

Online Explainable Ensemble of Tree Models Pruning for Time Series Forecasting^{*}

Amal Saadallah

Lamarr Institute for Machine Learning and AI, Dortmund, Germany

Abstract

Tree-based models are commonly used in time series forecasting due to their inherent interpretability, which makes them preferable to more complex black-box models. However, simple tree-based models are prone to overfitting, limiting their applicability in real-world scenarios. Ensembles of tree-based models are employed to mitigate this, but ensemble pruning is challenging, especially in the presence of dynamic time series data and concept drift. In this paper, we use TreeSHAP, a tree-specific explainability tool, to perform online tree-based ensemble pruning that adapts dynamically to changes in the time series, addressing the concept drift issue. Empirical evaluations on real-world time series datasets demonstrate that our method performs on par with or better than state-of-the-art techniques. In future research, we plan to automate the determination of the optimal number of clusters for ensemble pruning by leveraging ensemble properties like diversity, accuracy, and stability. This automation aims to enhance both the flexibility and explainability of the model selection process. Given that this work is in its early stages, we seek feedback and collaboration with experts to create a robust and explainable framework for ensemble-based time series forecasting.

Keywords

Tree Models, Online Ensemble Pruning, TreeSHAP, Time Series Forecasting, Concept-drift, Explainability

1. Introduction

Time series forecasting is crucial for real-time planning and decision-making across various fields like traffic management, weather prediction, and financial markets. However, it is also one of the most challenging tasks due to the complex and dynamic nature of time series data, which often involves non-stationary variations and is susceptible to concept drift [1]. This makes accurate forecasting inherently difficult, necessitating models that can adapt to changing data patterns [2, 3, 4, 5, 6, 7]. Given these challenges, explainability in forecasting models has become increasingly important, especially for safety-critical applications. Tree-based models are often favored for their intrinsic explainability, but identifying appropriate models for specific time series requires adaptability due to time-varying characteristics. Decision Trees and their ensembles, like Random Forests and Gradient-boosted Trees, are commonly used for time series forecasting. However, these models can struggle with dynamic data since they typically operate


Late-breaking work, Demos and Doctoral Consortium, colocated with The 2nd World Conference on eXplainable Artificial Intelligence: July 17–19, 2024, Valletta, Malta

^{*}This research has partly been funded by the Federal Ministry of Education and Research of Germany and the state of North-Rhine-Westphalia as part of the Lamarr-Institute for Machine Learning and Artificial Intelligence.

*Corresponding author.

✉ amal.saadallah@tu-dortmund.de (A. Saadallah)

ORCID [0000-0003-2976-7574](https://orcid.org/0000-0003-2976-7574) (A. Saadallah)

 © 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

in a static manner, not inherently considering variations in the underlying time series. In addition, combining multiple models into ensembles can improve forecasting accuracy, but at the cost of explainability. To address these issues, we propose an online ensemble pruning approach for time series forecasting, where the ensemble members are selected based on an adaptive clustering procedure that uses TreeSHAP values to group models with similar modeling paradigms. This methodology not only ensures diversity within the ensemble but also allows for an explainable selection process by indicating which aspects of the time series data contribute most to the predictions.

In our future research, one key goal is to automatically determine the optimal number of clusters, which corresponds to the ideal number of trees or ensemble members in the ensemble. This would involve using ensemble properties such as diversity, accuracy, and stability to guide the selection of the most suitable cluster count. By automating this process, we aim to improve the ensemble’s flexibility and effectiveness in adapting to dynamic time series data. Moreover, we intend to deepen the explainability aspect of our approach by explicitly demonstrating that selecting models based on different TreeSHAP values aligns with distinct modeling paradigms and hypotheses. This could be achieved by visualizing or analyzing how these varying TreeSHAP values translate into different interpretations of the underlying data, providing insights into the rationale behind model selection. Given that this is early-stage work, we plan to engage with experts in the field to exchange ideas and gather feedback. Collaboration with specialists will be instrumental in refining our methodology for selecting the optimal number of trees and enhancing explainability. By incorporating diverse perspectives, we hope to develop a robust and transparent approach that addresses the complexities of time series forecasting while maintaining clarity in model selection and ensemble pruning. This collaborative effort will contribute to building a reliable framework for ensemble-based forecasting, with a particular emphasis on explainability and adaptability.

