Markowitz random forest: weighting random forest trees with modern portfolio theory

Eleftherios Kouloumpris[a,*], Ioannis Vlahavas[a]

[a]*School of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece*

Abstract

Tree-based ensembles such as random forest (RF) are essential methods for supervised learning. Whereas the traditional RF gives equal weights to its trees, there is significant evidence that different tree weighting schemes can improve the traditional method. Previously proposed tree weighting methods rely on the predictive performance of each individual tree, with high performing trees receiving higher weights. However, the predictive power of RF stems not only from accurate individual trees, but also tree variety, which has not been considered in previous weighting schemes. In this paper, we propose Markowitz random forest (MRF), a weighting scheme that takes into account both tree performance and tree variety, using a tree covariance matrix for the latter. Our method is formulated as a constrained optimization problem, and is inspired by financial mathematics for creating high performing and well-diversified portfolios, particularly modern portfolio theory. Our experiments on four benchmark datasets show that MRF can significantly outperform RF in terms of F_1 score.

Keywords: Random Forest, Tree Weighting, Modern Portfolio Theory

1. Introduction

The recent advances in machine learning (ML) constitute a major accomplishment of artificial intelligence (AI) research. The further development of supervised learning, a well-studied ML paradigm, has important implications

*Corresponding author

Email addresses: elefthenk@csd.auth.gr (Eleftherios Kouloumpris), vlahavas@csd.auth.gr (Ioannis Vlahavas)

for both academia and industry. Tree-based ensembles (TBE) are currently among the most popular supervised learning methods.

Whereas the deep learning approach provides state-of-the-art results for multimedia data (e.g., image, video, sound), TBEs are still the best performing methods for tabular datasets. For instance, most Kaggle competitions have been won by a combination of feature engineering and a TBE solution. Two well known TBEs are random forest (RF) and gradient boosting (GB). RF combines multiple low-bias/high-variance trees in parallel in order to decrease the total variance, while GB sequentially combines high-bias/low-variance trees to decrease the total bias. Further on, we will consider how these trees should be weighted for optimal results. Given that GB inherently decides the appropriate tree weights, we will only consider RF for the remainder of this work.

The traditional RF algorithm gives equal weights to every tree, a strategy that we will denote as equally weighted trees (EWT). Previous works claim that EWT is an arbitrary strategy that does not consider factors such as tree performance. The same works have proposed different weight allocation strategies that aim to improve upon EWT. Whereas the proposed weight allocation strategies take into account tree performance, to the best of our knowledge, information relating to the covariance between trees has not been considered before. Therefore, weighting methods that rely only on tree performance run the risk of giving significant weights to a small set of high performing, but also significantly correlated trees. This is important because the predictive power of RF relies not only on the accuracy of independent trees, but also on having a diverse set of uncorrelated trees (tree diversity).

In this paper, taking into account the influence of weights on tree diversity, we propose a novel method for estimating the weights of RF trees. Our method is inspired by financial mathematics, in the same vein that neural networks and genetic algorithms are inspired by neuroscience and biology. We also note that while many ML methods are commonly used to solve problems in finance (i.e., ML-based finance), in the opposite direction, we propose the first finance-inspired ML method. More generally, our intention is to demonstrate that financial mathematics can lead to innovative AI research.

Specifically, we apply a method from modern portfolio theory called mean-variance analysis (MVA), first introduced by H. Markowitz in 1952 (Markowitz, 1952). MVA was originally designed for the construction of portfolios that have high expected returns, but must also include a diverse set of assets. Hence, we name our method Markowitz random forest (MRF).

As it will be explained in detail, MRF is the first tree weighting method that simultaneously takes into account both tree performance and tree covariance. Our experimental work on four datasets show how MRF can easily provide a predictive performance increase against the traditional RF.

The remainder of the paper has the following structure. Section 2 presents and discusses previously proposed tree weighting methods for RF. Section 3 provides the necessary background for RF and MVA. Section 4 presents the proposed method and the experimental setup. Section 5 reveals the experimental results and provides additional discussion. Finally, Section 6 concludes the paper.

