OBLIQUE RANDOM FOREST ALGORITHM USING LASSO REGRESSION FOR WIND POWER FORECASTING

by

Burak Tabak

B.S., Industrial Engineering, Boğaziçi University, 2018

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering

Boğaziçi University

2022

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my thesis supervisor, Assist. Prof. Mustafa Gökçe Baydoğan for his guidance, support and encouragement since day one.

Also, I would like to thank my coworkers Uğur, Harun and the others in Algopoly team for their helpfulness and good fellowship.

I am thankful to my dear family for their moral support from beginning to end. Special thanks to my father for motivating me by asking "How many pages are left?" each day. Last but not least, I have to thank my fiancee Şeyma for always being there for me.

ABSTRACT

OBLIQUE RANDOM FOREST ALGORITHM USING LASSO REGRESSION FOR WIND POWER FORECASTING

With the increasing trend towards the use of renewable energy sources, wind power has been the subject of many researches. Wind power has stochastic nature due to uncertainties in atmospheric conditions, especially in wind speed, which makes it hard to forecast accurately. To solve the problem, statistical methods using Numerical Weather Prediction (NWP) models as inputs are proposed in the literature.

Random Forest is a statistical model frequently used in wind power forecasting with proven success. Random Forest ensembles decision trees that partition the feature space over a single variable at each node. However, partitions based on a single variable may fail to provide a proper distinction. Thus, oblique decision tree algorithms evaluating the partitions over linear combinations of variables are proposed in the literature, especially on classification problems. There are a limited number of studies in the literature on oblique decision tree-based methods applied in time series regression problems.

This thesis proposes a novel strategy to be applied in regional wind power forecasting tasks that ensembles oblique decision trees. The proposed method is compared with its univariate counterparts in three wind power forecasting tasks. Computational results show that the proposed method performs better on all tasks.

ÖZET

RÜZGAR ENERJİSİ TAHMİNİ İÇİN LASSO REGRESYONU KULLANAN EĞİK RASSAL ORMAN ALGORİTMASI

Yenilenebilir enerji kaynaklarına yönelimin artmasıyla birlikte rüzgar enerjisi birçok araştırmaya konu olmuştur. Rüzgar enerjisi, atmosferik koşullardaki, özellikle rüzgar hızındaki belirsizlikler nedeniyle, doğru bir şekilde tahmin etmeyi zorlaştıran stokastik bir yapıya sahiptir. Bu problemi çözmek için literatürde Sayısal Hava Tahmini modellerini girdi olarak kullanan istatistiksel yöntemler önerilmiştir.

Rassal Orman, rüzgar enerjisi tahmininde sıklıkla kullanılan başarısı kanıtlanmış istatistiksel bir modeldir. Rassal Orman, veriyi her düğümde tek bir değişken üzerinden bölen karar ağaçlarını bir araya getirmektedir. Ancak tek bir değişken üzerinden yapılan bölünmeler doğru bir ayrım sağlamayabilir. Bu nedenle literatürde, özellikle sınıflandırma problemlerinde uygulanan ve veriyi doğrusal değişken kombinasyonları üzerinden bölen eğik karar ağacı algoritmaları önerilmektedir. Literatürde zaman serisi regresyon problemlerinde uygulanan eğik karar ağacı tabanlı yöntemler ile ilgili sınırlı sayıda çalışma bulunmaktadır.

Bu tez, eğik karar ağaçlarını bir araya getiren bölgesel rüzgar enerjisi tahmin probleminde uygulanacak yeni bir strateji önermektedir. Önerilen yöntem, üç rüzgar enerjisi tahmin probleminde tek bir değişken üzerinden bölmeler yapan muadilleri ile karşılaştırılmıştır. Hesaplanan sonuçlara göre önerilen yöntem üç veri setinin hepsinde daha iyi performans göstermektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

c	Degree of polynomial transformation function
d	Maximum depth parameter
D	Input data
g_1, g_2	Aggregation functions
I	Identity matrix
m	Number of random features selected at each node
n	Number of observations
p	Number of features
P	Polynomial transformation function
r	Repetition times for random selection of features
s	Number of sampled data point
T_i	Terminal nodes of the tree i
w	Weight vector
X	Feature matrix
Y	Target vector
β	Regression coefficients vector
ϵ	Random error component vector
λ	Penalization factor
ν	Number of Tree
ψ	Function returning the observations of terminal nodes

LIST OF ACRONYMS/ABBREVIATIONS

2D	2-dimensional
ANN	Artificial Neural Network
ARIMA	Auto-Regressive Integrated Moving Average
CNN	Convolutional Neural Network
CART	Classification and Regression Trees
CART-LC	Classification and Regression Trees with Linear Combinations
CO2	Continuous Optimization of Oblique Splits
CV	Cross Validation
DT-LASSO	Multivariate Decision Tree using LASSO Regression
DT-LM	Decision Tree using Linear Models
FACT	Fast Algorithm for Classification Trees
GFS	Global Forecasting System
GLM	Generalized Linear Model
HHCART	Householder CART
KWH	Kilowatt-Hour
LMDT	Linear Machine Decision Trees
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MRMR	Max Relevance-Min Redundancy
MSE	Mean Square Error
MWH	Megawatt-Hour
NWP	Numerical Weather Prediction
OC1	Oblique Classifier 1
ODT	Oblique Decision Tree
OLS	Ordinary Least Squares
ORF	Oblique Random Forest
QRF	Quantile Random Forest
SADT	Simulated-Annealing Decision Trees

SMT	Sparse Multivariate Tree
RF	Random Forest
RF-LASSO	Random Forest using LASSO Regression
ROFLMAO	Robust Oblique Forests with Linear Matrix Operations
SPORF	Sparse Projection Oblique Random Forest
SSE	Sum of Square Error
WMAPE	Weighted Mean Absolute Percentage Error
WODT	Weighted Oblique Decision Trees

1. INTRODUCTION

In recent years, the negative effects of traditional fuels on the environment have become an important agenda with the fast-growing energy need. The trend towards renewable energy sources has also increased from year to year to decelerate the problems posed by traditional fuels. Although it is a clean source of energy, the main challenge with renewables is their stochastic nature due to many uncontrollable factors, especially the uncertainties in the atmospheric conditions. The stochasticity creates a need for forecasting, especially for production planning, energy supply and trade operations. Wind energy also has an important share in renewables. Therefore, wind power forecasting is a subject that a lot of research is carried out today [1].

In wind power forecasting, the main aim is to predict the amount of electricity produced in a given time interval under certain weather conditions [2]. The key driver of the wind power is wind speed because the power of wind is formulated as

$$P = \frac{1}{2} \times \xi \times \rho \times \pi \times r^2 \times V^3 \tag{1.1}$$

theoretically where P is the wind power in Watts, ξ is the efficiency factor in percentage, ρ is the air density in kg/m^3, r is the radius of the wind turbine blade in m and V is the wind velocity in m/s [3]. However; in practice, the power output can be estimated using a power curve that is specific for each turbine [4]. Figure 1.1 illustrates the power curve of the wind turbine branded "Nordex-N90" that has blades with 45-meter radius and rated power of 2300 kWh [5].

It can be observed in Figure 1.1 that the power production starts at cut-in speed and reach its maximum at the rated speed [4]. Until cut-out speed, the power production is limited to rated power and does not increase with wind speed [4]. After cut-out speed, the production is interrupted due to safety issues [6]. Moreover, it is worth mentioning that the power output of a wind turbine is approximately proportional to

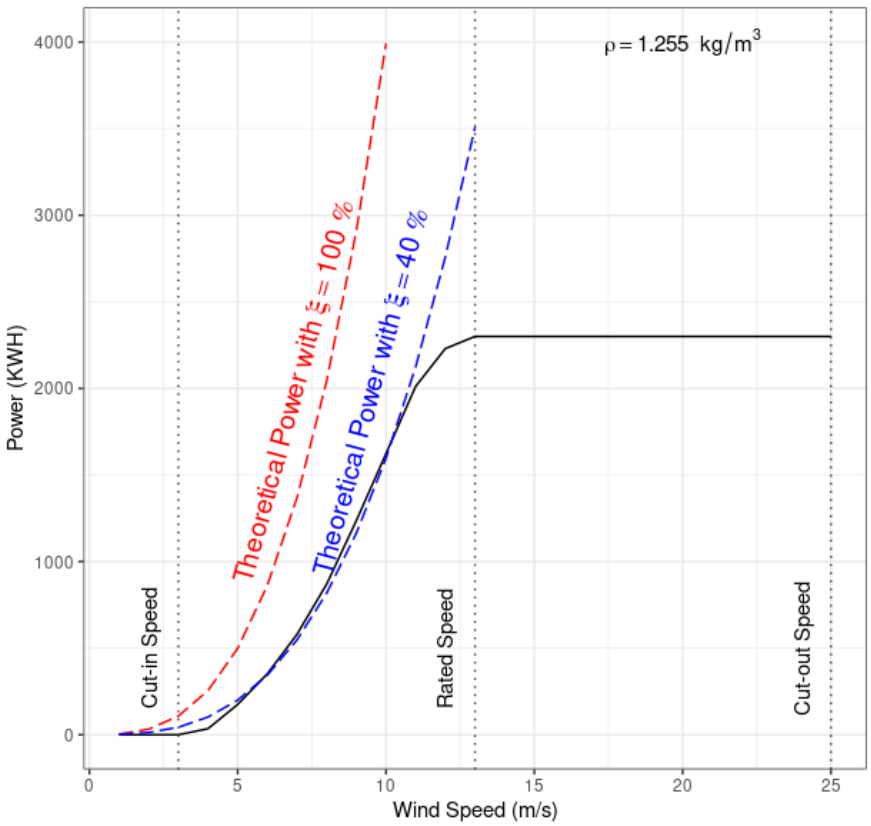


Figure 1.1. Power curve of Nordex-N90.

the cube of wind speed between cut-in and rated speed because the power curve is close to the theoretical power with a 40% efficiency factor.

Although the production for a single turbine can be predicted using its power curve, the aggregated production forecast for multiple turbines in a specific region requires more effort [7]. It is not always possible to reach turbine level information [7]. It can be technically erroneous, costly and labor-intensive to collect data from each turbine especially when the number of turbines in the region is large [7,8]. Also, actual wind speed information at turbine locations is not always available for forecasting tasks. To overcome the absence of perfect information, Numerical Weather Prediction (NWP) models can be used to receive wind speed forecasts for spatial grid points [1,9,10]. The spatial grid points (see Figure 1.2) are the intersection of equally spaced longitudes and latitudes that are subject to a specific region [11]. It is also clear that wind speed forecasts are needed since actual wind speed information is not available for

the forecasting of future production. The regional wind power forecasting problem is summarized in Figure 1.2. Suppose y_i is the production at wind turbine i, the aim is to develop a statistical model that predicts the sum of y_i's using NWP wind speed forecasts acquired at regular grid without knowing information about single y_i's. In the thesis, regional wind power forecasting task is chosen as a field of study.

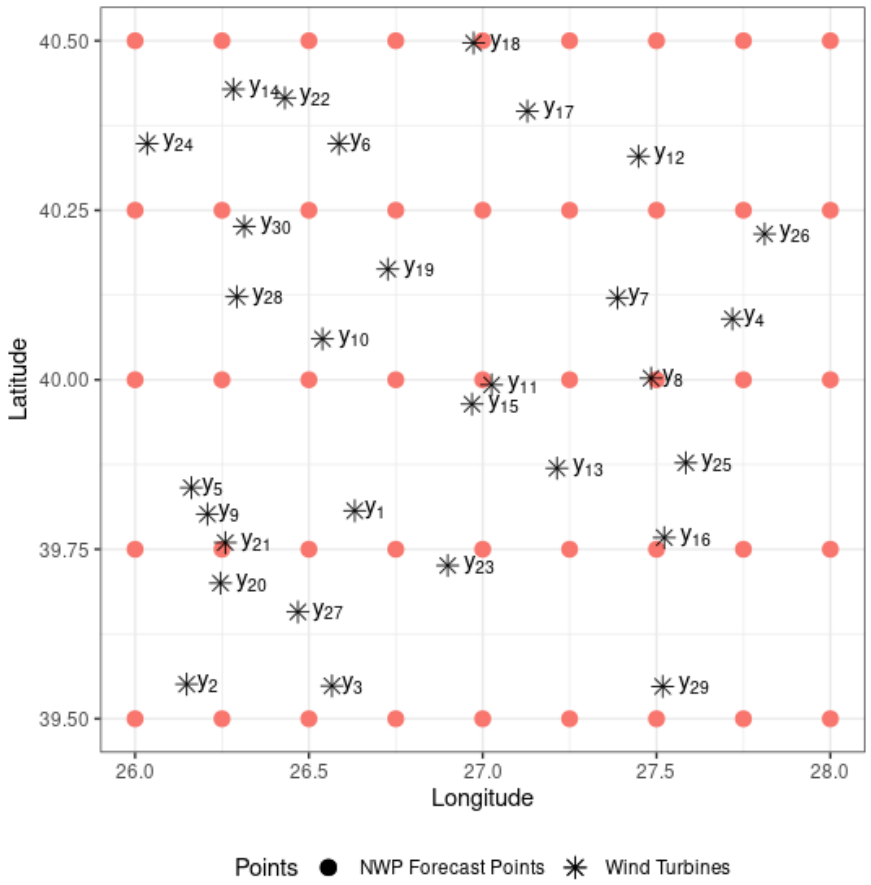


Figure 1.2. Regional wind power forecasting.

To forecast wind power, both physical and statistical approaches are proposed in the literature. The physical approaches derive wind power forecasts by applying power curve transformation of wind speed forecast at the exact location of wind turbines. However, statistical approaches predict wind power by building statistical models over historic wind power data and weather forecasts. Auto-regressive time series models, deep learning methods and tree-based ensembles are the widely used methods for wind power forecasting task [1, 12–15].

Random Forest (RF) [16] is a tree-based statistical model that is frequently used in wind power forecasting tasks because of its high accuracy and robustness compared to other methods and RF has proven success in wind power forecasting according to several experiments [12–15]. In addition, RF is powerful in modeling nonlinear relationships and it has embedded feature selection procedure. Therefore, RF seems suitable for the wind power forecasting task. RF works on the principle of ensembling univariate decision trees [16]. However, the split based on a single variable are not always optimal [17]. In order to compensate the disadvantages, tree-based methods utilizing more than one variable in the splitting have been proposed [18–20].

Oblique Decision Trees (ODT) are the methods that split data based on a linear combination of features [18–20]. Moreover, Oblique Random Forest (ORF) is a special type of RF that ensembles oblique decision trees [21, 22]. It has been observed in the several studies that tree-based methods with oblique splits are more successful than the univariate ones in terms of prediction accuracy [18–20]. However, oblique tree methods have generally been studied in classification tasks [22]. [22] claims that there is only one oblique tree study performed on time series tasks.

In the light of all these observations, a new oblique random forest method is proposed to be applied on the time series task. The new proposed method is referred to as Random Forest using LASSO regression (RF-LASSO) to find oblique splits and it ensembles multiple DT-LASSO's. As defined in the thesis, DT-LASSO is a special oblique decision tree method that uses Least Absolute Shrinkage and Selection Operator (LASSO) regression to find oblique splits at each node. LASSO is a penalized linear regression method that uses L_1 regularization [23]. The main purpose of choosing LASSO regression is to produce an affine hyperplane in a supervised manner because the output of linear regression is an affine hyperplane. Moreover, LASSO is more robust to multicollinearity as opposed to standard linear regression [23–27]. The robustness to multicollinearity is important for finding a proper affine hyperplane in splitting process especially for data with a highly correlated feature set.

Although it is possible to apply RF-LASSO to all regression problems, first of all, the study is carried out on the regional wind power forecasting task. The purpose of choosing the regional wind power forecasting task is that conventional RF method gives successful results in this domain and the task includes multivariate time series nature with high correlation between the series. [12–15]. The selection of the task allows for a valid and meaningful comparison between the proposed method RF-LASSO and conventional RF, while contributing to research on oblique tree methods for time series tasks.

This thesis is organized as follows: Section 2 provides the detailed literature review on tree-based models with oblique splits besides wind power forecasting methods with special interest on RF. Section 3 compiles the background information about the models used in the proposed method together with the performance metrics and NWP models. Section 4 starts with the motivation behind RF-LASSO. Then, the basics of RF-LASSO algorithm is explained. Lastly, the specialization of RF-LASSO for wind power forecasting task is described. Section 5 introduces experiments together with data and model results. Section 6 concludes the observations in the thesis.

2. LITERATURE REVIEW

In this section; firstly, the studies in the literature about tree-based learning methods utilizing oblique splits are summarized. Then, wind power forecasting literature is briefly introduced with emphasis on tree-based methods especially random forest. Lastly, the objective of the proposed method in the thesis is clarified alongside the existing studies in the literature.

2.1. Tree-Based Methods with Oblique Splits

The first oblique decision tree algorithm is named as Classification and Regression Trees with Linear Combinations (CART-LC) [28]. CART-LC finds the local optimum values for hyperplane coefficients in an iterative way by employing deterministic hill-climbing algorithm. Starting from the best orthogonal split option, CART-LC perturb the coefficients and stop when the increase in the goodness value of the split is lower than the predefined threshold at each iteration. Moreover, CART-LC applies backward selection method by eliminating the most irrelevant features one by one while keeping the goodness value in order to make a split simple and interpretable.

CART-LC can be stuck at local optima because it uses a deterministic hill-climbing heuristic. To escape local optimum, [29] proposes Simulated-Annealing Decision Trees (SADT) that applies random perturbations to coefficients at each iteration by the principles of simulated-annealing heuristic. The common problem of SADT that it is time-consuming to apply simulated-annealing heuristic at each node and it is an inefficient algorithm in terms of the time complexity.

To overcome the efficiency issue in SADT, [18] proposes Oblique Classifier 1 (OC1) algorithm that combines the idea of SADT and CART-LC. OC1 searches the local optimum at each iteration as CART-LC does and if it is stuck at local optimum, it perturbs the coefficients as SADT does. Moreover, OC1 introduce randomness by re-

peating the search with different initial solutions. Lastly, OC1 uses the best orthogonal split if it fails to find a better split option.