2. Methodology

Our proposed method uses TreeSHAP for online ensemble pruning using model clustering. First, we define the used notation. Second, we describe Shapley values with a focus on TreeSHAP values [8]. Third, we show how we generate the candidate tree-based models. Finally, we demonstrate how TreeSHAP values are used for model clustering to allow for efficient ensemble pruning and how the whole process is made adaptive to the changes in the time series.

2.1. Preliminaries

A time series X is a temporal sequence of values, where $X_{1:t} = \{x_1, x_2, \dots, x_t\}$ is a sequence of X until time t and x_i is the value of X at time i . Denote with $\mathbb{T} = \{T^1, T^2, \dots, T^M\}$ the pool of M tree-based models trained to approximate a true unknown function f that generated X . Let $\hat{x}_{t+h} = (\hat{x}_{t+h}^{T^1}, \hat{x}_{t+h}^{T^2}, \dots, \hat{x}_{t+h}^{T^M})$ be the vector of forecast values of X at a future time instant $t+h$, $h \geq 1$ (i.e. x_{t+h}) by each of the models in \mathbb{T} . An ensemble model $\bar{T}_{\mathbb{T}}$ of \mathbb{T} at time instant $t+h$ can be formally expressed as a convex combination of the forecasts of the models in \mathbb{T} : $\bar{T}_{\mathbb{T}}(\hat{x}_{t+h}) = \sum_{j=1}^M w_{t+h}^j \hat{x}_{t+h}^{T^j}$ where $w_{t+h}^j \in [0, 1]$ are the ensemble weights. The weights

are constrained to be positive and sum to one. In addition, it can be seen from the notation that the weights are time-dependent. This is one of the requirements in online ensemble learning, where the weights are required to be set in a timely manner to cope with the dynamic nature of the time series and the time-changing performance of the ensemble members [5, 6]. The goal of dynamic online ensemble pruning is to identify the subset of models $\mathbb{S} \subset \mathbb{T}$ that should compose the ensemble at each time step $t + h$ such that the expected prediction error of the pruned ensemble is reduced compared the full ensemble $\bar{T}_{\mathbb{T}}$ for each forecast.

$$\operatorname{argmax}_{\mathbb{S} \subset \mathbb{T}} \mathbb{E}[(x_{t+h} - \bar{T}_{\mathbb{T}}(\hat{x}_{t+h}))^2 | X_{1:t+h-1}] - \mathbb{E}[(x_{t+h} - \bar{T}_{\mathbb{S}}(\hat{x}_{t+h}))^2 | X_{1:t+h-1}] \quad (1)$$

2.2. TreeSHAP Ensemble Learning

2.2.1. Ensemble Pruning

We divide the time series $X_{1:t}$ into $X_{\omega}^{train} = \{x_1, x_2, \dots, x_{t-\omega}\}$ and $X_{\omega}^{val} = \{x_{t-\omega+1}, x_{t-\omega+2}, \dots, x_t\}$, with ω a provided window size. X_{ω}^{train} is used for training the models in \mathbb{T} and X_{ω}^{val} is used to compute the TreeSHAP values. For each tree-based model $T^j \in \mathcal{T}$, for each observation $x_{t-\omega+k} \in X_{\omega}^{val}$ with $k \in [1, \omega]$, we compute a TreeSHAP value $\phi_i^j(x_{t-\omega+k})$ for each lagged value, i.e., $i \in [1, l^j]$, where l^j is the number of lags on which the model T^j is trained. Then, we aggregate absolute SHAP values over all the observations in X_{ω}^{val} to acquire SHAP-based lag importance I_i^j for each lag $i \in [1, l^j]$ using the model T^j :

$$I_i^j = \frac{1}{\omega} \sum_{k=1}^{\omega} |\phi_i^j(x_{t-\omega+k})|, \forall i \in [1, l^j], \forall T^j \in \mathcal{T} \quad (2)$$

