A comprehensive study on the performance of different Multi-class Classification Algorithms and Hyperparameter Tuning Techniques using Optuna

Johnsymol Joy
Computer Science and Engineering
Sathyabama Institute of Science and Technology
Jeppiaar Nagar, Chennai – 600 119
johnsymoljoy07@gmail.com

Dr. Mercy Paul Selvan
Computer Science and Engineering
Sathyabama Institute of Science and Technology
Jeppiaar Nagar, Chennai – 600 119
mercypaulselvan.cse@sathyabama.ac.in

2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS) | 978-1-6654-6883-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IC3SIS54991.2022.9885695

Abstract— **Multi-class classification talks about classification tasks that have three or more classes. It takes the assumption that every data sample in the dataset is assigned to one and only one class. It is a key idea in machine learning that is used in a variety of applications. Machine Learning includes different types of multi-class classification algorithms like SVM, Decision Tree, Random Forest, etc. This paper will discuss the workings of these algorithms and also emphasize the performance analysis of these algorithms. For performance analysis, this paper takes four multi-class datasets, namely the fetal-health classification dataset, the LED display dataset, the mobile price classification dataset, and the E-Coli dataset. For analyzing the performance of above mentioned three algorithms, this paper uses a machine-learning automated tool for hyperparameter optimization called OPTUNA. Different hyperparameter tuning techniques like Tree-structured Parzen Estimator (TPE) optimization, Covariance Matrix Adaptation Evolution Strategy (CMA-ES) optimization, Random Search, Grid Search, etc. were used for analysis purposes.**

Keywords— Machine Learning, Multi-class classification algorithms, supervised learning, SVM classifier, Decision Tree Classifier, Random Forest Classifier, Hyperparameter, Kernel, OPTUNA.

I. INTRODUCTION

Machine learning (ML), data science, and Artificial Intelligence (AI) are the most relevant new technologies in our emerging world. ML is a subset of AI. Machine Learning-based systems are automated systems that recognise patterns in data and make choices using efficient machine learning algorithms. Machine Learning techniques are used in a wide range of applications, including decision making, sentiment analysis, image processing, and speech recognition.

Unsupervised learning and supervised learning are two types of machine learning algorithm [1]. There are no label features in unsupervised learning. This type of algorithm groups data based on some similarities. Some of the most prominent unsupervised learning techniques include clustering algorithms like K-mean, K-medoid etc. For supervised learning algorithms, we have a label attribute and based on that label attribute, data is classified. Regression and classification algorithms are the two categories that come under supervised machine learning methods.

Regression is able to produce a continuous-valued output for new input data given to the algorithm by learning from labeled datasets. When the intended output is a number, such as money or height, it is utilised. Linear regression and logistic regression are the two types of regression. There is a linear relationship between input (X) and output (Y) in linear regression. The independent variable is the input variable, and the dependent variable is the output variable. When the algorithm receives fresh data, it applies the function, estimates the values, and maps the inputs to a continuous value for the output. The logistic algorithm calculates discrete values for the set of independent variables that have been provided to it. Because the algorithm estimates the likelihood of new data, the outcome of logistic regression will fall between 0 and 1.

Classification classifies data into different groups based on a labeled attribute. Classification is of two types: binary classification and multiclass classification. A binary classification algorithm maps the new data to any one of the 2 classes that we have in our dataset. The classes must be mapped to either 0 or 1, which in real life is converted to 'Yes' or 'No'. The output will be one of the classes and not a number like in regression. Multi-class classification refers to the division of a dataset into multiple classes, with more than two classes defined by the class label property.

In this study, the performance of multi-class classification algorithms including SVM(Support Vector Machine), Decision Tree, and Random Forest classifiers was examined using hyperparameter optimization methodologies like Tree-Structured Parzen Estimator, Covariance Matrix Adaptation Evolution Strategy (CMA-ES) optimizer, Random Search, and Grid Search. This study uses an automated method called OPTUNA to optimise hyperparameters.