2. Related Work

There are several works that aim to improve the weight allocation scheme of RF. Li et al. (2010) proposed tree weighted random forest (TWRF) in order to weight trees according to their out-of-bag (OOB) performance in terms of classification accuracy. TWRF reduced the effect of noisy trees and was able to outperform RF and other traditional supervised learning methods. Winham et al. (2013) introduced another tree weighted RF in which weights reflected tree accuracy. Besides improving the predictive performance of the ensemble, the latter work also emphasized the computation of feature importance. El Habib Daho et al. (2014) introduced a variant of RF that, among other modifications, also used tree weighting. Similarly to the previously mentioned works, tree weights reflected OOB performance. The final ensemble was evaluated on several medical datasets.

Pham and Olafsson (2019) considered a modified voting scheme that used Cesáro sequence averaging instead of the traditional average voting. Essentially, this voting scheme relies on a sorted sequence of the trees, with trees at the beginning of the sequence receiving larger weights. They used two criteria to sort the trees: (a) OOB error rates and (b) accuracy on another training set. They provided both theoretical and empirical proof that Cesáro RF can outperform RF under certain conditions. Devi et al. (2019) proposed a tree weighting method in the context of imbalanced financial data. Specifically, they assigned the tree weights according to tree performance on the minority class. The proposed system outperforms traditional RF in an unbalanced fraud detection task.

Whereas the aforementioned works propose static weights that are estimated during training time, Jain et al. (2019) proposed the exponentially weighted random forest (EWRF) scheme for dynamic weight allocation at prediction time. For this purpose, they defined an observation-tree similarity function based on the exponential function. When predicting a new example, the trees that are most similar to this test example receive the largest weights. The merit of dynamic weight allocation at prediction time was proven by multiple experiments. Gajowniczek et al. (2020) experimented with weighting schemes for both observations and trees, with the latter relying on a combination of in-bag (INB) and OOB errors. Their solution was able to outperform traditional ensemble algorithms and decrease false alarms on data from Physionet/Computing in Cardiology Challenge, 2015.

Shahhosseini and Hu (2020) proposed tree weighting methods based on constrained optimization of accuracy and area under curve (AUC), including also several stacking-based solutions. The stacked solution trained a second RF based on the first RF's OOB predictions or probabilities. More recently, Zhang et al. (2023) worked with probability intervals to introduce the cautious weighted random forest (CWRF). Based on the theory of belief functions, they designed a convex optimization problem that takes into account both determinacy (i.e., the preciseness of a probability interval) and accuracy. Experiments on multiple benchmark datasets showed the appropriateness of the method for problems in which cautiousness is important or when data are of low quantity or quality.

In summary, a study of the literature reveals significant evidence that weighted tree RF methods can outperform the traditional algorithm. Yet, the majority of the reviewed methods rely on independent tree performance only (e.g. measured by OOB tree accuracy), without taking into account tree variety. The most recent works tend to formulate tree weight allocation as a constrained optimization problem (Shahhosseini and Hu, 2020; Zhang et al., 2023), which we also do in this work. However, our formulation is different as it has a double objective; instead of focusing only on independent tree performance, we also consider maximizing tree variety. The latter objective is achieved using a tree covariance matrix in our optimization formula to decrease the total variance of the ensemble.

3. Background

3.1. Random forest

RF is an integration of two ideas, namely the random subspaces method (RSM) and bootstrap aggregating (bagging). Ho (1998) proposed RSM, a method that trains independent estimators on different feature subspaces (i.e., columns of a design matrix) that are sampled with replacement. Breiman (1996) proposed bagging, which trains estimators on multiple datasets by sampling examples (i.e., rows of a design matrix) with replacement. Breiman (2001) also coined the term RF for a method that combines RSM/bagging and is specifically applied to decision trees.

RF starts by creating a number of samples via bagging, and then fits a tree on each sample. The algorithm used to fit each tree differs from the traditional decision tree algorithm. Specifically, for each internal node, only a subspace of features are considered for the best split. This randomization contributes to the decrease of the total variance of the ensemble, which results in an increased performance.

3.2. Mean-variance analysis

In portfolio management, an investor has to decide how to allocate a certain capital C to a number k of financial assets (e.g., stocks). Assuming that for each asset there is an expected future return $r_i, i \in [1, k]$, the allocation of capital is denoted by the portfolio weights $w_i, i \in [1, k]$ such that $w_i > 0$ and $\sum_{i=1}^{k} w_i = 1$. Using linear algebra notation, both the expected returns and weights can be concisely written as vectors $r \in \mathbb{R}^k$ and $w \in \mathbb{R}^k$.

