MASTER IN COMPUTER SCIENCE PROFESSIONAL FOCUS IN SOFTWARE ENGINEERING

Short-term time series forecasting of the electricity consumption in Spain using an Evolutionary Algorithm and an Ensemble Method

Gilson, Aude

Award date:
2018

Awarding institution:
University of Namur

Université de Namur
Faculty of Computer Science
Academic Year 2017–2018

Short-term time series forecasting of the electricity consumption in Spain using an Evolutionary Algorithm and an Ensemble Method

Aude Gilson

Internship mentor: Professor Federico Divina

Supervisor: _____ (Signed for Release Approval - Study Rules art. 40)
Professor Wim Vanhoof

A thesis submitted in the partial fulfillment of the requirements
for the degree of Master of Computer Science at the Université of Namur

Acknowledgements

This thesis is submitted for the Master's Degree in Computer Sciences at the University of Namur. The research described here was conducted under the supervision of Professor Wim Vanhoof from the University of Namur and Professor Federico Divina from the University Pablo de Olavide in Spain between September 2017 and June 2018.

I am very grateful to my Professor and supervisor Wim Vanhoof for making this incredible experience possible as well as for his help, advice, correction and guidance in writing this thesis.

I would also thank my internship mentor, Professor Federico Divina, for his hospitality and his advice during the internship.

A big thank you to the member of the Data Science & Big Data research lab: Ricardo Talavera-Llames, David Gutiérrez-Avilés, Antonio Galicia, Rubén Pérez-Chacón, José Francisco Torres, Antonio Fernández and Francisco Martínez-Álvarez for welcoming me with open arms in their lab for the duration of the internship. This allowed me to live a very beautiful experience at the university Pablo de Olavide and in Sevilla.

Abstract

The ability to predict short-term electricity consumption provides several benefits, both at the economic and environmental level. Indeed, it would allow for an efficient use of resources in order to face the actual demand, reducing the costs associated with the production as well as the emission of CO2.

The aim of this thesis is to propose two methodologies based on an Evolutionary Algorithm for regression trees and an Ensemble Method by Stacking strategy in order to tackle the short-term consumption forecasting problem.

The Ensemble Method uses a Stacking ensemble learning method, where the predictions produced by three bases learning methods (Random forest, Artificial Neural Networkand Evolutionary Algorithm for regression trees) are combined in a generalizer (Gradient Boosted Method) in order to produce final predictions.

The two methods are applied on a dataset reporting the electric consumption in Spain over more than nine years.

Résumé

La capacité de prédire la consommation d'électricité à court terme présente plusieurs avantages, tant sur le plan économique qu'environnemental. En effet, elle permettrait une utilisation efficace des ressources pour faire face à la demande réelle, en réduisant les coûts associés à la production ainsi que l'émission de CO2.

L'objectif de cette thèse est de proposer deux méthodologies basées sur un algorithme évolutif pour les arbres de régression et une méthode ensembliste par stratégie d'empilement afin d'aborder le problème de la prévision de consommation à court terme.

La méthode d'ensemble utilise une méthode d'apprentissage d'ensemble par empilement, où les prédictions produites par trois méthodes d'apprentissage de base (forêt aléatoire, réseau neuronal artificiel et algorithme évolutif pour les arbres de régression) sont combinées dans un généralisateur (méthode du gradient boosté) afin de produire des prédictions finales.

Les deux méthodes sont appliquées sur un ensemble de données rapportant la consommation électrique en Espagne sur plus de neuf ans.

Contents

Introduction

With the apparition of technologies for extracting information from large databases such as Big Data - Data Mining or data analyses, a large number of specialists have taken an interest in their usefulness for predicting the evolution of this data. Indeed, in certain fields such as economy, finance or, in the case of this thesis, electricity consumption, having access to a prediction of the evolution of data can make it possible to avoid overproduction, the loss of money and unnecessary pollution.

a Background and motivation

Electricity has become the motor of human development. It makes it possible to feed a large part of the everyday objects (cooker, heater, light bulb, etc.) that meets the most essential needs of man.

The world's energy demand is increasing day by day. The reasons are various, for example the rapid development of the human population, or the increase of the energy required by buildings and technology applications. It follows that efficient energy management and forecasting energy consumption are important in decision-making for effective energy saving and development in particular places, in order to decrease both the costs associated with the consumption and the environmental impact it has. In fact, energy consumption is important not only at the economic level, but also for the repercussions that it has on the environment. For example, the European Union has recently issued a directive (European Directive 2009/28/EC [1]) that requires that all EU countries will have to adopt a set of minimum energy efficiency requirements. More specifically, it stipulates a 20% reduction in energy consumption by 2020 relative to 1990 levels along with a 20% reduction in CO_2 emissions and 20% of all energy produced by renewable technologies.

b Time series and time series forecasting

In their book, *Introduction to Time Series and Forecasting* [2], P. Brickwell and R. Davis define time series as :"a set of observations x_t, each observation being recorded at a specific time t". In other words, a time series is a sequence of time-ordered observations measured at equal intervals of time. A time series can be discrete (measurements are taken at fixed intervals) or continuous (measurements are recorded continuously over a period of time).

Time series forecasting is the use of a model or a method to predict from the measurements already known the continuation of the time series.

A section of the chapter 1 : State of the art explains these two elements in detail.

c What is electric energy consumption forecasting?

Electric energy consumption forecasting algorithms can provide several improvements to the issues outlined in the section background and motivation. For example, in [3, 4] forecasting is used to assess what fraction of the generated power should be stored locally for later use and what fraction of it can instead be fed to the loads or injected into the network. Generally, forecasting can be divided into three categories, depending on the prediction horizon, i.e., the time scale of the predictions. Short-term load forecasting, characterised by prediction horizons going from one hour up to a week, medium-term load forecasting, with prediction from one month up to a year, and long-term load forecasting, for prediction involving a prediction horizon of more than one year [5]. Predictions are more accurate for short-term load forecasting, and so most of the recent work in this field focus on this kind of prediction horizon [5].

In fact, with reliable and precise prediction of short-term load, schedules can be generated in order to determine the allocation of generation resources, operational limitations, environmental and equipment usage constraints. Knowing the short-term energy demand can also help in ensuring the power system security since accurate load prediction can be used to determine the optimal operational state of power systems. Moreover, the predictions can be of help in preparing the power systems according to the future predicted load state. Precise predictions also have an economic impact, and may improve the reliability of power systems. The reliability of a power system is affected by abrupt variations of the energy demand. Shortage of power supply can be experienced if the demand is underestimated, while resources may be wasted in producing energy if such energy demand is overestimated. From the above observations, we can understand why short-term load forecasting has gained popularity. This thesis focuses on short-term load forecasting.

Basically, there are two main approaches to forecasting energy consumption, conventional methods, such as [6, 7] and, more recently, a method based on Machine Learning. Conventional methods, including statistical analysis [8], smoothing techniques [9] such as the autoregressive integrated moving average (ARIMA) [10] and exponential smoothing [11] and regression-based approaches [12], can achieve satisfactory results when solving linear problems. Machine Learning strategies, in contrast to traditional methods, are also suitable for non-linear cases (like in [13, 14, 15, 16]). Among the Machine Learning strategy approaches, strategies such as Artificial Neural Networks (NN) [17] or Support Vector Machines (SVM) [18] have been successfully exploited to forecast power consumption data, for example, [19, 20, 21].

d Thesis definition

The aim of this thesis is to design, using Machine Learning methods, an algorithm to obtain the best predictions of Spanish electricity consumption based on a large quantity of data. The core of the thesis is the design of a technique to explore trends in electricity consumption using Evolutionary Algorithms and Ensemble Methods to obtain accurate predictions in the short-term.

e Thesis contribution

This thesis proposes a methodology for forecasting electricity consumption. To achieve this goal, two Machine Learning (ML) techniques are used. One is the Evolutionary Algorithms (EAs) [22] for Regression Trees and the other is the ensemble learning [23, 24, 25].

EAs are population-based strategies which are mainly inspired by evolutionary biology in an attempt to use techniques such as inheritance, mutation, selection, and crossover. The goal behind all these techniques is to put enough "pressure" on the population to make it evolve towards a solution, like natural selection. Each individual of the population represents a candidate solution to a given problem.

Regression trees [26] are commonly used in regression-type problems, where we attempt to predict the values of a continuous variable from one or more continuous and/or categorical predictor variables. An advantage of using regression trees is that results can be easier to interpret.

Ensemble learning [27] is a ML paradigm where multiple learners are trained to solve the same problem. In contrast to ordinary ML approaches, which try to learn one hypothesis from training data, Ensemble Methods try to construct a set of hypotheses and combine them. This approach usually yields better results than the use of a single strategy, since it provides better generalizations, i.e. adaptation to unseen cases, better capabilities of escaping from local optima and superior search capabilities. Article *Stacking Ensemble Learning for Short-Term Electricity Consumption Forecasting* [28] is related to the research that led to this thesis and focuses on the Ensemble Method.

These three technologies are explained in more detail in chapter 1 : State of the art.

In order to assess the correctness of our proposal,a dataset regarding the electricity consumption in Spain registered over a period of more than nine years has been used. We use a fixed prediction horizon of four hours while we vary the historical window size i.e, the amount of historical data used in order to make predictions.Results show that an ensemble scheme can achieve better results than single methods, obtaining more precise predictions.

f Thesis organization

This thesis is organised as follows.

The first part presents the technologies used for this thesis as well as a state of the art of what has already been done in terms of electricity consumption prediction as well as the use of Evolutionary Algorithm or Ensemble Method for electricity consumption forecasting.

The second part presents the material (dataset and programming language) used for the thesis and the two prediction methods as well as how they work.

The third part presents the results obtained by the methods and a comparison between them and with other methods.

A conclusion followed by future improvements will then be proposed to the reader.

The code implementing the two methods is present in the appendix.

1 State of the art

This chapter gives the reader the basics of the main technologies used to realize the thesis. Once these technologies have been introduced, examples of their use in case of energy consumption forecasting will be provided.

1.1 Technology

1.1.1 Time series

Most of this section is based on the book *Introduction to Time Series and Forecasting* [2].

As said in the introduction, a time series is a sequence of time-ordered observations measured at equal intervals of time. In a time series consisting of T real value samples $x_1, \dots x_T$, x_i $(1 \leq i \leq T)$ represents the recorded value at time i. We can then define the problem of time series forecasting as the problem of predicting the values of $x_{w+1}, \dots, x_{w+h}$, given the previous $x_1, \dots x_w$ $(w + h \leq T)$ samples, with the objective of minimizing the error between the predicted value $\hat{x}_{w+i}$ and the actual value x_{w+i} $(1 \leq i \leq h)$. Here, we refer to w as the historical window (how many values we consider in order to produce the predictions) and to h as the prediction horizon (how far in the future one aims to predict).

R. Weber, professor at the University of Cambridge, explains in his course on time series [29], that they can be decomposed into 4 elements:

1. Trend - This term refers to the general tendency exhibited by the time series. A time series can present different types of trends, such as linear, logarithmic, exponential power, polynomial, etc.

2. Seasonal effects - This is a pattern of changes that represents periodic fluctuations of constant length. These variations are originated by effects that are stable along with time, magnitude and direction.

3. Cycles[30] - When data show increases and decreases that are not fixed time periods it is a cyclical pattern. These cycles can be observed when the time series has a very large number of measurements (more than two years of measurement).

4. Residual - This component represents the remaining, mostly unexplainable, parts of the time series. It also describes random and irregular influences that, in case of being high enough, can mask the trend and seasonality.

Although the components cycle and seasonal seem quite similar, there is a noticeable difference: the fluctuations are cycle if they do not appear at fixed periods. The fluctuations are seasonal if the period is immutable and associated with an aspect of the calendar [30].

According to the number of variables involved, time series analysis can be divided into univariate and multivariate analysis. In the univariate case, a time series consists of a single observation recorded sequentially. In contrast, in multivariate time series the value of more than one variable is recorded at each time stamp. The interaction among such variables should be taken into account.

There are different techniques that can be applied to the problem of time series forecasting. Such approaches can be roughly divided into two categories, linear and non-linear methods [31, 32].

Linear methods try to model the time series using a linear function. the basic idea is that although the random component of a time series may prevent accurate predictions, the strong correlation between the data suggests that the next observation can be determined by a linear combination of the previous observations. The basic idea is that even if the random component of a time series may prevent one from making any precise predictions, the strong correlation among the data makes it possible to assume that the next observation can be determined by a linear combination of the preceding observations, except for additive noise i.e. noise introduced into a time series to imitate the effect of many random processes that happen in nature and are added to the noise already present.[33].

Non-linear methods are currently in use in the Machine Learning domain. These methods try to extract a model, that can be non-linear, which describes the observed data, and then uses the so obtained model in order to forecast future values of the time series. Machine Learning techniques have gained popularity in the forecasting field, due to the fact that while conventional methods can achieve satisfactory results in linear problems, Machine Learning methods are suitable also for non-linear modelling [34].

Such approaches can be roughly divided into conventional methods and data-driven approaches [35]. Among the conventional methods, two widely used approaches are stochastic [36] and regression-based methods [12], which were very popular before the emergence of Machine Learning methods.