II. RELATED WORKS

To our knowledge, only a few works address the topic of tunability and the design of tuning search spaces. Philipp Probst, Anne-Laure Boulesteix, and Bernd Bischl [2] present concise and intuitive definitions for optimal defaults of machine learning algorithms, as well as the impact of tuning them jointly, individually, or in combinations, based on the general concept of surrogate empirical performance models. The tunability of elastic net, decision tree, k-nearest neighbours, SVM, random forest, and xgboost, as well as the tunability of their individual parameters, was investigated. One of the shortcomings of the above study is that it only evaluated binary classification datasets, whereas our work uses multi-class classification datasets.

Bergstra and Bengio [3] compute the significance of neural network hyperparameters and find that some are essential on all datasets, while others are only relevant on a few. Their conclusion is mostly visual, and it is used to demonstrate why

random search is preferable to grid search when tuning neural networks.

Hutter et al. [4] take an alternative strategy, using forward selection to determine the most essential hyperparameters. Fawcett and Hoos [5] describe an ablation analysis technique that tries to discover the hyperparameters that contribute the most to enhanced performance following tuning. They calculate the performance improvement that may be gained by altering the value of each of the considered hyperparameters from the original value to the value indicated in the target configuration set by the tuning strategy.

While previous research has focused on the importance of hyperparameters in binary classification algorithms, this paper focuses on multiclass classification and employs an automated tool to compare the performance of various classification algorithms and samplers.

III. MULTICLASS CLASSIFICATION ALGORITHMS

Multi-class classification means classifying the given datasets into multiple classes, which means the class label attribute has more than two classes [6]. Multiclass classification is probably the most common machine learning task. Consider a real-time example: Let's say a cybersecurity firm wants to be able to monitor a user's email and classify incoming emails as different phishing attacks. To do this, it might train a classification model on the email texts and email addresses of previous phishing scams to learn the model to predict which URLs have a tendency to carry threatening emails. In this paper, we will discuss some important multiclass algorithms like SVM, decision tree, and random forest and will analyse their performance.

A. SVM(Support Vector Machine)

SVM [7] is a supervised machine learning technique that is mostly used to solve classification and regression problems. It aids in the discovery of an ideal border between the various outputs. The goal is to create an n-dimensional hyper plane that splits instances or data points into potential classes. The hyper plane should be put as close to the data points as possible. Support vectors are the data points with the shortest distance to the hyper plane [8].

SVM performance is heavily influenced by the kernel function. Aside from the great performance of simulation research, there are no significant principles for kernel selection. Different kernel functions are linear, nonlinear, polynomial, Gaussian, sigmoid, Radial Basis Function (RBF) etc.

B. Decision Tree

Decision trees are flowchart-like tree structures where internal nodes contain features or attributes, branches indicate decision rules, and leaf nodes represent the result class. The root node is located at the top of the hierarchy. To create a decision tree, first determine the optimal attribute using any of the various attribute selection metrics such as Information Gain, Gain Ratio, Gini Index, and so on. The data is partitioned into sub-trees based on the given attribute value. Partitions will happen recursively, so it is called recursive partitioning. The final structure will be a flowchart-like framework that closely resembles human thought. As a result, decision trees are simple to grasp. For multiclass classification, a decision tree classifier [9] is an effective method.

Advantages of decision trees include their ability to easily capture non-linear patterns and their applications in processes like feature engineering. The disadvantages of decision trees are that they are biased with an imbalanced dataset, so it is advised that the dataset must be balanced before starting the procedure. They are also sensitive to noisy data and sometimes over fit noisy data. The smallest variation in data can result in a different decision tree.

C. Random Forest Classifier

A random forest is a tree-based ensemble method. It's a supervised machine learning system that focuses on classification and regression. This classifier combines numerous decision trees into a forest and averages predictions from multiple trees. To create an effective uncorrelated forest of trees, the Random Forest classifier employs bagging and the random subspace approach in the construction of each tree [10]. The final conclusion or prediction will be made using the majority of votes from each node in the decision tree. [11] [12].