According to modern portfolio theory, an investor should aim for a portfolio with high expected return but low overall risk. The risk (volatility) is typically expressed as the total variance (standard deviation) of the portfolio's total return. The expected return M and the total variance V of a portfolio are given in Equations 1-2.

$$M = \sum_{i=1}^{k} w_i r_i = w^T r \tag{1}$$

$$V = \sum_{i=1}^{k} \sum_{j=1}^{k} w_i w_j \sigma_{ij} = w^T \Sigma w \tag{2}$$

In Equation 2, σ_{ij} is the expected covariance between the returns of the i-th and j-th assets, and $\Sigma \in \mathbb{R}^{k \times k}$ denotes the expected covariance matrix for the returns of the k assets. MVA computes the efficient frontier, a set of portfolios that optimize Equation 3 for different parameters of λ, which controls the return/risk trade-off. As a side note, if only V is minimized we derive the minimum volatility portfolio that also belongs to the efficient frontier.

$$\begin{aligned} \max_{w} \quad & w^T r - \lambda w^T \Sigma w \\ \text{s.t.} \quad & \sum_{i=0}^{k} w_i = 1 \\ & 0 \leq w_i \leq 1 \end{aligned} \tag{3}$$

The above is a convex optimization problem that can be solved with Quadratic Programming (QP). The efficient frontier contains a portfolio named max Sharpe portfolio, which has the optimal Sharpe ratio (Sharpe, 1998). The optimization criterion that leads to this portfolio is given in Equation 4.

$$\begin{aligned} \max_{w} \quad & \frac{w^T r}{w^T \Sigma w} \\ \text{s.t.} \quad & \sum_{i=0}^{k} w_i = 1 \\ & 0 \leq w_i \leq 1 \end{aligned} \tag{4}$$

This is a Fractional Programming (FP) problem that cannot be solved directly with QP. However, it is possible to solve it with QP after a certain transformation that uses variable substitution. For a more detailed explanation of optimization methods in finance, consider the work of Cornuéjols and Tutuncu (2007).

4. Methods and Data

4.1. Data

We performed our experiments on four datasets from the University of California Irvine (UCI) ML Repository (Dua and Graff, 2017): (a) Cylinder Bands Data Set (bands) (Evans and Fisher, 1994), (b) Online Shoppers Intentions (shoppers) (Sakar et al., 2019), (c) Default of Credit Card Clients (credit) (Yeh and hui Lien, 2009), and (d) Bank Marketing (bank-marketing) (Moro et al., 2014). Our pre-processing pipeline performs a removal of duplicate records, one-hot encoding of categorical variables and the filling of missing values with the mean value of each column. Statistics for the processed datasets appear in the Table 1.

Table 1: Dataset statistics

dataset	rows	columns
bands	541	96
shoppers	12330	28
credit	30000	24
bank-marketing	41176	63

4.2. Technologies and Programming Packages

All experiments have been conducted with the Python programming language, using the RF implementation of Scikit-learn. For the MVA algorithm we used the implementation of the package PyPortfolioOpt. In the context of Sharpe ratio optimization, the package is able to transform the related FP problem to a QP problem. Furthermore, PyPortfolioOpt relies on the cvxopt and cvxpy packages for the solution of convex optimization problems.

4.3. Markowitz random forest

Our main goal is to apply an optimization framework, originally intended to solve portfolio management problems, as a solution for the RF tree weighting problem. Specifically, we consider the MVA framework, which has been explained in Section 3.2. As mentioned in Section 1, we name this tree-weighted RF method as MRF. In order to apply this financial mathematics solution to our supervised learning problem, which is a case of reducing one problem to another, we consider the mapping of concepts presented in Table 2.