Data-driven approaches, on the other hand, are based on analysing the data without an explicit knowledge of the physical behaviour of the system.

1.1.2 Evolutionary Algorithms

Much of this section is based on the book *Introduction to Evolutionary Computing* [22].

As said earlier, Evolutionary Algorithms (EAs) are population-based strategies that use techniques inspired by evolutionary biology such as inheritance, mutation, selection and crossover. Each individual i of the population represents a candidate solution to a given problem and is assigned a fitness value, which is a measure of the quality of the solution represented by i. Typically EAs start from an initial population consisting of randomly initialised individuals. Each individual is evaluated in order to determine

its fitness value. Then a selection mechanism is used in order to select a number of individuals. Usually the selection is based on the fitness, so that fitter individuals have a higher probabilities of being selected. Selected individuals generate offspring, i.e., new solutions, by means of the application of crossover and mutation operators. This process is repeated over a number of generations or until a good enough solution is found. Figure 1.1 shows the general principle of an EA in the form of a flowchart.

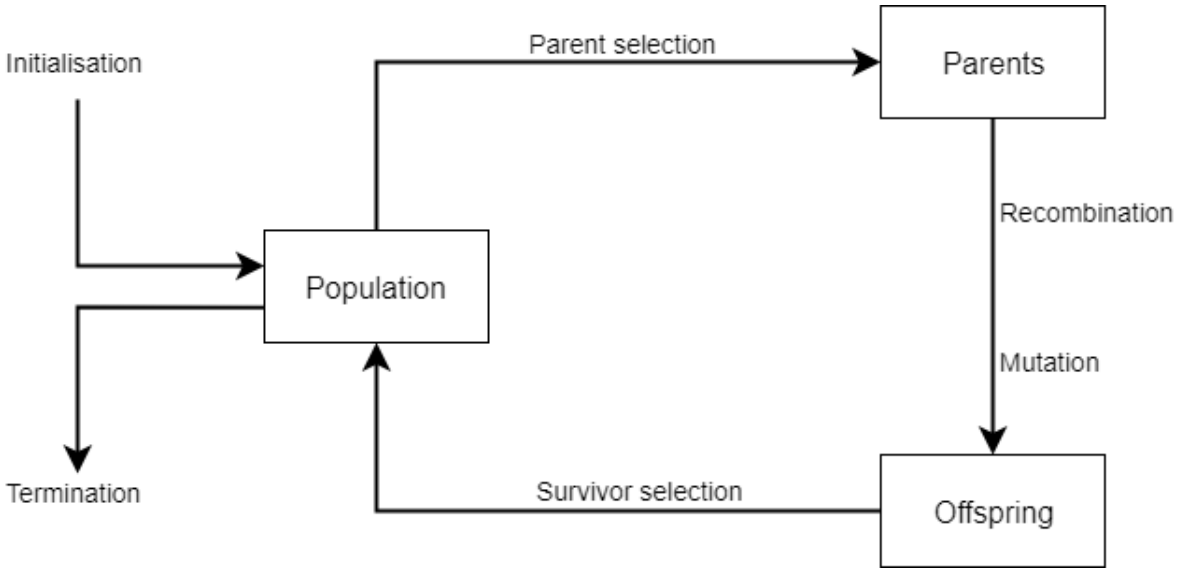


Figure 1.1: Flowchart representation of the general operation of an EA from the book *Introduction to Evolutionary Computing* of A. Eiben and J. Smith [22].

The idea is that better and better solutions will be found at each generation. Moreover, the use of stochastic operators, such as mutation, allows EAs to escape from local optima.

1.1.3 Decision tree

For the problem tackled in this thesis, each individual of the EAs encodes a regression tree. A regression tree is a decision tree [26] similar to a classification tree [37].

Both classification and regression trees aim at modeling a response variable Y by a vector of P predictor variables $X = (X_1, \ldots, X_P)$. The difference is that for classification trees, Y is qualitative and for regression trees Y is quantitative. In both cases X_i can be continuous and/or categorical variables.

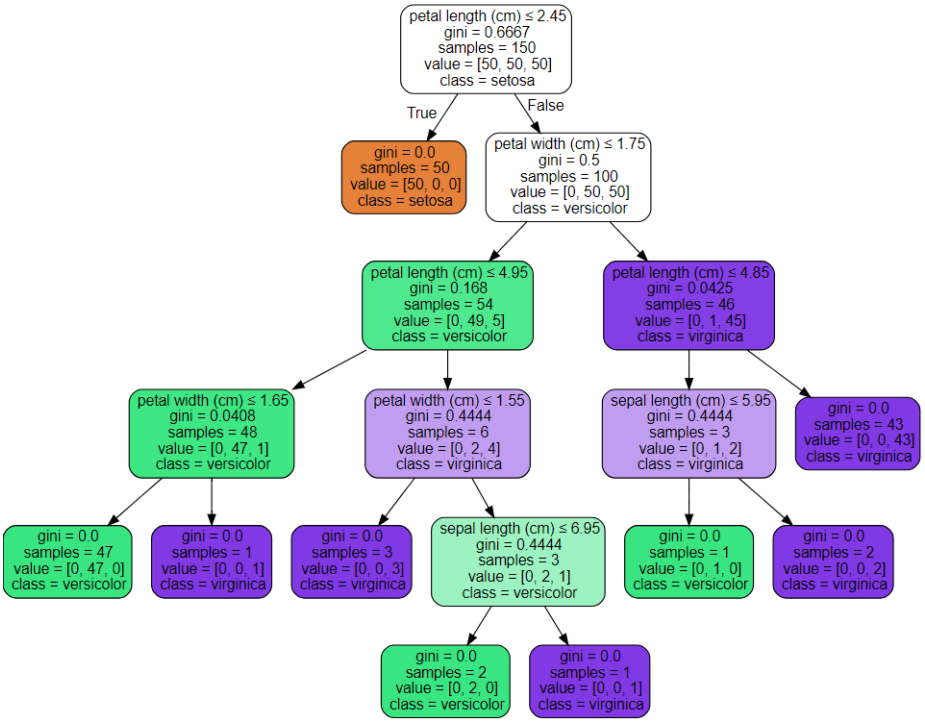


Figure 1.2: Example of a classification tree from the scikit-learn.org website [38].

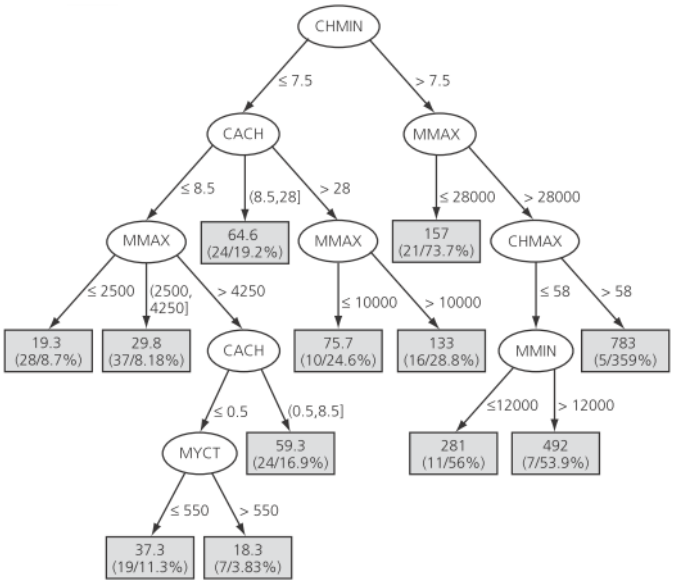


Figure 1.3: Example of a regression tree from the book *DATA MINING : Practical Machine Learning Tools and Techniques* of I. Witten and E. Frank [26].

Figure 1.2 represents a classification tree. This classification tree determines the type of an iris according to the length and width of the sepal as well as the length and width of the petals of the iris. Here, the output of the classification tree is a category i.e. setosa, versicolor or virginica.

Figure 1.3 shows the regression tree that determines the relative performance of computer processing power (CPU) on the basis of a number of relevant attributes. Here, the output of the tree is a value.

Other greedy strategies have been used in order to obtained regression trees, for example [39, 40]. The main challenge of such strategies is that the search space is typically huge, rendering full-grid searches computationally infeasible. Due to their search capabilities, EAs have proven that they can overcome this limitation.

1.1.4 Ensemble Method

Ensemble Method consists in combining different learning models in order to improve the results obtained by each individual model.

The earliest works on ensemble learning were carried out in 90's ([41, 42, 43]), where it was proven that multiple weak learning algorithms could be converted into a strong learning algorithm. In a nutshell, ensemble learning [44, 27] is a procedure where multiple learner modules are applied on a data set to extract multiple predictions.

Such predictions are then combined into one composite prediction. Usually two phases are employed [45]. In a first phase a set of base learners are obtained from training data, while in the second phase the learners obtained in the first phase are combined in order to produce a unified prediction model. Thus, multiple forecasts based on the different base learners are constructed and combined into an enhanced composite model superior to the base individual models. This integration of all good individual models into one improved composite model generally leads to higher accuracy levels.

The most used and well-known of the basic Ensemble Methods are bagging, boosting and stacking [45].

Bagging stands for **B**ootstrap **Agg**regat**ing**. The main objective of boostrapping is to reduce variance. The method consists of a choice of n observations among the original data of size n. Imagine a training set containing 5 observations $x1$, $x2$, $x3$, $x4$ and $x5$. The method will, for example, create a set of $x1$, $x1$, $x3$, $x5$, $x5$ and another set of $x2$, $x2$, $x3$, $x4$, $x5$. Once these different sets are acquired, a classifier is trained on each of these sets. Majority voting or averaging concepts allow the final prediction to be obtained from the various predictions make by the classifiers [46]. Figure 1.4 illustrates how Bagging works.

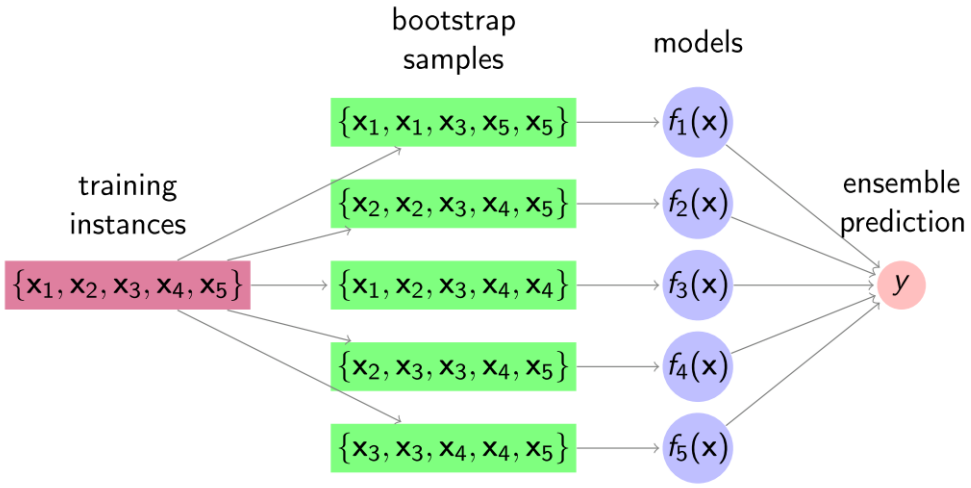


Figure 1.4: illustration of the mechanism of Bagging from the *Machine Learning : from Neurals Networks to Big Data* course given by B. Frénay at the University of Namur[47].

Boosting is similar to bagging, but with one conceptual modification. Instead of assigning equal weighting to models, boosting assigns different weights to classifiers, and derives its ultimate result based on weighted voting. In case of regression a weighted average is usually the final output.

Stacking This method works on two layers. In the first layer, we put as many learning algorithms as we need. Each of these algorithms will work on the data and produce results. These results are then transmitted to the second layer, which contains a combiner algorithm that combines the results to get better results. In summary, this method will build models using different learning algorithms, then a combiner algorithm is formed to make the ultimate predictions using the predictions generated by the basic algorithms. Figure 1.5 shows an abstract view of a staking method.

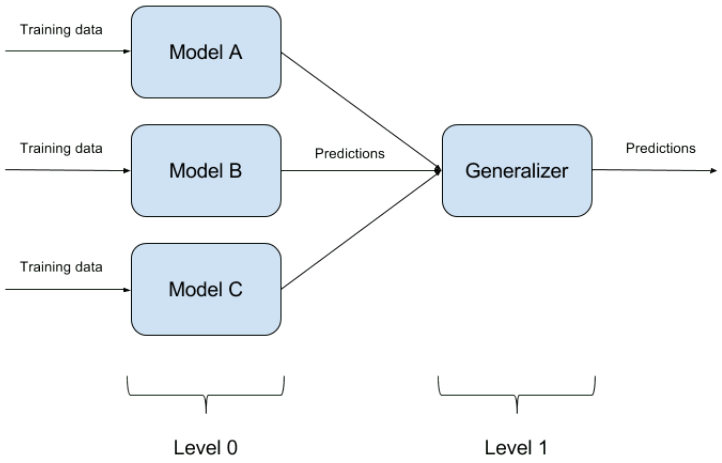


Figure 1.5: An example scheme of Stacking ensemble learning from the article *Stacking Ensemble Learning for Short-Term Electricity Consumption Forecasting* of F. Divina et al. [28].