IV. HYPERPARAMETER TUNING TECHNIQUES

Machine learning comprises of predicting and classifying data, and to do this, different well-defined machine learning models can be used. Machine learning models are normally parameterized, and therefore it is possible to tune the parameters for a given problem. Every model contains a large number of parameters, and finding the best combination of parameters is referred to as a search problem. Parameters can be model parameters and hyperparameters [13]. Model parameters are internal model configuration variables whose values can be determined using the data provided. They are learned from past data and they can usually not be set manually. They are essential for the model when building prediction models. They are internal variables of a machine learning model. Some examples of model parameters of machine learning algorithms are support vectors in SVM, weights of neural networks etc.

Variables that are external to the model are called hyperparameters [13]. It is usually well-defined manually before the training of the model. Its value cannot be assessed from the datasets. It is initially impossible to determine the appropriate value for hyperparameters for a particular context. However, the trial-and-error method can be used to achieve it. Hyperparameters have a big influence on the model's speed and accuracy. The number of hyperparameters and the type of hyperparameters used are determined by the algorithms. In Random Forest Classifiers, hyperparameters such as n_estimators, which is the number of trees in the forest; and max_depth, which is the maximum number of levels in each decision tree, and the value of K in k-nearest neighbours are examples of hyperparameters.

There are various hyperparameter optimization/tuning methods available. Some of them are random search, grid search, meta-heuristic approaches like genetic algorithms, swarm optimization algorithms, Tree-structured Parzen

Estimator [14], CMA-ES [15], etc. Grid search, random search, TPE, and CMA-ES are highlighted in this study. A search space is defined by random search as a bounded domain of hyperparameter values with randomly sampled points within it. Grid Search assesses every point in the grid and defines a search space as a grid of hyperparameter values . A Sequential Model-Based Optimization (SMBO) method is the Tree-structured Parzen Estimator (TPE). This method develops models progressively to estimate hyperparameter performance from historical measurements, then selects new hyperparameters. The most powerful system for black-box optimization is the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). A multivariate Gaussian distribution is sampled via CMA-ES. After assessing all solutions, the solutions are sorted by assessment values, then updating the distribution parameters based on the ranking of assessment values.

V. Performance Analysis

A. Datasets and Methodology used

Four multi-class datasets are utilized for evaluating the performance of multiclass algorithms like SVM, Decision Tree, and Random Forest. They are the fetal-health classification dataset, the LED display dataset, the mobile price classification dataset, and the E-coli dataset. The fetal-health dataset from Kaggle contains 2126 cases of features extracted from cardiotocogram exams [16]. These instances were classified into 3 classes: normal, suspect, and pathological. LED display datasets [17] are UCI data repository datasets. It has seven Boolean attributes and 10 classes. The class attribute values are integers ranging between 0 and 9. The mobile price classification dataset [18] is a Kaggle dataset that classifies the mobile price range. The E. coli dataset [19] was acquired from Kaggle and is a Probabilistic Classification System for Predicting the Cellular Localization Sites of Proteins.

On the given datasets, we must next apply hyperparameter tuning approaches such as Random Search, Grid Search, Tree-structured Parzen Estimator optimization, and CMA-ES optimization, and compare the results of these three algorithms. Each algorithm has its hyperparameters. Some of them may be more important than others. This paper uses an automated tool called Optuna for performing hyperparameter optimization [20]. Optuna is an open-source framework built on python language for hyperparameter optimization. This framework was developed by a Japanese AI company named Preferred Networks and for automating the search space of hyperparameters, it uses a Bayesian method. It facilitates the use of several hyperparameter optimization methods such as Grid search, Random Search, TPE, and CMA-ES algorithms. In optuna technical terms, they are called samplers. That means Grid sampler, Random sampler, TPE Sampler, and CMA-ES sampler.

B. Test results and analysis

First, we have a Kernel-based approach SVM, Kernels, and penalty (C) are some of the most important hyperparameters of SVM. For getting a better result, the proper selection of Kernel is important. The kernel hyperparameter is used to set the mathematical functions utilized in the Support Vector Machine, which gives the user a window to adjust data [21] [22]. Kernel Function turns the training set of data into a linear equation in a higher number of dimension spaces, allowing a non-linear decision surface to be updated. The important kernel functions are linear, polynomial, and RBF [23]. Here we check Grid search, Randomized Search, TPE, and CMA-ES hyperparameter tuning method using optuna. The results are shown in Table 1. Fig. 1 shows a graphical representation of accuracy values. When we analyses the result, we know that random sampler perform better compared to others.