The aforementioned oblique decision tree methods find the split by applying heuristic algorithms. However, it is possible to find a linear hyperplane used for splitting in a supervised manner at each node. For example in [30], the proposed algorithm Sparse Multivariate Tree (SMT) solves the classification problem with logistic regression using L_1 regularization at each node to find a linear combination of the features for split evaluation.

Morever; Linear Machine Decision Trees (LMDT) [31], Weighted Oblique Decision Trees (WODT) [32], Householder CART (HHCART) [20], Continuous Optimization of Oblique Splits (CO2) [33] and Fast Algorithm for Classification Trees (FACT) [34] are the other examples of oblique decision trees in the literature.

Along with the oblique decision trees, the ensemble of them is also addressed in the literature. In [21]; Oblique Random Forest (ORF) method is proposed, influenced by [35] and [36]. The method differentiates from conventional RF [16] in the process of split search at each node. The bagging and random feature selection procedure is the same as in [16]. At each node, the method solves the classification problem with regression using L_2 regularization and randomly selected feature set. The penalization factor λ is chosen according to the performances on out-of-bag sample. Then, a possible split is searched on the fitted value of the ridge regression model by considering the maximum decrease in Gini impurity measure.

Heterogeneous ORF [37], Manifold ORF [38], ORF based on partial least squares [39], Robust Oblique Forests with Linear Matrix Operations (ROFLMAO) [40] and Sparse Projection ORF (SPORF) [41] are the other alternatives that ensemble oblique decision trees in order to get more robust and accurate predictions.

Although there are several oblique tree ensemble algorithms, almost all of them are applied and compared for classification tasks. It is claimed in [22] that their proposed method ORF via Least Square Estimation is the single ORF method in the scope of time series forecasting. The proposed method in [22] transforms the regression task to classification problem by labeling the target value as -1 if it is below the median value and +1 if it is above the median at each node. Then, least square estimation is applied to find linear hyperplane and the observations are separated into child nodes according to this hyperplane. There is no extra split point selection procedure applied on the fitted value. As it is also stated in [22], the proposed method has some limitations. The method uses historical data points as features and it is tested only for univariate time series problems. In practice, time series can be affected by other time series variables. So, the further investigation on multivariate time series forecasting models is suggested.

2.2. Wind Power Forecasting

With the growing interest in renewable energy sources, a vast amount of research has emerging on wind power forecasting. In [1], the studies on wind power forecasting are explained and compiled comprehensively. See [1] for comprehensive background on wind power forecasting task.

Wind power forecasting methods can be divided into two categories as physical and statistical approaches. The physical approaches aim to derive wind speed at the exact locations of wind turbines with the help of numerical weather prediction models and terrain characteristics. Then, wind power is estimated by predefined power curve transformations of the wind turbines [1].

On the other hand, statistical approaches intend to estimate the wind power by constructing statistical models that represent the relation between historical wind power observations and weather forecasts. In the literature, there are both parametric and non-parametric approaches applied for wind power forecasting such as auto-

regressive time series models, deep learning methods and tree-based ensembles [1]. For time series models, Auto-Regressive Integrated Moving Average (ARIMA) models are widely applied especially for short term wind power forecasting [42–44]. Artificial Neural Network (ANN), Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) are deep learning methods that are widely used for wind power forecasting due to their capability of learning complex nonlinear relations [45–47].

Tree-based ensembles such as random forests and gradient-boosted trees are also in the mainstream of wind power forecasting. The proposed method in this thesis is a random forest method with oblique splits. In [12–15], random forest algorithms are compared with alternative methods such as ANN, support vector machines, polynomial regression, gradient-boosted trees and Bayesian networks. Random forest proves its predictive strength in the wind power forecasting domain according to the performance comparisons in [12–15].

In addition to conventional random forest algorithm [16], [48] proposes the optimized version of RF which employs dimensionality reduction and weighted ensembling of resulting trees. Optimized RF uses wind speed forecasts generated from neural network model and the feature selection is applied by correlation elimination algorithm which is referred as Max Relevance-Min Redundancy (MRMR). The performances are compared for at most one day-ahead forecast. [48] concludes that RF has ability to forecast wind power more precisely compared to conventional wind power forecasting methods due to its ensembling capability.

In [49], an improved RF algorithm utilizing 2-stage feature selection obeying MRMR and the elimination of decision trees with weak generalization performance in the ensembling phase is proposed. The performances of both conventional and improved RF are compared against support vector machine with radial basis function and neural network model. [49] concludes that RF is more successful in terms of accuracy, efficiency, and robustness as compared to the alternatives.

In [50], RF algorithm using Poisson sampling instead of random bootstrapping in the bagging phase is proposed. Moreover, [50] transform wind power regression task into a classification task by discretizing target variable into several bins using chi-square test. Then, the target bin of a new observation is estimated by using weighted k-nearest neighborhood method based on wind speed and direction values. Lastly, RF model using Poisson sampling is built upon train data that falls into estimated target bin. The performances are compared against gradient-boosted trees and neural networks. [50] shows that both conventional RF using random sampling and the proposed RF perform better than two opponents.

Lastly; [51] proposes generalized random forest algorithm that ensembles honest regression trees introduced in [52]. Honest regression trees use each observation either for building the tree or determining the weights [52]. According to the performance evaluations on five different wind farm located in France and Turkey, both conventional and generalized RF algorithms show significant performance in comparison with gradient-boosted trees, Gaussian process regression and several support vector machine models using different kernel functions.

As a consequence of all these studies in tree-based learning with oblique splits and wind power forecasting, there is a clear lack of research on the oblique tree applications for time series problems [22]. On the other hand, RF proves its success in wind power forecasting tasks and it is widely used in several studies [12–15]. Therefore, the aim of this thesis to propose a new oblique random forest method RF-LASSO for a time series problem and measure its capability by comparing with conventional RF algorithm in wind power forecasting domain. Wind power forecasting is a proper domain for the comparison because of its multivariate time series nature and conventional RF is an appropriate method for the power forecast.

3. BACKGROUND

In this section, the background information about the proposed methodology including models, input set and performance metrics is provided.

3.1. Regression Methods

The main aim of regression methods is to find a proper function using statistical techniques that can explain and generalize relationship between the target and explanatory variables. There are parametric and non-parametric approaches for finding the explanatory function in regression. Parametric approaches such as least squares regression make several assumptions about the model and try to find model parameters under these assumptions. However, non-parametric approaches such as decision tree learn the model structure from data itself without given prior assumptions [53, 54].

3.1.1. Least Squares Regression

Least squares fitting is a parametric method to find best-fitted curve for given data points [24, 55–57]. The best-fitted curve is defined as the curve with the least sum of squares error. Least squares method aims at finding model parameters $\hat{\beta}$ of the explanatory function f_β such that

$$\hat{\beta} = \underset{\beta \in R^p}{\operatorname{argmin}} E(\beta) \tag{3.1a}$$

$$E(\beta) = \sum_{i=1}^{n} (Y_i - f_\beta(X_i))^2 \tag{3.1b}$$

where $Y \in \mathbb{R}^{n \times 1}$ is a target vector with size n equal to the number of observations, $X \in \mathbb{R}^{n \times p}$ is a feature matrix with p equal to the number of features. Y_i denotes the ith observation of Y and X_i denotes the ith row of X where $i = 1, 2, ..., n-1, n$.

3.1.1.1. Ordinary Least Squares. Ordinary Least Squares (OLS) is a special type of least squares method with the assumption of linearity and f_β given as

$$f_\beta(X) = X\beta \tag{3.2}$$

where $\beta \in \mathbb{R}^{p\times 1}$ is the regression coefficients vector. OLS is also called as a linear regression and tries to find the best-fitted linear line for given data [24]. It is a fast, simple and interpretable method. However, linear regression suffers from multicollinearity if there are correlated explanatory features [24]. Moreover; with the increasing p when n is kept constant, the space becomes more sparse and this sparsity may cause to problems in fitting [24]. This phenomenon is called as the curse of dimensionality [24].

3.1.1.2. Regularized Least Squares. To reduce the effects of the problems with OLS, the regularization term $R(\beta)$ is introduced into the error function $E(\beta)$ [23–27]. The error function is formulated as

$$E(\beta) = \sum_{i=1}^{n}(Y_i - f_\beta(X_i))^2 + R(\beta). \tag{3.3}$$

$R(\beta)$ is a function of β and the idea of regularization is that the complexity of the model can be controlled by proper $R(\beta)$. Regularized Least Squares method is also referred to as penalized regression.

Least Absolute Shrinkage and Selection Operator, namely LASSO is a penalized linear regression method utilizing L_1 norm for regularization [23–25]. The parameters $f_\beta(X)$ and $R(\beta)$ of LASSO regression are given as

$$f_\beta(X) = X\beta;\ R(\beta) = \lambda\sum_{j=1}^{p}|\beta_j| \tag{3.4}$$

where $\lambda \geq 0$ is a penalization factor. LASSO regression penalizes the sum of the absolute value of the model coefficients. So, the coefficients of some features with

less extra information tend to be zero [23–27]. LASSO regression has an embedded feature selection capability and it is more robust to multicollinearity as opposed to OLS [23–27]. To be more generic in definition, $R(\beta)$ can be extended into

$$R(\beta) = \lambda \sum_{j=1}^{p} |\theta_j|;\ \theta = D\beta \tag{3.5}$$

where $D \in \mathbb{R}^{z \times p}$ is a penalization matrix with z which is a custom dimension parameter [58–60]. This extension is called as generalized LASSO regression [58–60]. For simple LASSO regression, the penalization matrix is an identity matrix where $D = I$.

2-Dimensional (2D) Fused LASSO is a special type of generalized LASSO regression where the absolute coefficient differences between neighbor features are penalized by custom D [58–60]. For OLS, explanatory features are assumed to be independent but in practice, features may have spatial or temporal relations. For example, an image data has pixel values as features and each pixel has specific position on the image. Consider the 2×2 image data P and penalization matrix D as

$$P = \begin{bmatrix} p_1 & p_2 \\ p_3 & p_4 \end{bmatrix} \quad D = \begin{bmatrix} -1 & +1 & 0 & 0 \\ 0 & 0 & -1 & +1 \\ -1 & 0 & +1 & 0 \\ 0 & -1 & 0 & +1 \end{bmatrix} \tag{3.6}$$

for 2D Fused LASSO. Also, β_i is the coefficient of p_i where $i = 1, 2, 3, 4$. $|\theta|$ becomes

$$|\theta| = |D\beta| = \begin{bmatrix} |\beta_2 - \beta_1| \\ |\beta_4 - \beta_3| \\ |\beta_3 - \beta_1| \\ |\beta_4 - \beta_2| \end{bmatrix} \tag{3.7}$$

in the regularization component. So, it can be observed that the absolute coefficient differences between neighbor pixel values are penalized during parameter learning.

3.1.2. Tree-based Learning

Tree-based learning is a non-parametric learning method that tries to approximate the explanatory function between target and features via the combination of if-then-else rules [28, 61–63]. The resulting relationship between target and features is represented as a tree structure. Tree-based learning methods can be used both for classification and regression tasks [28]. Moreover, trees are generally easy to interpret because of the simple splitting rules.

3.1.2.1. Decision Tree. Decision tree (DT) is a greedy recursive partitioning algorithm that aims to minimize the variance in data at each partition [28]. Decision tree used for regression tasks is also called as a regression tree [28]. In general, decision tree employs the univariate split which is a split based on the value of a single feature [64, 65]. Therefore, the univariate decision tree divides feature space into rectangular regions and each split is orthogonal to the one of the features at each partition. Eventual regions after the consecutive splits are terminal nodes. Prediction for a new observation is the mean of the observations in the corresponding terminal node that the new observation falls into [28]. The univariate regression tree searches for the best splitting option of a single feature that maximize the reduction in the sum of squared errors (SSE) which is defined as

$$SSE = \Sigma_{i \in S_1}(y_i - \bar{y}_1)^2 + \Sigma_{i \in S_2}(y_i - \bar{y}_2)^2 \tag{3.8}$$

at each partition where S_1 and S_2 are the samples after the split, $\bar{y}_1$ and $\bar{y}_2$ are corresponding sample averages [24, 28, 62]. Although decision tree methods utilize the univariate split in general, it is possible to find splits based on multiple feature values [17]. This type of decision tree methods are called as the multivariate decision tree [17]. Oblique Decision Tree (ODT) is an example of the multivariate decision tree where linear or affine combinations of multiple features are evaluated for the splitting option at each partition [18–20].

Because decision tree is a greedy algorithm, it is prone to over-fitting [24, 66, 67]. To prevent over-fitting, several stopping conditions can be introduced such as maximum tree depth, minimum number of the observations in a node for further split. Without any stopping condition, it is possible to get complete discrimination in train set where each terminal node has single observation.

3.1.2.2. Random Forest. The good estimator can be defined as an estimator with low bias and low variance [24]. However, decision trees have low bias but high variance and they are prone to over-fitting because of the greedy nature [24, 66, 67]. Because the averaging reduces the variance, Random Forest (RF) ensembles multiple decision trees in a randomized manner to obtain an estimator with lower variance by keeping the low bias [16]. To ensemble multiple trees, random forest utilize bootstrap aggregating, namely bagging [68]. For each tree, random subset of observations are sampled and the tree is built upon this subset. Moreover, random forest algorithm evaluates random m features for split selection at each node in order to diversify the trees [16]. Figure 3.1 shows the overall schema of RF.

3.2. Numerical Weather Prediction (NWP)

Numerical Weather Prediction (NWP) is a physical simulation method used for weather forecasting by solving a set of differential equations numerically regarding the flow of fluids and atmospheric dynamics [69, 70]. NWP predicts future weather occurrences according to given initial conditions [69, 70]. In general, NWP models provide forecasts for various weather parameters including wind speed, temperature, pressure at different levels of the atmosphere such as surface, 10-meter above ground, 100-meter above ground [69–72]. Moreover, they provide forecasts on spatio-temporal grid points which are the intersections of equally spaced longitudes and latitudes for time indices of particular resolution [69, 70]. Figure 3.2 shows an example for the scope of a general NWP model.

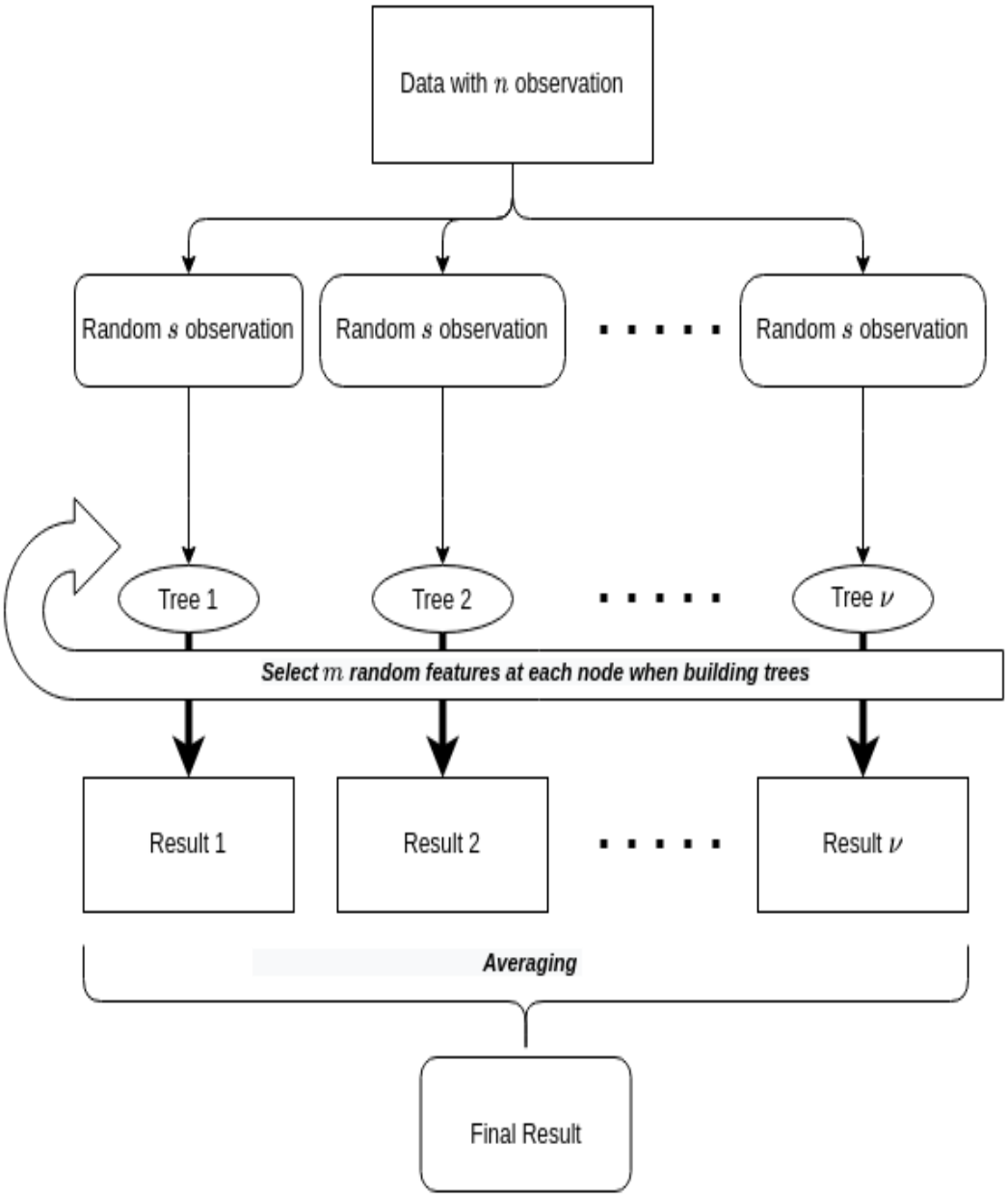


Figure 3.1. Overall schema of Random Forest.

Global Forecasting System (GFS) is a NWP model that provides weather forecasts on global scale with 0.25° spatial and hourly temporal resolution up to 120 hour ahead for each model run [71, 72]. GFS 0.25° hourly model is updated four times a day and each update is called as a model run [71, 72]. In the study for wind power forecasting, GFS model is used as a source of wind speed forecasts in this thesis. For each time point, forecasts from the latest model run of GFS are used as feature values.