For this thesis, the Stacking approach has been used. Thus the first method of EAs for regression trees can be incorporated into the Ensemble Method and the results obtained by this method can be improved by being mixed with other methods.

1.2 Related work

In this section, existing work on the electricity consumption forecasting will be presented.

The number of studies addressing the electricity consumption forecasting is increasing due to several reasons, such as gaining knowledge about the demand drivers [48], or comprehending the different energy consumption patterns in order to adopt new policies according to demand response scenarios [49], or, again, measuring the socio-economic and environmental impact of energy production for a more sustainable economy [50].

1.2.1 Electricity consumption forecasting with conventional approach

In the conventional approach, the Auto-Regressive and Moving Average (ARMA) [9] is a very common technique that arises as a mix of the Auto-Regressive (AR) and the Moving Average (MA) models. In [6] Nowicka-Zagrajek and Weron applied the ARMA model to the California power market. In another work, Chujai et al. [51] compared the Auto-Regressive *Integrated* Moving Average (ARIMA) [9] with ARMA on household electric power consumption. The results showed that the ARIMA model performed better than ARMA at forecasting longer periods of time, while ARMA is better at shorter periods of time. The ARIMA methods were applied in [52] by Mohanad et al. to predict short-term electricity demand in Queensland (Australia) market. ARMA is usually applied on stationary stochastic processes [6] while ARIMA on non-stationary cases [53].

Regression based methods are also popular in energy consumption studies. The use of the simple regression model of the ambient temperature was proposed by Schrock and Claridge [54], where the authors investigated a supermarket's electricity use. In later studies, however, the use of multiple regression analysis is preferred, due to the capability to handle more complex models. Lam et al. [55] used such an approach to analyse office buildings in different climates in China. In another work, Braun et al. [56] performed multiple regression analysis on gas and electricity usage in order to study how the change in the climate affects the energy consumption in buildings. In a more recent work Mottahedia et al [57] investigated the suitability of the multiple-linear regression to model the effect of building shape on total energy consumption in two different climate regions.

1.2.2 Electricity consumption forecasting with Machine Learning

A significant part of recent studies in the literature is focussed on time series forecasting using Machine Learning techniques. Among these techniques, Artificial Neural Networks (ANN) have been extensively applied. In an early work presented by Nizami and Al-Garni [58], the authors developed a two-layered fed-forward ANN to analyse the relation between electric energy consumption and weather-related variables. In another work, Kelo and Dudul [59] proposed to use a wavelet Elman neural network to forecast short-term electrical load prediction under the influence of ambient air temperature. In [60] Chitsaz et al. combined the wavelet and ANN for short-term electricity load forecasting in micro-grids. In a more recent work, Zheng et al. [61] developed a hybrid algorithm that combines similar days selection, empirical mode decomposition, and long short-term memory neural networks to construct a prediction model for short-term load forecasting.

Despite the popularity of ANN, other novel-techniques are lately gaining attention. For instance, Talavera-Llames et al. [62] adapted a Nearest Neighbours-based strategy to address the energy consumption forecasting problem in a Big Data environment. Torres et al. [63] developed a novel strategy based on Deep Learning to predict times series and tested such strategy on electricity consumption data recorded in Spain from 2007 to 2016. Zheng et al. [64] also presents a Deep Learning approach to deal with forecasting short-term electric load time series. Galicia et al. [65] compared Random Forest with Decision Trees, Linear Regression and the gradient-boosted trees on Spanish electricity load data with a ten-minute frequency. Furthermore, Evolutionary Algorithms have been applied to short-term forecasting energy demand by Castelli et al. in [66, 67]. Burger and Moura [68] tackled the forecasting of electricity demand by applying an ensemble learning approach that uses Ordinary Least Squares and k-Nearest Neighbors. In [69], Papadopoulos and Karakatsanis explore the ensemble learning approach and compare four different methods: seasonal autoregressive moving average (SARIMA), seasonal autoregressive moving average with exogenous variable (SARIMAX), random forests (RF) and Gradient Boosting regression trees (GBRT). Finally, Li et al. [70] proposed a novel Ensemble Method for loads forecasting based on wavelet transform, extreme learning machine (ELM) and partial least squares regression.

The reader can find a more exhaustive review about time series forecasting in the

articles of Martínez-Álvarez et al. [71] (about Machine Learning methods) and Daut et al. [72] and Deb et al. [34] (about conventional and artificial intelligence methods).

1.2.3 Electricity consumption forecasting with Evolutionary Algorithms

In electricity consumption forecasting, Evolutionary Algorithms are often coupled with other methods [73, 74]. We use them especially when we are dealing with a non-linear time series. In their article [75], Y. Lee and L. Tong proposes a hybrid forecasting model mixing ARIMA and genetic programming i.e. a type of algorithm used in EAs. This hybrid model achieves very good results, whether the databases were large or small. It was tested on energy consumption data of China. The hybrid model obtained a lower error rate than those of the other models with which it was comparing. In this case the authors used genetic programming to improve forecasting of non-linear time series. C. Unsihuay-Vila et al. [76] prefer to use a genetic algorithm to determine the optimal parameters for the nonlinear chaotic dynamic based predictor that is charged to forecast electricity loads and prices of the New England data set consists of hourly electricity loads and Alberta data set consists of hourly electricity loads.

Although Evolutionary Algorithms are most often used in combination with other algorithms, studies using directly Evolutionary Algorithms for electricity consumption prediction were also conducted. In their article [77], A. Azadeh and S. Tarverdian seeks to predict a month of energy consumption in Iran through three methods : a genetic algorithm, the ARIMA approach for time series and simulated-data i.e. simulation of one month of consumption data based on the stochastic behaviour of the raw data. The genetic algorithms stand out from the three as the one with the smallest relative error.

Harun Kemal Ozturka et al. [78] and H. Ceylan and H. Ozturk [79] use Evolutionary Algorithms to predict electricity consumption in Turkey. In article [78] the objective is twofold, on the one hand the authors seek estimated the Turkish electricity demand and on the other hand, show that genetic algorithms are a very good way of doing so.

Article [79] seeks to prove the effectiveness of two forms of equations, one linear and the other exponential through genetic algorithms.

1.2.4 Electricity consumption forecasting with Ensemble Methods

Ensemble Methods have been successfully applied for solving pattern classification, regression and forecasting in time series problems [80, 81]. For example, Adhikari [82] proposed a linear combination method for time series forecasting that determines the combining weights through a novel neural network structure. Bagnal et al. [83] proposed a method using an ensemble of classifiers on different data transformations in order to improve the accuracy of time-series classification. Authors demonstrated that the simple combination of all classifiers in one ensemble obtained better performance than any of its components. Jin and Dong [81] proposed a deep neural network-based Ensemble Method that integrates filtering views, local views, distorted views, explicit and implicit training, subview prediction, and Simple Average for classification of biomedical data. In particular, they used the Chinese Cardiovascular Disease cardiogram database.

Chatterjee et al. [84] developed an ensemble support vector machine algorithm for reliability forecasting of a mining machine. This method is based on least square support vector machine (LS-SVM) with hyper parameters optimized by a Genetic Algorithm (GA). The output of this model was generalized from a combination of multiple SVM predicted results in time series dataset. Additionally, the advantages of Ensemble Methods for regression from different viewpoints such as strength-correlation or biasvariance was also demonstrate in the literature [85].

Ensemble learning based methods have been also applied in energy time series forecasting context. For example, Zang et al. [86] proposed a method, called extreme learning machine (ELM), which was successfully applied on the Australian National Electricity Market data.

Another example was presented by Tan et al. in [87] where the authors proposed a price forecasting method based on wavelet transform combined with ARIMA and GARCH models. The method was applied on Spanish and PJM electricity markets. Fan et al. [88] proposed a ensemble Machine Learning model based on Bayesian Clustering by Dynamics (BCD) and SVM. The proposed model was trained and tested on the data of the historical load from New York City in order to forecasts the hourly electricity consumption. Tasnim et al. [89] proposed a cluster-based ensemble framework to predict wind power by using an ensemble of regression models on natural clusters within wind data. The method was tested on a large number of wind datasets of locations across spread Australia.

Ensembles of ANNs have been recently applied in the literature with the aim of energy consumption or price forecasting. For instance, the authors in [90] presented a building-level neural network-based ensemble model for day-ahead electricity load forecasting. The method showed that it outperforms the previously established best performing model by up to 50%, in the context of load data from operational commercial and industrial sites. Jovanovic et al. [91] used three artificial neural networks for prediction of heating energy consumption of a university campus. The authors tested the neural networks with different parameter combinations, which, when used in an ensemble scheme, achieved better results.

2 | Methodology

The main objective of this chapter is to present the functioning of the two prediction methodologies : an Evolutionary Algorithm for regression trees and an Ensemble Method by Stacking strategy. Firstly, a section is dedicated to present to the reader the data on which the methods have been applied and the language chooses to implement the methods. The reader will then have an explanation of how these two methods work.

2.1 Material

2.1.1 Dataset

The dataset used in this thesis records the general electricity consumption in Spain (expressed in megawatts) over a period of 9 years and 6 months, with a 10 min period between each measurement. Thus, what is measured is the electricity consumption taken as a whole, not relative to a specific sector. In total, the dataset is composed of 497.832 measurements, which go from 1 January 2007 at midnight till 21 June 2016 at 11:40 p.m.

The original dataset has been pre-processed in order to be used, as in [63]. First, the attribute corresponding to consumption has been extracted, and a consumption vector has been obtained. After that, the consumption vector has been redistributed in a matrix depending on a historical window, w, and a prediction horizon, h. The historical window, or data history (w) represents the number of previous entries taken into consideration in order to train a model that will be used to predict the subsequent values (h). This process is detailed in Figure 2.1.

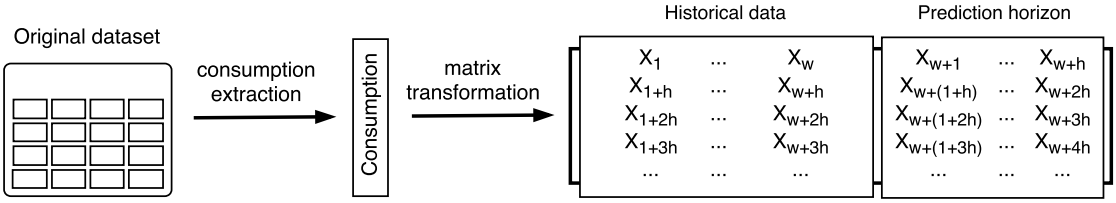


Figure 2.1: Dataset pre-processing from the article *Stacking Ensemble Learning for Short-Term Electricity Consumption Forecasting* of F. Divina et al. [28]. w determines the amount of the historical data used, while h determines the prediction horizon.

The prediction horizon (h) has been set to 24, corresponding to a period of 4 h. Moreover, different values of the data history have been used. In particular, w has been set to the values 24, 48, 72, 96, 120, 144 and 168, corresponding to 4, 8, 12, 16, 20, 24 and

28 h, respectively. The resulting datasets have been divided into 70% for the training set and 30% for the test set.

Table 2.1 provides the details of each dataset. Notice that for all the obtained datasets, the last 24 columns represent the values to be predicted, and thus are not considered for training purposes.

w	**#Rows**	**#Columns (w+h)**
24	20,742	48
48	20,741	72
72	20,740	96
96	20,739	120
120	20,738	144
144	20,737	168
168	20,736	192

Table 2.1: Dataset information depending on the value of w.

2.1.2 R software and language

Figure 2.2: R logo from the website R [92]

This part is based on the book *Introduction à la programmation en R* [1] by V. Goulet [93]. R is an integrated data manipulation, calculation and graphing environment created by Robert Gentleman and Ross Ihaka[94]. It is also a complete and autonomous programming language. It is free, open source and multi-platform i.e it can be installed on Linux, Window or Mac. It is a very powerful and complete tool, well adapted for the computer implementation of statistical methods. One of the most practical features of this software is the integrated documentation system that allows direct access to the documentation of the functions it offers. It also has efficient procedures and data storage capabilities. It offers some treatment for tables and matrices, which is ideal for our data presented above. The R language also provides tools for the learning machine [95]. R's internal function library is divided into sets of functions and related data sets called packages. Some are present by default, but it is easy to add packages for very specific treatments.

R has been chosen to implement both methods as it offers a large number of packages for both evolutionary and learning algorithms. In the next section, the two methods will be presented and the R packages will be quoted, as well as the parameters used in them.

[1] Introduction to R programming

2.2 Methods

This section details how work the methods and which R package have been used to implement them.

2.2.1 EAs for regression trees

The approach for the EAs prediction method is as follows: the population consists of regression trees. Each individual, i.e. a tree, is trained on the training set according to certain parameter e.g. the maximum depth of the tree. At each iteration of the Evolutionary Algorithm, individuals will be selected according to the error between the prediction and the real value and the complexity of the tree. These individuals will then be modified. For example, we will mix parameters from two regression trees to produce a third. Individuals are trained on the data and so on.