Table 2: Mapping of concepts

Portfolio Management	**Tree Weighting**
financial assets	decision trees
assets weights	tree weights
portfolio	ensemble (forest)
portfolio total return	ensemble accuracy
portfolio total risk	ensemble variance
diversification	uncorrelated trees

We elaborate on why MVA is an attractive framework for tree weighting. Traditionally, investors are interested in portfolios with high return potentials. However, given two portfolios with equal return potentials, the investor will choose the most diversified portfolio (i.e., the one that has less correlated assets). The other, less diversified portfolio comes with additional and unnecessary risk (volatility) for the same expected return. MVA is designed to optimize towards portfolios with high returns, that are also well-diversified. Similarly, RF works best when it consists of accurate trees, which are also less correlated to each other. Consequently, our application of MVA to tree weighting can simultaneously consider both (a) tree accuracy and (b) promote the diversification of trees by weighting uncorrelated trees.

As mentioned in Section 3.2, MVA requires an expected return and covariance matrix. We will explain how these are designed for the tree weighting problem. Initially, an RF model with k trees is fitted for a binary classification task on a training dataset X with m examples. Then, each tree RF_j predicts probabilities for each example X_i. The positive class probability prediction of the j-th tree on the i-th example is denoted as P_{ij}. Then, for each combination of example and tree we use the actual labels $y_i \in \{0, 1\}$ to compute the soft accuracy metric given in Equation 5.

$$softAccuracy(y_i, P_{ij}) = 1 - |y_i - P_{ij}| \in [0, 1] \tag{5}$$

However, because MVA works with returns that are unbounded, we perform the following log-odds transformation to derive log soft accuracy (LSA) as in Equation 6.:

$$LSA(y_i, P_{ij}) = log(\frac{softAccuracy(y_i, P_{ij})}{1 - softAccuracy(y_i, P_{ij})}) \in \mathbb{R} \tag{6}$$

The LSA for each example/tree combination are collected in a performance matrix Q, as depicted in Equation 7:

$$Q = \begin{bmatrix} LSA(1,1) & \dots & LSA(1,k) \\ \vdots & LSA(i,j) & \vdots \\ LSA(m,1) & \dots & LSA(m,k) \end{bmatrix} \in \mathbb{R}^{m \times k} \tag{7}$$

Finally, the M and V parameters for the MVA analysis can be extracted directly from this performance matrix Q. Specifically, M is the average row of Q, while V is the covariance matrix of Q, as presented in Equations 8-9.

$$M = E_i[Q_{i,:}] \in \mathbb{R}^k \tag{8}$$

$$V = Cov(Q) \in \mathbb{R}^{k \times k} \tag{9}$$

Finally, we apply MVA for maximum Sharpe ratio to obtain the tree weights w that solve the optimization problem in Equation 10:

$$\begin{aligned} \max_{w} \quad & \frac{w^T M}{w^T V w} \\ \text{s.t.} \quad & \sum_{i=0}^{k} w_i = 1 \\ & 0 \leq w_i \leq 1 \end{aligned} \tag{10}$$

4.4. Hyper-parameter tuning

We use grid search to tune both RF and MRF. For both methods, we tune the number of trees (k) and the maximum depth of each tree (d). Additionally, for MRF we also consider the minimum weight at which MRF can set the weight of a tree. This weight is expressed and tuned as a percentage $p \in [0, 100]$ of the tree weight of an equally weighted RF, which is $\frac{1}{k}$. Therefore, the minimum weight for a tree in MRF is $\frac{p}{100}\frac{1}{k}$. The search space for each hyper-parameter is provided in Table 3. Finally, we use the class balancing option of Scikit-learn as some of the datasets are imbalanced.

Our hyper-parameter search aims for highly accurate models that are not over-fitting the data. Therefore, as optimization criterion we use the metric presented in Equation 11 to score a hyper-parameter configuration ($params$) based on the corresponding validation and train accuracy ($train\_acc$,val_acc).

$$score(params) = \frac{val\_acc}{|train\_acc - val\_acc|} \tag{11}$$

Table 3: Hyper-parameter search space

method	hyper-parameter	search space
RF, MRF	number of trees (k)	[100, 150, 200, 250]
RF, MRF	max tree depth (d)	[3, 5, 7, 9, 12, 15]
MRF	min weight percentage (p)	[90%, 80%, 70%, 60%, 50%, 40%, 30%, 20%, 10%]

5. Experimental Results

We performed our experiments on the four datasets mentioned in Section 4.1. For every dataset, we split the data into train-test-validation at the respective ratios of 64%, 20% and 16%. To begin with, we train all model configurations mentioned in Section 4.4 and evaluate them on the validation set. We provide the best configurations for each dataset in Table 4, according to the evaluation score also described in Section 4.4.

