

A SURVEY OF RANDOM FOREST PRUNING TECHNIQUES

Minu Rose and Dr.Hani Ragab Hassen

Department of Mathematical and Computer Sciences, Heriot Watt University,
Dubai, UAE

ABSTRACT

Random Forest is an ensemble machine learning method developed by Leo Breiman in 2001. Since then, it has been considered the state-of-the-art solution in machine learning applications. Compared to the other ensemble methods, random forests exhibit superior predictive performance. However, empirical and statistical studies prove that the random forest algorithm generates unnecessarily large number of base decision trees. This may cost high computational efficiency, predictive time, and occasional decrease in effectiveness. In this paper, Authors survey existing random forest pruning techniques and compare the performance between them. The research revolves around both the static and dynamic pruning technique and analyses the scope of improving the performance of random forest by techniques including generating diverse and accurate decision trees, selecting high performance subset of decision trees, genetic algorithms and other state of art methods, among others.

KEYWORDS

Ensemble Learning, Random Forest, Pruning Techniques.

1. INTRODUCTION

Over the past few decades, data mining as well as machine learning techniques have been a popular area of research. This is mostly due to the advances in computational efficiency and the overwhelming data created day by day. Data mining can be referred to as a process of extracting useful information from unstructured data which in turn helps the client to make business decisions. It incorporates techniques from different domains such as statistics, high-performance computing, pattern recognition, information retrieval, machine learning, and many more [1].

Due to this interdisciplinary nature of data mining and the significant overlap of techniques used between data mining and machine learning, both these terms are often confused with each other. Machine learning can be defined as algorithms that provide the computing device with the ability to automatically learn complex patterns and make intelligent insights [2].

Machine learning algorithms can be broadly classified into supervised and unsupervised learning algorithms [3]. This classification is based on whether output of the training data is known prior to training. In supervised learning, models build from labelled datasets are used to predict outcomes of the future data. On the other hand, unsupervised learning tries to extract hidden patterns from the unlabelled data [2]. Classification is one of the applications for supervised machine learning where the model learns from labelled training data and builds classifier which can be deployed after validating using testing data [4]. And, Clustering is an application of unsupervised machine learning. Clustering groups' similar data points into clusters of varying size [5].

There exist several classification techniques such as Decision trees, Naïve Bayes, Support Vector Machine [2]. Among these, Decision trees are the most popular and powerful approach due to its simplicity, comprehensibility and high predictive performance. Even though decision trees have many benefits over other classifiers, it suffers from various drawbacks such as instability and many more [17]. One way to utilize maximum potential of these decision tree algorithms is to construct a random forest [6].

Random Forest algorithm builds multiple decision tree classifiers and combines the prediction of these classifiers into a unified output [7]. Recently, Fernandez-Delgado et al. [2014] compared 179 classifiers from 19 families on 121 data sets and concluded that the random forest tends to outperform all other classifiers. It is worth noting that the high predictive performance of random forests cost high computational time. In real time application like stream mining it is important to reduce the time complexity without compromising the predictive performance [9].

Usually random forest algorithms build 100 to 500 base decision trees for the prediction [10]. But the predictive performance of random forests depends only on the strength of the base classifiers and diversity among the base classifiers [7]. Theoretical and empirical studies prove that adding on base classifiers beyond certain limit do not contribute to the accuracy of the ensemble but may lead to correlated classifiers that degrades the performance of random forests [11]. Many researchers have proposed several random forest enhancements that exploit this concept to boost the performance of the classifier. These enhancements that focus on finding the optimal random forest subset is known as pruning of random forest [12].

In our paper, we review techniques that can be used to prune random forest. Some of them are general ensemble pruning techniques whereas other are random forest specific.

2. BACKGROUND

This section provides an overview of the core concepts and machine learning techniques used in pruning of random forest and concludes by investigating and critically analysing the related research work done since 2001 in the field of ensemble pruning and random forest enhancements.