Each model $T^j \in \mathcal{T}$ can then be characterized by a vector $\mathbf{I}^j = \{I_1^j, I_2^j, \dots, I_{l^j}^j\}$. The models can thus be clustered using their SHAP-based lag importance vectors \mathbf{I}^j . However, different models in \mathcal{T} might be trained using different lag values. As a result, the length of the vectors \mathbf{I}^j can vary between l_{min} and l_{max} . It exists clustering distance measure that can handle vectors of different lengths [9]. However, we are mainly interested in grouping models based on the way they represent the relationship between the input lagged values and the output. Therefore, we assume that the models that are trained using a lag value l^j lower than l_{max} ignore the importance and the contribution of lagged features that are greater than l^j . In other words, if the mode T^j is trained on $l^j \leq l_{max}$, for each i , such that $l^j \leq i \leq l_{max}$, the value of its corresponding SHAP-based lag importance I_i^j on i is set to zero. In this manner, we bring all the vectors \mathbf{I}^j for all the models $T^j \in \mathcal{T}$ to the same length l_{max} , and we use K-means with Euclidean distance for model clustering. Models belonging to different clusters are expected to have different modeling paradigms of the contributions of different lagged values to the predictions, which contributes to boosting the ensemble diversity. We select only cluster representatives to take part in the ensemble. We simply select the closest model to each cluster center.

2.2.2. Ensemble Adaptation

Streaming time series data is prone to significant changes, leading to concept drifts. To account for these shifts, the selection of ensemble members must be updated, allowing for the inclusion of

models that can better address newly emerging patterns. Concept drift is detected by monitoring deviations in the mean of the time series over time, using the Hoeffding Bound to evaluate if these deviations are significant. If a drift is detected, an alarm is triggered, the TreeSHAP-based model clustering is updated, and the ensemble is adjusted to reflect the new patterns in the data.

3. Experiments

Our method is denoted in the following as **OEP-TT**: Online explainable Ensemble Pruning of Tree models for Time series forecasting.

3.1. Experimental Setup

We use 100 univariate time series datasets from various application domains, including financial, weather, and synthetic data. These datasets are provided by the Monash Forecasting Repository [10]. We process each time series X by using the first 50% for training (X_{ω}^{train}), the following 25% for validation (X_{ω}^{val}) and the remaining 25% for testing. Due to this way of splitting the time series, we discard series that are shorter than 250 to allow enough training and validation data. All experiments have been performed on consumer hardware, namely on a 2022 MacBook Pro in R.

3.2. OEP-TT Setup

Tree-based models set-up: We construct a pool \mathcal{T} of tree-based models using different parameter settings that are summarized in Table 1. The list of parameters and their value ranges

Tree-based Model	Configurations	
Decision Tree (DT)	Maximum Depth	$d_{max} \in \{4, 8, 16\}$
Random Forest (RF)	Number of trees	$n_{trees} \in \{50, 100, 150, 200\}$
	Num. of variables sampled at each split	$mtry \in \{3, 5, 7\}$
	Minimum size of terminal nodes	$nodesize \in \{5, 10, 15\}$
Gradient Boosted DT (GBDT)	Number of trees	$n_{trees} \in \{50, 100, 150, 200\}$
	Maximum depth of each tree	$interaction.depth \in \{5, 7, 15\}$
	Shrinkage parameter	$shrinkage \in \{0.001, 0.01, 0.1\}$
eXtreme Gradient Boosting (Xgboost)	Max number of iterations	$nrounds \in \{50, 100, 150, 200\}$
Light GBM (LGBM)	Step size of each boosting step	$eta \in \{0.001, 0.01, 0.1\}$
	Maximum Depth	$max.depth \in \{5, 7, 15\}$
Light GBM (LGBM)	Metric	$metric \in \{L_1, L_2\}$ -Regularization
	Max number of iterations	$num_iterations \in \{50, 100\}$
	Maximum depth of each tree	$max_depth \in \{5, 7, 15\}$

Table 1

Hyper-parameters values of the tree-based models. Different configurations are generated by taking some combinations of these hyper-parameters as described in the last column.

in Table 1 is not exhaustive, and further parameters and values can be considered to generate more base learners. We also vary the lag parameter l on which the tree-based models are trained,

i.e., $l \in \{3, 5, 7, 10, 15, 20\}$. Considering different combinations of all the parameters, we train a total of 294 tree-based models.