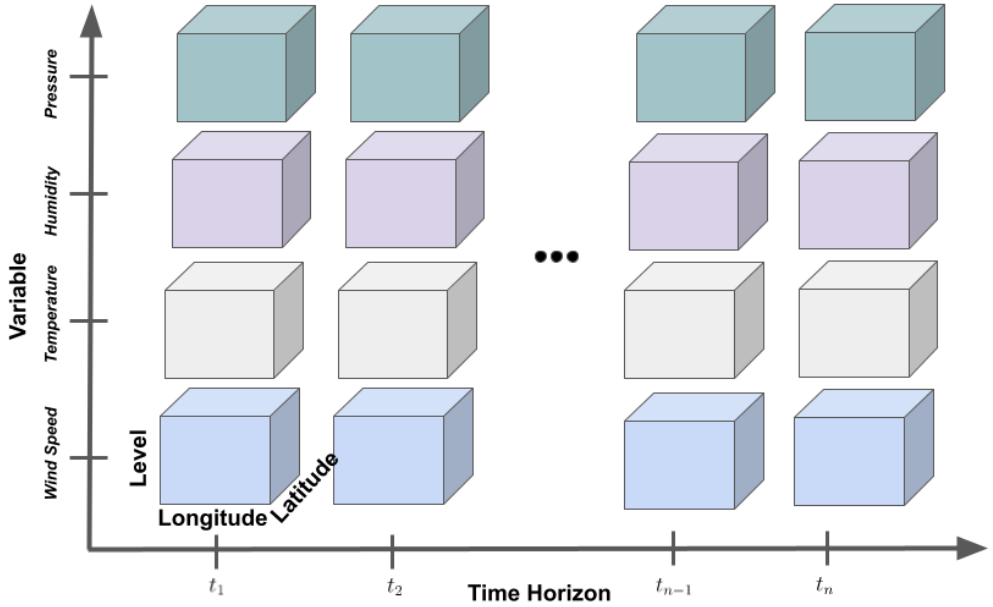


Figure 3.2. The scope of a NWP model.

3.3. Performance Metrics

This section introduces various performance measures used to compare and evaluate the models proposed in the study.

3.3.1. Mean Absolute Error (MAE)

Mean Absolute Error (MAE) measures the performance of predictor by its closeness to actual value in original scale and the formula is given as

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} \tag{3.9}$$

where y_i and $\hat{y}_i$ denotes the actual and the predicted value of the ith observation.

3.3.2. Mean Squared Error (MSE)

Mean Squared Error (MSE) measures the performance of predictor by penalizing the error with its square. MSE avoids larger errors as compared to MAE. It is a measure of variance for the predictor and a general loss function of both linear regression and regression trees. MSE is calculated as

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n} \tag{3.10}$$

where y_i and $\hat{y}_i$ denotes the actual and the predicted value of the ith observation.

3.3.3. Weighted Mean Absolute Percentage Error (WMAPE)

Weighted Mean Absolute Percentage Error (WMAPE) is a measure in percentage as compared to MAE which is a measure in original scale of the actual values. So, WMAPE is more interpretable and generic measure for the performance of the predictor compared to MAE. WMAPE is formulated as

$$WMAPE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{\sum_{i=1}^{n}|y_i|} \tag{3.11}$$

where y_i and $\hat{y}_i$ denotes the actual and the predicted value of the ith observation.

3.3.4. Bias

The quality of an estimator is a function of bias and variance. So, good estimator can be defined as having unbiased and low variance estimations. Bias metric is formulated as

$$BIAS = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{\sum_{i=1}^{n}|y_i|} \tag{3.12}$$

where y_i and $\hat{y}_i$ denotes the actual and the predicted value of the ith observation.

4. METHODOLOGY

This section summarizes the motivation and basics of the proposed method, namely RF-LASSO. Additionally, the specialization of RF-LASSO on regional scale aggregate wind power forecasting task is discussed in detail.

4.1. Motivation

In the wind power forecasting task, there is a nonlinear relationship between wind speed and wind power. Moreover, there are multiple wind speed forecast features at neighbor grid points. Therefore, it is important to have a model that is capable of learning nonlinear relationships between target and selected features. DT is an effective learning method used for both classification and regression tasks [28]. DT is capable of learning nonlinear relationships between target and explanatory variables [61–63]. Moreover, DT employs a built-in feature selection method when evaluating possible splits [61–63]. So, DT seems a proper choice for regional wind power forecasting.

Although its capability for finding nonlinear relations, the most widely used DT algorithms consider univariate splits in which single feature at each node is used [64,65]. Univariate decision trees employ orthogonal hyperplanes to feature space at each node, but orthogonal hyperplanes may not be suitable in some cases [17]. For example, Figure 4.1 shows the performances of oblique and orthogonal splits for a simple classification task where each class is indicated by different colors [24].

To show the deficiency of orthogonal split in the regression task, a simple regression problem is proposed where $Y \sim X_1 + X_2 + \epsilon$ and $X_1, X_2 \sim U(-1, 1); \epsilon \sim \mathcal{N}(0, 1)$. Figure 4.2 shows both orthogonal and oblique splits and their performances for a synthetic dataset. It can be observed that orthogonal splits are not the best option to reduce impurity in data because MSE when utilizing oblique splits is decreasing more sharply compared to the case with orthogonal splits.

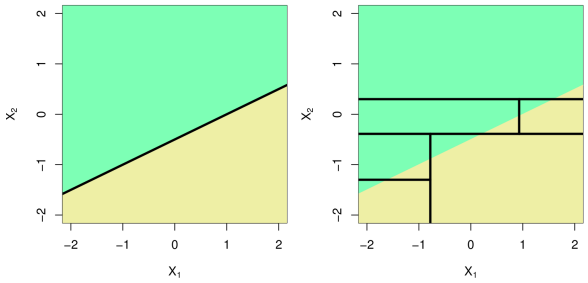


Figure 4.1. Oblique vs orthogonal split for classification task.

To avoid potential problems with orthogonal splits; Oblique Decision Tree (ODT) algorithm utilizing multivariate splits is proposed [18–20]. In ODT; oblique splits involving multiple features, which are affine hyperplanes, are evaluated at each node [18–20]. To find a proper reference hyperplane, which is a hyperplane that is perpendicular to candidate hyperplanes used for splitting, multivariate linear regression can be used because its output is also an affine hyperplane [24]. Figure 4.3 illustrates two perpendicular affine hyperplanes; the reference hyperplane with purple and a possible hyperplane that can be used for splitting with light grey.

The use of standard linear regression to find reference hyperplane can be problematic when there are multiple highly correlated features [24]. It is shown in Figure 4.4 that the wind speed feature set is highly correlated and this may create an issue of multicollinearity when finding a proper oblique hyperplane. Therefore, penalized regression algorithms can be used instead of standard linear regression because penalized regression algorithms are more robust to multicollinearity by penalizing the complexity of the model [23–27].

LASSO is a penalized regression method that uses L_1 regularization [23–27]. L_1 regularization penalizes the sum of absolute values of coefficients and this force some coefficients of the model to be zero. Therefore, LASSO can be a convenient candidate for finding affine hyperplane because it is more robust to multicollinearity and it has also an embedded feature selection procedure as decision tree does [23–27].

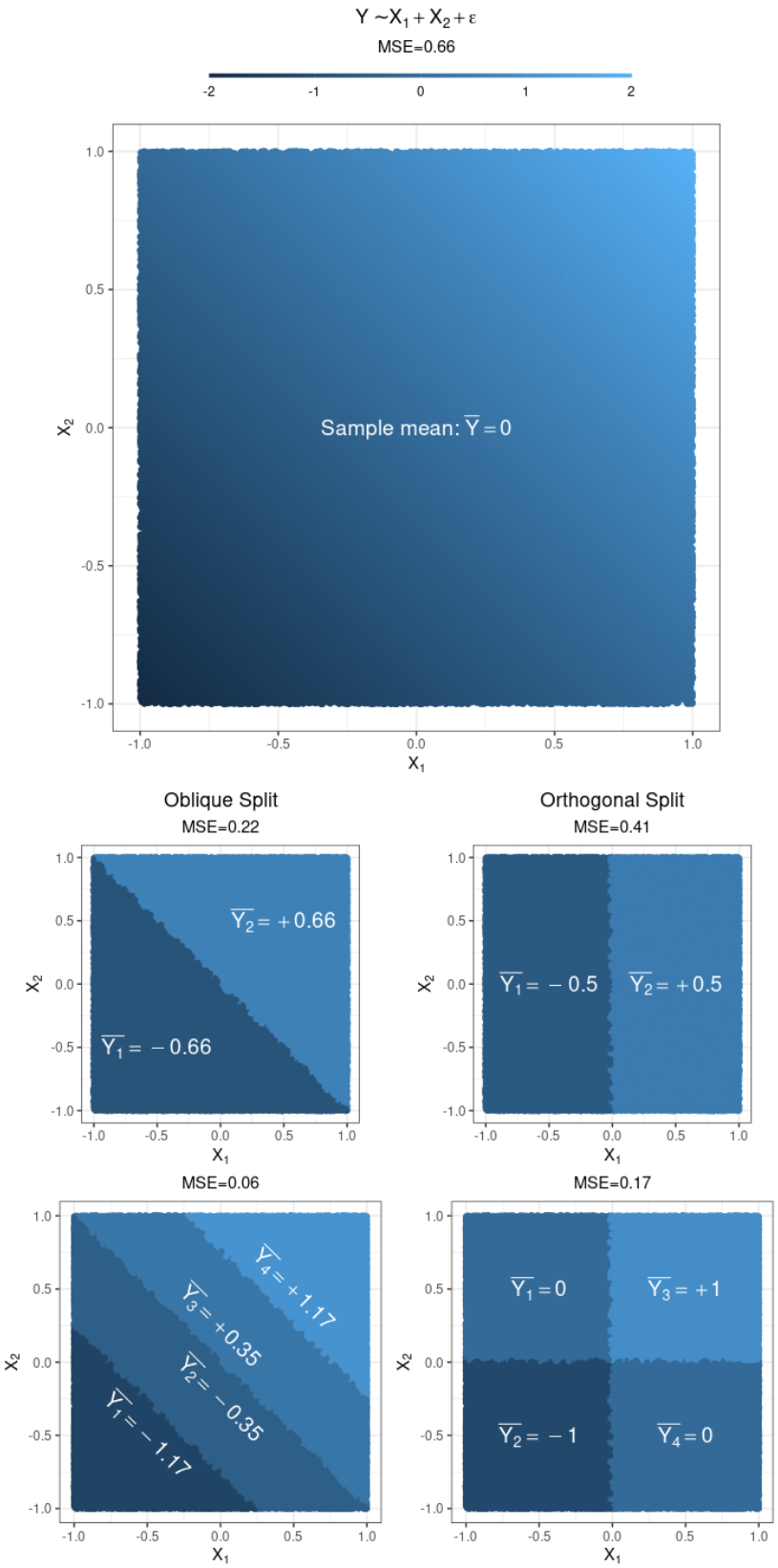


Figure 4.2. Oblique vs orthogonal split for regression task.

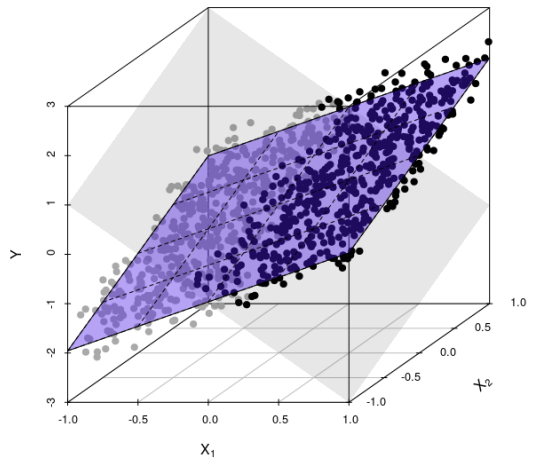


Figure 4.3. Affine hyperplane of linear regression for reference to candidates.

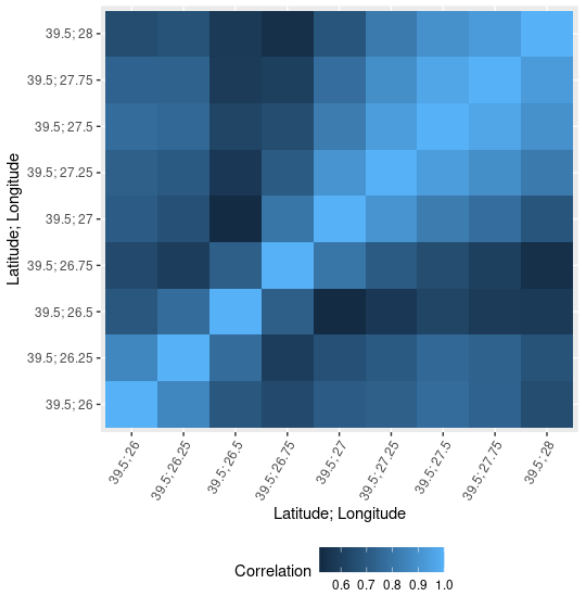


Figure 4.4. Correlations between Global Forecasting System (GFS) 0.25° hourly model wind speed forecasts.

Despite the aforementioned advantages; tree constructing algorithms are generally greedy heuristics, where the best splitting option is found at each node [66, 67]. The common problem with greedy heuristic algorithms is that it is possible to be stuck at the local optimum. In order to avoid this problem, randomization may be introduced [73]. Moreover, DT algorithms are prone to over-fitting because it is theoretically possible to reach complete discrimination in the train set by a fully-grown tree [24]. In the light of these problems, Random Forest (RF) algorithm is proposed [16]. RF is a tree-based ensemble method that utilizes bootstrap aggregating, or bagging [68]; in order to reduce over-fitting, improve accuracy and stability by decreasing variance with ensembling multiple trees [16]. In addition to bagging, RF selects random feature subset at each node for split evaluation in order to reduce dependency between trees by introducing randomization [16].

Oblique Random Forest (ORF) algorithm is the special type of RF that ensembles multivariate decision trees with oblique splits instead of the univariate trees with orthogonal ones [21, 22]. Consequently, the proposed method RF-LASSO is also a specialized ORF algorithm that ensembles decision trees with oblique splits found by LASSO regression.

4.2. Description of RF-LASSO Algorithm

RF-LASSO is a random forest method that ensembles multivariate trees with oblique splits in order to find empirical function f for problem formulated as

$$Y = f(X) + \epsilon \tag{4.1}$$

where $Y \in \mathbb{R}^{n\times 1}$ is a target vector with size n equal to the number of observations, $X \in \mathbb{R}^{n\times p}$ is a feature matrix with p equal to the number of features and $\epsilon \in \mathbb{R}^{n\times 1}$ is a random error component vector that follows standard normal distribution. Lastly; in general, $A_{i,j}$ denotes the entry at ith row and jth column of a matrix A, and B_i is the ith entry of a vector B.

RF-LASSO algorithm can be divided into two parts. The first one is the multivariate regression trees with oblique splits. Each multivariate tree is a learning algorithm on its own, namely DT-LASSO. DT-LASSO integrates LASSO regression that uses L_1 regularization in the splitting process [23, 24]. However, LASSO is a powerful tool if target and explanatory variables have a linear relationships but it has difficulty if the relationship is nonlinear [24]. Thus, DT-LASSO has an option to utilize nonlinear transformation of feature space by P^c in penalized regression fitting phase. P^c is a polynomial function with a concatenation operator, degree c and all terms equal to 1 except for the constant term which is 0. It can be defined as $P^c : X \rightarrow [X, X^2, ..., X^{c-1}, X^c]$ where P^1 is the identity function.

Unlike the classical decision tree methods that search at each node for the best splitting point of a single feature that maximize the reduction in the sum of squared errors; DT-LASSO firstly solves the problem

$$\text{Minimize:} \sum_{i=1}^{n}(Y_i - \sum_{j=1}^{cp} P^c(X_{ij})\beta_j)^2 + \lambda \sum_{j=1}^{cp} |\beta_j| \tag{4.2}$$

at each node in order to find regression coefficients vector $\beta \in \mathbb{R}^{cp\times 1}$ for λ^* value

$$\lambda^* = \underset{\lambda}{\text{argmax}}\{\lambda \mid \frac{\sum_{i=1}^{k} E_i}{k} < \min_{\lambda}(\frac{\sum_{i=1}^{k} E_i}{k}) + \frac{\sigma_E}{\sqrt{k}}\} \tag{4.3}$$

chosen by using k-fold cross-validation (CV) from candidates of possible λ values sequence where E_i is the mean square error (MSE) in the ith fold for the corresponding λ value and σ_E is the standard deviation of E_i where i is from 1 to k. λ^* is also denoted as $\lambda_{1.se}$ [74].

After finding β, the best splitting point that maximizes the reduction in SSE is searched on the fitted value which corresponds to $P^c(X)\beta$. In this way, the linear combination of features is used as a reference hyperplane that is perpendicular to the possible candidate affine hyperplanes in order to divide feature space. For the stopping criteria, maximum depth parameter d is used. Trees are grown until maximum depth is

reached. There is no restriction on the minimum number of observations in leaf nodes. The complexity is also controlled by maximum depth parameter d. In Figure 4.5, the pseudo-code of DT-LASSO algorithm is depicted.

Input:

- D : Input data
- p : Dimension of feature space
- P^c : Polynomial transformation function
- m : Number of random features selected at each node, p as default
- r : Repetition times for random selection of features, 1 as default
- d : Maximum tree depth

Output:

- T_i Terminal nodes of the tree

1: **for** $j = 1$ to r **do**
2: Randomly select m features out of p features
3: Fit LASSO on the current node of D using m features transformed by P^c
4: Find $\lambda_{1.se}$ value for LASSO using k-fold CV
5: $F_j \leftarrow$ fitted values of LASSO model for $\lambda_{1.se}$ value at step 4
6: **end for**
7: Find the best split point F^* over $F_1, F_2, ... F_r$ that maximizes SSE reduction
8: Split current node of D into 2 child nodes according to F^*
9: Repeat steps 1-8 for each child node until the tree is grown with depth d.
10: $T_i \leftarrow$ Keep terminal nodes of tree i
11: **return** T_i

Figure 4.5. DT-LASSO algorithm.