2.1. Decision Trees

Most common application of decision trees is classification. Decision trees are represented by leaves, internal nodes, and root nodes. The node without an incoming edge are root node and nodes with incoming and outgoing edges are internal nodes. The leaves are nodes which are characterized by only incoming edge. Each internal node represents a test condition that is to be checked on an instance. Thus, internal nodes are also called test nodes. C4.5 and CART are the commonly used algorithm for decision tree construction. The advantages of decision tree include self-explanatory behaviour, capability to handle diverse input data [6].

2.2. Random Forest

A random forest is a classification technique developed by Breiman as an extension to the bagging technique. It is one of the most reliable ensemble techniques with base classifiers as decision trees [6]. Since random forest constitutes of decision tree as base classifier, bagging is usually used to generate random forest [13]. Bagging builds decision tree with the help of bootstrap replicates from training data. Since the decision tree exhibit low bias and high variance, bagging is considered to be the best suitable method to create random forest. When an instance is to be classified, the prediction of the base classifier is combined using majority voting [14]. Random forest exhibits several advantages when compare to other ensemble techniques. While building random forest, there is no much role for the test datasets as random forest itself uses out

of bag instances to train the classifier. That is generalization capability is internally estimated using out of bag instances. Random forest also incorporates internal methods to handle missing data and unbalanced data. Finally, when compared to different ensemble methods, random forest exhibits less sensitivity towards noise data [15].

3. GENERAL PRUNING TECHNIQUES

3.1. Ensemble Pruning Techniques

The concept of ensemble method was originally proposed as supervised machine learning technique by Nilsson in 1965 [16]. It is treated as one of the state-of-the-art solutions for many machine learning challenges. Ensemble method works by combining the predictions made by a collection of individually trained base classifiers [17].

Generally, there are several variants of ensemble learning methods that include Bagging [18], Boosting [19], Random Subspace method [20], and many more [15]. Bagging builds ensemble classifiers using training sets formed by bootstrap replicates of the original training set [13]. The prediction of base classifiers is mostly made by simple majority voting paradigm [18]. Bagging is considered as the best method if the base classifiers have low bias and high variance or if the base classifier is unstable [21]. The unstable classifiers can be referred to as a base classifier whose output changes even with the slightest change in the training data. Decision trees and neural network are two of the examples that exhibit instability in classification behaviour [22]. Boosting builds a series of base classifiers successively and the weights for the training set are adjusted in accordance to the accuracy of the preceding base classifier. AdaBoost is one of the well-known algorithms for boosting [19]. Boosting is considered as the ideal method if the training set is considerably large and if the base classifiers have high bias and low variance. For better performance boosting requires the base classifiers to be stable [21].

Random Subspace method modifies the training dataset and this modified data is utilized for building base classifiers. Parallel learning algorithm is generally used for random subspace approach where the generation of base classifiers are independent of each other. Thus, this approach helps in parallel processing which is suitable for real time applications. The output prediction of each classifier is usually integrated by majority voting paradigm. Random subspace is considered the best method when the training data available is critical as well as small and when the base classifiers are weak linear classifiers [20].

The comparison study by Skurichina et al. [2002] on the above-mentioned ensemble variants utilizing several real and artificial datasets conclude that, boosting is preferred method when considerably large training data is available, while random subspace and bagging is considered beneficial when the available training data is critical and small.

Empirical and theoretical studies show that the overall performance and generalization ability of the ensemble is often highly accurate than any of the individual base classifiers in the ensemble [24, 25]. Still, it is worth noting that the high predictive performance accuracy come along with some drawbacks like high computation requirement and high time complexity [9].

For applications like stream mining and remote sensing, it becomes crucial to reduce the run time of the ensemble methods. Oza et al. [2008] has reflected on the need of developing ensemble methods that uses the richness of the big data without the cost of high computational power. Zhou et al. [2003] has also suggested that it is always better to classify with a set of carefully selected classifiers instead of all the classifiers in the base pool ensemble. This can be achieved by ensemble pruning.

The purpose of the ensemble pruning technique is to yield a subset of diverse ensemble that retains or outperforms the accuracy of original ensemble method [10]. The performance of the pruned ensemble subset depends on the trade-off between the individual predictive performance and diversity among the base classifier [17]. Over the last few decades several enhancements were made to optimize the performance measure of ensemble classifiers [26, 27, 28].