To train the method means that the method will run with certain parameters and its outputs will be compared to the real values. The model will then adjust these parameters to a better approximation of its predictions compared to the actual values. The method with the best parameters become the output model that will produce predictions on data whose real value is unknown [96].

For the implementation, the R package *Evtree* [97] has been used with the following parameters:

- *minbucket*: 8 (minimum number of observations in each terminal node)
- *minsplit*: 100 (minimum number of observations in each internal node)
- *maxdepth*: 15 (maximum tree depth)
- *ntrees*: 300 (number of trees in the population)
- *niterations*: 1000 (maximum number of generations)
- *alpha*: 0.25 (complexity part of the cost function)
- *operatorprob*: with this parameter, we can specify, in list or vector form, the probabilities for the following variation operators:
  - *pmutatemajor*: 0.2 (Major split rule mutation, selects a random internal node r and changes the split rule, defined by the corresponding split variable v_r, and the split point s_r [97])
  - *pmutateminor*: 0.2 (Minor split rule mutation is similar to the major split rule mutation operator. However, it does not alter v_r and only changes the split point s_r by a minor degree, which is defined by four cases describes in [97])
  - *pcrossover*: 0.8 (Crossover probability)

- *psplit*: 0.2 (Split selects a random terminal-node and assigns a valid, randomly generated, split rule to it. As a consequence, the selected terminal node becomes an internal node r and two new terminal nodes are generated)
- *pprune*: 0.4 (Prune chooses a random internal node r, where r $>$ 1, which has two terminal nodes as successors and prunes it into a terminal node [97])

For more information on how *Evtree* works, the reader is invited to consult *evtree: Evolutionary Learning of Globally Optimal Classification and Regression Trees in R* written by T. Grubinger, A. Zeileis and K. Pfeiffer [97]. In this document, the creators of *Evtree* explain the mechanics behind their algorithm.

Functioning

Before using the *Evtree* method, processing is carried out on the input data.

Data preparation

For reasons of calculation time, it was decided to predict the columns separately from each other to be able to run the code in parallel. This choice requires data preparation.

Once the data in matrix form has been obtained as shown in Figure 2.1 page 15 (it can be of different sizes depending on the historical window: 24, 48, 72, 96, 120, 144 or 168 values representing 4, 8, 12, 16, 20, 24 or 28 hours respectively), the matrix is divided into two sets: the training set and the test set. Once it is done, the column I to be predicted, divided in the same way into training and test set, is added to them. The column I is one of the 24 columns of the horizon prediction representing the 4 hours following the historical data.

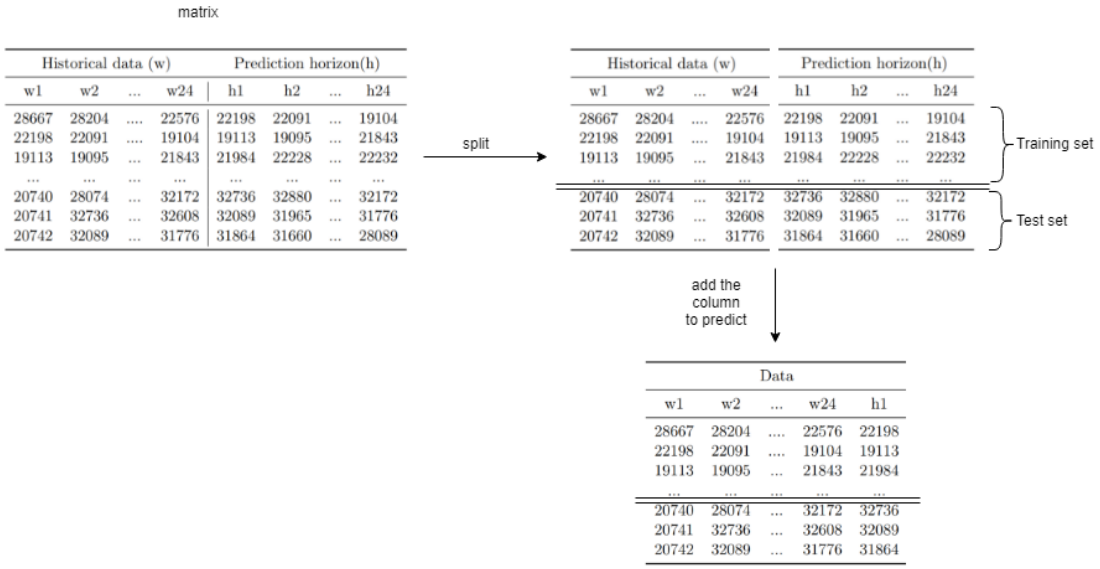


Figure 2.3: Example of data preparation with $w = 24$.

Figure 2.3 illustrates how data preparation works with a w equal to 24. First the matrix is formed with the historical data and the prediction horizon, then it split into training and test set (70/30 %) and final the column to predict (in the Figure 2.3 it is the columns $h1$) is added.

EAs method

The training set is given to the *Evtree* method so it can train and produces a model i.e. the tree with the best result. This model is then verified with the data from the test set. Figure 2.4 illustrates how the method works for one column I. The process must therefore be repeated for all the 24 measurements. The results obtained by these models are presented in chapter 3 : Results.

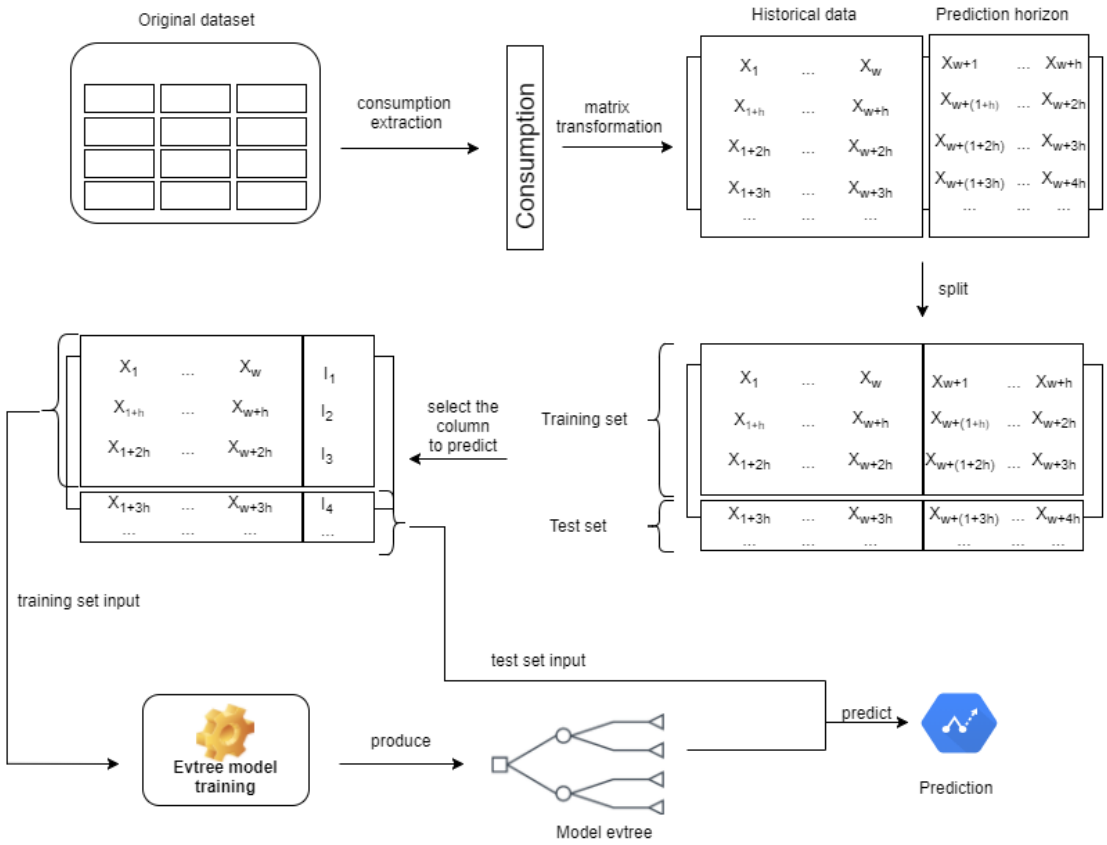


Figure 2.4: A scheme of Eas method mechanism from the article *Stacking Ensemble Learning for Short-Term Electricity Consumption Forecasting* of F. Divina et al. [28]. w determines the size of the historical window used, while h determines the prediction horizon.

The code of this method is available in appendix A Code source of the EA method.

2.2.2 Ensemble Method by Stacking strategy

As already stated in chapter 2 : Methodology, we used a stacking ensemble scheme for the Ensemble Method. In particular, a scheme formed by three base learning methods and a top method is employed. The basic learning methods are Evolutionary Algorithm for Regression Tree, Artificial Neural Networks and Random Forests. At the top level, we have used the Generalized Boosted Regression Models in order to combine the predictions produced by the bottom level. The employed scheme is graphically shown in Figure reffig:ensemble.

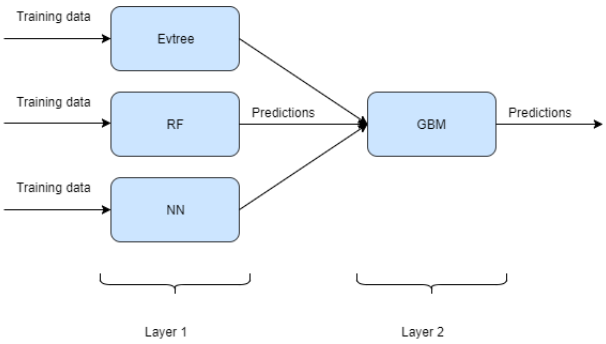


Figure 2.5: A graphical representation of the ensemble scheme from the article *Stacking Ensemble Learning for Short-Term Electricity Consumption Forecasting* of F. Divina et al. [28]. NN: Artificial Neural Network; RF: Random Forests; GBM: Generalized Boosted Regression Models.

Before detailing how the method works, an explanation of the Artificial Neural Network, Random Forests, and Generalized Boosted Regression Models learning methods is given in the following subsection.

Artificial Neural Network

Artificial Neural Network (ANNs) [98] are computational models inspired by the structure and functions of biological Neural Networks. The basic unit of computation is the neuron, also called node, which receives input from other nodes or from an external source and computes an output. In order to compute such output, the node applies a function f called the *Activation Function*, which has the purpose of introducing non-linearity into the output. Furthermore, the output is produced only if the inputs are above a certain threshold.

Basically, an ANN creates a relationship between input and output values and is composed of interconnected nodes grouped in several layers. Among such layers we can distinguish the outer ones, called input and output layers, from the "internal" ones, called hidden layers. In contrast to biological neuron networks, ANNs usually consider only one type of node, in order to simplify the model calculation and analysis. Figure 2.6 shows

a 2-layers Neural Network.

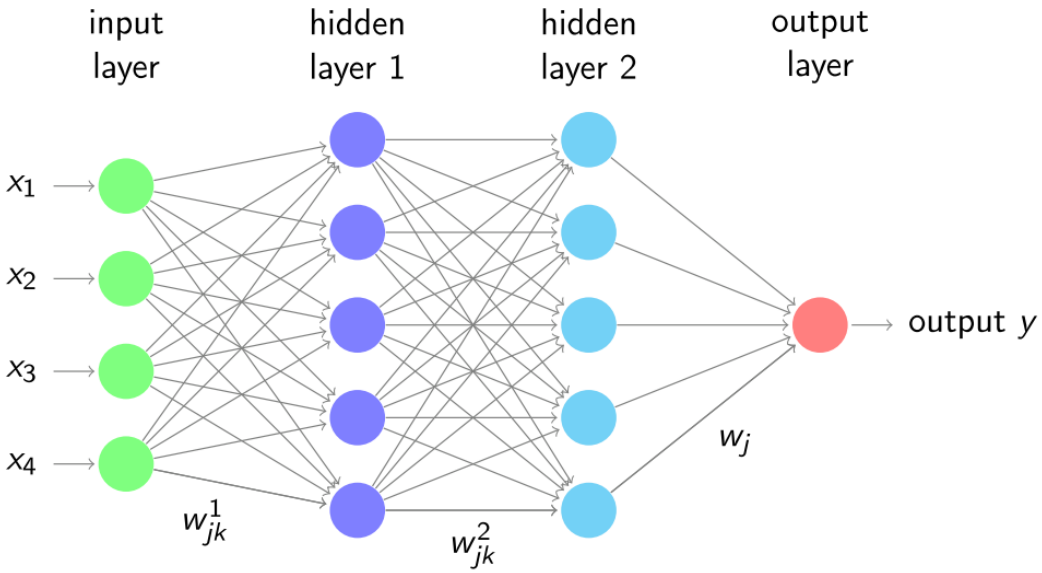


Figure 2.6: illustration of a 2-layers Neural Network from the *Machine learning : from Neural Network to big data* course given by B. Frénay at the University of Namur [47]

The intensity of the connection between nodes is determined by weights, which are modified during the learning process. Therefore, the learning process consists to adapting the connections to the data structure that model the environment and to characterize its relations.