The second part of RF-LASSO algorithm is based on the ensembling of multivariate oblique decision trees in order to dilute the greedy behaviour of decision trees and prevent over-fitting by applying bagging [16,68]. RF-LASSO builds ν independent oblique regression trees on a randomly selected s data point that is sampled without replacement where $\frac{s}{n} \approx 0.632$ [68]. In the random sampling process, data points are selected with probabilities proportional to the weight vector $w \in \mathbb{R}^{n\times 1}$.

Input:

- n : Number of observations in data
- p : Dimension of feature space
- Y : Target vector
- X : Feature Matrix
- ν : Number of Tree
- w : Weight vector
- s : Number of randomly selected observations for each tree
- m : Number of random features selected at each node
- r : Repetition times for random selection of features
- d : Maximum tree depth

Output:

- $T_1, T_2, ... T_{\nu-1}, T_\nu$ Terminal nodes of each of ν trees

1: **for** $i = 1$ to ν **do**
2: $S_i \leftarrow$ Sample s out of n from $[Y, X]$ without replacement considering w
3: $T_i \leftarrow$ DT-LASSO$(D = S_i, p = p, m = m, r = r, d = d)$
4: **end for**
5: **return** $T_1, T_2, ... T_{\nu-1}, T_\nu$

Figure 4.6. RF-LASSO algorithm.

Introducing random noise into ensemble modeling improves accuracy if the correlations between individual learners are minimized while the prediction powers of the learners are kept significant [16]. To introduce randomness in the ensembling process,

in addition to bagging, m random features are selected at each node of trees to find the split point [16]. In RF-LASSO algorithm, m random features are selected and LASSO regression is fitted over these m features. This process is repeated r times at each node. Consequently, the split point is evaluated on r fitted value. In Figure 4.6, the pseudo-code of RF-LASSO algorithm is depicted.

Lastly, RF-LASSO algorithm produces prediction over ν trees for the new observation y_{new} by two consequent aggregation functions g_1 and g_2 such as mean, median, etc. The predicted output is formulated as

$$\hat{y}_{new} = g_1(g_2(\psi(T_j))); \quad y_{new} \in \psi(T_j) \tag{4.4}$$

where T_j is the terminal node observations at jth tree for $j = 1, 2, ..., \nu$ and ψ is a function returning the observations of terminal nodes that the new instance fall into together. $g_2(\psi(T_j))$ is a vector of size ν. Figure 4.7 illustrates the prediction process of RF-LASSO for the new observation y_{new}.

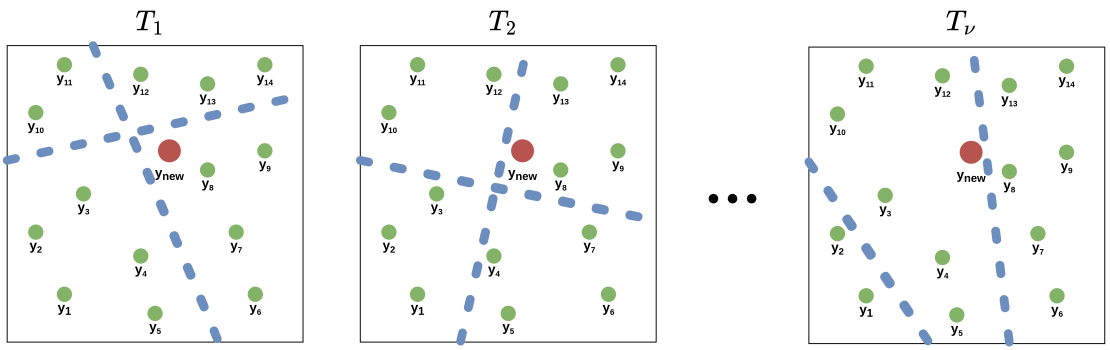


Figure 4.7. The prediction process of RF-LASSO.

4.3. RF-LASSO for Wind Power Forecasting Tasks

The basics of RF-LASSO algorithm have been explained in Section 4.2. In this part, the specialization of RF-LASSO algorithm on wind power forecasting task is discussed in detail. Firstly, the data used in determining the specifications of RF-LASSO algorithm is described. Then, the details of multivariate decision tree with oblique splits and ensembling parts of RF-LASSO are clarified respectively.

4.3.1. Data

The studies for determining RF-LASSO specifications are conducted on unlicensed wind power production data in the responsibility area of Uludag Electricity Consumption Company [75]. Although there are multiple wind turbines in this region, data only includes hourly aggregated wind production values in MWH from 2018-05-01 to 2021-09-30. There is no available information about the exact locations and productions of single turbines but, it is known that all turbines are located between 39.5 - 40.75 North latitudes and 25.75 - 29.75 East longitudes [75]. This boundary box is the minimal rectangle that covers the region. Data from 2021-01-01 to 2021-09-30 is used as a test and evaluation period and the rest is used for training possible alternatives. In Figure 4.8, hourly wind production values over time in the interested region are depicted.

The main driver of wind power is the wind speed variable [3]. Therefore; for explanatory variables, wind speed forecasts at 80 m above ground with the hourly resolution are retrieved from GFS 0.25° Hourly model [71, 72]. GFS 0.25° hourly model provides forecasts with a spatial resolution of 0.25° [71,72]. Therefore; there are $6 \times 17 = 102$ forecast points that cover 39.5 - 40.75 North latitudes and 25.75 - 29.75 East longitudes. Table 4.1 shows the structure of data used in the experiments.

Table 4.1. Sample Data Structure.

Time Indices		**Target**	**Features: Wind Speed at Lat_Lon**			
Date	**Hour**	**Wind Power**	**39_25.5**	**39_26**	**39.5_25.5**	**39.5_26**
5/2/2018	0	5.57	5.20	3.93	3.90	5.11
5/2/2018	1	3.19	5.12	3.63	3.51	4.64
5/2/2018	2	2.28	5.04	3.34	3.15	4.18
5/2/2018	3	3.17	4.97	3.04	2.81	3.74
5/2/2018	4	4.43	5.20	3.06	2.60	3.50
5/2/2018	5	4.98	5.43	3.10	2.40	3.27
5/2/2018	6	4.58	5.66	3.18	2.22	3.04
5/2/2018	7	3.27	5.76	3.16	2.13	2.84
5/2/2018	8	3.13	5.86	3.14	2.06	2.67
5/2/2018	9	2.18	5.96	3.12	2.01	2.54
5/2/2018	10	2.35	5.67	2.73	1.62	2.08
5/2/2018	11	3.71	5.42	2.44	1.28	1.80
5/2/2018	12	5.03	5.23	2.30	1.03	1.77
5/2/2018	13	6.27	5.20	2.56	0.59	1.74
5/2/2018	14	6.15	5.25	3.03	0.81	1.93
5/2/2018	15	5.97	5.37	3.64	1.41	2.27
5/2/2018	16	7.33	5.58	3.80	1.28	2.06
5/2/2018	17	6.52	5.81	4.01	1.32	1.86
5/2/2018	18	5.76	6.06	4.28	1.52	1.68
5/2/2018	19	3.83	5.66	3.76	1.15	1.23
5/2/2018	20	6.33	5.31	3.32	0.86	0.84
5/2/2018	21	5.90	5.01	2.99	0.77	0.62
5/2/2018	22	5.44	3.82	2.12	1.37	1.86
5/2/2018	23	3.82	3.06	2.37	2.67	3.30

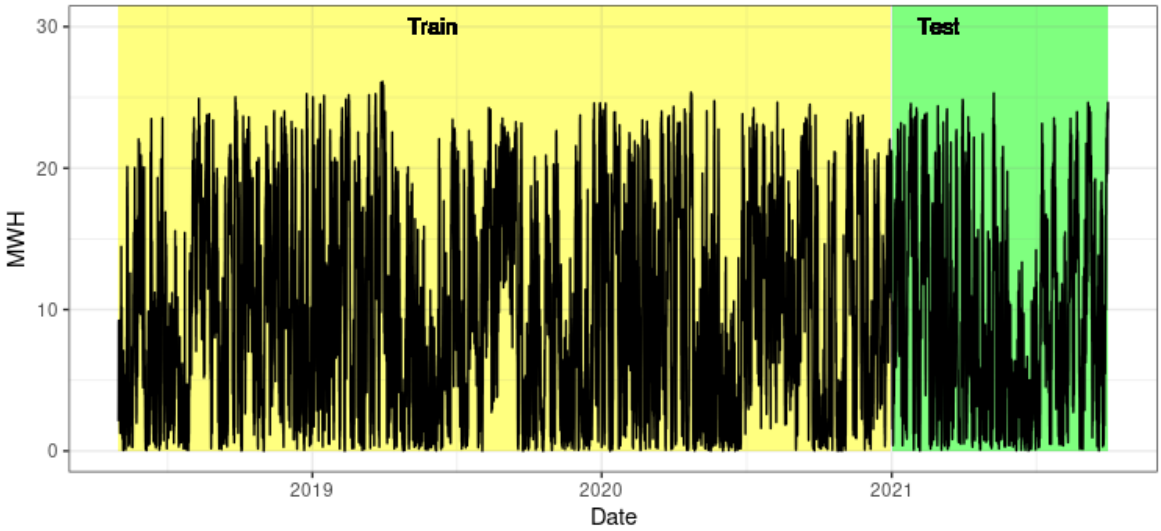


Figure 4.8. Hourly wind production values in Uludag region.

4.3.2. Multivariate Decision Tree with Oblique Splits

RF-LASSO algorithm ensembles multivariate oblique decision trees. At each node of the decision tree, linear combinations of feature values, which is affine hyper-plane, are evaluated for splitting. So, this combination is determined by applying LASSO regression at that node [23]. In this part, input transformations and parameter selection for LASSO regression are discussed. Moreover, 2D Fused LASSO and linear regression are tested as alternative methods for LASSO [24, 58–60]. In the experiments; R [76] software packages "*glmnet*" [74],"*rpart*" [77],"*genlasso*" [58] and "*ranger*" [78] are used for LASSO, DT, 2D Fused LASSO and RF respectively.

4.3.2.1. Determination of P^c. There is a nonlinear relation between wind speed and wind production (see Figure 1.1). Therefore, it is expected that nonlinear transformation of feature space provides better learning because LASSO is a linear regression method and its capability for learning nonlinear relations is weak [24]. In order to test the feature transformation effect, LASSO regression is applied to full data using P^c transformation for $c = 1, 2, 3, 4, 5$. $\lambda_{1.se}$ determined by 5-fold CV out of 50 candidate λ values. Figure 4.9 shows WMAPE performances on test data. The minimum WMAPE is reached with $c = 3$ which is in accordance with the wind power formula.

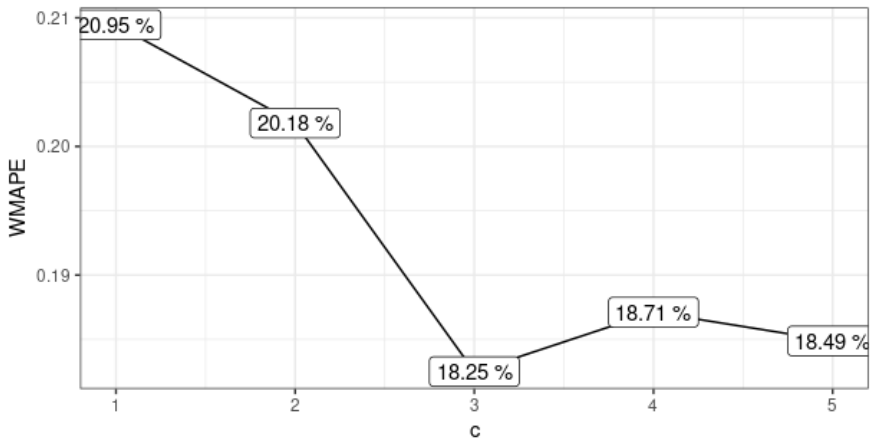


Figure 4.9. Polynomial input transformation LASSO regression performances.

4.3.2.2. Alternatives to LASSO. As alternative methods for finding reference affine hyperplanes in the splitting process; 2D Fused LASSO and standard multivariate linear regression are also tested against LASSO regression [24, 58–60]. The reason behind proposing 2D Fused LASSO as an alternative is that the feature set is a 2D grid with neighbor points. 2D Fused LASSO penalizes coefficient differences between neighbor points and this may lead to neater fitted values [58–60]. It is shown in Figure 4.4 that there are strong correlations between features which may lead to multicollinearity problem in finding the proper affine hyperplane. The main aim is to validate the effect of L_1 regularization on reducing multicollinearity issue by comparing it with alternative linear models [23–27]. Moreover, how the integration of linear models into the splitting process performs is analyzed. In order to test the effects; DT-LM algorithm which is the generic version of DT-LASSO that can use alternative linear regression models in addition to LASSO in splitting is proposed. DT-LM algorithm builds a single decision tree on full data using all of the feature set which corresponds to the default setting with $m = p$ and $r = 1$. Table 4.2 shows WMAPE performances of the following models on test data:

- DT : Conventional decision tree evaluates orthogonal splits [28]
- $DT - LASSO1 : DT$ using LASSO and P^1 transformation
- $DT - LASSO2 : DT$ using LASSO and P^2 transformation

- $DT - LASSO3 : DT$ using LASSO and P^3 transformation
- $DT - LR : DT$ using linear regression with P^3 transformation
- $DT - 2DFL : DT$ using 2D Fused LASSO with P^1 transformation

Table 4.2. WMAPE performances of alternative models.

Model	$d = 2$	$d = 4$	$d = 6$	$d = 8$	$d = 10$	$d = 12$
DT	30.33%	23.33%	21.53%	21.11%	21.42%	21.84%
DT-LASSO1	23.66%	18.12%	18.02%	18.18%	17.97%	17.98%
DT-LASSO2	22.77%	18.00%	17.92%	17.90%	17.87%	18.05%
DT-LASSO3	22.60%	17.95%	17.98%	**17.65%***	17.99%	17.89%
DT-LR	23.08%	19.69%	22.51%	26.69%	28.25%	28.29%
DT-2DFL	23.68%	18.64%	18.58%	18.70%	18.99%	19.32%

According to results in Table 4.2, the integration of LASSO into decision tree increases test performance significantly. The best performance is achieved by $DT - LASSO3$ with 17.65% which is in accordance with the inference from the polynomial degree selection part where the best result is reached with $c = 3$. It can be seen that $DT - LR$ performs poorly with increasing tree depth. Multicollinearity issue may cause comparably poor performance for $DT - LR$ because there is no penalization in linear regression [23–27]. Also, $DT - 2DFL$ performance is decreasing with increasing depth but it is still better than classical orthogonal DT. Lastly; $DT - LASSO1$, $DT - LASSO2$ and $DT - LASSO3$ have close performances in contrast to pure LASSO regression performances that differ significantly for cases $c = 1, 2, 3$ as shown in Figure 4.9. So, it can be derived that $DT - LASSO$ model is more robust as compared to LASSO in terms of c when the deep trees are grown.

4.3.2.3. Family Selection. In linear regression, it is assumed that $E[Y|X] = X^T\beta$, which is the conditional expectation of the target equals to the affine combination of features [79–82]. However; for cases where Y has no linear relationship with X, this assumption can be generalized into $E[Y|X] = G^{-1}(X^T\beta)$ where G is a one-to-one

and monotonic link function and Y is from the exponential family [79–82]. For linear regression, G is an identity link function. In the wind power forecasting case, it is known that there is no direct linear relationship between our feature and target variables. So, using alternative link functions can improve the goodness of fit. Therefore, logit function, the inverse of the sigmoid function, can be used for the link function because the shape of the sigmoid is similar to the wind power curve as shown in Figure 4.10.

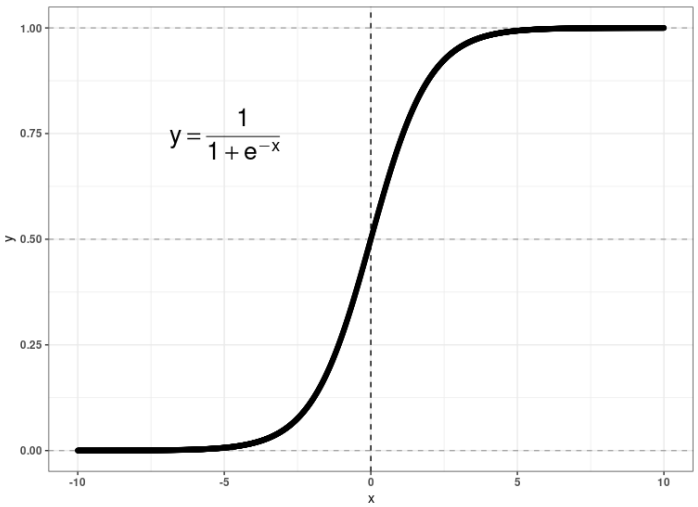


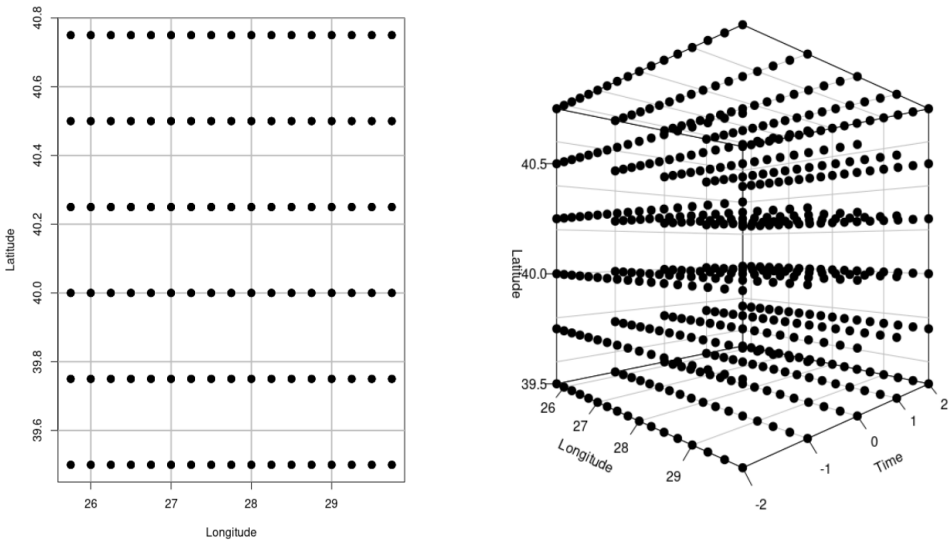
Figure 4.10. Sigmoid function.