Table 4: Validation set performance and optimal hyper-parameters for RF and MRF on four datasets

dataset	model	trees	max.depth	min.weight	train accuracy	validation accuracy
Bands	RF	200	3		0.8174	0.7701
	MRF	200	3	20%	0.8058	0.7816
Shoppers	RF	100	3		0.8699	0.8702
	MRF	200	5	20%	0.8756	0.8753
Credit	RF	100	5		0.7776	0.7777
	MRF	100	5	90%	0.7814	0.7815
Bank-Marketing	RF	100	5		0.8417	0.8402
	MRF	100	3	90%	0.8573	0.8564

From the train-validation results, we make three important observations. First, the best number of trees and maximum tree depth have small differences between RF and MRF. Second, the validation accuracy of MRF is better in all cases. Third, the difference between train and validation accuracy is closer for MRF. For instance, for the bands dataset, MRF outperforms RF in validation accuracy by 1.15%, while the gap between train and validation accuracy is decreased by 2.42%.

For MRF, we also observe that the minimum tree weight parameter seems related to the size of the dataset; it is lower for the smallest datasets and higher for the largest datasets. One interpretation is that the larger the size of the training set, the less modification is required to the original tree weights. On the contrary, trees built on small datasets need significant reweighting of trees. For instance, some trees made from smaller datasets are prone to having a smaller contribution due to noise and overfitting.

Figure 1 provides a graphical representation of weight allocation for the bands and credit datasets. For both RF and MRF, we plot the gross percentage difference for the weight of each tree compared to equally balanced weights. For k trees and a specific tree with weight w, the gross percentage difference is computed as $100\frac{w}{k}\%$. Obviously, the RF weights are all set to

