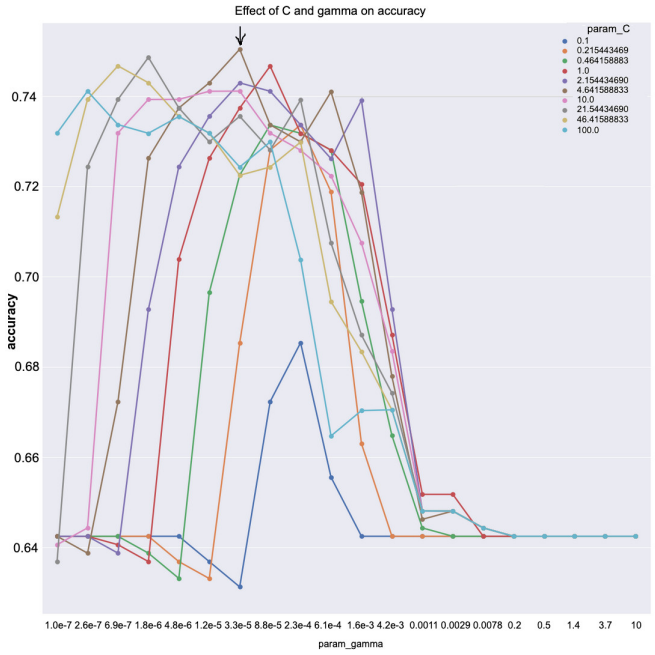


Fig. 4: Training SVM on Diabetes dataset.

of parameter-tuning is shown in Figure 4, where optimal values are $gamma = 0.00003$ and $C = 4.64$.

Results from the training stage are provided in the Table VIII. On MNIST dataset Random Forest has the best results, while SVM outperforms other models on IRIS and Diabetes

	Iris	MNIST	Diabetes
DT	96.7%	79.0%	75.4%
Random Forest	97.7%	**97.2%**	76.5%
MLP	83.3%	94.9%	75.7%
SVM	**98.7%**	96.0%	**77.1%**
Other Ref.	98.7% [26]	99.7% [27]	76.0% [28]

TABLE VIII: Accuracy Performance of Trained Models with references. (Highest accuracy values in bold).

dataset. As a comparison the results of Random Forest model from the Scikit Learn library were provided.

Model Selection for Regression: Following two approaches to improve interpretability are discussed in this paper: 1) model is replaced by another simpler model and 2) the same model is simplified by tuning its hyperparameters (e.g. reducing the number of layers in MLP).

Results of the first approach are shown in Figure 5 and the idea behind is to show how different models behave when they are optimized for interpretability using the trade-offs methodology. The point $(0, 0)$ corresponds to the baseline accuracy and SOC score of the estimator on the given dataset. These are the first (top most) entries of each algorithm and dataset pair on Table IX. For example, for LMT and Servo, baseline score is $(0.133, 19)$. The percentage of the increase or decrease is calculated with respect to the baseline score. Next entry in LMT and Servo on Table IX is $(0.134, 17)$, which corresponds to $(0.75 = (0.134 - 0.133)/0.133 \times 100, 10.52 = (19 - 17)/19 \times 100)$ point on the blue graph (encircled with red) on Figure 5.

All estimators (MLP, SVR and LMT) behave similarly on regression task. For example, on Servo dataset SOC is reduced significantly (by around $85\%, 17\%$ and 11%) for small increases in error rate ($2.1\%, 1.6\%$ and 0.75% respectively). MLP is the most accurate estimator for Servo (with 0.096 MAE value). If 2% reduction in accuracy is feasible for Servo dataset, MLP's interpretability could be increased by 85%. Alternatively, if MAE value of 0.133 is acceptable for Servo dataset, LMT algorithm with SOC value of only 19 could be used instead of MLP.

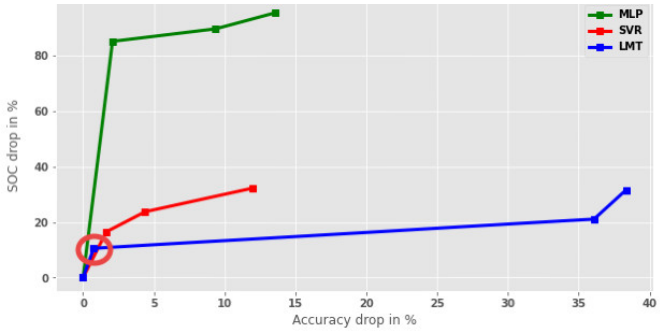


Fig. 5: Comparison of Models in Accuracy and Interpretability on the Servo dataset.