TABLE I. SVM- ACCURACY FOR DIFFERENT SAMPLERS

Datasets	Optimization Methods			
	TPE Sampler	CMA-ES Sampler	Random Sampler	Grid Sampler
LED	72	72	72.8	72
Fetal-health	84.38	84.76	84.8	83.3
Mobile	88.4	88.2	88.45	88.2
E-coli	84.2	81.52	86.29	76.49

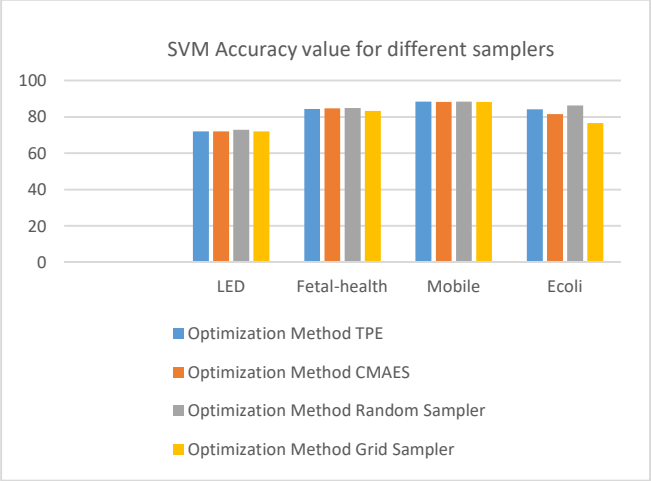

Fig. 1. SVM accuracy values for different datasets and different samplers

The next classifier is the decision tree. Some hyperparameters relevant to decision tree classifiers include max_depth, max_features, min_samples_leaf, and criterion. Max_depth specifies the maximum depth to which you wish the tree to grow. Max_features is the size of the random subsets of features taken while splitting a node. The smallest number of samples required on the leaf node is referred to as min_sample_leaf. The criterion dictates how the impurity of a split will be determined [24].

Here we check the Random search, Grid search, TPE, and CMA-ES hyperparameter tuning method. The results are shown in Table 2. Here also random sampler perform better compared to others. Fig. 2 is a graphical representation of accuracy values for different samplers in different datasets.

TABLE 2. DECISION TREE-ACCURACY VALUE FOR DIFFERENT SAMPLERS

Datasets	Optimization Method			
	TPE Sampler	CMA-ES Sampler	Random Sampler	Grid Sampler
LED	70.4	69.4	70.2	70.6
Fetal-health	85.37	85.65	86.1	86.5
Mobile	84.85	85	85.4	84.35
E-coli	80.66	80.07	80.95	79.76

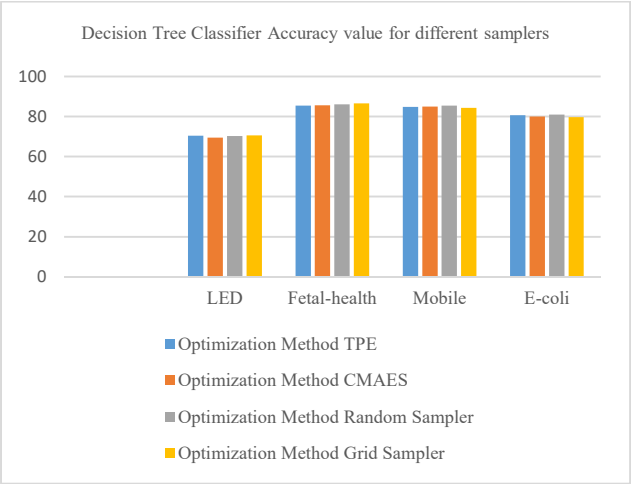

Fig 2. Decision Tree accuracy values for different datasets and different samplers.