OEP-TT set-up: **OEP-TT** has also a number hyper-parameters: M is the Size of the Pool of the tree-based models \mathbb{T} : 294, ω is the Size of the validation time window :25% of the data length, $|S|$ is the Number of final selected models: 6.

3.3. State-of-the-Art Methods Setup

We compare **OEP-TT** against State-of-the-Art (SoA) methods for online ensemble pruning, tree-based ensembles, and time series forecasting in general. These models include: Auto-Regressive Integrated Moving Average (**ARIMA**) [11], Exponential Smoothing (**ETS**) [11], Long Short-Term Memory (**LSTM**) [12], Multi-Layer Perceptron (**MLP**) [12], Convolutional Neural Network with LSTM (**CNN-LSTM**, **Bi-LSTM**) [13], Random Forest (**RF**) [14], Gradient-Boosted Decisions Trees (**GBDT**) [15], eXtrem Gradient Boosting (**XGBoost**) [16], and Light Gradient-Boosting Machine (**LGBM**) [17].

To enable a fair comparison with **OEP-TT**, we feed to these ensemble pruning methods the same pool of tree-based models \mathbb{T} that was used for **OEP-TT**: **Ens**: Ensemble of all the base modes in \mathbb{T} ; **OCL** [5]: Online drift-aware clustering of the tree-based models in \mathbb{T} using covariance-based clustering; **OTOP** [5]: Online drift-aware Top best-performing tree-based models ranking using temporal correlation analysis; **DEMISC** [5]: Dynamic Ensemble Members Selection using Clustering; Online drift-aware Top best-performing models ranking using temporal correlation analysis combined with covariance-based clustering; **ADE** [18, 19] was recently developed for an online dynamic ensemble of forecasters construction. A meta-learning strategy that specializes the tree-based models across the input time series. A sequential weighting schema is developed to automatically select ensemble members by setting their weights to zero.

We also compare **OEP-TT** to its variants: **OEP-TT-ST**: Static variant of **OEP-TT**. Pruning is decided at the initial forecasting instant and kept fixed along testing; **OEP-TT-Per**: Pruning is updated periodically in a blind manner (i.e. without considering the occurrence of the drift).

3.4. Results

3.4.1. Predictive Performance

Table 2 presents the average ranks and their deviation for **OEP-TT** and its variants and SoA methods for time series forecasting and online ensemble pruning. For the paired comparison, we compare our method **OEP-TT** against each of the other methods. We counted wins and losses for each dataset using the RMSE scores. We use the non-parametric Wilcoxon Signed Rank test to compute significant wins and losses, which are presented in parenthesis (significance level 0.05). In the results in Table 2, **OEP-TT** outperforms almost all the baseline methods in terms of ranks and wins/loses in pairwise comparison.

In this part, we show how initially **OEP-TT** supports explainability for the reason behind specific tree-based model selection to construct the ensemble at a specific time instant or interval, for model performance, and for the importance of input lagged time series observations.

Method	Avg. Rank	Std Deviation	Wins	Losses
ETS	16.60	4.53	89 (74)	11 (9)
ARIMA	15.39	5.90	88 (79)	12 (10)
MLP	13.48	5.04	80 (71)	20 (20)
LSTM	11.03	3.35	78 (70)	22 (21)
CNN-LSTM	7.29	2.68	70 (68)	30 (18)
Bi-LSTM	10.32	4.35	75 (74)	25 (20)
RF	10.76	5.04	75 (72)	25 (20)
GBDT	11.84	3.35	78 (78)	22 (18)
LGBM	12.61	2.68	80 (78)	20 (18)
Xgboost	15.54	2.68	87 (84)	13 (12)
ADE	12.68	3.61	79 (75)	21 (21)
OTOP	14.15	3.78	84 (81)	16 (15)
Ens	7.03	3.90	70 (601)	30 (28)
DEMISC	3.13	3.11	58 (57)	42 (30)
OCL	3.02	3.67	63 (61)	37 (27)
OEP-TT-St	2.74	2.85	54 (52)	46 (41)
OEP-TT-Per	1.95	2.78	- (-)	- (-)
OEP-TT	2.18	2.19	- (-)	- (-)