Table IX is an extension of Figure 5 by adding the results of Auto MPG and Forest Fire datasets. Similar to the figure,

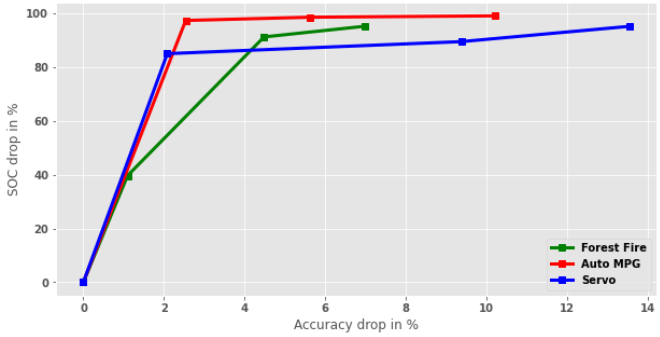


Fig. 6: Trade-off between Accuracy and Interpretability in MLP estimator.

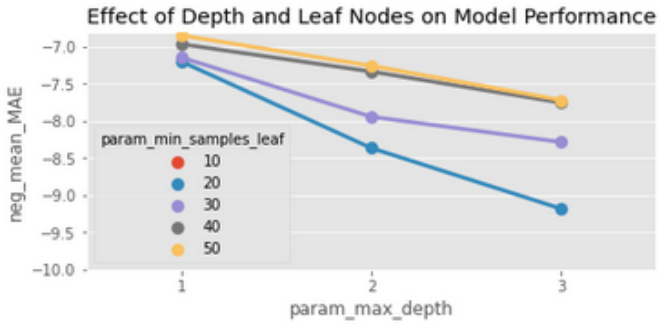


Fig. 7: Performance of LMT on Forest Fire dataset.

significant improvement in interpretability can be achieved with small reduction in accuracy. In the case of LMT on Auto MPG and Forest Fire datasets, the most accurate results are obtained with tree depth of 1 (see Figure 7), hence model cannot be simplified further.

	LMT acc., SOC	MLP acc., SOC	SVR acc., SOC
Servo	0.133, 19 0.134, 17 0.181, 15 0.184, 13	0.096, 339371 0.098, 50583 0.105, 35403 0.109, 16103	0.183, 2280 0.186, 1905 0.191, 1740 0.205, 1545
Auto MPG	1.889, 13 - - - - - -	1.890, 45393 1.938, 1163 1.996, 613 2.083, 383	1.830, 6264 1.876, 1494 1.918, 1242 2.006, 1062
Forest F.	6.847, 27	5.376, 806 5.435, 486 5.617, 70 5.751, 38	5.212, 10550 5.347, 7425 5.479, 6850 5.725, 5700

TABLE IX: Comparison of models in terms of accuracy and interpretability (acc. is short for accuracy).

Results of the second approach are shown in Figure 6 using MLP on the three datasets. The estimator behaves similarly on all the datasets - SOC can be decreased significantly with small reduction in error rate. The elbow points can be suitable candidates for effective trade-offs between accuracy and interpretability, since after that point (when moving from left to right) a slope of the graph sharply decreases. For example, for the Auto MPG dataset (red line) interpretability can be improved (reduction of SOC) by 97% with only 2.5% drop (increase in MAE) in accuracy. The similar concept applies to other datasets also.

Model Selection for Classification: For the classification task, similar approaches as for regression were applied; and Figure 8 summarizes the results of the first approach where trade-offs between accuracy and interpretability for all the models (DT, MLP and SVM) on MNIST dataset are depicted. It could be seen from Figure 8 that all graphs start at point (0, 0), which is baseline scores for accuracy and interpretability. These baseline score correspond to last entries (with highest accuracy and SOC values) of each algorithm and dataset pair on Table X. For instance, for MLP and MNIST baseline score is (94.9%, 10719). Percentage change in SOC or accuracy are calculated with respect to the baseline score, for example, next entry in MLP and MNIST on Table X is (93.4%, 3879), and it gives a red point (1.5% = (94.9% - 93.4%), 64% = (1-(3879/10719))*100) on Figure 8. Overall, models perform in the same way on the classification task. For instance, interpretability could be increased dramatically (by 7.3%, 64%, and 40%) in exchange to small decrease in accuracy (6.62%, 1.5%, and 3.2% correspondingly). SVM is the most accurate model (accuracy 96%) for MNIST dataset, since DT is interpretable intrinsically, its interpretability could not be improved effectively with trade-offs method.