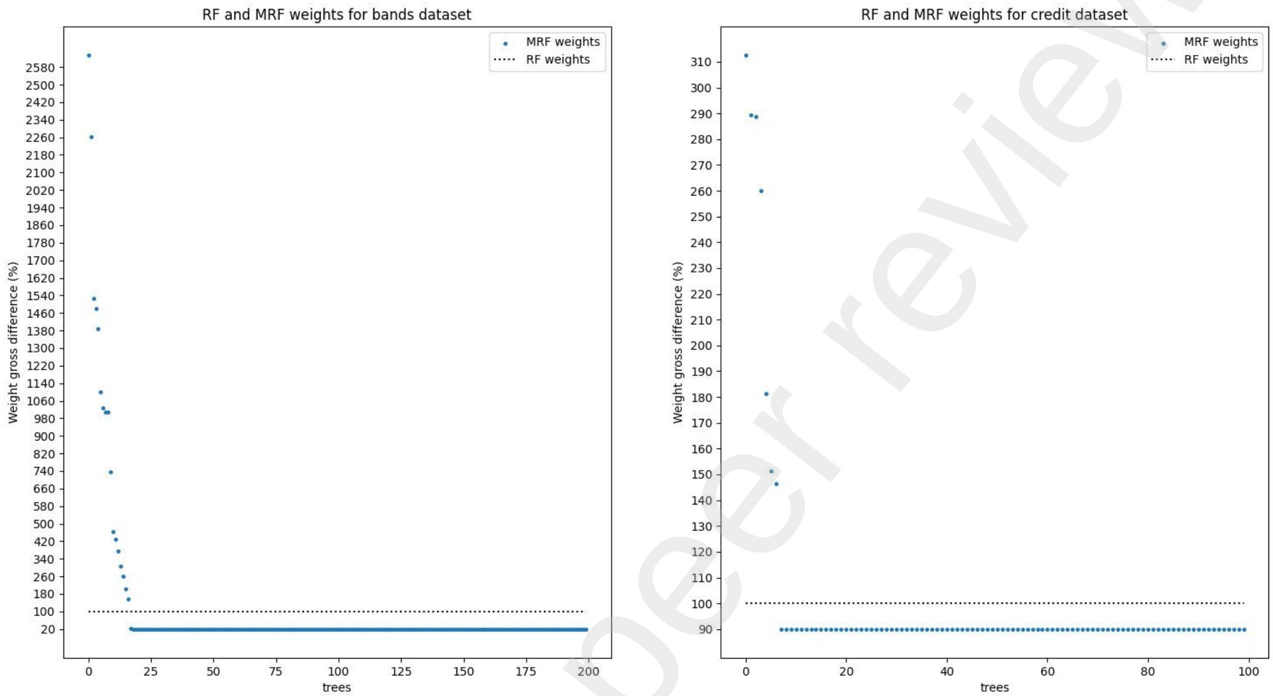

Figure 1: Distribution of weights for RF and MRF

100% as it uses equally weighted trees. In the next paragraph, we discuss the graphical results for MRF's weight allocation.

For the bands dataset, MRF increases the weight of 17 trees by 1.5 to 25.8 times. The weights for the remaining trees drop to 20% of the original weights. For the credit dataset, MRF promotes 7 trees by increasing their weights by 1.45 to 3.15 times, setting the remaining trees at about 90% of their original weights. Besides having high predictive performance, we note that the 17 (7) trees that are promoted in the bands (credit) problem also form a forest with high tree variety and low correlation between trees, as we considered the tree covariance matrix in our optimization task.

We applied the best performing RF and MRF models to the corresponding test sets, and the results are provided in Table 5. At this point, in addition to accuracy, we also consider F_1, precision and recall. The first important observation is that, in terms of accuracy and F_1 score, MRF is the best performing model across all datasets. Whereas the performance gains are marginal for accuracy, for F_1 score, the increase in performance ranges from 0.5% to 1.5%.

Table 5: Test set performance metrics for RF and MRF on four datasets

dataset	model	accuracy	F_1	precision	recall
Bands	RF	0.7890	0.7294	0.8611	0.6327
	MRF	**0.7982**	**0.7442**	**0.8649**	**0.6531**
Shoppers	RF	0.8892	0.6234	**0.7459**	0.5355
	MRF	**0.8905**	**0.6311**	0.7452	**0.5473**
Credit	RF	0.8240	0.4772	**0.6666**	0.3716
	MRF	**0.8248**	**0.4840**	0.6662	**0.3801**
Bank-Marketing	RF	0.9086	0.5170	**0.6807**	0.4168
	MRF	**0.9087**	**0.5228**	0.6765	**0.4261**

The F_1 score results can be further explained by observing precision and recall. Specifically, we observe that MRF has better recall in all test sets. Furthermore, while RF has better precision in three out of four datasets, we observe that these improvements are small. MRF offers an improvement in recall ranging from 0.8% to 2%, at a much smaller cost, a decrease in precision which on average is about -0.037%. Given that F_1 is the harmonic mean of precision and recall, it becomes clear why MRF has better F_1 in all datasets. To summarize, our experimental findings on unseen data are very similar across all datasets. In every case, MRF provides an improved recall score, which in turn increases F_1, while accuracy and precision are comparable.

Finally, we note that an improvement in recall has even greater importance when False Negatives (FNs) are more costly than False Positives (FPs). For instance, there are problems in finance for which FNs are more expensive, such as early warning systems for financial fragility. Specifically, an undetected financial crisis (FN) has more severe economic effects compared to the costs of misreporting an upcoming financial crisis (FP). In medical diagnosis as well, undetected disease (FN) can often be more costly than an incorrect positive diagnosis (FP). Thus, we consider that the performance improvements of MRF can be valuable in this class of problems.

6. Conclusion

The traditional RF algorithm assigns equal weights for each decision tree. In this paper, we propose an improved version of RF that relies on a different weighting method. While tree weighting schemes have been proposed before, to the best of our knowledge, these methods only consider the predictive performance of each independent tree.

In our approach, we use an additional criterion for tree weighting; to decrease the total variance of the ensemble by promoting diversified trees. Our method is based on financial mathematics which aim to simultaneously improve portfolio return and total portfolio variance. Thus, this is also a finance-inspired ML method.

Our experimental results on four public datasets provide evidence that our method can provide notable improvements against the traditional RF algorithm. Specifically, it provides significantly improved F_1 and recall scores, with similar results in accuracy and precision.