According to the structure, there are different types of ANN. The suitability of the structure depends on several factors as, for example, the quality and the volume of the input data. The simplest type of ANN is the so-called feedforward Neural Network. In such networks, nodes from adjacent layers are interconnected and each connection has a weight associated to it. The information moves forward from the input to the output layer through the hidden nodes. There is only one node at the output layer, which provides the final result of the network, being it a class label or a numeric value.

It is the *nnet* package of R [99] that is used to implement a Neural Network in the stacking method, a package for feed-forward Neural Networks with a single hidden layer, and for multinomial log-linear models. The following parameters were used in this paper:

- *size*: 10 (number of nodes in the hidden layer)
- *skip*: true (add skip-layer connections from input to output)
- *MaxNWts*: 10,000 (maximum number of weights allowed)
- *maxit*: 1000 (maximum number of iterations)

Random Forest

The term Random Forest (RF) was introduced by Breinman and Cutle in [100], and refers to a set of decision trees which form an ensemble of predictors.

Thus, RF is basically an ensemble of decision trees, where each tree is trained separately on an independent randomly selected training set. It follows that each tree depends on the values of an input dataset sampled independently, with the same distribution for all trees.

In other words, the trees generated are different since they are obtained from different training sets from a bootstrap subsampling and different random subsets of features to split on at each tree node. Each tree is fully grown, in order to obtain low-bias trees. Moreover, at the same time, the random subsets of features result in low correlation between the individual trees, so the algorithm yields an ensemble that can achieve both, low bias and low variance [101]. For classification, each tree in the RF casts a unit vote for the most popular class at the input. The final result of the classifier is determined by a majority vote of the trees. For regression, the final prediction is the average of the predictions from the set of decision trees.

The method is less computationally expensive than other tree-based classifiers that adopt bagging strategies, since each tree is generated by taking into account only a portion of the input features [102].

the *randomForest* package of R [103] have used for the implementation, which provides an R interface to the original implementation by Breiman and Cutle [100]. In the code, the algorithm is used with the following parameters:

- *ntree* : 100 (number of trees to be built by the algorithm).
- *maxnodes*: 100 (maximum number of terminal nodes trees in the forest can have).

Generalized Boosted Regression Models

The following part is based on *Greedy Function Approximation: A Gradient Boosting Machine* and *Stochastic Gradient Boosting* from J. Friedman [104, 105].

This method iteratively trains a set of decision trees. The current ensemble of trees is used in order to predict the value of each training example. The prediction errors are then estimated, and poor predictions are adjusted, so that in the next iterations the previous mistakes are corrected.

Gradient Boosting involves three elements:

- A loss function to be optimized. Such function is problem dependent. For instance, for regression a squared error [106] can be used and for classification we could use a logarithmic loss [107].
- A weak learner to make predictions. Regression trees are used to this aim, and a greedy strategy is used in order to build such trees. This strategy is based on using a scoring function used each time a split point has to be added to the tree. Other

strategies are commonly adopted in order to constrain the trees. For example, one may limit the depth of the tree, the number of splits or the number of nodes.

- An additive model to add trees to minimise the loss function. This is done in a sequential way, and the trees already contained in the model built so far are not changed. In order to minimise the loss during this phase, a gradient descend procedure is used. The procedure stops when a maximum number of trees has been added to the model or once there is no improvement in the model.

Overfitting is common in Gradient Boosting, and usually, some regularization methods are used in order to reduce it. These methods basically penalize various parts of the algorithm. A model is overfitting when it sticks so much to the training data that it learns the details and the noise present in it. This has a negative impact on model performance when it faces new data [108].

Usually some mechanisms are used in order to impose constraints on the construction of decision trees. For example : limit the depth of the trees, the number of nodes or leafs or the number of observations per split.

Another mechanism is shrinkage, which is basically weighting the contribution of each tree to the sequential sum of the predictions of the trees. This is done with the aim of slowing down the learning rate of the algorithm.

As a consequence the training takes longer, since more trees are added to the model. In this way a trade-off between the learning rate and the number of trees can be reached.

The GBM package of R [109] is used in the stacking method with the following parameters:

- *distribution*: Gaussian (function of the distribution to use)
- *n.trees*: 3000 (total number of trees, i.e., the number of Gradient Boosting iteration)
- *interaction.depth*: 40 (maximum depth of variable interactions)
- *shrinkage*: 0.9 (learning rate)
- *n.minobsinnode*: 3 (minimum number of observations in the trees terminal nodes)

Functioning of the Ensemble Method

The choice of predicting the 24 measurements separately was also made for this method. The data goes through the same preparation that was described in paragraph data preparation on page 18.

Once the preparation is complete, the training set is given to these three methods (*NN*, *RF* and *Evtree*) constituting level 0 of the Ensemble Method. Each of these three methods train a model on these data and produce predictions. These predictions are given to the *GBM* method (the level 1 of the Ensemble Method) which uses them to train its own model. Once the GBM model is obtained, the test set is given an input and it produces the predictions. Figure 2.7 shows the mechanism of the method. The results obtained by these models are presented in chapter 3 : Results.

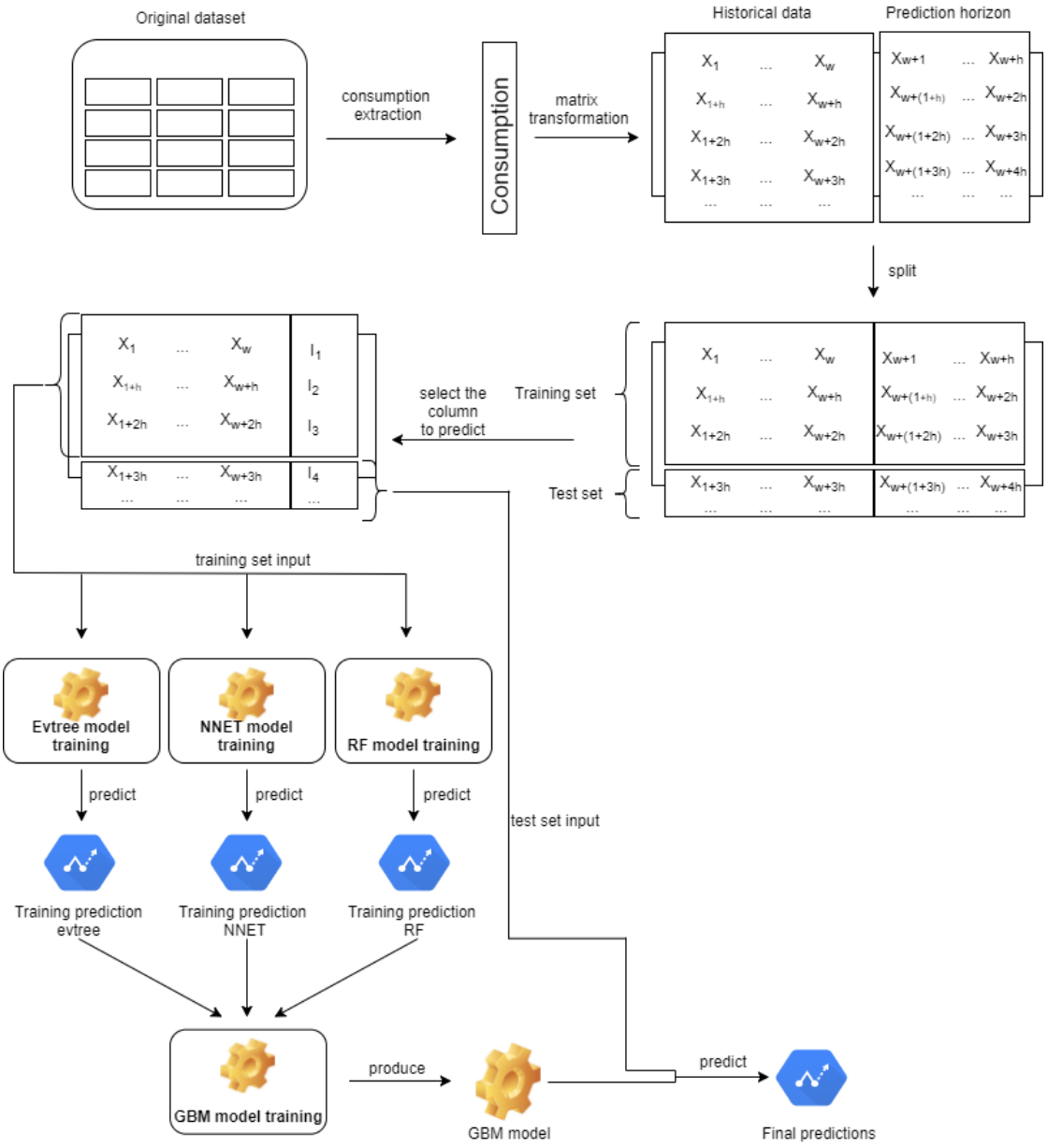


Figure 2.7: stacking ensemble process from the article *Stacking Ensemble Learning for Short-Term Electricity Consumption Forecasting* of F. Divina et al. [28].

The code of this method is available in appendix B Code source of the ensemble method.

3 | Results

This chapter provides the results obtained by the two prediction methodologies, an Evolutionary Algorithm for regression trees and an Ensemble Method by Stacking strategy, on the dataset described in chapter 2 as well as some observations about them.

In order to assess the performances of both the Evolutionary Algorithm and the Ensemble Method (for the different methods it uses), the mean relative error (MRE) was used. The formula of the MRE is :

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i - \hat{Y}_i|}{Y_i} \tag{3.1}$$

Where, $\hat{Y}_i$ is the predicted value, Y_i the real value and n the number of lines in the test set.

3.1 Notations

The following notations are used throughout this chapter:

- w corresponds to the historical window and has as value 168, 144, 120, 96, 72, 48 and 24.
- h corresponds to the prediction horizon i.e. the 24 columns that contain the values to forecast.
- i is one of the 24 columns of h and the column with the values that the method (the Eas or the Ensemble Method) has to predict. It has as values 1 to 24.

3.2 Result of the method Evolutionary Algorithm for regression trees

In this section, the results obtained by the EAs method are exposed.

Figure 3.1 and Table 3.1 show the MRE obtained by the *Evtree* method for the different value of i and historical window (w). We can see that the longer the historical window, the smaller the error. Indeed, when w is equal to 168 the corresponding average MRE is 3.09 %. In comparison when w is equal to 24 the corresponding average MRE is 4.49 %. It can also be seen that the further away the column to be predicted is from the prediction window (the higher the i), the greater the MRE. For example, when w equal to 168 and i equal to 1 the MRE is 1.15 % but when the i equal to 24 the MRE is of 4.13 %.

i \ w	**168**	**144**	**120**	**96**	**72**	**48**	**24**
1	1.15	0.98	1.08	1.01	1.11	0.93	0.94
2	1.63	1.58	1.44	1.36	1.3	1.39	1.43
3	1.81	1.75	1.83	1.79	1.77	1.86	1.79
4	2.1	2.09	1.97	2.08	2.1	2.1	2.09
5	2.31	2.11	2.3	2.22	2.29	2.34	2.58
6	2.57	2.46	2.34	2.53	2.37	2.79	2.76
7	2.71	3.23	2.7	2.73	2.77	2.87	3.17
8	2.63	2.8	2.73	2.92	2.99	3.23	3.49
9	2.9	2.99	2.98	2.99	3.2	3.37	3.9
10	2.92	3.45	3.15	3.33	3.3	3.58	3.75
11	3.23	3.21	3.4	3.57	3.27	3.86	4.15
12	3.24	3.58	3.47	3.62	3.66	4.3	4.55
13	3.26	3.21	3.71	3.72	3.9	4.19	4.79
14	3.58	3.29	3.5	3.75	4.12	4.59	4.99
15	3.45	3.41	3.59	4.05	3.91	4.79	5.63
16	3.54	3.51	3.88	4.24	4.25	5.21	5.4
17	3.52	3.89	3.77	4.24	4.24	5.07	5.74
18	3.75	3.7	3.78	4.02	4.12	5.37	6.03
19	3.55	3.65	4.2	3.97	4.46	5.37	6.28
20	3.75	3.99	3.95	4.24	4.5	5.55	6.14
21	4.02	4.05	4.1	4.94	4.89	5.49	6.55
22	4.02	4.06	4.14	4.79	4.93	5.98	6.85
23	4.38	4.14	4.4	5.14	5.01	5.71	7.21
24	4.13	4.46	4.21	4.89	4.98	5.68	7.52
average	3.09	3.15	3.19	3.42	3.48	3.98	4.49

Table 3.1: Value of the MRE in % by w and for each value of i.