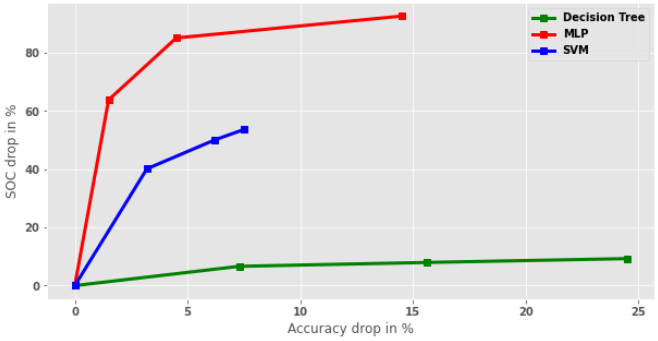


Fig. 8: Comparison of Models in Accuracy and Interpretability on the MNIST dataset.

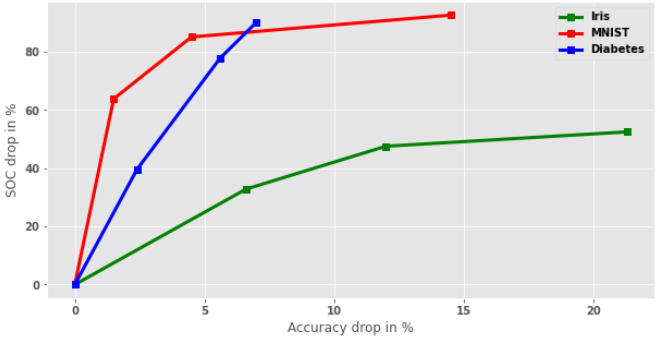


Fig. 9: Trade-off between Accuracy and Interpretability in MLP estimator for classification.

Table X is a more detailed version of Figure 8, and it shows

significant increases of interpretability by allowing some drops in accuracy.

	DT acc., SOC	MLP acc., SOC	SVM acc., SOC
Iris	66.7%, 11	62.0%, 29	96.7%, 117
	93.3%, 13	71.3%, 32	97.3%, 162
	96.0%, 15	76.7%, 41	98.0%, 198
	96.7%, 17	83.3%, 61	98.7%, 252
Mnist	54.5%, 137	80.4%, 794	88.5%, 45220
	63.4%, 139	90.4%, 1599	89.8%, 48720
	71.7%, 141	93.4%, 3879	92.8%, 58380
	79.0%, 151	94.9%, 10719	96.0%, 97500
Diabetes	72.3%, 19	68.7%, 103	75.7%, 5185
	73.7%, 21	70.1%, 231	76.6%, 5338
	73.8%, 23	73.3%, 627	77.1%,5712
	75.4%, 27	75.7%, 1038	- -

TABLE X: Comparison of models in terms of accuracy and interpretability (acc. is short for accuracy) for classification.

The second approach was to test one of the models on three datasets, for example MLP (Figure 9). The model performs similarly on all datasets and SOC could be improved at cost of lowering accuracy. For example, 64% increase of interpretability would require 1.5% reduction in accuracy on MNIST.

V. Conclusion

In this paper, we proposed a trade-offs method between accuracy and interpretability by adopting the model agnostic quantitative metric, SOC. LMT algorithm, due to its simple yet sophisticated structure (combination of decision tree and linear regression models), is the most interpretable model among the evaluated regression estimators, having almost the same accuracy with MLP in relatively simple-medium datasets such as Servo, Auto MPG. In addition, for all types of estimator interpretability can be improved more by allowing accuracy to drop/sacrifice to a certain threshold.

Decision Tree algorithm is the most interpretable model among evaluated estimators due to its simplicity on classification task. It outperforms MLP on Iris and shows competitive results on Diabetes dataset. However it has the lowest accuracy on MNIST with a large gap from other algorithms.

This paper demonstrates the tradeoff method between accuracy and interpretability using SOC metric. SOC is a model agnostic quantitative metric, hence it allows fair comparison between different types of estimators. In our experiments decreasing SOC leads to simpler model with less memory requirement and faster inference speed. However, in general lower SOC may not always result in model with small memory requirement (i.e. replacing complex operation with simpler one) and faster inference speed (parallelizable model with high SOC can be faster than purely sequential model with low SOC on parallel hardwares).

Acknowledgement

This work was partly supported by the Nazarbayev University (NU), Kazakhstan, under FDCRGP grant 021220FD0851.

References

[1] Z. Nazir, D. Kaldykhanov, K. -K. Tolep and J. -G. Park, "A Machine Learning Model Selection considering Tradeoffs between Accuracy and Interpretability," 2021 13th International Conference on Information Technology and Electrical Engineering (ICITEE), 2021, pp. 63-68, doi: 10.1109/ICITEE53064.2021.9611872.