Table 2

Comparison (in terms of average rank achieved over 100 datasets) between our method and the baselines. The rank column presents the average rank and its standard deviation across different time series. An average rank of 1 means the model was the best performing on all time series.

3.4.2. Explainability Aspects

Figure 1 shows the TreeSHAP values clusters of the Saugeen River Flow data set. The dots on each lag value stand for the TreeSHAP values taken by the models belonging to the same cluster, while the line connects the mean values to show the TreeSHAP values for each lag of the representative selected model on each cluster (only for visualization purposes). Note that we show the name of the model and the value of the first hyper-parameter plus the lag value on which it is trained to distinguish between selected models belonging to the same family of tree-based models, e.g., RF200(Lag10) and RF50(Lag7). It can be seen that on different clusters different patterns of lagged values contributions to the target time series observations are observed. This confirms that our clustering procedure promotes ensemble diversity by enforcing the selection of models that have different modeling paradigms and distinct views on the importance of specific lag values. For example, while models in cluster 6 favor higher lag values and emphasize the contribution of their corresponding value to the output forecast value, models in cluster 5 are built on the assumption of restricting the memory of the models to lower lag values ($l = 3$). We can notice that in 3 clusters out of 6, models rely on restricted lagged values (clusters 2, 3, and 5). Even with this limited width of memory, i.e., historical data, they can excel in terms of predictive performance.

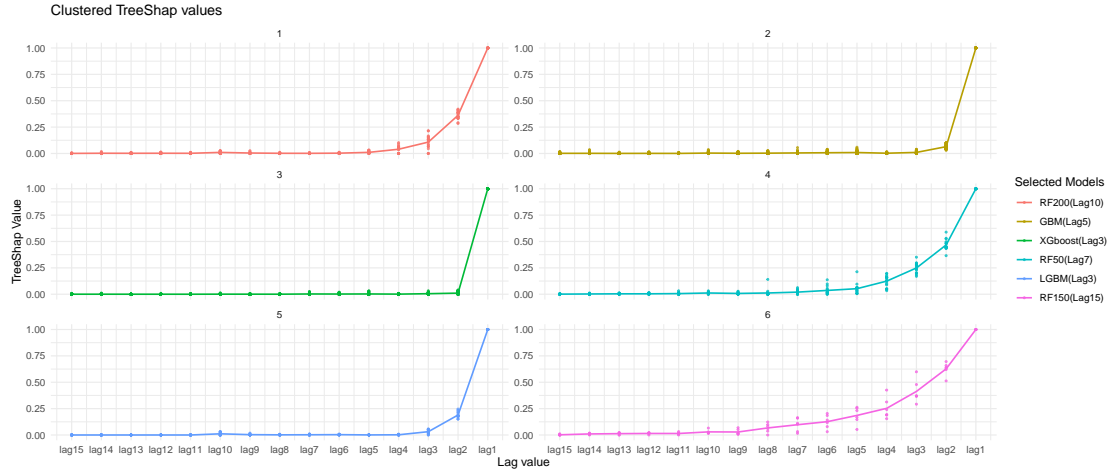


Figure 1: Comparison of TreeSHAP-based models clusters on the Saugeen River Flow data set.