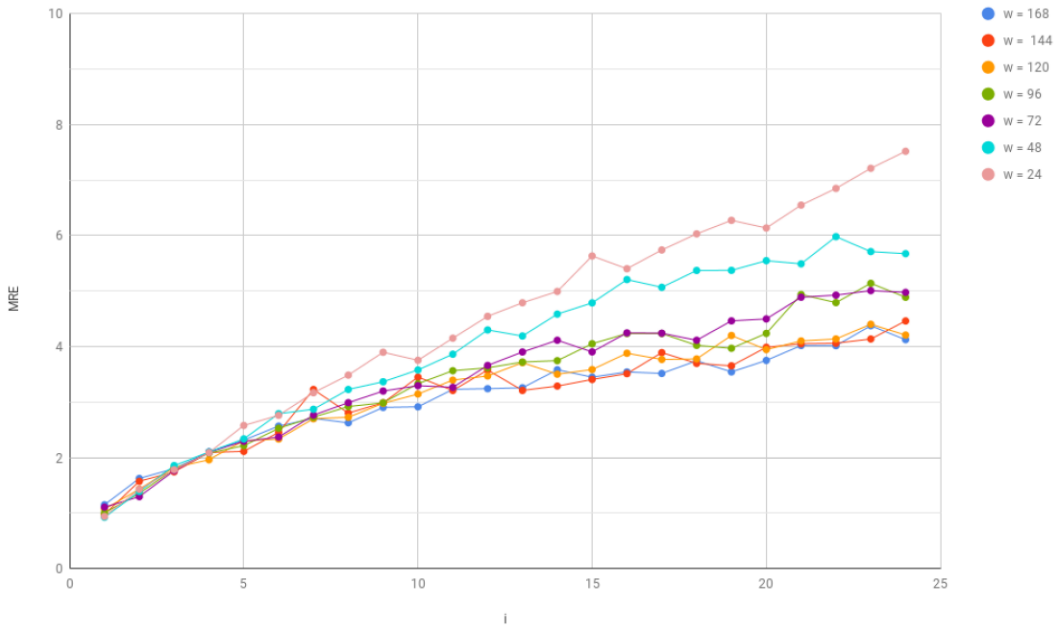


Figure 3.1: Comparison of the MRE obtained by *Evtree* on each value of i for each value of w used.

3.3 Result of the method Ensemble Method by Stacking strategy

This section exposes the results of the Ensemble Method and its different learning algorithms: Artificial Neural Network (NN), Random Forest (RF), EAs for regression trees and Generalized Boosted Regression Models (GBM).

Figures 3.2 to 3.5 show the results obtained on all the value of i for each historical window (w) by the different learning algorithms. The average results obtained are also shown in the bar graph show by Figure 3.6. In the graph, GBM results are labels as ensemble.

Results are summarized in Table 3.2 where a ranking of the methods is shown, according to the MRE obtained.

In Table 3.3, the averages MRE by historical window (w) are detailed.

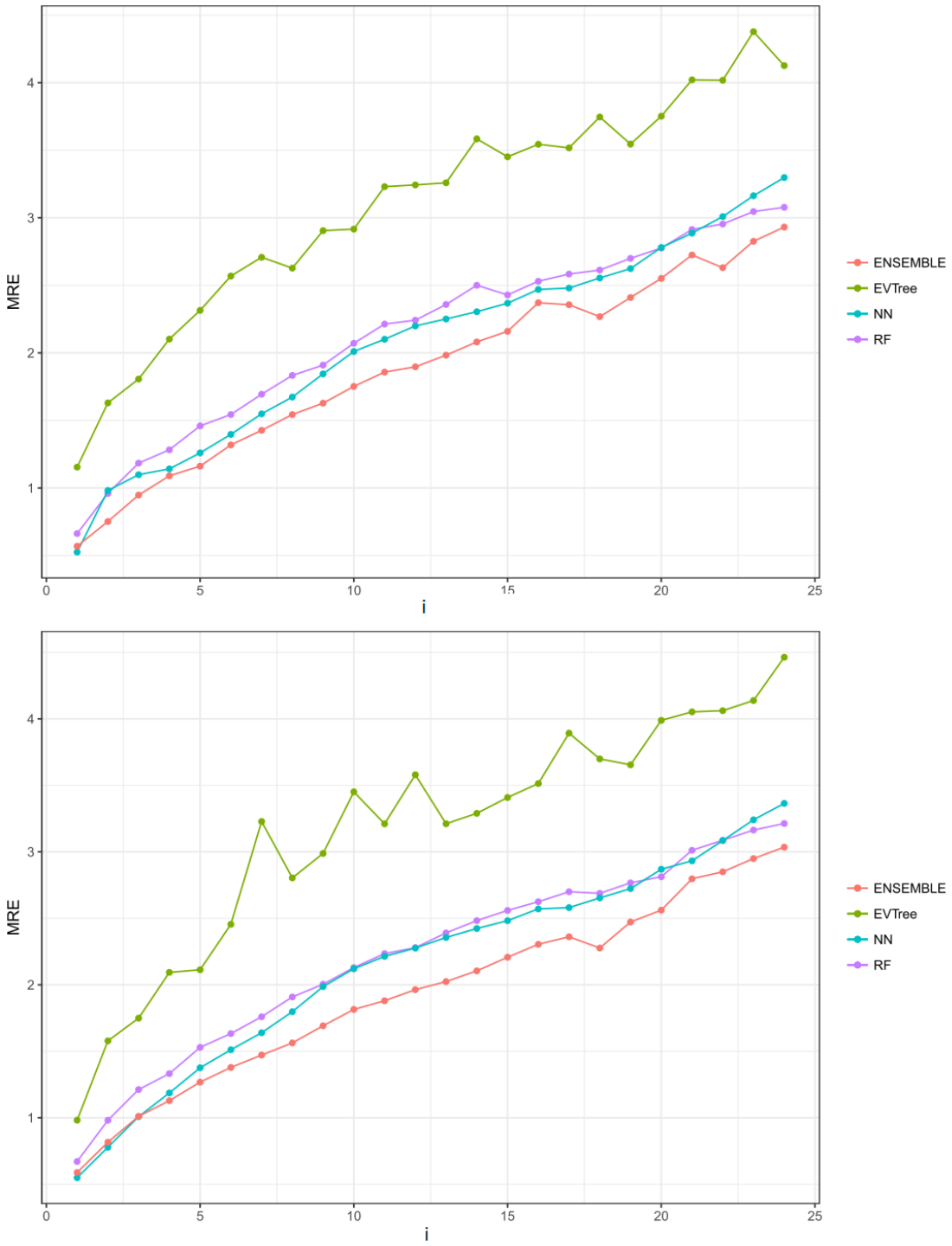


Figure 3.2: Comparison of the MRE obtained by the base algorithms and the ensemble scheme on each value of i when $w = 168$ (above) and $w = 144$ (bottom).

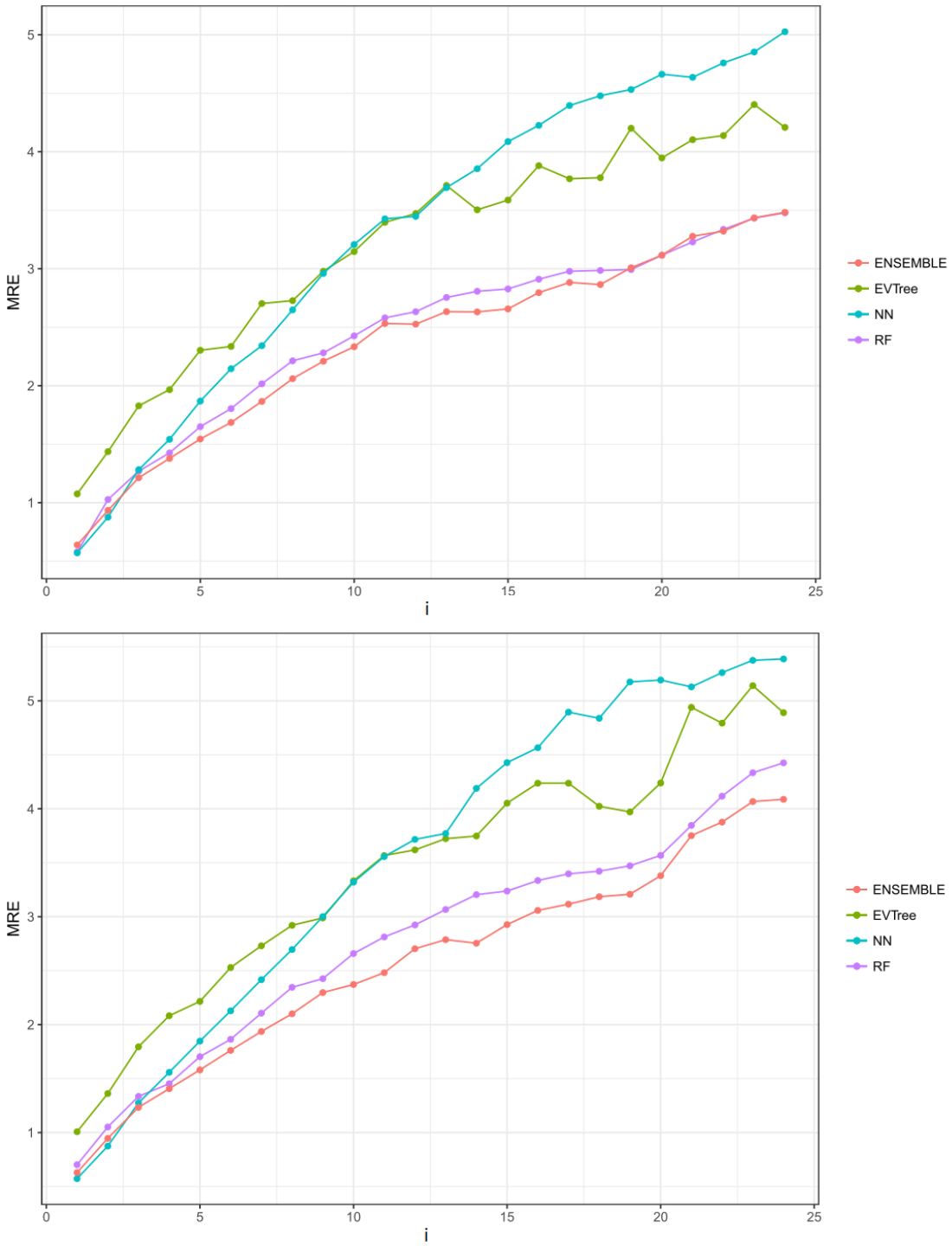


Figure 3.3: Comparison of the MRE obtained by the base algorithms and the ensemble scheme on each value of i when $w = 120$ (above) and $w = 96$ (bottom).

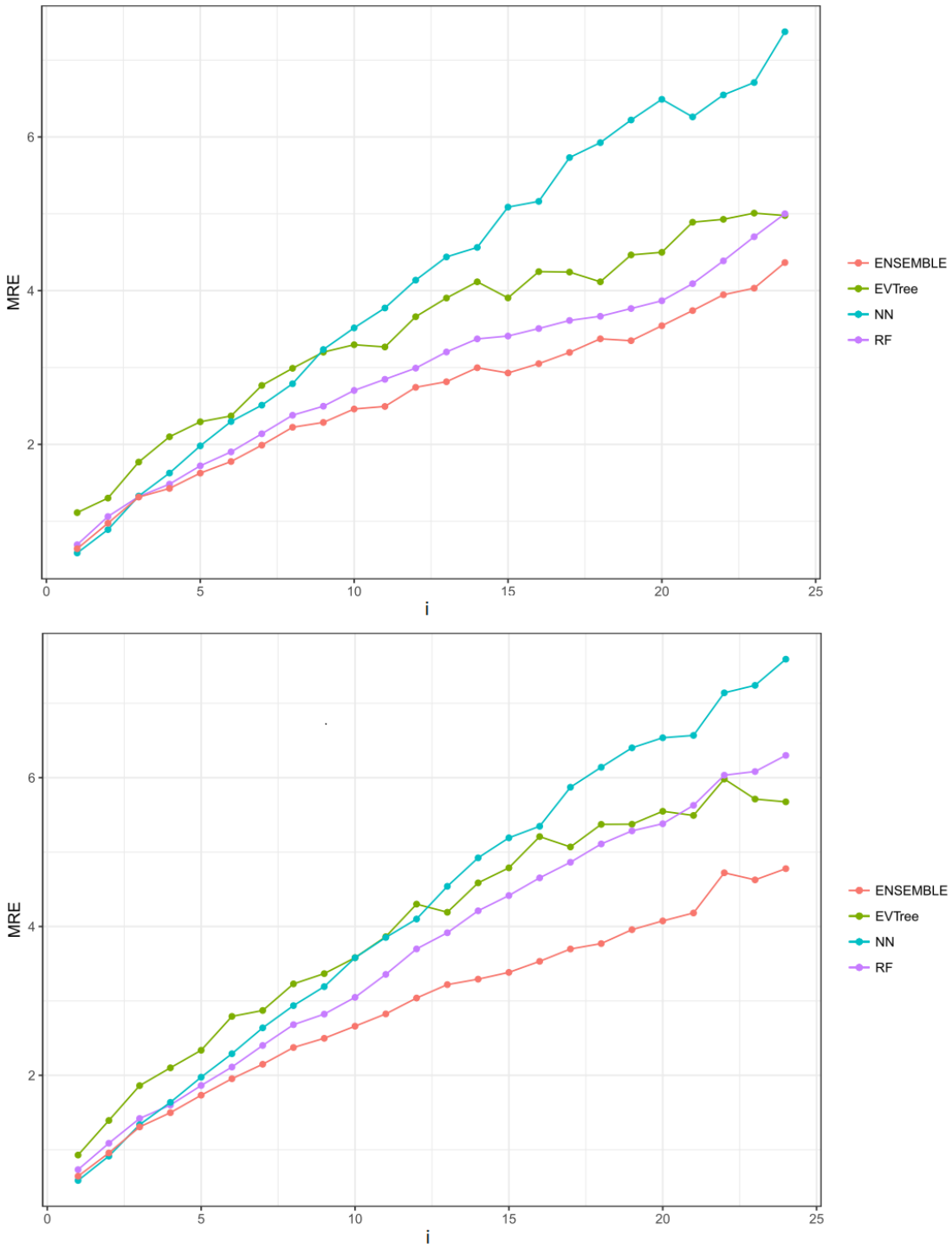


Figure 3.4: Comparison of the MRE obtained by the base algorithms and the ensemble scheme on each value of i when $w = 72$ (above) and $w = 48$ (bottom).