In the recent survey, Kulkarni et al. [2012] classified the ensemble pruning into static and dynamic approach. Usually, pruning methods employs overproduce and choose strategy. The overproduce phase constructs a large pool of base classifiers and choose phase selects the base classifiers from the constructed classifier pool with the help of specific evaluation metric. The static approach follows overproduce and choose strategy and keeps ensemble subset fixed for predicting all the classification patterns. The dynamic approach eliminates 4 the overproduce phase and dynamically generates optimized ensemble based on specific criterion for classification [12].

Most of the pruning effort in the literature are targeted towards static approach [10, 14, 28, 29]. While recently some effort towards dynamic pruning can also be noticed [30, 31, 32, 33].

In static approach, Tsoumakas et al. [2009] categorized different classifier selection procedure for choose phase such as ranking-based, clustering-based, and optimization-based.

3.1.1. Ranking-Based Method

In the ranking based technique, the individual classifiers in the ensemble are ranked based on specific evaluation measures like orientation pruning, or kappa pruning [26]. The highest ranked classifiers are then selected for further classification. [28, 29] employs this technique for their pruning process.

Guo et al. [2018] uses margin and diversity measure (MDM) as evaluation metric to rank the classifiers. And also, Guo et al. [2018] focuses on ranking the individual classifiers based on low margin, high diversity and high accuracy. It is important to note that the issue of choosing the final number of classifiers can be solved by selecting a fixed user specific number of classifiers or by dynamically selecting classifiers according to their predictive performance [26]. However, the presence of redundant classifier in the pruned ensemble may reduce the efficiency of this ranking based technique

3.1.2. Clustering-Based Method

Clustering based ensemble pruning techniques, which is the most relevant topic to our research comprise of two stages. In the initial stage a clustering technique is employed so as to discover groups of similar predictive model. And in the final stage, sub-ensembles are selected from each cluster and grouped together to form the pruned ensemble that can be employed for classification [26]. This increases the diversity and reduces the correlated classifiers in the resultant subset of the ensemble. Diverse clustering algorithms such as k-means [34], deterministic annealing [35] have been proposed for the initial stage.

Different from the above-mentioned clustering approach Zhang et al. [2014] proposed a static approach that utilized the spectral clustering-based scheme to prune the ensemble. They have used bagging technique to create ensemble classifiers with high accuracy and diversity. Based on the accuracy of individual classifiers and the diversity of the ensemble, spectral clustering is utilized to prune the ensemble. Q statistics is used to calculate the pair-wise diversity within the ensemble. The results show that this approach maintains standard accuracy level but may incur

huge time complexity up to $O(n^2 + knt)$ for computing the accuracy vectors, accuracy similarity and diversity similarity.

3.1.3. Optimization-Based Method

The different types of optimization-based method involve semi-definite programming [9], hill climbing [36, 37, 38] and genetic algorithm [32, 33].

Semi-definite programming (SDP): Zhang et al. [2006] defines ensemble pruning as a mathematical problem that can be solved by applying semi-definite programming techniques. Although the algorithm proved to get better results than kappa pruning and diversity based pruning technique, it had to run in polynomial times which increased the time complexity.

Hill climbing: Hill climbing algorithm applies greedy search along the neighbourhood of the current state to discover the next state to be visited. Each state constitutes to a subset and the neighbourhood of a state contains subsets that can be created by addition or removal of model from the subset. Depending on the greedy search pattern from one end to the other end, search patterns can be classified as forward selection or backward elimination. Forward selection involves starting with empty end and adding models based on their ability to improve the ensemble performance. Backward elimination involves starting with a complete set and deleting the models that do not contribute to the predictive performance of the ensemble [26]. [37, 38] uses Forward selection and [36] uses Backward selection procedure for ensemble pruning. Diversity and performance are the main evaluation technique utilized to measure the success of pruned ensemble in hill climbing method. In spite of the performance attained by this method, the traversal pattern can lead to high time complexity in real time applications [26].