The next Classifier is Random Forest Classifier [25]. The parameters n_estimators (number of trees in the forest), max_ features (maximum number of features considered for splitting a node), max_depth (maximum number of levels in each decision tree), min_samples_split (minimum number of data points positioned in a node before the node is split), min_ samples_leaf (minimum number of data points permitted in a leaf node), bootstrap (method for sampling data points) etc., are considered as hyperparameters of Random Forest Classifier. Here we check the Random search, Grid search, TPE, and CMA-ES hyperparameter tuning method. The results are shown in Table 3. Here for the LED dataset and E-coli dataset, the Random sampler shows a higher accuracy value but for the fetal-health dataset, CMA-ES shows a high accuracy value and for the mobile price dataset, TPE shows the highest accuracy. Fig 3. Shows graphical representation of Random Forest accuracy values for different datasets and different samplers.

TABLE 3. RANDOM FOREST-ACCURACY VALUE FOR DIFFERENT SAMPLERS

Datasets	Optimization Method			
	TPE Sampler	CMAES Sampler	Random Sampler	Grid Sampler
LED	73	73.2	73.8	72.2
Fetal-Health	87.4	87.63	87.54	86.83
Mobile	88.4	87.65	86.9	87.95
E-coli	86.89	86.9	86.9	86.9

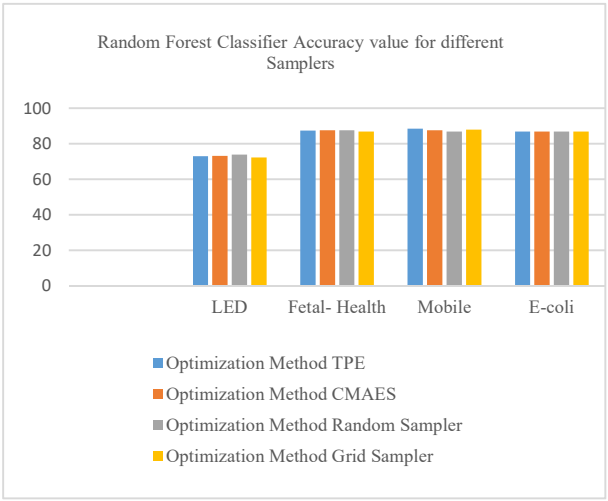

Fig 3. Random Forest accuracy values for different datasets and different samplers.

Random Forest Classifier shows better accuracy for all these datasets. For the majority of datasets, the random search sampler shows better performance.

V. CONCLUSION

Machine learning algorithms have tremendous use in different fields and for different applications. Multi-class classification is a major type of supervised machine learning technique. With the use of different hyperparameter tuning strategies like Random Search, Grid Search, TPE optimization, and CMA-ES, this paper examined the performance of different multiclass algorithms like SVM, decision tree, and random forest classifier on four multi-class datasets. This paper used an automated hyperparameter tuning tool called Optuna. We may conclude from the findings of this study that using hyperparameter tweaking approaches, it is possible to improve the performance of machine learning algorithms. In the future, we can use additional meta-heuristic methods to hyperparameter tuning strategies to test the performance of

these algorithms. When compared to various samplers employed on the same datasets, this research concludes that the Random Forest Classifier has the best accuracy, and the Random Search sampler shows the best performance for the supplied datasets.

REFERENCES

[1] L. Bottou *et al.*, "Comparison of classifier methods: a case study in handwritten digit recognition," in *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3 - Conference C: Signal Processing (Cat. No.94CH3440-5)*, Oct. 1994, vol. 2, pp. 77–82 vol.2. doi: 10.1109/ICPR.1994.576879.

[2] P. Probst, B. Bischl, and A.-L. Boulesteix, "Tunability: Importance of Hyperparameters of Machine Learning Algorithms," *arXiv: 1802.09596 [stat]*, Oct. 2018, Accessed: Jun. 12, 2021. [Online]. Available: http://arxiv.org/abs/1802.09596

[3] J Bergstra, Bardenet ,Bengio,"Implementations of algorithms for hyper-parameter optimization," in 24th International Conference on Neural Information Processing Systems, Granada, Spain, December 2011, pp.12–15.

[4] F. Hutter, H. Hoos, and K. Leyton-Brown, "An Efficient Approach for Assessing Hyperparameter Importance," in *International Conference on Machine Learning*, Jan. 2014, pp. 754–762. Accessed: Jun. 12, 2021.