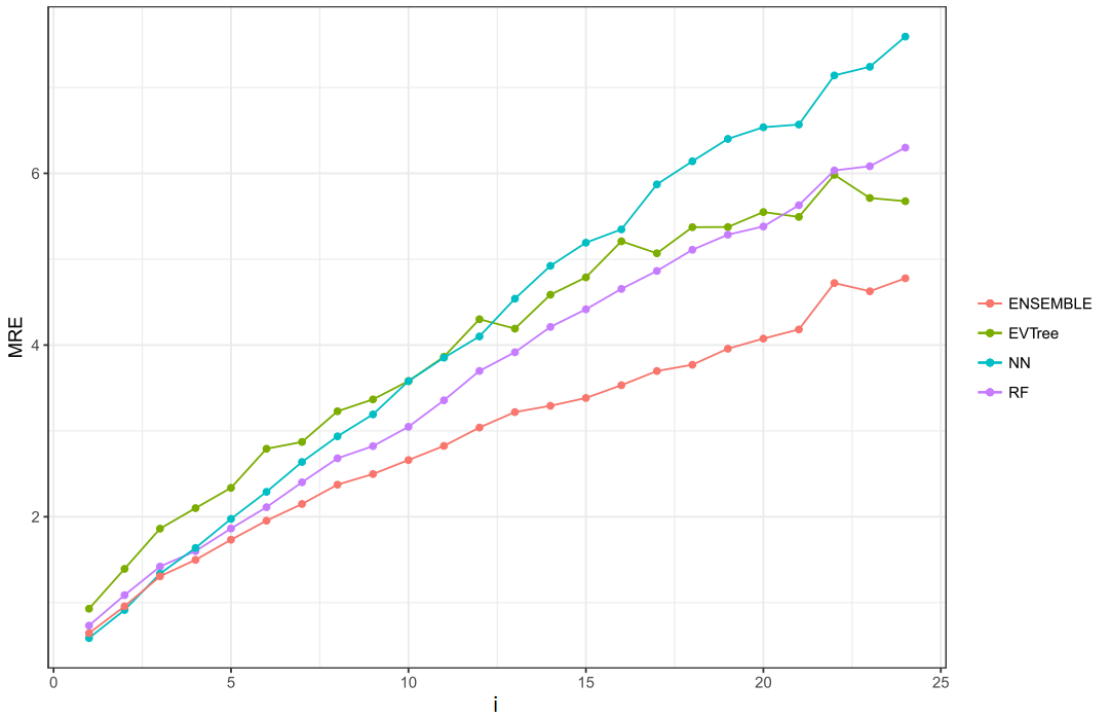


Figure 3.5: Comparison of the MRE obtained by the base algorithms and the ensemble scheme on each value of i when $w = 24$.

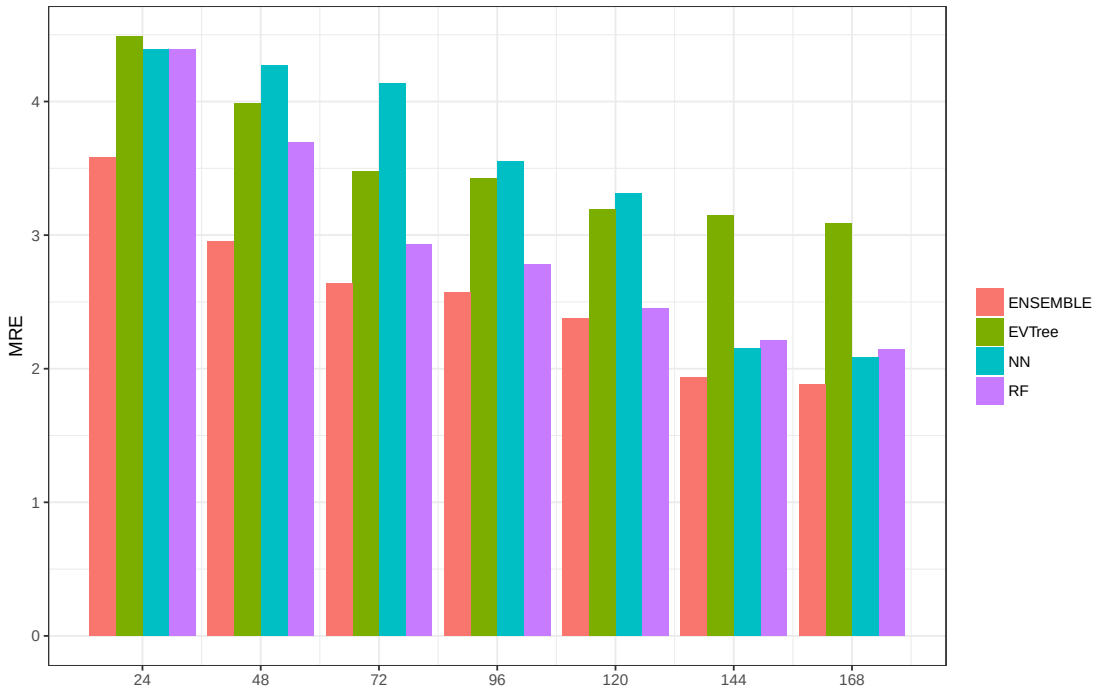


Figure 3.6: Average MRE obtained in each value of w.

168	**144**	**120**	**96**	**72**	**48**	**24**
GBM	GBM	GBM	GBM	GBM	GBM	GBM
NN	NN	RF	RF	RF	RF	RF,NN,EV
RF	RF	NN,EV	NN,EV	EV	NN,EV	
EV	EV			NN		

Table 3.2: Ranking of the methods according to their performances obtained on different values of w, according.

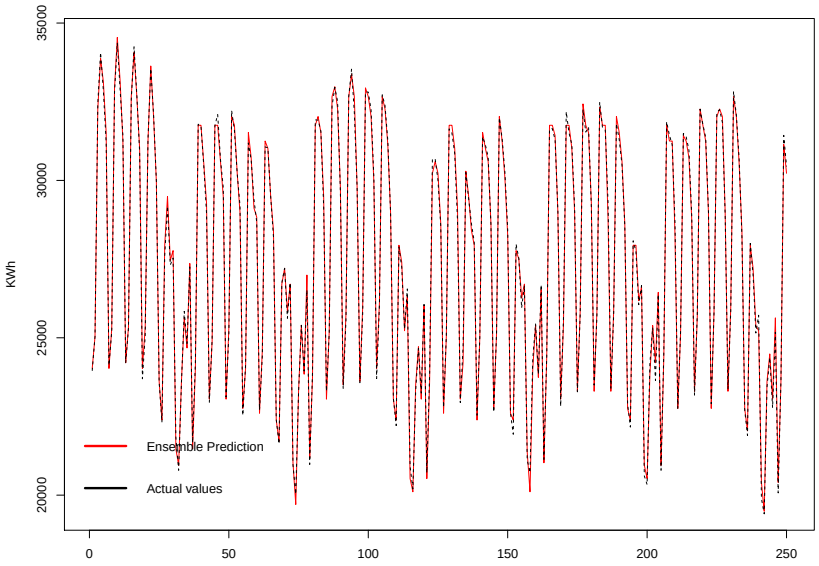
w	**Evtree**	**RF**	**NNET**	**GBM - ENSEMBLE**
168	3.09	2.15	2.08	1.88
144	3.15	2.22	2.16	1.94
120	3.19	2.45	3.31	2.38
96	3.42	2.78	3.55	2.57
72	3.48	2.93	4.13	2.64
48	3.98	3.7	4.27	2.95
24	4.49	4.39	4.39	3.58
average	3.62	3.08	3.64	2.68

Table 3.3: Average value of the MRE in % by w and *method*.

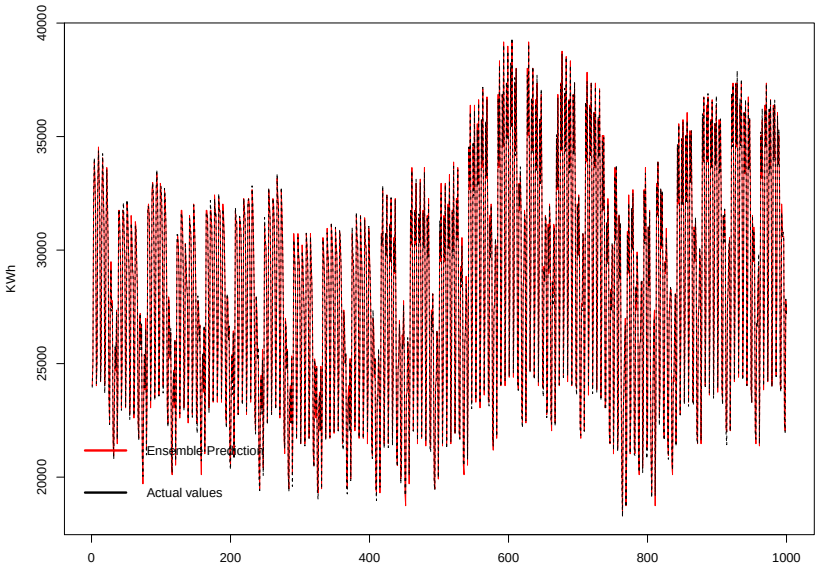
As for the EAs method, the longer the historical window that is used for the method ensemble, the smaller the MRE obtained. As we can see when we use a historical window of 168 measurements, the average MRE obtained is 1.88 %.

In general, we can also notice the degradation of performances of NN when the historical window used is reduced. In fact, for a historical window of 168, NN obtains the best results among the three base methods (*Evtree, NN, RF*), while for smaller historical windows, starting from 120 measurements, the predictions obtained by this method are always worse than those obtained by RF, and are comparable or worse than the predictions produced by *Evtree*.

We can also notice that the predictions are less and less accurate for increasing values of i, meaning that it is easier to predict the very near future demand than the medium-far future demand. In this sense, we can also observe that NN performs really well when i equalling 1 or 2. In fact, for these two values of i, in many cases the predictions obtained by NN are superior to those obtained by the GBM i.e. the method combining the results of the base algorithms. However, as the value of i increments, the results obtained by GBM are much better than the results achieved by the three base methods. Basically the real difference is made when the value of i become higher and higher. It's mostly visible in Figure 3.6



(a)



(b)

Figure 3.7: Comparison of real and predicted values for a subset of the time series, for $w = 168$. (**a**) 250 readings; (**b**) 1000 readings.

Finally, Figure 3.7 presents a comparison of the real and predicted values for a subset of the time series when a historical window of 168 was used.

For readability reasons (it was not possible to graphically represent the comparison of all data given the number of observations present in the time series), the selection of two subsets of 250 and 1000 readings, respectively shown in Figure 3.7 a,b has been made. We can notice that the predictions are very accurate, and that they can describe in a very precise way the original time series.

3.4 Comparison

3.4.1 Comparison of the two methods with each other

The EAs method for regression trees has an acceptable MRE. Without being very close to 0, it is not more than 5 %. However Table 3.3 shows that the EAs method is less efficient than RF or NN. On the contrary, the Ensemble Method with GBM (that combines the results of the other three bases methods) shows excellent results.