Genetic Algorithm: Genetic algorithms are evolutionary algorithm that go through different stages. In the first stage, an initial set of individual classifier population are generated. From the initial population, classifiers that exhibits good performance are selected using a fitness function. Then the selected ensemble subset goes through crossover and different mutations to produce classifiers. This process continues till the termination condition is satisfied [32]. [32] and [33] employs genetic algorithm for ensemble pruning.

Due to non-linear optimization process, most of the above-mentioned optimization-based techniques has high time complexity [28].

In dynamic approach, dynamic classifier selection and dynamic ensemble selection are most commonly known techniques. Dynamic classifier selection process dynamically selects a single base classifier with higher predictive performance for each of the test pattern [39]. Woods et al. [1997] proposed a dynamic classifier selection technique that utilized local accuracy for selecting the classifier. The approach estimates the predictive accuracy of base classifiers in the local region defined by K nearest neighbour and selects the most accurate classifier. This strategy is suitable when there is a significant difference between performance of base classifiers and least recommended otherwise.

Inspired from dynamic classifier selection strategy, Xiao et al. [2009] proposed a novel approach GAES (Adaptive Classifier Ensemble Selection) by combining group method for data handling theory with dynamic classifier selection. Then, Xiao et al. [2009] extended GAES and proposed GDES (Dynamic Ensemble Classifier) algorithm. The time complexity of this approach is usually higher as GAES algorithm consumes time depending on the number of cyclic operations required to find the optima. Experiments with these dynamic ensemble selection approach proved to be slightly better than K nearest oracle (KNORA) [41].

To improve the performance of GDES, Xiao et al. [2010] extended the GDES approach GDES-AD. The approach considers both diversity and accuracy for the ensemble selection process. The accuracy of ensemble was measured based on symmetric regularity criterion. The analysis of the experimental results show that the noise immunity and predictive performance of the proposed method GDES-AD is higher when compared to above strategy. The application area of dynamic pruning includes medical field [30] and many more but this research focuses on static methods for ensemble pruning.

3.2. Random Forest-Specific Pruning Techniques

Compared to many other ensemble learning techniques, random forests have proved to have higher accuracy and predictive performance [8]. It has outperformed most of the existing techniques including SVM as well as AdaBoost [42].

Even though random forests exhibit high performance, they are not utilized to their full potential [15]. Thus, many researchers believe that there is possibility to enhance random forest for improved performance. Performance enhancement of random forest can be achieved by pruning and non-pruning techniques. More specifically, reported literature research focuses on two main aspects for boosting the performance of random forests. First aspect is to construct base classifiers either by altering the count of features used at every node or by using different evaluation criteria to determine best split at every node. Second aspect is to identify the best combination of base classifiers from the base pool of classifiers [15].

Robnik-Sikonja [2004] have investigated both the above-mentioned aspects to improve the overall performance in random forest. Robnik-Sikonja [2004] focused on better performance of random forests by strengthening the individual classifiers without compromising the diversity among them. This was achieved by using various attribute evaluation and estimation methods like ReliefF algorithm instead of Gini index [43].

Robnik-Sikonja [2004] achieved another promising improvement that stemmed from using weighted voting instead of majority voting technique for prediction. Weighted voting approach uses internal estimates to find the similar instances to the instance that is to be predicted and votes are weighted according to strength demonstrated by the trees on the near instances. The experiments proved the weighted voting mechanism to provide better results than ReliefF algorithm.

By replacing simple majority voting approach with more advanced dynamic integration, Tsymbal et al. [2006] points out an enhanced method to improve the performance of random forest. In accordance with the local performance estimators, this method utilizes three different dynamic integration techniques namely dynamic selection, dynamic voting and dynamic voting along with selection. Initially, dynamic selection technique selects the classifier with least error and dynamic voting weights the classifier proportional to its performance. Finally, dynamic voting with selection discards the base classifiers with error higher than specified limit and the rest of the base classifiers are forwarded to dynamic voting phase. Dynamic integration strategy showed improvement in 12 out of 27 datasets.