Generalized Linear Models (GLM) with binomial and quasi-binomial families are the proper alternative choices to ordinary linear regression because they utilize logit link function [79–82]. In order to use binomial family models, the target variable should lie between 0 and 1 because binomial GLM can be used if Y is defined as $n_{success}/n_{total}$ [83, 84]. It is possible to reduce the wind production to [0,1] by applying the following transformation $Y'_{n,1} = Y_{n,1}/max(Y_{n,1})$. Table 4.3 shows WMAPE performances of different GLM families on test data for $DT - LASSO3$ model. According to data, Gaussian family still performs better. Gaussian family is the ordinary regression with identity link function and assuming error component distributed with normal distribution [79–82].

Table 4.3. WMAPE performances of alternative GLM families.

Family	$d = 2$	$d = 4$	$d = 6$	$d = 8$	$d = 10$	$d = 12$
Gaussian	22.60%	17.95%	17.98%	**17.65%***	17.99%	17.89%
Binomial	22.77%	18.03%	18.03%	18.24%	17.95%	18.23%
Quasibinomial	22.77%	17.95%	18.22%	18.14%	17.98%	18.08%

4.3.2.4. Temporal Feature Integration. In wind power forecasting task, both target and features have a time series nature. So, it is reasonable to integrate auto-regressive components into the modeling phase. In order to include temporal effects, it is shown in Figure 4.11 that the feature set is extended to $[X_{t-2}, X_{t-1}, X_t, X_{t+1}, X_{t+2}]$ where X_t is the wind speed forecasts for time t. Table 4.4 shows WMAPE performances of $DT - LASSO3$ algorithm with two different feature sets on test data. It can be deduced that the integration of temporal features improves model performance.



(a) Spatial feature set. (b) Spatio-temporal feature set.

Figure 4.11. Temporal extension of feature set.

Table 4.4. WMAPE performances of temporal extension in feature set.

Feature	$d = 2$	$d = 4$	$d = 6$	$d = 8$	$d = 10$
Spatial	22.60%	17.95%	17.98%	17.65%	17.99%
Spatio-temporal	22.12%	17.40%	17.53%	17.46%	**17.24%***

4.3.3. Ensembling

This section specifies bagging and ensembling strategies of RF-LASSO algorithm for wind power forecasting.

4.3.3.1. Aggregation Method. To get predictions over trained weak learners, RF uses the mean operator for both g_1 and g_2 in Equation (4.4) [16, 24, 68]. However, mean operator is suitable for symmetrical distributions and it has disadvantages if the distribution is skewed. For skewed distributions, median is a better measure of central tendency [85, 86]. Figure 4.12 shows that production values have right-skewed distribution and there is a gap between mean and median values. Therefore, different combinations of g_1 and g_2 are tested using RF with $\nu = 100$, $m = 24$ and spatio-temporal feature set as input. Additionally, quantile random forest (QRF) is trained for comparison purposes [87]. In QRF; $\psi(T_j)$ for ν trees are united under a single vector and the prediction is the median value of that vector for the new observation [87]. Table 4.5 summarizes WMAPE results for the combinations of g_1 and g_2 and QRF strategies in aggregation. The results show that using the median operator for both g_1 and g_2 performs the best in the test period. Also, it can be concluded that combining terminal nodes under a single vector and applying the median operator over a combined set is less effective because the local distributions of terminal node observations are ignored in this case contrary to the best scenario where each terminal node distribution is evaluated separately by applying median operator over each of them.

4.3.3.2. Temporal Bagging. In random forest algorithm; the probability of being chosen is distributed uniformly for each data point [16, 24, 68]. In other words, observa-

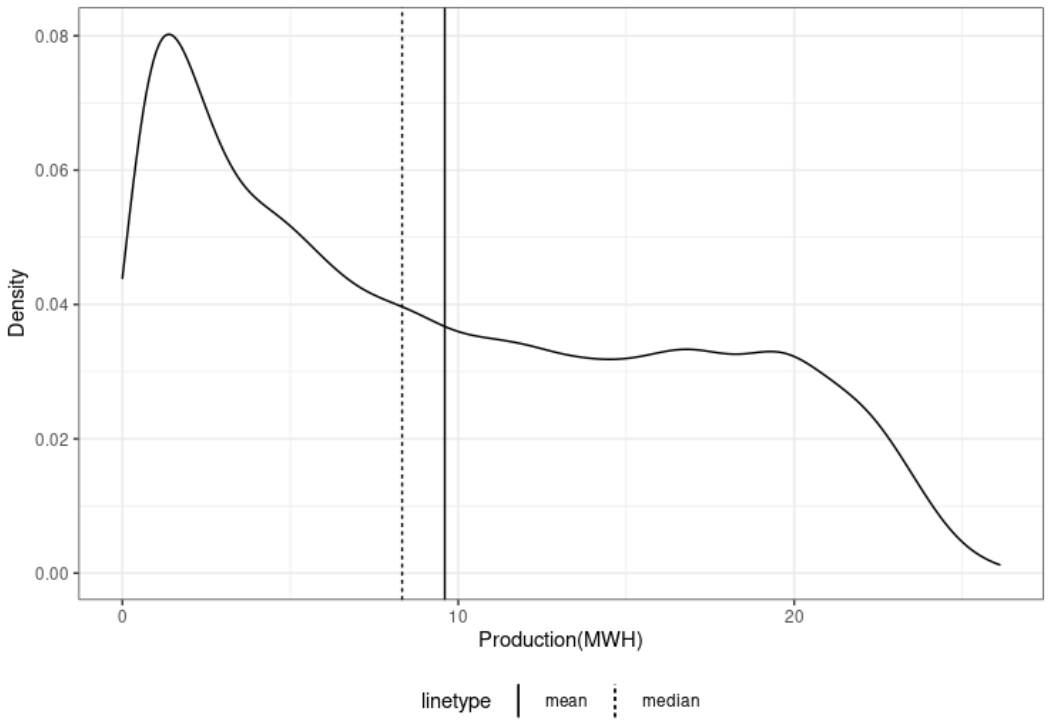


Figure 4.12. Production distribution.

Table 4.5. WMAPE performances of aggregating functions.

Model	g_1	g_2	$d = 4$	$d = 6$	$d = 8$	$d = 10$	$d = 12$
RF	**Mean**	**Mean**	19.71%	18.26%	17.60%	17.22%	17.08%
RF	**Mean**	**Median**	18.96%	17.83%	17.35%	17.07%	17.00%
RF	**Median**	**Mean**	19.46%	18.13%	17.47%	17.05%	16.85%
RF	**Median**	**Median**	18.95%	17.83%	17.32%	16.98%	**16.83%***
QRF	**-**		19.10%	17.88%	17.45%	17.15%	16.97%

tions are randomly selected in the bagging phase when building ν independent decision trees [16, 24, 68]. In time series nature, more recent observations have a larger probability of carrying useful information for forecasting of future [88].

The wind power forecasting task also includes temporal effects as it is discussed in the previous section. So, giving more weight to recent observations in the bagging

phase may help to extract more information. "T-Bagging" algorithm which employs temporal bagging for time series problems is proposed in [88]. In this part, "T-Bagging" alternatives have been experimented for wind power forecasting. Weights that change according to the time index for alternative strategies are depicted in Figure 4.13 [88]. The probability of being selected in bagging is proportional to weight magnitude. The sampling is applied without replacement. In order to test the strategies, RF algorithm with $\nu = 100$, $m = 24$ and spatio-temporal feature set as input is trained for different tree depths. For aggregation, median function is used for both g_1 and g_2. Table 4.6 summarizes WMAPE results for six types of bagging strategies. According to the results, all strategies giving more weight to recent data perform better than Type1 strategy which corresponds to pure random selection. Also, strategies with exponentially increasing weights to recent observations perform slightly better than logarithmically increasing ones. With minimum average WMAPE over different tree depths, Type4 strategy is selected for bagging operation.

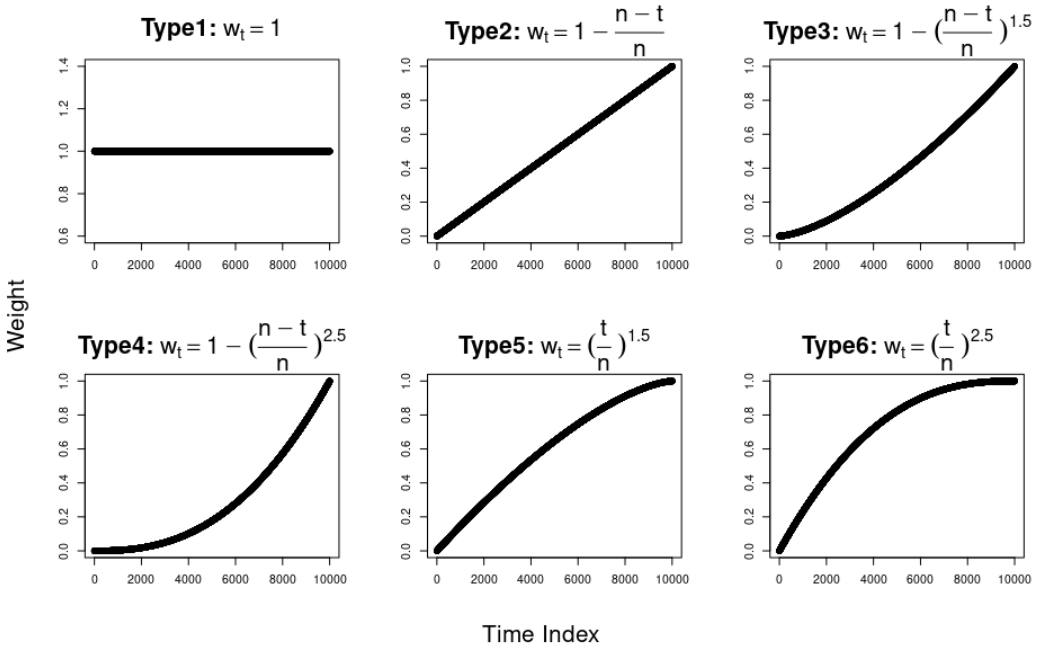


Figure 4.13. Temporal weights for bagging.

Table 4.6. WMAPE performances of weighting strategies for bagging.

Bagging	$d = 4$	$d = 6$	$d = 8$	$d = 10$	$d = 12$	**Mean**
Type1	18.95%	17.83%	17.32%	16.98%	16.83%	17.58%
Type2	18.79%	17.75%	17.13%	16.88%	16.67%	17.44%
Type3	18.83%	17.64%	17.12%	16.81%	16.73%	17.43%
Type4	18.63%	17.66%	17.18%	16.85%	16.74%	**17.41%***
Type5	18.72%	17.66%	17.14%	16.89%	16.76%	17.43%
Type6	18.84%	17.77%	17.25%	16.95%	16.84%	17.53%

5. EXPERIMENTS AND RESULTS

In this section, datasets used in the experiments and the experimental setup are explained. Then, the results of the experiments are presented. Finally, future works about the possible improvements for the proposed method are discussed.

5.1. Descriptions of the Datasets

The experiments are conducted with a total of three different datasets, with two new datasets containing the total production of multiple wind farms in addition to the data described in Section 4.3.1 (Dataset 1). The first of the additional datasets (Dataset 2) consists of the following 12 wind farms; Mazı, Aliağa Bergama, Soma12, Zeytineli, Pitane, Bergres, Kuyucak, Kırkağaç, Yuntdağ, Kocadağ, Geres and Düzova. The second (Dataset 3) consists of 6 wind farms; Bares, Üçpınar, Kocalar, Kızılcaterzi, Edincik and Şadıllı. In the selection of the datasets, factors such as having different production levels, being located in different regions and having different data lengths are taken into consideration. The additional datasets are accessed through the EPIAS Transparency Platform [89].

In all three datasets, the main aim is to estimate the hourly total wind production. As the explanatory features, GFS 80-meters above ground wind speed forecasts at each 0.25° point in a boundary box are used [71, 72]. The boundary box is defined as the smallest rectangle covering the interested region. Although wind speed forecasts are in hourly resolution as of now, they are presented in 3-hour resolution before 2020-10-05. The wind speed forecast data in the 3-hour resolution period are converted into hourly resolution by linear interpolation method.

In the experiments, each dataset is divided into two parts as train and test. Model learning studies are carried out on the train data, and the performances of the corresponding models are evaluated on the test set. Figure 5.1 shows the boundary

boxes covering the respective datasets. In Figure 5.2, hourly wind productions over time are shown with additional information about the train and test periods.

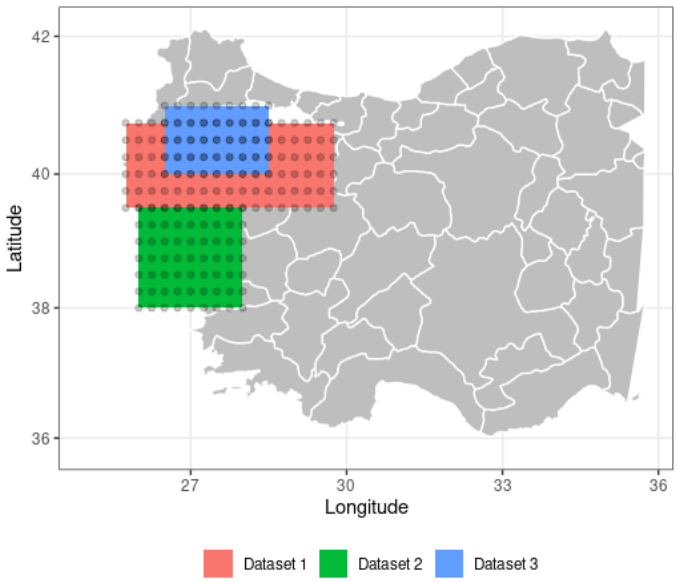


Figure 5.1. Boundary boxes of the datasets.

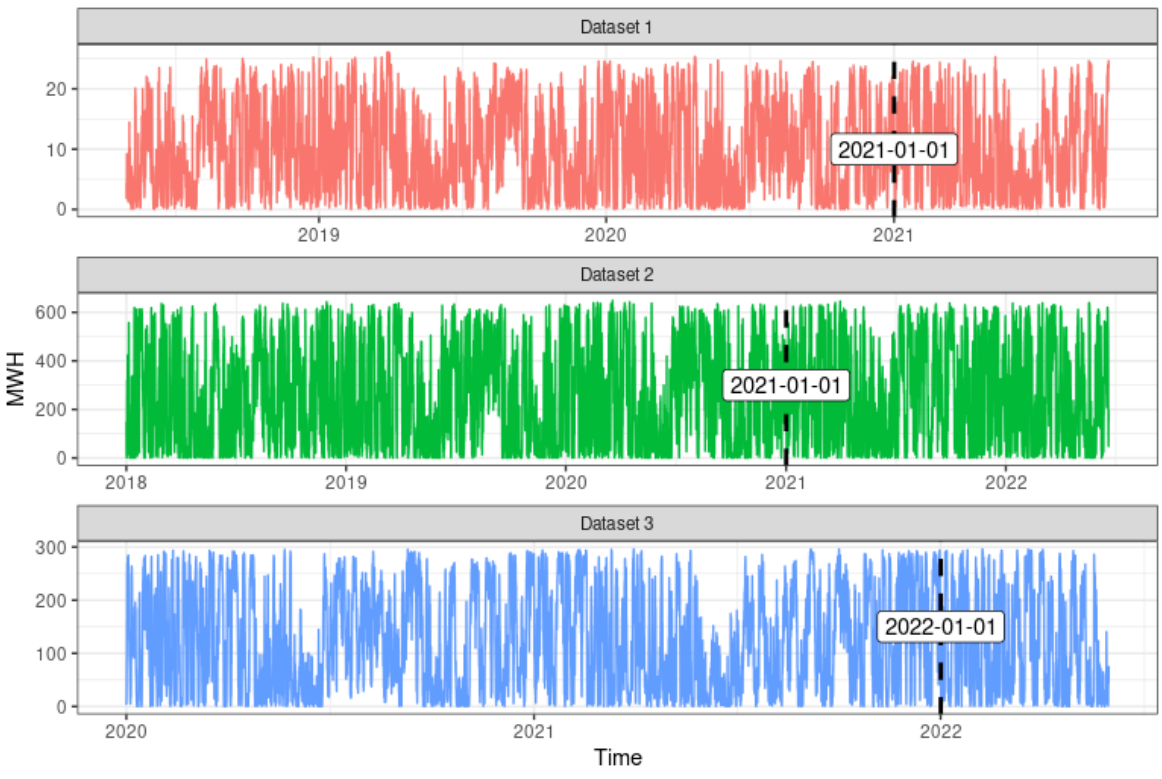


Figure 5.2. Productions over time with test start dates.

5.2. Experimental Setup

The experiments are primarily designed to measure the predictive performance of the proposed RF-LASSO algorithm against the conventional RF algorithm. Therefore, the performances of each model over the test periods of each datasets are accepted as an anchor point. WMAPE is selected as the primary performance metric in comparisons. The main purpose of this selection is to facilitate comparisons on different datasets, since the WMAPE metric is scale-free.

In the experiments, the performance of the DT-LASSO algorithm compared to the conventional DT and LASSO regression algorithms is also evaluated. Moreover, an extended version of RF algorithm called RF-EXT is proposed to measure the effects of alternative aggregation functions and temporal bagging. So, the models given with the following details are used in the experiments:

- LASSO: Penalized linear regression model with L_1 regularization [23]
  - (i) X, $P^3(X)$, $[X_{t-2}, X_{t-1}, X_t, X_{t+1}, X_{t+2}]$, and $P^3([X_{t-2}, X_{t-1}, X_t, X_{t+1}, X_{t+2}])$ are used for the experiments as an input data. X_t is the wind speed forecasts for time t.
  - (ii) $\lambda_{1.se}$ is used for prediction. $\lambda_{1.se}$ is determined by 5-fold CV out of 50 candidate λ values.
  - (iii) Gaussian family is chosen for the GLM family parameter.
  - (iv) "*glmnet*" [74] package is used for training.
- DT: Conventional decision tree evaluates orthogonal splits [28]
  - (i) X and $[X_{t-2}, X_{t-1}, X_t, X_{t+1}, X_{t+2}]$ are used for the experiments as an input data.
  - (ii) Trees are grown until maximum depth parameter d is reached without additional limitation on complexity.
  - (iii) $d = \{2, 4, 6, 8, 10, 12\}$ are selected for the experiments.
  - (iv) "*rpart*" [77] package is used for training.
- DT-LASSO: The proposed ODT algorithm

(i) X, $P^3(X)$, $[X_{t-2}, X_{t-1}, X_t, X_{t+1}, X_{t+2}]$, and $P^3([X_{t-2}, X_{t-1}, X_t, X_{t+1}, X_{t+2}])$ are used for the experiments as an input data.