We plan several directions for future work. First, to re-formulate MRF for multi-class classification problems, as the current version is specifically designed for binary classification. Second, to also design and evaluate a variation of MRF for regression problems. Third, to modify the static MRF weighting mechanism towards a dynamic weighting scheme that depends on the observation. Finally, to continue our search for financial mathematics that can lead to finance-inspired ML methods.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declarations of Interest

None

References

Breiman, L., 1996. Bagging predictors. Machine learning 24, 123–140.

Breiman, L., 2001. Random forests. Machine learning 45, 5–32.

Cornuéjols, G., Tutuncu, R., 2007. Optimization methods in finance. doi:10.1017/CBO9780511753886.

Devi, D., Biswas, S.K., Purkayastha, B., 2019. A cost-sensitive weighted random forest technique for credit card fraud detection, in: 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–6. doi:10.1109/ICCCNT45670.2019.8944885.

Dua, D., Graff, C., 2017. UCI machine learning repository. URL: http://archive.ics.uci.edu/ml.

El Habib Daho, M., Settouti, N., El Amine Lazouni, M., El Amine Chikh, M., 2014. Weighted vote for trees aggregation in random forest, in: 2014 International Conference on Multimedia Computing and Systems (ICMCS), pp. 438–443. doi:10.1109/ICMCS.2014.6911187.

Evans, B., Fisher, D., 1994. Overcoming process delays with decision tree induction. IEEE Expert 9, 60–66. doi:10.1109/64.295130.

Gajowniczek, K., Grzegorczyk, I., Zabkowski, T.S., Bajaj, C.L., 2020. Weighted random forests to improve arrhythmia classification. Electronics 9.

Ho, T.K., 1998. The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20, 832–844. doi:10.1109/34.709601.

Jain, V., Sharma, J., Singhal, K., Phophalia, A., 2019. Exponentially weighted random forest, in: Pattern Recognition and Machine Intelligence, Springer International Publishing, Cham. pp. 170–178.

Li, H.B., Wang, W., Ding, H.W., Dong, J., 2010. Trees weighting random forest method for classifying high-dimensional noisy data, in: 2010 IEEE 7th International Conference on E-Business Engineering, pp. 160–163. doi:10.1109/ICEBE.2010.99.

Markowitz, H., 1952. Portfolio selection. The Journal of Finance 7, 77–91. URL: http://www.jstor.org/stable/2975974.

Moro, S., Cortez, P., Rita, P., 2014. A data-driven approach to predict the success of bank telemarketing. Decision Support Systems

62, 22–31. URL: https://www.sciencedirect.com/science/article/pii/S016792361400061X, doi:https://doi.org/10.1016/j.dss.2014.03.001.

Pham, H., Olafsson, S., 2019. On cesáro averages for weighted trees in the random forest. Journal of Classification 37. doi:10.1007/s00357-019-09322-8.

Sakar, C.O., Polat, S., Katircioglu, M., Kastro, Y., 2019. Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and lstm recurrent neural networks. Neural Computing and Applications 31. doi:10.1007/s00521-018-3523-0.

Shahhosseini, M., Hu, G., 2020. Improved weighted random forest for classification problems. ArXiv abs/2009.00534.

Sharpe, W.F., 1998. The sharpe ratio. Streetwise–the Best of the Journal of Portfolio Management , 169–185.

Winham, S.J., Freimuth, R.R., Biernacka, J.M., 2013. A weighted random forests approach to improve predictive performance. Statistical Analysis and Data Mining: The ASA Data Science Journal 6, 496–505. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/sam.11196, doi:https://doi.org/10.1002/sam.11196, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/sam.11196.

Yeh, I.C., hui Lien, C., 2009. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications 36, 2473–2480. URL: https://www.sciencedirect.com/science/article/pii/S0957417407006719, doi:https://doi.org/10.1016/j.eswa.2007.12.020.

Zhang, H., Quost, B., Masson, M.H., 2023. Cautious weighted random forests. Expert Systems with Applications 213, 118883. URL: https://www.sciencedirect.com/science/article/pii/S0957417422019017, doi:https://doi.org/10.1016/j.eswa.2022.118883.