The latest research of dynamic pruning has led to FIRE-DES++ method. This strategy removes both noise and overlap between classes. Thus, defining an area of competence with unbiased validation data instead of biased data. The results show significant improvement when compared to the existing dynamic pruning methods [46]

Bernard et al. [2009] proposed a different approach to add the trees to random forest based on feature selection strategy. The aim was to point out that random forest subsets performs better than original random forest. This approach utilizes optimization based sequential forward or

backward selection (SFS or SBS) approach to build the random forest subset. Experiments on resultant random forest subset demonstrates better performance but suffers from few drawbacks. Some of the drawbacks include redundant classifiers, less diverse decision trees, reduced generalization capability and high time complexity because of its iterative selection procedure.

Inspired by the random forest optimization technique proposed by Bernard et al. [2009], Tripoliti et al. [2013] investigated on selecting strong individual classifiers along with maintaining diversity in the random forest subset. Tripoliti et al. [2013] uses updated version of sequential forward and backward selection (Modified SFS RF and Modified SBS RF) where the difference is in the criteria that should be fulfilled while adding or discarding a tree in the random forest. Experiments show that the modified algorithm exhibits greater generalized performance than original SFS and SBS [15].

Zhang et al. [2009] proposed different method to determine a sub-forest from the parent random forest without compromising the accuracy. They focused on three measures to determine the importance of a tree in the sub-forest and to find out the optimal sub-forest. The first method is to remove trees that has minimal overall impact on the accuracy of the random forest. The other two methods are based on similarity measure between the trees. Most of the redundant trees are removed from the parent forest and pruning results show improved accuracy [45].

Research done by Bader-El-Den et al. [2012] and Zhou et al. [2003] explains the genetic algorithm based pruning method. The approach developed by Bader-El-Den et al. [2012] used indirect genetic algorithm and was known as Genetic algorithm-based RFs (GARF). The extensive experiments prove that GARF outperforms several classification algorithms including SVM (Support Vector Machine) and AdaBoost [32].

Genetic algorithm-based technique investigated by Zhou et al. [2003] uses the modified GASEN (Genetic algorithm based selective ensemble) algorithm with bit representation known as GASEN-b. This GASEN-b method trains several base classifiers and then uses genetic algorithm to selects a set of optimal classifiers that will constitute to an ensemble. The results show a comparable performance against AdaBoost and Arc-x4 algorithms [33].

Fawagreh [1997,2015] proposed extensions to random forest algorithm that utilised techniques like local outlier factor and clustering to prune random forest. The research based on LOF (Local Outlier Factor) focuses at identifying the diversity of the individual classifiers and ranks the individual classifiers accordingly. Experimental results of LOFB-DRF shows comparable performance accuracy with original random forest, and high pruning level [29].

The clustering approach proposed by Fawagreh et al. [2015] utilizes K-Modes algorithm to group similar decision trees in the random forest. The diversity between two decision trees is calculated in accordance with the mismatch between their output vectors. Higher mismatch between the base classifiers results in more diverse classifiers. Out-of-bag (OOB) techniques is used to select the representatives from each cluster. The experiments show that this technique outperformed the original random forest and maintained healthy pruning level of at least 94% [10]. However, one challenge faced by this technique is that the K-Modes clustering algorithm demands to specify the number of clusters in advance. Finally, Kulkarni et al. [2012] points out some pruning techniques that focuses on pruning of random forest which is good to have a look at.

Table 1. SUMMARY OF RELATED WORK.

Pruning Technique	Base Classifiers	Ensemble Construction Method	Performance	Taxonomy
CLUB-DRF [10]	Decision Tree	Bootstrap	Outperforms the parent RF with accuracy of 92% and above.	Clustering
LOFB-DRB [29]	Decision Tree	Bootstrap	Outperforms the RF with 99% pruning level.	Ranking
GASEN [33]	C4.5 Decision Tree	Genetic algorithm	Strong generalization ability with fewer base classifiers & Performance better than algorithms such as AdaBoost, Arc-x4.	Optimization Genetic algorithm
GARF [32]	C4.5 Decision Tree	Indirect genetic algorithm	Better performance than SVM, AdaBoost algorithms	Optimization Genetic algorithm
SDP [9]	C4.5 Decision Tree	AdaBoost	Better performance than kappa pruning and diversity based pruning but algorithm runs in polynomial times.	Optimization SDP
Spectral clustering [14]	Standard Cart Tree	Bootstrap	Better performance with high time complexity.	Clustering