(ii) $m = p$ and $r = 1$ default settings are used. All features are the input for LASSO regression at each split without random selection.

(iii) $\lambda_{1.se}$ is used for acquiring fitted values in splitting process. $\lambda_{1.se}$ is determined by 5-fold CV out of 50 candidate λ values. Gaussian family is chosen for the GLM family parameter.

(iv) Trees are grown until maximum depth parameter d is reached without additional limitation on complexity.

(v) $d = \{2, 4, 6, 8, 10, 12\}$ are selected for the experiments.

(vi) It is implemented in R software [76]. The implementation can be found in the author's repository [90].

- RF: Conventional RF algorithm that ensembles univariate DT's [16]
  (i) The feature set X is extended to $[X_{t-2}, X_{t-1}, X_t, X_{t+1}, X_{t+2}]$
  (ii) $\nu = 100$ trees are ensembled.
  (iii) $d = \{4, 6, 8, 10, 12, 16, 24\}$ and $m = \{8, 16, 24, 32, 64\}$ are selected for the experiments.
  (iv) Both g_1 and g_2 are the mean operator.
  (v) Random sampling is applied with replacement in the bagging. The probability of being chosen for each observation is equal.
  (vi) "*ranger*" [78] package is used for training.
- RF-EXT: The extended version of RF algorithm except that:
  (i) Both g_1 and g_2 are the median operator.
  (ii) Random sampling s observation out of n without replacement is applied in the bagging. The probability of being chosen for each observation is proportional to Type4 strategy.
  (iii) "*ranger*" [78] package is used for training. But, the prediction function utilizing median operators is implemented additionally in R.
- RF-LASSO: The proposed ORF algorithm
  (i) The feature set X is extended to $[X_{t-2}, X_{t-1}, X_t, X_{t+1}, X_{t+2}]$

(ii) P^3 transformation is applied in LASSO regression.

(iii) $\lambda_{1.se}$ determined by 5-fold CV out of 50 candidate λ values. Gaussian family is chosen for the GLM family parameter.

(iv) $\nu = 100$ trees are ensembled.

(v) $d = \{4, 6, 8, 10\}$ and $(m, r) = \{(p, 1), (16, 1), (24, 1), (32, 1), (4, 16), (4, 24), (4, 32), (8, 16), (8, 24), (8, 32), (16, 16), (16, 32), (32, 16), (32, 32)\}$ are selected for the experiments.

(vi) Both g_1 and g_2 are the median operator.

(vii) Random sampling s observation out of n without replacement is applied in the bagging. The probability of being chosen for each observation is proportional to Type4 strategy.

(viii) It is implemented in R software [76]. The implementation can be found in the author's repository [90].

5.3. Results

In this section, first of all, the performances of DT-LASSO, LASSO and DT models are compared over the test period of each dataset. Afterwards, detailed performance analysis of RF and RF-EXT models is performed. In addition, a comprehensive comparison of the RF-LASSO model with RF and RF-EXT is conducted. Finally, the results of all models with the best working parameter versions are summarized.

5.3.1. DT-LASSO Comparisons

In this section, the forecasting performances of DT-LASSO, LASSO and DT models are reported and discussed. The main purpose of comparing these three models is that the proposed model DT-LASSO is an oblique decision tree method and uses LASSO regression when finding oblique splits. Therefore, the performance of DT-LASSO against the conventional orthogonal decision tree method is also be determinant for the random forest models that ensemble these trees. Figures 5.3, 5.4 and 5.5 show the performance of the models in terms of WMAPE on all three datasets, respectively.

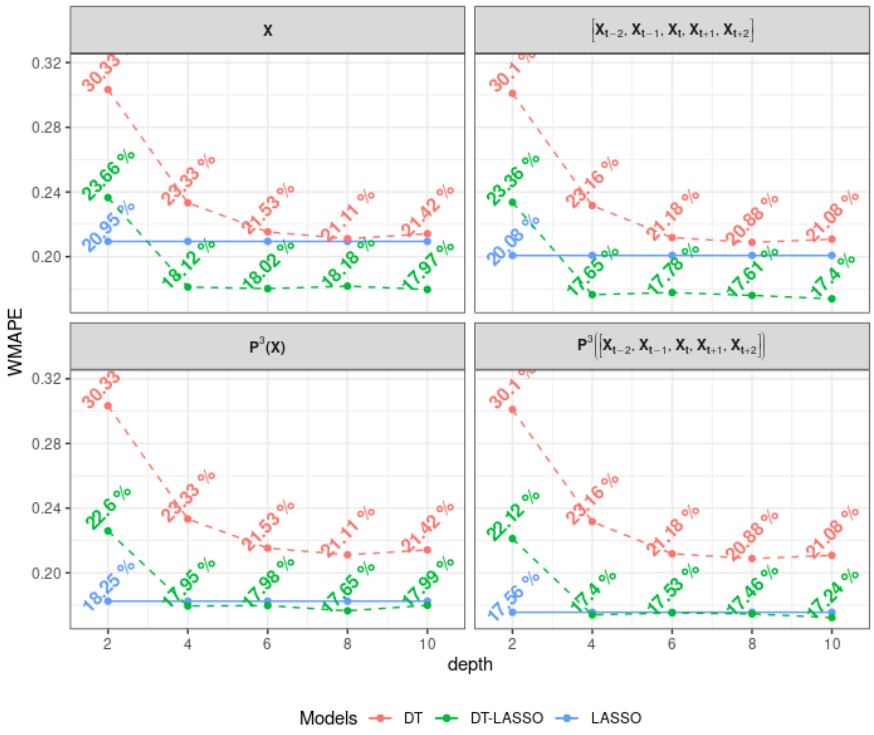


Figure 5.3. DT-LASSO performance comparisons for Dataset 1.

DT-LASSO model performs best in all three datasets and all input combinations. The best results are achieved with DT-LASSO using $P^3([X_{t-2}, X_{t-1}, X_t, X_{t+1}, X_{t+2}])$ as an input. Moreover, the performance of LASSO for no input transformation setting lags significantly behind the case with P^3 polynomial transformation setting. However, this situation is not the same for DT-LASSO. Although DT-LASSO utilizes LASSO regression with no input transformation, it still seems to be successful especially for the capturing of nonlinearity. Additionally, the integration of temporal dimension into feature set improves the predictions for all models. While DT-LASSO has preserved the strength of conventional DT method for learning nonlinear relations, It has also preserved the predictive power of LASSO regression.

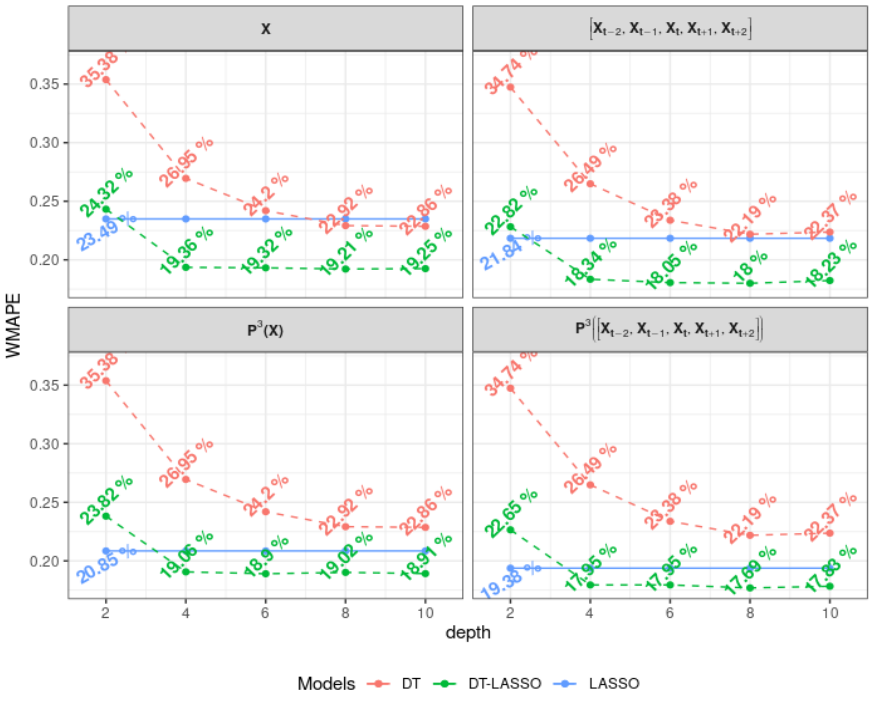


Figure 5.4. DT-LASSO performance comparisons for Dataset 2.

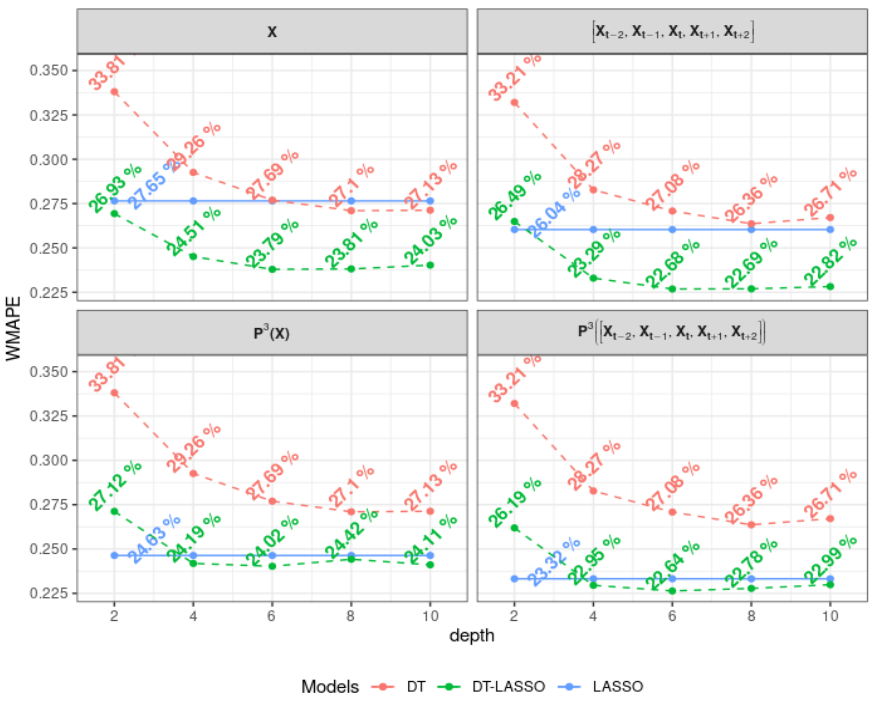


Figure 5.5. DT-LASSO performance comparisons for Dataset 3.

5.3.2. RF and RF-EXT

This section compares the results of RF and RF-EXT models. In time series tasks, current observations are more informative than previous ones for forecasting [88]. Based on this assumption, RF-EXT model gives more weight to current observations in the bagging , unlike RF model which gives equal weight to each observation [16]. Therefore, the probability of using current observations in individual decision trees is higher for RF-EXT as compared to RF. Moreover, the median operation estimates the central tendency better than the mean operation for non-symmetrical distributions [86]. Therefore, RF-EXT model uses median values instead of mean in the prediction process in addition to temporal bagging. To analyze the effects of extensions of RF-EXT, the performances of RF and RF-EXT model over the test period of each dataset are compared in terms of WMAPE. Then, BIAS and WMAPE values at different wind power levels are reported to analyze the distributional effects of wind power data for prediction . Figures 5.6, 5.7 and 5.8 show the performance of the models for different d and m values for all three datasets, respectively. Table 5.1 summarizes the best performing RF and RF-EXT with corresponding parameter setting for each dataset.

RF-EXT outperformed RF in all parameter combinations for each dataset. Especially for Dataset 2 and Dataset 3, the performance difference between the best parameter combinations of RF and RF-EXT is significant. To evaluate the performance of median as compared to mean in aggregation, wind power is divided into five equal quantiles after sorting. Quantile ranges of the productions are denoted as groups. Then, BIAS and WMAPE values of RF and RF-EXT in the quantiles are analyzed. Table 5.2 summarizes the quantiles with their corresponding group number and Table 5.3 shows the performances of the models in each range. As observed, BIAS of RF-EXT model is closer to 0 compared to BIAS of RF, especially in the low and high wind power ranges. The results indicate that median operation estimates the central tendency of wind power distribution better than mean operation does.

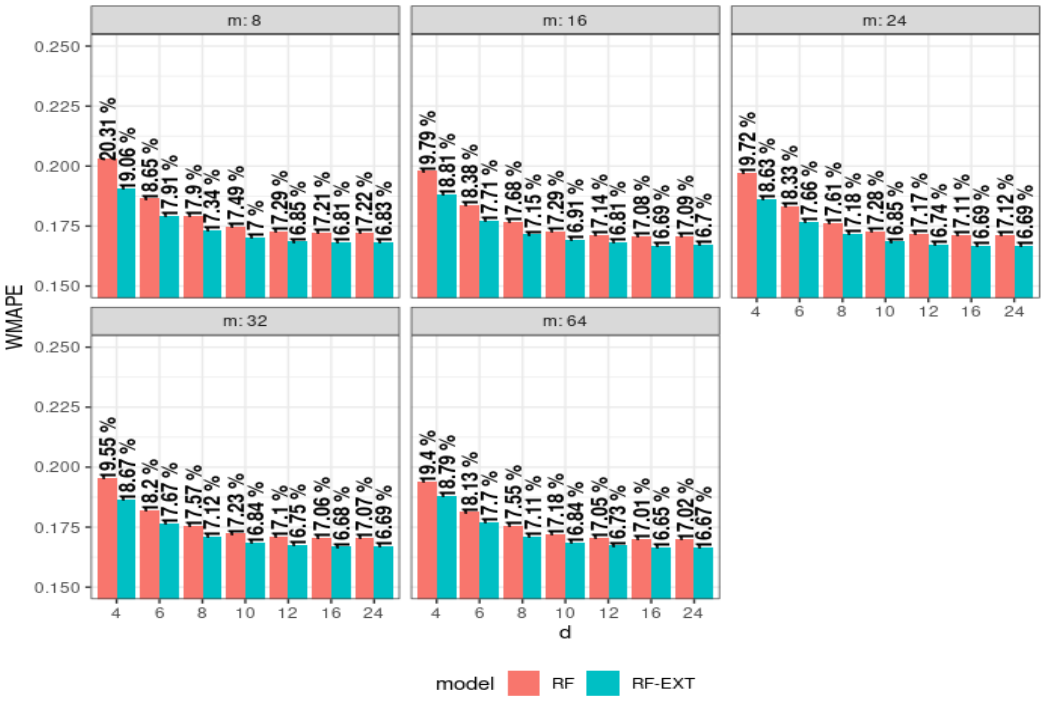


Figure 5.6. RF vs RF-EXT performance comparison for Dataset 1.

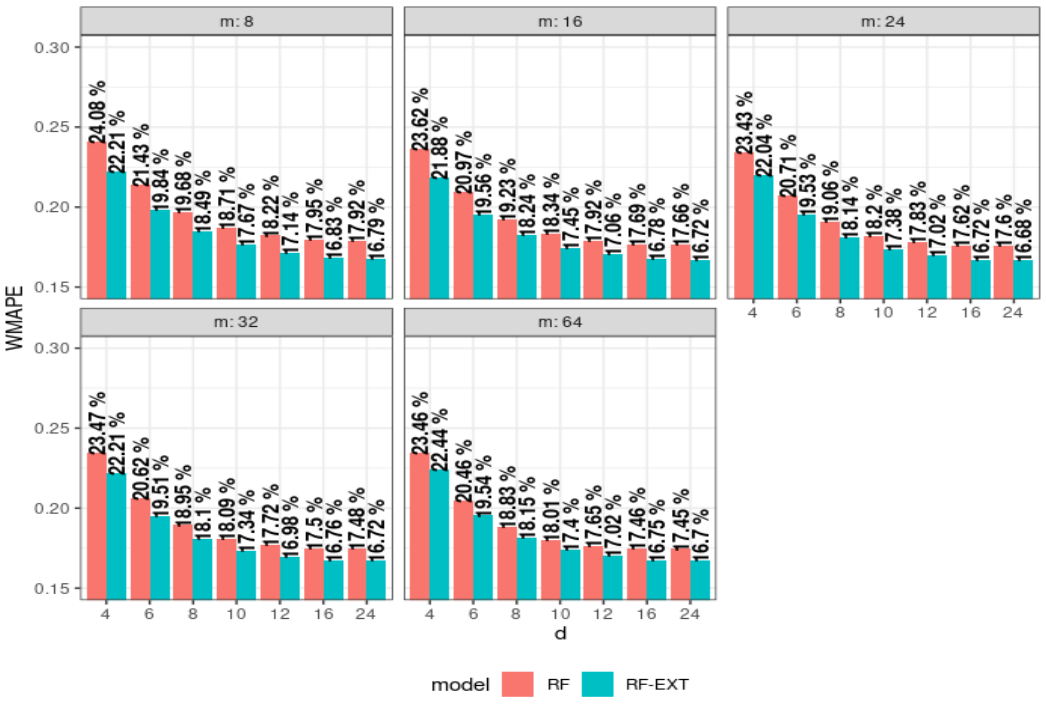


Figure 5.7. RF vs RF-EXT performance comparison for Dataset 2.