4. Concluding Remarks and Future Work

This paper introduces **OEP-TT** a novel method for online adaptive ensemble of tree-based models pruning. Through the use of TreeSHAP values, we are able to gain insight into its decision-making process, both for model selection, as well as for the input time series points relevance. We showed the advantages of **OEP-TT** on 100 real-world datasets, both in terms of predictive performance as well as its explainability aspects. In future work, we plan to extend our method to hybrid model pools by using the most efficient Shapley value estimation methods for each model family, such as TreeSHAP for tree-based models, DeepSHAP [8] for Neural Networks, as well as KernelSHAP [8] for remaining models, tune the size of the ensemble and dive further into the explainability aspects. Given that this is early-stage work, we plan to engage with experts in the field to exchange ideas and gather feedback.

References

- [1] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, *ACM computing surveys (CSUR)* 46 (2014) 1–37.
- [2] A. Saadallah, M. Jakobs, K. Morik, Explainable online deep neural network selection using adaptive saliency maps for time series forecasting, in: N. Oliver, F. Pérez-Cruz, S. Kramer, J. Read, J. A. Lozano (Eds.), *Machine Learning and Knowledge Discovery in Databases. Research Track*, Springer International Publishing, Cham, 2021, pp. 404–420.
- [3] A. Saadallah, M. Jakobs, K. Morik, Explainable online ensemble of deep neural network pruning for time series forecasting, *Machine Learning* 111 (2022).
- [4] A. Saadallah, H. Mykula, K. Morik, Online adaptive multivariate time series forecasting, in: *Joint European conference on machine learning and knowledge discovery in databases*, Springer, 2022.
- [5] A. Saadallah, F. Priebe, K. Morik, A drift-based dynamic ensemble members selection

- using clustering for time series forecasting, in: Joint European conference on machine learning and knowledge discovery in databases, Springer, 2019.
- [6] A. Saadallah, M. Tavakol, K. Morik, An actor-critic ensemble aggregation model for time-series forecasting, in: IEEE ICDE, 2021.
 - [7] M. Jakobs, A. Saadallah, Explainable adaptive tree-based model selection for time series forecasting, arXiv preprint arXiv:2401.01124 (2024).
 - [8] S. M. Lundberg, S.-I. Lee, A Unified Approach to Interpreting Model Predictions, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 4765–4774. URL: <http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf>.
 - [9] D. J. Berndt, J. Clifford, Using dynamic time warping to find patterns in time series., in: KDD workshop, volume 10, 1994, pp. 359–370.
 - [10] R. Godahewa, C. Bergmeir, G. I. Webb, R. J. Hyndman, P. Montero-Manso, Monash time series forecasting archive, in: Neural Information Processing Systems Track on Datasets and Benchmarks, 2021. Forthcoming.
 - [11] G. E. Box, G. M. Jenkins, G. C. Reinsel, G. M. Ljung, Time series analysis: forecasting and control, John Wiley & Sons, 2015.
 - [12] F. A. Gers, D. Eck, J. Schmidhuber, Applying lstm to time series predictable through time-window approaches, in: Neural Nets WIRN Vietri-01, Springer, 2002, pp. 193–200.
 - [13] P. Romeu, F. Zamora-Martínez, P. Botella-Rocamora, J. Pardo, Time-series forecasting of indoor temperature using pre-trained deep neural networks, in: International conference on artificial neural networks, Springer, 2013, pp. 451–458.
 - [14] L. Breiman, Random forests, Machine learning 45 (2001) 5–32.
 - [15] S. B. Taieb, R. J. Hyndman, A gradient boosting approach to the kaggle load forecasting competition, International journal of forecasting 30 (2014) 382–394.
 - [16] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, et al., Xgboost: extreme gradient boosting, R package version 0.4-2 1 (2015) 1–4.
 - [17] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, Lightgbm: A highly efficient gradient boosting decision tree, Advances in neural information processing systems 30 (2017).
 - [18] V. Cerqueira, L. Torgo, F. Pinto, C. Soares, Arbitrated ensemble for time series forecasting, in: Joint European conference on machine learning and knowledge discovery in databases, Springer, 2017, pp. 478–494.
 - [19] V. Cerqueira, L. Torgo, F. Pinto, C. Soares, Arbitrage of forecasting experts, Machine Learning (2018).