4. CONCLUSIONS

Ensemble machine learning imitate human behaviour and final outcome is determined by combining the decision of several models. Thus, ensemble learning algorithms are proved to have superior performance than most of the other machine learning techniques. Random Forest algorithms are considered as one of the states of art method in ensemble machine learning. The intention of this survey was to point out different pruning techniques and compare the performance between them. The detailed description of each method in this survey would help the researchers to get better intuition about the future research directions.

ACKNOWLEDGEMENTS

I would like to sincerely thank my family and supervisor Dr. Hani Ragab for his valuable time, and guidance he gave for the preparing this survey paper.

REFERENCES

- [1] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, p. 37, 1996.
- [2] S. R. Guruvayur and R. Suchithra, "A detailed study on machine learning techniques for data mining," in *2017 International Conference on Trends in Electronics and Informatics (ICEI)*. IEEE, 2017, pp. 1187–1192.
- [3] S. Raschka, *Python machine learning*. Packt Publishing Ltd, 2015.
- [4] K. Fawagreh, "On pruning and feature engineering in random forests." 2016.
- [5] G. Gan, C. Ma, and J. Wu, *Data clustering: theory, algorithms, and applications*. Siam, 2007, vol. 20.
- [6] L. Rokach, "Decision forest: Twenty years of research," *Information Fusion*, vol. 27, pp. 111–125, 2016.
- [7] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [8] M. Fernandez-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3133–3181, 2014.
- [9] Y. Zhang, S. Burer, and W. N. Street, "Ensemble pruning via semi-definite programming," *Journal of Machine Learning Research*, vol. 7, no. Jul, pp. 1315–1338, 2006.
- [10] K. Fawagreh, M. M. Gaber, and E. Elyan, "Clubdrf: A clustering approach to extreme pruning of random forests," in *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer, 2015, pp. 59–73.
- [11] S. Bernard, L. Heutte, and S. Adam, "On the selection of decision trees in random forests," in *2009 International Joint Conference on Neural Networks*. IEEE, 2009, pp. 302–307.
- [12] V. Y. Kulkarni and P. K. Sinha, "Pruning of random forest classifiers: A survey and future directions," in *2012 International Conference on Data Science & Engineering (ICDSE)*. IEEE, 2012, pp. 64–68.
- [13] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine learning*, vol. 40, no. 2, pp. 139–157, 2000.
- [14] H. Zhang and L. Cao, "A spectral clustering based ensemble pruning approach," *Neurocomputing*, vol. 139, pp. 289–297, 2014.
- [15] E. E. Tripoliti, D. I. Fotiadis, and G. Manis, "Modifications of the construction and voting mechanisms of the random forests algorithm," *Data & Knowledge Engineering*, vol. 87, pp. 41–65, 2013.
- [16] N. J. Nilsson, "Learning machines." 1965.
- [17] T. G. Dietterich et al., "Ensemble learning," *The handbook of brain theory and neural networks*, vol. 2, pp. 110–125, 2002.
- [18] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