[2] Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5), 206-215.

[3] Zech, J. R., Badgeley, M. A., Liu, M., Costa, A. B., Titano, J. J., Oermann, E. K. (2018). Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. PLoS medicine, 15(11), e1002683.

[4] Molnar, C., Casalicchio, G., Bischl, B. (2020). Interpretable Machine-Learning–A Brief History, State-of-the-Art and Challenges. arXiv preprint arXiv:2010.09337

[5] Park, J.-G., Dutt, N., and Lim, S.-S. An Interpretable Machine Learning Model Enhanced Integrated CPU-GPU DVFS Governor. ACM Trans. Embed. Comput. Syst. (TECS), 20(6): 108:1-108:28 (2021).

[6] Farquad, M. A. H., Ravi, V., Raju, S. B. (2010). Support vector regressionbased hybrid rule extraction methods for forecasting. Expert Systems with Applications, 37(8), 5577-5589

[7] Doshi-Velez, F., and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608

[8] Slack, D., Friedler, S. A., Scheidegger, C., Roy, C. D. (2019). Assessingthe Local Interpretability of Machine Learning Models. arXiv preprint arXiv:1902.03501

[9] Quinlan, J. R. (1992). "Learning with continuous classes." Proc., 5thAustralian Joint Conf. on Artificial Intelligence, Adams Sterling,eds., World Scientific, Singapore, 343–348

[10] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[11] Servo dataset (1993). Link: https://archive.ics.uci.edu/ml/datasets/Servo

[12] Auto Mpg dataset. Link: http://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.data-original

[13] Forest Fire dataset. Link: https://archive.ics.uci.edu/ml/datasets/forest+fires

[14] Engineering Statistics Handbook. Nist Sematech. https://www.itl.nist.gov/div898/handbook/eda/section3/eda35h.htm

[15] O'brien, R. M. (2007). A caution regarding rules of thumb for variance inflation factors. Quality quantity, 41(5), 673-690.

[16] Cortez, P., Embrechts, M. J. (2013). Using sensitivity analysis and visualization techniques to open black box data mining models. Information Sciences, 225, 1-17. page 17.

[17] Birnbaum, L. A. (Ed.). (2014). Machine Learning Proceedings 1993: Proceedings of the Tenth International Conference on Machine Learning, University of Massachusetts, Amherst, June 27-29, 1993. Morgan Kaufmann. page 240.

[18] Stanford-Moore, A., Moore, B. Wildfire Burn Area Prediction. page 5.

[19] Johansson U, Sönströd C, Norinder U, Boström H. Trade-off between accuracy and interpretability for predictive in silico modeling. Future Med Chem. 2011 Apr;3(6):647-63. doi: 10.4155/fmc.11.23. PMID: 21554073.

[20] Mori, T., Uchihira, N. Balancing the trade-off between accuracy and interpretability in software defect prediction. Empir Software Eng 24, 779–825 (2019). https://doi.org/10.1007/s10664-018-9638-1

[21] Uzair, M., Jamil, N. (2020, November). Effects of Hidden Layers on the Efficiency of Neural networks. In 2020 IEEE 23rd International Multitopic Conference (INMIC) (pp. 1-6). IEEE.

[22] Dillard, L. lmt.py (2017). Link: https://gist.github.com/logandillard/lmt.py.

[23] A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," in IEEE Access, vol. 6, pp. 52138-52160, 2018, doi: 10.1109/ACCESS.2018.2870052.

[24] Iris dataset. Link: https://archive.ics.uci.edu/ml/machine-learning-databases/iris/

[25] MNIST dataset. Link: http://yann.lecun.com/exdb/mnist/

[26] Pima Indian Diabetes dataset. Link: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

[27] Hussain, Z. F., Ibraheem, H. R., Alsajri, M., Ali, A. H., Ismail, M. A., Kasim, S., Sutikno, T. (2020). A new model for iris data set classification based on linear support vector machine parameter's optimization. International Journal of Electrical and Computer Engineering (IJECE), 10(1), 1079.

[28] Cire¸san, D.C.; Meier, U.; Gambardella, L.M.; Schmidhuber, J. Deep, big, simple neural nets for handwritten digit recognition. Neural Comput. 2010, 22,3207–3220. Link: http://dx.doi.org/10.1162/NECO_a_00052

[29] B. Chandra and V. P. Paul, "A Robust Algorithm for Classification Using Decision Trees," 2006 IEEE Conference on Cybernetics and Intelligent Systems, 2006, pp. 1-5, doi: 10.1109/ICCIS.2006.252336.