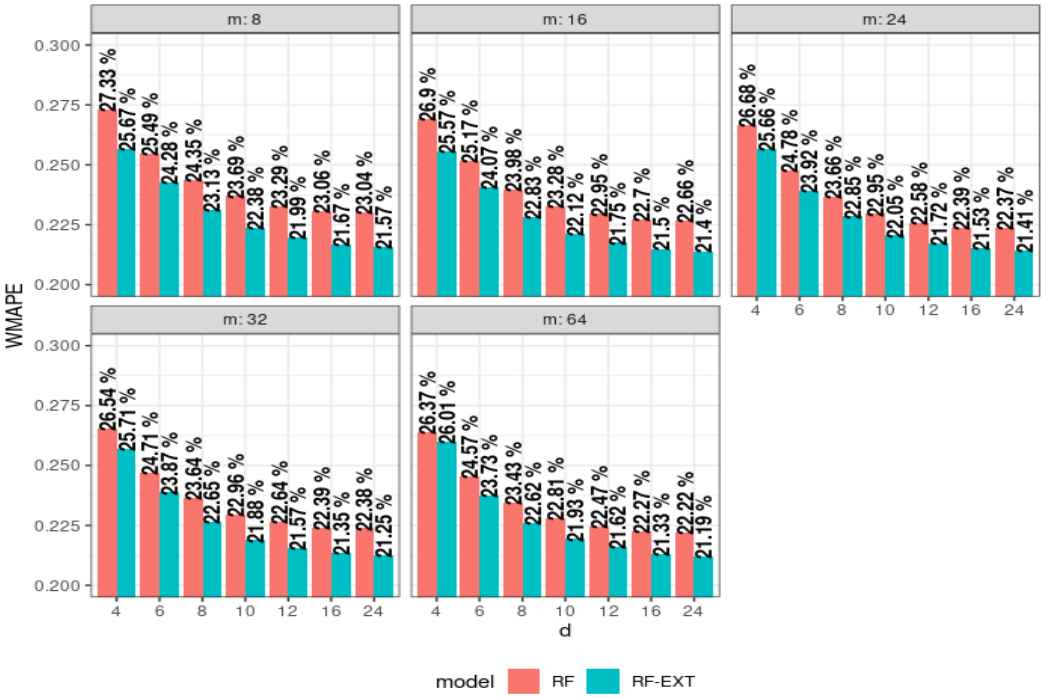


Figure 5.8. RF vs RF-EXT performance comparison for Dataset 3.

Table 5.1. Parameter settings of the best RF and RF-EXT.

Input	Model	d	m	WMAPE
Dataset 1	**RF-EXT**	16	64	16.65%
	RF	16	64	17.01%
Dataset 2	**RF-EXT**	24	24	16.68%
	RF	24	64	17.45%
Dataset 3	**RF-EXT**	24	64	21.19%
	RF	24	64	22.22%

Table 5.2. Wind power quantile intervals of the datasets.

Group	**Wind Power Intervals in MWH**		
	Dataset 1	**Dataset 2**	**Dataset 3**
1	[0.019, 2.140)	[0, 44.640)	[0, 17.550)
2	[2.140, 5.500)	[44.640, 141.952)	[17.550, 56.444)
3	[5.500, 10.217)	[141.952, 306.392)	[56.444, 137.780)
4	[10.217, 16.364)	[306.392, 487.670)	[137.780, 225.600)
5	[16.364, 25.299]	[487.670, 645.870]	[225.600, 295.940]

Table 5.3. RF vs RF-EXT performance comparison for wind power intervals.

Input	**Group**	**RF**		**RF-EXT**	
		BIAS	**WMAPE**	**BIAS**	**WMAPE**
Dataset 1	1	-69.66%	81.71%	-43.33%	67.29%
	2	-12.82%	33.17%	-7.13%	33.23%
	3	-0.96%	20.48%	0.42%	20.84%
	4	3.79%	16.42%	3.18%	17.04%
	5	7.34%	9.49%	5.97%	8.91%
Dataset 2	1	-70.60%	90.31%	-28.13%	65.03%
	2	-18.24%	42.88%	-3.93%	40.25%
	3	-3.41%	27.02%	-0.86%	28.58%
	4	6.13%	15.76%	4.44%	16.22%
	5	7.12%	8.42%	4.78%	6.99%
Dataset 3	1	-165.47%	170.22%	-98.72%	120.04%
	2	-20.91%	52.01%	-3.97%	49.27%
	3	4.06%	33.33%	7.83%	35.59%
	4	8.70%	18.80%	7.30%	19.09%
	5	10.21%	12.25%	7.92%	10.85%

5.3.3. RF-LASSO Comparisons

In this section, the performances of RF-LASSO model are compared with other tree-ensembling algorithms RF and RF-EXT. The detailed comparison of RF and RF-EXT models is conducted in the previous section. As a result, it is observed that RF-EXT algorithm outperforms RF. As the method proposed in this thesis, RF-LASSO model combines oblique decision trees instead of univariate ones unlike the other two algorithms,. Additionally, RF-LASSO uses median operation like RF-EXT in aggregation phase. Moreover, temporal weighting of the observations is applied in the bagging phase like RF-EXT does. For this reason, RF-EXT results are reported as a benchmark in detailed analysis. In Tables 5.4, 5.5, and 5.6; RF-LASSO performances for all parameter settings for the three datasets are reported in WMAPE, respectively. At the same time, the best performances of RF and RF-EXT models at the corresponding tree depth are also included in these tables. In Table 5.7, together with RF-EXT, the best RF-LASSO parameter setting and related WMAPE performance in each dataset are reported.

RF-LASSO outperforms RF and RF-EXT at the same depth in all datasets. Because RF-LASSO is a multivariate tree ensembling method, it achieves better performances at less depths as compared to univariate alternatives. Also, the best RF-LASSO model in all three datasets outperforms the best RF-EXT version. While this difference is minimal for Dataset 3, it is significant for Dataset 1. Since RF-LASSO algorithm is coded from scratch, it does not yet have an efficient implementation like other random forest algorithms. Therefore, experiments on deeper trees could not be performed as the time complexity changes with depth. The situation is discussed in more detail in the discussion section. In addition, Table 5.8 shows the performances of RF-LASSO and RF-EXT in each wind power range. As observed, BIAS of RF-LASSO model is closer to 0 compared to BIAS of RF-EXT, especially in the low wind power ranges.

Table 5.4. RF-LASSO performance results for Dataset 1.

RF-LASSO		***d***			
m	***r***	4	6	8	10
p	1	16.72%	16.61%	16.60%	16.61%
4	16	17.71%	17.11%	16.82%	16.78%
4	24	17.72%	17.03%	16.72%	16.61%
4	32	17.58%	16.96%	16.66%	16.59%
8	16	17.40%	16.73%	16.55%	16.52%
8	24	17.38%	16.76%	16.49%	16.46%
8	32	17.32%	16.69%	16.44%	16.43%
16	1	17.35%	16.88%	16.83%	16.88%
16	16	16.98%	16.46%	16.39%	16.35%
16	32	16.93%	16.44%	16.33%	16.28%
24	1	17.00%	16.67%	16.64%	16.64%
32	1	16.83%	16.58%	16.55%	16.55%
32	16	16.59%	16.36%	16.21%	16.21%
32	32	16.56%	16.22%	16.17%	16.15%
RF		19.40%	18.13%	17.55%	17.18%
RF-EXT		18.63%	17.66%	17.11%	16.84%

Table 5.5. RF-LASSO performance results for Dataset 2.

RF-LASSO		***d***			
m	***r***	4	6	8	10
p	1	17.65%	17.44%	17.47%	17.45%
4	16	19.81%	18.03%	17.39%	17.12%
4	24	19.66%	18.01%	17.30%	17.05%
4	32	19.61%	18.00%	17.23%	16.95%
8	16	18.60%	17.41%	16.96%	16.85%
8	24	18.51%	17.36%	16.82%	16.74%
8	32	18.46%	17.34%	16.88%	16.64%
16	1	18.39%	17.61%	17.50%	17.48%
16	16	17.79%	16.87%	16.69%	16.68%
16	32	17.72%	16.82%	16.53%	16.54%
24	1	17.86%	17.29%	17.22%	17.25%
32	1	17.55%	17.07%	17.06%	17.09%
32	16	17.13%	16.60%	16.49%	16.48%
32	32	17.10%	16.53%	16.42%	16.44%
RF		23.43%	20.46%	18.83%	18.01%
RF-EXT		21.88%	19.51%	18.10%	17.34%

Table 5.6. RF-LASSO performance results for Dataset 3.

RF-LASSO		***d***			
m	***r***	4	6	8	10
p	1	22.47%	22.38%	22.40%	22.34%
4	16	24.26%	22.67%	22.07%	21.89%
4	24	24.05%	22.53%	22.00%	21.64%
4	32	23.95%	22.32%	21.70%	21.70%
8	16	23.13%	21.93%	21.67%	21.54%
8	24	23.08%	21.86%	21.52%	21.44%
8	32	23.00%	21.77%	21.41%	21.28%
16	1	23.10%	22.32%	22.31%	22.29%
16	16	22.21%	21.55%	21.29%	21.42%
16	32	22.13%	21.36%	21.13%	21.21%
24	1	22.45%	22.03%	21.93%	21.91%
32	1	22.22%	21.81%	21.78%	21.75%
32	16	21.70%	21.26%	21.26%	21.17%
32	32	21.57%	21.24%	21.13%	21.16%
RF		26.37%	24.57%	23.43%	22.81%
RF-EXT		25.57%	23.73%	22.62%	21.88%

Table 5.7. Parameter settings of the best RF-LASSO and RF-EXT.

Input	**Model**	d	m	r	**WMAPE**
Dataset 1	**RF-EXT**	16	64	-	16.65%
	RF-LASSO	10	32	32	16.15%
Dataset 2	**RF-EXT**	24	24	-	16.68%
	RF-LASSO	8	32	32	16.42%
Dataset 3	**RF-EXT**	24	64	-	21.19%
	RF-LASSO	8	16	32	21.13%

Table 5.8. RF-LASSO performance comparison for wind power intervals.

Input	**Group**	**RF-EXT**		**RF-LASSO**	
		BIAS	**WMAPE**	**BIAS**	**WMAPE**
Dataset 1	1	-43.33%	67.29%	-40.34%	63.58%
	2	-7.13%	33.23%	-5.01%	31.70%
	3	0.42%	20.84%	1.06%	20.61%
	4	3.18%	17.04%	2.57%	16.21%
	5	5.97%	8.91%	6.03%	8.87%
Dataset 2	1	-28.13%	65.03%	-21.43%	62.15%
	2	-3.93%	40.25%	0.70%	39.73%
	3	-0.86%	28.58%	0.38%	28.47%
	4	4.44%	16.22%	3.85%	15.99%
	5	4.78%	6.99%	4.24%	6.79%
Dataset 3	1	-98.72%	120.04%	-88.55%	107.67%
	2	-3.97%	49.27%	-0.17%	50.56%
	3	7.83%	35.59%	7.68%	37.65%
	4	7.30%	19.09%	6.27%	19.01%
	5	7.92%	10.85%	6.92%	10.27%

5.3.4. Model Performances with Best-Performing Parameters

In this section, the performances of the models for the parameter setting in which they show the best performance are summarized. Firstly, the best performances are obtained with $P^3([X_{t-2}, X_{t-1}, X_t, X_{t+1}, X_{t+2}])$ input data for all models. Table 5.9 reports the results of the models in all datasets in WMAPE and MAE.

Table 5.9. Performance summary with best performing parameters.

Input	Model	WMAPE	MAE
Dataset1	**DT**	20.88%	1.915
	DT-LASSO	17.24%	1.580
	LASSO	17.56%	1.610
	RF	17.01%	1.559
	RF-EXT	16.65%	1.527
	RF-LASSO	16.15%	1.480
Dataset2	**DT**	22.19%	56.985
	DT-LASSO	17.69%	45.445
	LASSO	19.38%	49.772
	RF	17.45%	44.819
	RF-EXT	16.68%	42.857
	RF-LASSO	16.42%	42.189
Dataset3	**DT**	26.36%	30.603
	DT-LASSO	22.64%	26.276
	LASSO	23.32%	27.070
	RF	22.22%	25.793
	RF-EXT	21.19%	24.597
	RF-LASSO	21.13%	24.533

RF-LASSO stands out as the model with the best performance in all three datasets. Additionally, RF-EXT model performs better than RF model. All tree

ensembling methods give more successful results compared to other alternatives such as DT, LASSO and DT-LASSO. Although DT-LASSO model grows a single tree, it performs close to RF. On the other hand, DT shows the worst performance in all datasets according to the relevant metrics.

Table 5.10. Daily WMAPE performances of the best performing models.

Input	Model	Q10	Q25	Q50	Q75	Q90
Dataset1	**DT**	10.62%	15.25%	26.92%	39.62%	66.05%
	DT-LASSO	9.18%	13.32%	20.77%	33.83%	48.44%
	LASSO	8.68%	12.89%	21.05%	34.27%	58.75%
	RF	8.14%	12.32%	21.65%	32.23%	49.92%
	RF-EXT	7.60%	12.29%	21.32%	32.19%	47.00%
	RF-LASSO	7.76%	12.24%	19.99%	31.53%	45.59%
Dataset2	**DT**	9.81%	17.11%	29.56%	48.52%	70.24%
	DT-LASSO	7.09%	13.19%	23.68%	38.41%	53.27%
	LASSO	8.44%	13.63%	24.62%	43.68%	80.42%
	RF	7.35%	12.93%	23.18%	37.92%	55.83%
	RF-EXT	6.60%	12.80%	22.73%	35.64%	50.17%
	RF-LASSO	6.08%	12.22%	21.92%	36.32%	48.70%
Dataset3	**DT**	13.57%	19.39%	38.17%	56.37%	82.76%
	DT-LASSO	11.30%	16.23%	28.11%	47.19%	67.08%
	LASSO	12.49%	17.06%	28.62%	48.44%	84.55%
	RF	11.15%	16.09%	29.32%	44.60%	72.05%
	RF-EXT	9.85%	15.04%	28.45%	46.45%	70.35%
	RF-LASSO	9.13%	15.04%	26.90%	45.52%	64.24%

The results in Table 5.9 are calculated over the whole period. However, it is important to analyze the performance of the models on a daily basis in order to compare their robustness. For this purpose, daily WMAPE values of each model are calculated. Then, the daily WMAPE values in different quantiles are reported. Table 5.10 re-

ports 10%, 25%, 50%, 75% and 90% quantile performances of the models. RF-LASSO outperforms other models for different quantile ranges, especially in 50% and 90%.

5.4. Discussion

RF-LASSO and DT-LASSO outperform their univariate counterparts for the regional wind forecasting problem in the selected datasets. Moreover; median aggregation and temporal bagging, which are proposed as an extension to RF, also show more successful results. These methods can also be applied and tested in different time series regression problems. Considering the lack of research on oblique decision tree-based methods especially in time series regression problems, the successful performance of the proposed models shows that alternative oblique decision tree-based model studies can be made.

Although RF-LASSO gives satisfactory performance, it lags behind RF in terms of running times. This is due to the fact that the time complexity of finding a supervised multivariate split at each node is higher than finding a split based on a single variable. Therefore, improvements in the multivariate split searching directly affect the efficiency of the model. Therefore, suggestions that can be studied as future work are summarized as follows:

- The proper λ value in LASSO regression is determined using k-fold cross validation. As an alternative, performances of different λ options on the out-of-bag sample can be taken as a basis instead of using k-fold cross validation. Thus, multiple model fitting required by cross validation can be avoided.
- In addition, the proper λ value determined in the previous node can be used as the initial point for the next node.
- It is observed that decision trees using oblique split achieve better performances at lower depth values as compared to orthogonal trees. Therefore, oblique splits can be utilized up to a certain depth, and then orthogonal splits can be used. A faster and more accurate convergence can be achieved with a hybrid method.

6. CONCLUSION

This thesis proposes a new oblique tree-based ensembling algorithm RF-LASSO for the regional wind power forecasting task. RF-LASSO ensembles oblique decision trees, namely DT-LASSO which splits data at each node over a linear combination of features found in a supervised manner using LASSO regression. Because there is a lack of research in oblique tree-based algorithms applied in time series regression tasks, the main aim is to compare RF-LASSO with RF that has proven success in wind power forecasting domain in order to validate that oblique tree-based algorithms can be applied for time series regression.

First of all, DT-LASSO algorithm is compared with the conventional decision tree and LASSO regression methods in the experiments. It is observed that while DT-LASSO maintain the success of decision trees in learning nonlinear relationships, it shows more successful results compared to the other two models. Afterwards, the analyzes of median aggregation and temporal bagging, which are presented as an extension to random forest method, are performed. It is observed that the extensions to random forest method significantly improve model performance and better represent local distributions. Finally, the proposed model RF-LASSO is compared with other methods used in the experiments. In these comparisons, RF-LASSO outperforms the others. Moreover, RF-LASSO model shows better results at less depths, hence multiple variables are used in each split. So, RF-LASSO offers a faster convergence in terms of tree depth compared to other univariate alternatives. As a result, it is shown that oblique decision tree-based methods can also produce successful results in time series regression problems.

Although RF-LASSO produces successful results in regional wind power estimation, it can also be applied to different time series regression problems. Additionally, the time efficiency of RF-LASSO can be improved with the enhancements in the optimization problem solved for multivariate split search. In the light of this thesis,

alternative oblique decision tree-based methods that can be applied in time series regression problems, together with the improvements that can be made on RF-LASSO model, can be suggested as possible future works.

REFERENCES

1. Hanifi, S., X. Liu, Z. Lin and S. Lotfian, "A Critical Review of Wind Power Forecasting Methods—Past, Present and Future", *Energies*, Vol. 13, No. 15, p. 3764, 2020.

2. De Giorgi, M. G., A. Ficarella and M. Tarantino, "Assessment of the Benefits of Numerical Weather Predictions in Wind Power Forecasting based on Statistical Methods", *Energy*, Vol. 36, No. 7, pp. 3968–3978, 2011.

3. Kusiak, A., H. Zheng and Z. Song, "On-line Monitoring of Power Curves", *Renewable Energy*, Vol. 34, No. 6, pp. 1487–1493, 2009.

4. Lydia, M., S. S. Kumar, A. I. Selvakumar and G. E. Prem Kumar, "A Comprehensive Review on Wind Turbine Power Curve Modeling Techniques", *Renewable and Sustainable Energy Reviews*, Vol. 30, No. C, pp. 452–460, 2014.