- [19] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [20] I. Barandiaran, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, 1998.
- [21] N. C. Oza and K. Tumer, "Classifier ensembles: Select real-world applications," *Information Fusion*, vol. 9, no. 1, pp. 4–20, 2008.
- [22] T. G. Dietterich, "Machine-learning research," *AI magazine*, vol. 18, no. 4, pp. 97–97, 1997.
- [23] M. Skurichina and R. P. Duin, "Bagging, boosting and the random subspace method for linear classifiers," *Pattern Analysis & Applications*, vol. 5, no. 2, pp. 121–135, 2002.
- [24] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of artificial intelligence research*, vol. 11, pp. 169–198, 1999.
- [25] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: many could be better than all," *Artificial intelligence*, vol. 137, no. 1-2, pp. 239–263, 2002.
- [26] G. Tsoumakas, I. Partalas, and I. Vlahavas, "An ensemble pruning primer," in *Applications of supervised and unsupervised ensemble methods*. Springer, 2009, pp. 1–13.
- [27] M. A. O. Ahmed, L. Didaci, B. Lavi, and G. Fumera, "Using diversity for classifier ensemble pruning: an empirical investigation," *Theoretical and Applied Informatics*, vol. 29, 2017.
- [28] H. Guo, H. Liu, R. Li, C. Wu, Y. Guo, and M. Xu, "Margin & diversity based ordering ensemble pruning," *Neurocomputing*, vol. 275, pp. 237–246, 2018.
- [29] K. Fawagreh, M. M. Gaber, and E. Elyan, "An outlierdetection-based tree selection approach to extreme pruning of random forests," *arXiv preprint arXiv:1503.05187*, 2015.
- [30] E. E. Tripoliti, D. I. Fotiadis, and G. Manis, "Dynamic construction of random forests: Evaluation using biomedical engineering problems," in *Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine*. IEEE, 2010, pp. 1–4.
- [31] J. Xiao, C. He, X. Jiang, and D. Liu, "A dynamic classifier ensemble selection approach for noise data," *Information Sciences*, vol. 180, no. 18, pp. 3402–3421, 2010.
- [32] M. Bader-El-Den and M. Gaber, "Garf: towards selfoptimised random forests," in *International Conference on Neural Information Processing*. Springer, 2012, pp. 506–515.
- [33] Z.-H. Zhou and W. Tang, "Selective ensemble of decision trees," in *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*. Springer, 2003, pp. 476–483.
- [34] A. Lazarevic and Z. Obradovic, "Effective pruning of neural network classifier ensembles," in *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, vol. 2. IEEE, 2001, pp. 796–801.
- [35] B. Bakker and T. Heskes, "Clustering ensembles of neural network models," *Neural networks*, vol. 16, no. 2, pp. 261–269, 2003.
- [36] I. Partalas, G. Tsoumakas, and I. P. Vlahavas, "Focused ensemble selection: A diversity-based method for greedy ensemble selection." in *ECAI*, 2008, pp. 117–121.
- [37] W. Fan, F. Chu, H. Wang, and P. S. Yu, "Pruning and dynamic scheduling of cost-sensitive ensembles," in *AAAI/IAAI*, 2002, pp. 146–151.

- [38] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in Proceedings of the twenty-first international conference on Machine learning. ACM, 2004, p. 18.
- [39] K. Woods, W. P. Kegelmeyer, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," IEEE transactions on pattern analysis and machine intelligence, vol. 19, no. 4, pp. 405–410, 1997.
- [40] J. Xiao and C. He, "Dynamic classifier ensemble selection based on gmdh," in 2009 International Joint Conference on Computational Sciences and Optimization, vol. 1. IEEE, 2009, pp. 731–734.
- [41] A. H. Ko, R. Sabourin, and A. S. Britto Jr, "From dynamic classifier selection to dynamic ensemble selection," Pattern recognition, vol. 41, no. 5, pp. 1718–1731, 2008.
- [42] M. Robnik-Sikonja, "Improving random forests," in European conference on machine learning. Springer, 2004, pp. 359–370.
- [43] M. Robnik-Sikonja and I. Kononenko, "Theoretical and empirical analysis of relieff and rrelieff," Machine learning, vol. 53, no. 1-2, pp. 23–69, 2003.
- [44] A. Tsymbal, M. Pechenizkiy, and P. Cunningham, "Dynamic integration with random forests," in European conference on machine learning. Springer, 2006, pp. 801–808.
- [45] H. Zhang and M. Wang, "Search for the smallest random forest," Statistics and its Interface, vol. 2, no. 3, p. 381, 2009.
- [46] Cruz, Rafael MO and Oliveira, Dayvid VR and Cavalcanti, George DC and Sabourin, Robert, "FIRE-DES++: Enhanced online pruning of base classifiers for dynamic ensemble selection," Pattern Recognition, vol. 85, pp. 149--160, 2019.