[5] C. Fawcett and H. H. Hoos, "Analysing differences between algorithm configurations through ablation," *J Heuristics*, vol. 22, no. 4, pp. 431–458, Aug. 2016, doi: 10.1007/s10732-014-9275-9.

[6] P. Jain and A. Kapoor, "Active learning for large multi-class problems," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 762–769. doi: 10.1109/CVPR.2009.5206651.

[7] Y. Guermeur, "Combining Discriminant Models with New Multi-Class SVMs," *Pattern Anal Appl*, vol. 5, no. 2, pp. 168–179, Jun. 2002, doi: 10.1007/s100440200015.

[8] "Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond | MIT Press eBooks | IEEE Xplore." https://ieeexplore.ieee.org/book/6267332 (accessed Jun. 12, 2021).

[9] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 3, pp. 660–674, May 1991, doi: 10.1109/21.97458.

[10] A. Liaw and M. Wiener, "Classification and Regression by RandomForest," *Forest*, vol. 23, Nov. 2001.

[11] P. Kaur, R. Kumar, and M. Kumar, "A healthcare monitoring system using random forest and internet of things (IoT)," *Multimed Tools Appl*, vol. 78, no. 14, pp. 19905–19916, Jul. 2019, doi: 10.1007/s11042-019-7327-8.

[12] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[13] J. N. van Rijn and F. Hutter, "Hyperparameter Importance Across Datasets," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, New York, NY, USA, Jul. 2018, pp. 2367–2376. doi: 10.1145/3219819.3220058.

[14] "Tree-structured Parzen Estimator — Optunity 1.1.0 documentation." https://optunity.readthedocs.io/en/latest/user/solvers/TPE.html (accessed Jun. 04, 2022).

[15] I. Loshchilov and F. Hutter, "CMA-ES for Hyperparameter Optimization of Deep Neural Networks," *arXiv:1604.07269 [cs]*, Apr. 2016, Accessed: Jul. 10, 2021. [Online]. Available: http://arxiv.org/abs/1604.07269

[16] "Fetal Health classification." https://www.kaggle.com/andrewmvd/fetal-health-classification. (accessed May 06, 2022).

[17] "UCI Machine Learning Repository: LED Display Domain Data Set." https://archive.ics.uci.edu/ml/datasets/LED+Display+Domain (accessed Jun. 04, 2022).

[18] "Mobile Price Classification | Kaggle." https://www.kaggle.com/datasets/iabhishekofficial/mobile-price-classification (accessed Jun. 04, 2022).

[19] "Ecoli Data Set." https://www.kaggle.com/elikplim/ecoli-data-set (accessed Jun. 04, 2022).

[20] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," *arXiv: 1907.10902 [cs, stat]*, Jul. 2019, Accessed: Jul. 10, 2021. [Online]. Available: http://arxiv.org/abs/1907.10902.

[21] S Ali and K A Smith-Miles, "A meta-learning approach to automatic kernel selection for support vector machines," *Neurocomputing*, vol. 70, no. 1, pp. 173–186, Dec. 2006, doi: 10.1016/j.neucom.2006.03.004.

[22] L C Padierna, M. Carpio, A. Rojas, H. Puga, R. Baltazar & H. Fraire, "Hyper-Parameter Tuning for Support Vector Machines by Estimation of Distribution Algorithms," *Nature-Inspired Design of Hybrid Intelligent Systems*, Springer International Publishing, 2017, pp. 787–800. doi: 10.1007/978-3-319-47054-2_53.

[23] K R. Muller, S. Mika, G. Ratsch, K. Tsuda & B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Transaction on Neural Network.*, volume. 12, issue. 2, pp. 181–201, March. 2001, doi-10.1109/72.914517.

[24] R. Mantovani, T. Horvath, R. Cerri, J. Vanschoren & A. de Carvalho, "Hyper-Parameter Tuning of a Decision Tree Induction Algorithm", Oct. 2016.

[25] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Variable selection using random forests," *Pattern Recognit. Lett.*, vol. 31, no. 14, pp. 2225–2236, Oct. 2010, doi: 10.1016/j.patrec.2010.03.014.