5. Staffell, I., "Wind Turbine Power Curves", `https://www.academia.edu/1489838/Wind_Turbine_Power_Curves`, accessed on 4 Jul, 2022.

6. Manwell, J. F., J. G. McGowan and A. L. Rogers, *Wind Energy Explained: Theory, Design and Application*, John Wiley & Sons, Chichester, 2010.

7. Monteiro, C., H. Keko, R. Bessa, V. Miranda, A. Botterud, J. Wang and Conzelmann, *A Quick Guide to Wind Power Forecasting : State-Of-The-Art 2009*, Tech. rep., United States, 2009.

8. Wang, Z., W. Wang and B. Wang, "Regional Wind Power Forecasting Model with NWP Grid Data Optimized", *Frontiers in Energy*, Vol. 11, No. 2, pp. 175–183, 2017.

9. Andrade, J. R. and R. J. Bessa, "Improving Renewable Energy Forecasting With

a Grid of Numerical Weather Predictions", *IEEE Transactions on Sustainable Energy*, Vol. 8, No. 4, pp. 1571–1580, 2017.

10. Foley, A. M., P. G. Leahy, A. Marvuglia and E. J. McKeogh, "Current Methods and Advances in Forecasting of Wind Power Generation", *Renewable Energy*, Vol. 37, No. 1, pp. 1–8, 2012.

11. Collins, S. N., R. S. James, P. Ray, K. Chen, A. Lassman and J. Brownlee, "Grids in Numerical Weather and Climate Models", *Climate Change and Regional/Local Responses*, IntechOpen, Rijeka, 2013.

12. Lahouar, A. and J. Ben Hadj Slama, "Hour Ahead Wind Power Forecast based on Random Forests", *Renewable Energy*, Vol. 109, No. C, pp. 529–541, 2017.

13. Fischer, A., L. Montuelle, M. Mougeot and D. Picard, "Statistical Learning for Wind Power: A Modeling and Stability Study Towards Forecasting", *Wind Energy*, Vol. 20, No. 12, pp. 2037–2047, 2017.

14. Lin, Y., U. Kruger, J. Zhang, Q. Wang, L. Lamont and L. E. Chaar, "Seasonal Analysis and Prediction of Wind Energy Using Random Forests and ARX Model Structures", *IEEE Transactions on Control Systems Technology*, Vol. 23, No. 5, pp. 1994–2002, 2015.

15. Fugon, L., J. Juban and G. Kariniotakis, "Data Mining for Wind Power Forecasting", *European Wind Energy Conference & Exhibition EWEC 2008*, p. 6, Brussels, Belgium, 2008.

16. Breiman, L., "Random Forests", *Machine Learning*, Vol. 45, No. 1, pp. 5–32, 2001.

17. Brodley, C. E. and P. E. Utgoff, "Multivariate Decision Trees", *Machine Learning*, Vol. 19, No. 1, pp. 45–77, 2004.

18. Murthy, S. K., S. Kasif and S. Salzberg, "A System for Induction of Oblique

Decision Trees", *Journal of Artificial Intelligence Research*, Vol. 2, No. 1, pp. 1–32, 1994.

19. Chaturvedi, S. and S. Patil, "Oblique Decision Tree Learning Approaches - A Critical Review", *International Journal of Computer Applications*, Vol. 82, No. 13, pp. 6–10, 2013.

20. Wickramarachchi, D., B. Robertson, M. Reale, C. Price and J. Brown, "HHCART: An Oblique Decision Tree", *Computational Statistics & Data Analysis*, Vol. 96, pp. 12–23, 2016.

21. Menze, B. H., B. M. Kelm, D. N. Splitthoff, U. Koethe and F. A. Hamprecht, "On Oblique Random Forests", *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II*, p. 453–469, Springer-Verlag, Berlin, Heidelberg, 2011.

22. Qiu, X., L. Zhang, P. Nagaratnam Suganthan and G. A. Amaratunga, "Oblique Random Forest Ensemble via Least Square Estimation for Time Series Forecasting", *Information Sciences*, Vol. 420, No. C, pp. 249–262, 2017.

23. Tibshirani, R., "Regression Shrinkage and Selection via the Lasso", *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 58, No. 1, pp. 267–288, 1996.

24. James, G., D. Witten, T. Hastie and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*, No. 103 in Springer Texts in Statistics, Springer, New York, 2013.

25. Muthukrishnan, R. and R. Rohini, "LASSO: A Feature Selection Technique in Predictive Modeling for Machine Learning", *2016 IEEE International Conference on Advances in Computer Applications*, pp. 18–20, 2016.

26. Dormann, C., J. Elith, S. Bacher, C. Buchmann, G. Carl, G. Carré, T. Diekötter,

J. García Márquez, B. Gruber, B. Lafourcade, P. Leitão, T. Münkemüller, C. Mcclean, P. Osborne, B. Reineking, B. Schröder, A. Skidmore, D. Zurell and S. Lautenbach, "Collinearity: A Review of Methods To Deal With It and A Simulation Study Evaluating Their Performance", *Ecography*, Vol. 36, No. 1, pp. 27–46, 2013.

27. Chan, J. Y.-L., S. M. H. Leow, K. T. Bea, W. K. Cheng, S. W. Phoong, Z.-W. Hong and Y.-L. Chen, "Mitigating the Multicollinearity Problem and Its Machine Learning Approach: A Review", *Mathematics*, Vol. 10, No. 8, p. 1283, 2022.

28. Breiman, L., J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees*, Wadsworth and Brooks, 1984.

29. Heath, D., S. Kasif and S. Salzberg, "Induction of Oblique Decision Trees", *Journal of Artificial Intelligence Research*, Vol. 2, No. 2, pp. 1–32, 1993.

30. Deng, H., M. G. Baydogan and G. Runger, "SMT: Sparse Multivariate Tree", *Statistical Analysis and Data Mining*, Vol. 7, No. 1, pp. 53–69, 2014.

31. Brodley, C. E. and P. E. Utgoff, *Multivariate Versus Univariate Decision Trees*, Tech. rep., United States, 1992.

32. Yang, B. B., S. Q. Shen and W. Gao, "Weighted Oblique Decision Trees", *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, No. 01, pp. 5621–5627, 2019.

33. Norouzi, M., M. D. Collins, D. J. Fleet and P. Kohli, "CO2 Forest: Improved Random Forest by Continuous Optimization of Oblique Splits", arXiv:1506.06155 [cs], 2015.

34. Loh, W.-Y. and N. Vanichsetakul, "Tree-Structured Classification via Generalized Discriminant Analysis.", *Journal of the American Statistical Association*, Vol. 83, No. 403, pp. 715–725, 1988.

35. Rodriguez, J., L. Kuncheva and C. Alonso, "Rotation Forest: A New Classifier Ensemble Method", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 10, pp. 1619–1630, 2006.

36. Tan, P. J. and D. L. Dowe, "Decision Forests with Oblique Decision Trees", *Proceedings of the 5th Mexican International Conference on Artificial Intelligence*, p. 593–603, Springer-Verlag, Berlin, Heidelberg, 2006.

37. Katuwal, R., P. Suganthan and L. Zhang, "Heterogeneous Oblique Random Forest", *Pattern Recognition*, Vol. 99, No. C, p. 107078, 2020.

38. Perry, R., A. Li, C. Huynh, T. M. Tomita, R. Mehta, J. Arroyo, J. Patsolic, B. Falk and J. T. Vogelstein, "Manifold Oblique Random Forests: Towards Closing the Gap on Convolutional Deep Networks", arXiv:1909.11799 [cs], 2019.

39. Correia, A. J. L. and W. R. Schwartz, "Oblique Random Forest Based On Partial Least Squares Applied to Pedestrian Detection", *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 2931–2935, 2016.

40. Tomita, T. M., M. Maggioni and J. T. Vogelstein, "ROFLMAO: Robust Oblique Forests with Linear MAtrix Operations", *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 498–506, Society for Industrial and Applied Mathematics, 2017.

41. Tomita, T. M., J. Browne, C. Shen, J. Chung, J. Patsolic, B. Falk, C. E. Priebe, J. Yim, R. C. Burns, M. Maggioni and J. T. Vogelstein, "Sparse Projection Oblique Randomer Forests", *Journal of Machine Learning Research*, Vol. 21, No. 104, pp. 104:1–104:39, 2020.

42. Wu, B., M. Song, K. Chen, Z. He and X. Zhang, "Wind Power Prediction System for Wind Farm Based On Auto Regressive Statistical Model and Physical Model", *Journal of Renewable and Sustainable Energy*, Vol. 6, No. 1, p. 013101, 2014.

43. Hodge, B.-M., A. Zeiler, D. Brooks, G. Blau, J. Pekny and G. Reklatis, "Improved Wind Power Forecasting with ARIMA Models", E. Pistikopoulos, M. Georgiadis and A. Kokossis (Editors), *21st European Symposium on Computer Aided Process Engineering*, Vol. 29 of *Computer Aided Chemical Engineering*, pp. 1789–1793, Elsevier, 2011.

44. Chen, P., T. Pedersen, B. Bak-Jensen and Z. Chen, "ARIMA-Based Time Series Model of Stochastic Wind Power Generation", *Power Systems, IEEE Transactions on*, Vol. 25, No. 2, pp. 667 – 676, 2010.

45. Singh, S., T. S. Bhatti and D. P. Kothari, "Wind Power Estimation Using Artificial Neural Network", *Journal of Energy Engineering*, Vol. 133, No. 1, pp. 46–52, 2007.

46. Hong, Y. Y. and C. L. P. P. Rioflorido, "A Hybrid Deep Learning-based Neural Network for 24 Hour Ahead Wind Power Forecasting", *Applied Energy*, Vol. 250, No. C, pp. 530–539, 2019.

47. Zhang, J., J. Yan, D. Infield, Y. Liu and F. sang Lien, "Short-Term Forecasting and Uncertainty Analysis of Wind Turbine Power Based On Long Short-Term Memory Network and Gaussian Mixture Model", *Applied Energy*, Vol. 241, No. C, pp. 229–244, 2019.

48. Sun, Z., H. Sun and J. Zhang, "Multistep Wind Speed and Wind Power Prediction Based on a Predictive Deep Belief Network and an Optimized Random Forest", *Mathematical Problems in Engineering*, Vol. 2018, No. 4, pp. 1–15, 2018.

49. Shi, K., Y. Qiao, W. Zhao, Q. Wang, M. Liu and Z. Lu, "An Improved Random Forest Model of Short-Term Wind Power Forecasting to Enhance Accuracy, Efficiency, and Robustness", *Wind Energy*, Vol. 21, No. 12, pp. 1383–1394, 2018.

50. Hao, J., C. Zhu and X. Guo, "Wind Power Short-Term Forecasting Model Based on the Hierarchical Output Power and Poisson Re-Sampling Random Forest Algo-

rithm", *IEEE Access*, Vol. 9, pp. 6478–6487, 2021.

51. Lee, J., W. Wang, F. Harrou and Y. Sun, "Wind Power Prediction Using Ensemble Learning-Based Models", *IEEE Access*, Vol. 8, pp. 61517–61527, 2020.

52. Athey, S., J. Tibshirani and S. Wager, "Generalized Random Forests", *Annals of Statistics*, Vol. 47, No. 2, pp. 1179–1203, 2019.

53. Gautam, A. and V. Singh, "Parametric Versus Non-Parametric Time Series Forecasting Methods: A Review", *Journal of Engineering Science and Technology Review*, Vol. 13, No. 3, pp. 165–171, 2020.

54. Mahmoud, H. F. F., "Parametric Versus Semi and Nonparametric Regression Models", *International Journal of Statistics and Probability*, Vol. 10, No. 2, pp. 1–90, 2021.

55. Howell, J., U. S. N. Aeronautics, S. Administration and L. R. Center, *A Least-square Distance Curve-fitting Technique*, NASA Technical Note, National Aeronautics and Space Administration, 1971.

56. Rifkin, Ryan M. and Lippert, Ross A., "Notes on Regularized Least Squares", `https://dspace.mit.edu/handle/1721.1/37318`, accessed on 20 Jun, 2022.

57. Miller, S. J., *Chapter 24. The Method of Least Squares*, pp. 625–635, Princeton University Press, 2017.

58. Arnold, T. B. and R. J. Tibshirani, *genlasso: Path Algorithm for Generalized Lasso Problems*, 2019, `https://CRAN.R-project.org/package=genlasso`, r package version 1.4.

59. Arnold, T. B. and R. J. Tibshirani, "Introduction to the genlasso Package", `https://mran.microsoft.com/snapshot/2015-12-09/web/packages/genlasso/vignettes/`, accessed on 20 Jun, 2022.

60. Tibshirani, R. J. and J. E. Taylor, "The Solution Path of the Generalized Lasso", *Annals of Statistics*, Vol. 39, pp. 1335–1371, 2011.

61. Yohannes, Y. and J. Hoddinott, "Classification and Regression Trees: An Introduction", `https://pdf.usaid.gov/pdf_docs/Pnach725.pdf`, accessed on 20 Jun, 2022.

62. Timofeev, R. A., *Classification and Regression Trees Theory and Applications*, Master's Thesis, Humboldt University, 2004.

63. Rokach, L. and O. Maimon, "Decision Trees", *The Data Mining and Knowledge Discovery Handbook*, Vol. 6, pp. 165–192, 2005.

64. Czajkowski, M. and M. Kretowski, "The Role of Decision Tree Representation in Regression Problems – An Evolutionary Perspective", *Applied Soft Computing*, Vol. 48, No. C, pp. 458–475, 2016.

65. Sheth, N. S. and A. R. Deshpande, "A Review of Splitting Criteria for Decision Tree Induction", *Fuzzy Systems*, Vol. 7, No. 1, pp. 1–4, 2015.

66. Alkhalid, A., I. Chikalov and M. Moshkov, "Comparison of Greedy Algorithms for Decision Tree Construction", *KDIR 2011 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, pp. 438–443, 2011.

67. Murthy, S. and S. Salzberg, "Decision Tree Induction: How Effective is the Greedy Heuristic?", *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, p. 222–227, Association for the Advancement of Artificial Intelligence Press, 1995.

68. Breiman, L., "Bagging Predictors", *Machine Learning*, Vol. 24, No. 2, pp. 123–140, 2004.

69. Pu, Z. and E. Kalnay, "Numerical Weather Prediction Basics: Models, Numeri-

cal Methods, and Data Assimilation", *Handbook of Hydrometeorological Ensemble Forecasting*, pp. 1–31, Springer Berlin Heidelberg, Berlin, Heidelberg, 2018.

70. Stull, R. B., *Practical Meteorology: An Algebra-Based Survey of Atmospheric Science*, UBC Press, Vancouver, 2017.

71. National Centers for Environmental Prediction, "NCEP GFS 0.25 Degree Global Forecast Grids Historical Archive", `https://rda.ucar.edu/datasets/ds084.1/`, accessed on 20 Jun, 2022.

72. National Centers for Environmental Prediction, "Data Products GFS ad GDAS", `https://www.nco.ncep.noaa.gov/pmb/products/gfs/`, accessed on 20 Jun, 2022.

73. Gao, W., T. Friedrich, F. Neumann and C. Hercher, "Randomized Greedy Algorithms for Covering Problems", *Proceedings of the Genetic and Evolutionary Computation Conference*, p. 309–315, Association for Computing Machinery, 2018.

74. Friedman, J., T. Hastie and R. Tibshirani, "Regularization Paths for Generalized Linear Models via Coordinate Descent", *Journal of Statistical Software*, Vol. 33, No. 1, pp. 1–22, 2010.

75. "Elektrik Dağıtım Şirketleri", `https://www.enerjiatlasi.com/elektrik-dagitim-sirketleri/`, accessed on 5 Jul, 2022.

76. R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2020, `https://www.R-project.org/`.

77. Therneau, T. and B. Atkinson, *rpart: Recursive Partitioning and Regression Trees*, 2019, `https://CRAN.R-project.org/package=rpart`, r package version 4.1-15.

78. Wright, M. N. and A. Ziegler, "ranger: A Fast Implementation of Random Forests

for High Dimensional Data in C++ and R", *Journal of Statistical Software*, Vol. 77, No. 1, pp. 1–17, 2017.

79. Nelder, J. A. and R. W. M. Wedderburn, "Generalized Linear Models", *Journal of the Royal Statistical Society. Series A (General)*, Vol. 135, No. 3, p. 370, 1972.

80. Müller, M., *XploRe — Learning Guide*, pp. 205–228, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

81. Dobson, A. and A. Barnett, *An Introduction to Generalized Linear Models, Third Edition*, 2008.

82. Hardin, J. and J. Hilbe, *Generalized Linear Models and Extensions, 4th Edition*, 2018.

83. Dunn, P. K. and G. K. Smyth, "Chapter 9: Models for Proportions: Binomial GLMs", *Generalized Linear Models With Examples in R*, pp. 333–369, Springer New York, 2018.

84. Chen, K., Y. Cheng, O. Berkout and O. Lindhiem, "Analyzing Proportion Scores as Outcomes for Prevention Trials: a Statistical Primer", *Prevention Science*, Vol. 18, No. 3, 2016.

85. Manikandan, S., "Measures of Central Tendency: The Mean", *Journal of Pharmacology and Pharmacotherapeutics*, Vol. 2, No. 2, pp. 140–142, 2011.

86. Manikandan, S., "Measures of Central Tendency: Median and Mode", *Journal of Pharmacology and Pharmacotherapeutics*, Vol. 2, No. 3, pp. 214–215, 2011.

87. Meinshausen, N., "Quantile Regression Forests", *Journal of Machine Learning Research*, Vol. 7, No. 35, pp. 983–999, 2006.

88. Tüysüzoğlu, G., D. Birant and V. Kıranoğlu, "Temporal Bagging: A New Method

for Time-Based Ensemble Learning", *Turkish Journal of Electrical Engineering and Computer Sciences*, Vol. 30, No. 1, p. 279 – 294, 2022.

89. "Gerçek Zamanlı Üretim - Energy Exchange Istanbul", `https://seffaflik.epias.com.tr/transparency/uretim/gerceklesen-uretim/gercek-zamanli-uretim.xhtml`, accessed on 07 Jul, 2022.

90. Tabak, B., "RF-LASSO", `https://github.com/Burakt94/RF-LASSO`, accessed on 01 Aug, 2022.