Figure 3.8 shows the MRE for each method according to different values of i and the historical window (w).

i \ w	168				144				120				96				72				48				24			
	EV	**RF**	**NN**	**GBM**	**EV**	**RF**	**NN**	**GBM**	**EV**	**RF**	**NN**	**GBM**	**EV**	**RF**	**NN**	**GBM**	**EV**	**RF**	**NN**	**GBM**	**EV**	**RF**	**NN**	**GBM**	**EV**	**RF**	**NN**	**GBM**
1	1.15	0.66	0.53	0.57	0.98	0.67	0.55	0.59	1.08	0.58	0.57	0.64	1.01	0.70	0.57	0.63	1.11	0.69	0.59	0.64	0.93	0.73	0.59	0.64	0.94	0.74	0.59	0.64
2	1.63	0.96	0.98	0.75	1.58	0.98	0.78	0.82	1.44	1.03	0.88	0.93	1.36	1.05	0.88	0.95	1.30	1.06	0.89	0.97	1.39	1.09	0.91	0.96	1.43	1.15	0.92	1.00
3	1.81	1.18	1.10	0.95	1.75	1.21	1.01	1.01	1.83	1.27	1.28	1.21	1.79	1.34	1.27	1.23	1.77	1.32	1.33	1.31	1.86	1.42	1.34	1.31	1.79	1.53	1.35	1.37
4	2.10	1.28	1.14	1.09	2.09	1.33	1.19	1.13	1.97	1.43	1.54	1.38	2.08	1.45	1.56	1.41	2.10	1.48	1.63	1.43	2.10	1.60	1.64	1.50	2.09	1.79	1.66	1.65
5	2.31	1.46	1.26	1.16	2.11	1.53	1.38	1.27	2.30	1.65	1.87	1.54	2.22	1.70	1.85	1.58	2.29	1.72	1.98	1.63	2.34	1.86	1.98	1.73	2.58	2.11	2.00	1.85
6	2.57	1.54	1.40	1.32	2.46	1.63	1.51	1.38	2.34	1.80	2.15	1.69	2.53	1.86	2.13	1.76	2.37	1.90	2.30	1.78	2.79	2.11	2.29	1.95	2.76	2.42	2.34	2.14
7	2.71	1.69	1.55	1.43	3.23	1.76	1.64	1.47	2.70	2.02	2.34	1.87	2.73	2.11	2.42	1.94	2.77	2.14	2.51	1.99	2.87	2.40	2.64	2.15	3.17	2.76	2.66	2.38
8	2.63	1.83	1.67	1.54	2.80	1.91	1.80	1.56	2.73	2.21	2.65	2.06	2.92	2.35	2.70	2.10	2.99	2.38	2.79	2.22	3.23	2.68	2.94	2.37	3.49	3.11	2.97	2.76
9	2.90	1.91	1.84	1.63	2.99	2.00	1.99	1.69	2.98	2.28	2.96	2.21	2.99	2.43	3.00	2.30	3.20	2.50	3.23	2.29	3.37	2.82	3.19	2.50	3.90	3.34	3.36	2.89
10	2.92	2.07	2.01	1.75	3.45	2.13	2.12	1.81	3.15	2.43	3.21	2.33	3.33	2.66	3.32	2.37	3.30	2.70	3.52	2.46	3.58	3.05	3.58	2.66	3.75	3.71	3.68	3.07
11	3.23	2.21	2.10	1.86	3.21	2.23	2.21	1.88	3.40	2.58	3.43	2.53	3.57	2.81	3.56	2.48	3.27	2.85	3.77	2.49	3.86	3.36	3.85	2.83	4.15	4.00	4.02	3.30
12	3.24	2.24	2.20	1.90	3.58	2.28	2.28	1.96	3.47	2.63	3.45	2.53	3.62	2.92	3.72	2.70	3.66	2.99	4.14	2.74	4.30	3.70	4.10	3.04	4.55	4.33	4.33	3.56
13	3.26	2.36	2.25	1.98	3.21	2.39	2.36	2.02	3.71	2.75	3.69	2.63	3.72	3.07	3.77	2.79	3.90	3.20	4.44	2.82	4.19	3.92	4.54	3.22	4.79	4.69	4.71	3.84
14	3.58	2.50	2.30	2.08	3.29	2.48	2.42	2.10	3.50	2.81	3.86	2.63	3.75	3.20	4.19	2.75	4.12	3.37	4.56	3.00	4.59	4.21	4.92	3.29	4.99	4.95	5.04	3.90
15	3.45	2.43	2.37	2.16	3.41	2.56	2.48	2.21	3.59	2.83	4.09	2.66	4.05	3.24	4.43	2.93	3.91	3.41	5.09	2.93	4.79	4.42	5.19	3.38	5.63	5.19	5.42	4.29
16	3.54	2.53	2.47	2.37	3.51	2.62	2.57	2.30	3.88	2.91	4.23	2.80	4.24	3.34	4.57	3.06	4.25	3.51	5.16	3.05	5.21	4.65	5.35	3.53	5.40	5.46	5.76	4.24
17	3.52	2.58	2.48	2.36	3.89	2.70	2.58	2.36	3.77	2.98	4.40	2.88	4.24	3.40	4.90	3.12	4.24	3.61	5.73	3.20	5.07	4.86	5.87	3.70	5.74	5.82	6.09	4.51
18	3.75	2.61	2.55	2.27	3.70	2.69	2.65	2.28	3.78	2.99	4.48	2.86	4.02	3.42	4.84	3.18	4.12	3.67	5.93	3.37	5.37	5.11	6.14	3.77	6.03	6.00	6.45	4.75
19	3.55	2.70	2.62	2.41	3.65	2.77	2.72	2.47	4.20	2.99	4.53	3.01	3.97	3.47	5.18	3.21	4.46	3.77	6.22	3.35	5.37	5.29	6.40	3.96	6.28	6.19	5.74	4.91
20	3.75	2.78	2.78	2.55	3.99	2.81	2.87	2.56	3.95	3.12	4.66	3.12	4.24	3.57	5.19	3.38	4.50	3.87	6.49	3.54	5.55	5.38	6.54	4.08	6.14	6.45	6.89	5.17
21	4.02	2.91	2.89	2.72	4.05	3.01	2.93	2.80	4.10	3.23	4.64	3.28	4.94	3.85	5.13	3.75	4.89	4.09	6.26	3.74	5.49	5.63	6.57	4.18	6.55	6.79	6.84	5.45
22	4.02	2.95	3.01	2.63	4.06	3.09	3.08	2.85	4.14	3.34	4.76	3.32	4.79	4.12	5.26	3.88	4.93	4.39	6.55	3.95	5.98	6.03	7.14	4.72	6.85	7.16	7.16	5.63
23	4.38	3.05	3.16	2.83	4.14	3.16	3.24	2.95	4.40	3.43	0.85	3.44	5.14	4.33	5.38	4.07	5.01	4.70	6.71	4.03	5.71	6.08	7.24	4.63	7.21	7.62	7.53	6.10
24	4.13	3.08	3.30	2.93	4.46	3.21	3.36	3.04	4.21	3.48	5.03	3.48	4.89	4.43	5.39	4.09	4.98	5.00	7.37	4.37	5.68	6.30	7.59	4.78	7.52	7.97	7.86	6.60
avg	3.09	2.15	2.08	1.88	3.15	2.22	2.16	1.94	3.19	2.45	3.15	2.38	3.42	2.78	3.55	2.57	3.48	2.93	4.13	2.64	3.98	3.70	4.27	2.95	4.49	4.39	4.39	3.58

Figure 3.8: Results of each learning method for each historical window considered and each value of i. In the table EV stands for *Evtree*, RF for *Random Forest*, NN for *Neural Network* and GBM stand for the *Gradient Boost Models* and it is used on the results of the other three bases methods [28].

3.4.2 Comparison with other method

In this subsection, the two forecasting methods are compared with four methods cited in the state of the art: linear regression (LR), ARMA and ARIMA, Deep Learning (DL) and a decision tree algorithm (DT). More details on how these methods were used in the context of the problematic of this thesis are given in the section *Results* of the article *Stacking Ensemble Learning for Short-Term Electricity Consumption Forecasting* [28].

Table 3.4 shows the average MRE obtains by the methods for each value of w. The GBM column corresponds to the use of the GBM method on the time series data while the ENSEMBLE column corresponds to the GBM method used on the results of the basic algorithms.

As see in Table 3.4, the Ensemble Method by Stacking strategy has much better results than all the other methods. It also obtained the best results in all the historical window values considered. Another observation can be made on this table: LR and NN obtain good results which are also quite similar.

The RF method obtains good results, especially for smaller values of historical window. In comparison with NN and LR method, RF produces better results except when w has as values 144 or 168. But in every case it outperforms DL and EV.

The EV method gets fairly good results for the different values of w without really standing out. In general, the classical strategies ARMA and ARIMA and DT method do not perform well on this problem.

In conclusion, the results obtained on this problem by the ensemble scheme are satisfactory, as they achieve more accurate predictions for this short-term electricity consumption forecast problem than all the other methods considered.

w	LR	ARMA	ARIMA	DL	DT	GBM	RF	EV	NN	ENSEMBLE
168	2.07 (0.77)	2.43 (0.97)	6.92 (2.97)	2.46 (0.29)	8.79 (0.96)	4.45 (1.56)	2.15 (0.69)	3.09 (0.84)	2.08 (0.74)	1.88 (0.67)
144	2.15 (0.77)	2.57 (0.91)	7.63 (2.54)	2.32 (0.29)	8.86 (1.01)	4.49 (1.54)	2.22 (0.71)	3.15 (0.90)	2.16 (0.78)	1.94 (0.69)
120	3.33 (1.37)	5.21 (1.87)	6.79 (2.53)	2.98 (0.28)	9.08 (1.12)	5.02 (1.81)	2.45 (0.79)	3.19 (0.95)	3.15 (1.41)	2.38 (0.81)
96	3.57 (1.57)	4.66 (1.81)	14.03 (13.00)	3.12 (0.42)	9.40 (1.45)	5.33 (2.08)	2.78 (1.04)	3.42 (1.15)	3.55 (1.56)	2.57 (0.97)
72	4.20 (2.11)	8.08 (4.54)	11.37 (10.43)	3.39 (0.30)	9.33 (1.39)	5.73 (2.23)	2.93 (1.16)	3.48 (1.18)	4.13 (2.05)	2.64 (0.99)
48	4.28 (2.15)	8.67 (4.71)	8.26 (4.73)	3.46 (0.33)	9.45 (1.48)	6.59 (2.71)	3.69 (1.71)	3.98 (1.52)	4.27 (2.16)	2.95 (1.19)
24	4.44 (2.27)	7.67 (5.37)	8.82 (5.31)	4.51 (0.52)	9.52 (1.55)	8.07 (3.82)	4.39 (2.13)	4.49 (1.91)	4.39 (2.23)	3.58 (1.65)

Table 3.4: Average results for different historical window values. Standard deviation between brackets [28].

3.5 Conclusion

Of the two forecasting methods, it is the Ensemble Method by Stacking strategy that obtains the best results. The fact that the Ensemble Method has really good result is confirmed by the comparison with the four methods of the state of the art. Therefore, to have the best results, it is recommended to use the Ensemble Method. The computation time is also a crucial factor: where the Evolutionary Algorithm method trains only one model, the Ensemble Method trains four. In both cases, the electricity consumption of the previous 24 hours at least is required to have good results.

Conclusion

The ability to predict electricity consumption in a short-term is a real financial and environmental benefit. From a financial point of view, companies supplying electricity can predict with good accuracy future consumption and thus avoid losses by not producing too much. The fact that these companies produce only what is necessary allows unnecessary pollution avoidance.

Through this thesis, two forecasting methods were offered and analysed as a response for short-term electricity consumption forecasting problem. Both methods and their operation were explained. Finally, the results they obtain was presented and a comparison between each other and with other methods was offered.

The first method proposed is an Evolutionary Algorithm for regression trees. The second is an Ensemble Method by Stacking strategy using three base algorithms: Random Forest, Artificial Neural Networks and Evolutionary Algorithm for regression trees and a generalize method: Gradient Boost Model.

The purpose of this thesis was to design a methodology to obtain the best prediction of Spanish electricity consumption. Of the two proposed methods, the Ensemble Method of Stacking strategy is closer to this goal. The Evolutionary Algorithm for regression trees provided good result too, but compared to the Ensemble Method by Stacking or the four methods presented in the state of the art it is the Ensemble Method that is the most accurate.

The Ensemble Method could be improved by adding or replacing some method in the base algorithms. For example, replace the Evolutionary Algorithm for regression trees by a method based on the deep learning method [110] or support vector machines. Or keep the Evolutionary Algorithm and add one of these methods or both of them.

Using other datasets/time series of electricity consumption would also be a good way to ensure that this methodology can be generalized to this type of problem.

Bibliography

[1] *Directive 2009/28/EC of the European parliament and of the council of 23 April 2009 on the promotion of the use of energy from renewable sources and amending and subsequently repealing directives 2001/77/EC and 2003/30/ECOff.* J. Eur. Union, L140. 2009.

[2] P. J. Brockwell and R. A. Davis. *Introduction to Time Series and Forecasting.* Heidelberg, Germany: Springer, 2002.

[3] Balakrishnan Narayanaswamy, T. S. Jayram, and Voo Nyuk Yoong. "Hedging strategies for renewable resource integration and uncertainty management in the smart grid". In: *3rd IEEE PES Innovative Smart Grid Technologies Europe, ISGT.* 2012, pp. 1–8. DOI: 10.1109/ISGTEurope.2012.6465718. URL: https://doi.org/10.1109/ISGTEurope.2012.6465718.

[4] R. Haque et al. "Smart management of PHEV and renewable energy sources for grid peak demand energy supply". In: *2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT).* 2015, pp. 1–6. DOI: 10.1109/ICEEICT.2015.7307497.

[5] Muhammad Qamar Raza and Abbas Khosravi. "A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings". In: *Renewable and Sustainable Energy Reviews* 50 (2015), pp. 1352 –1372. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2015.04.065. URL: http://www.sciencedirect.com/science/article/pii/S1364032115003354.

[6] J. Nowicka-Zagrajek and R. Weron. "Modeling electricity loads in California: ARMA models with hyperbolic noise". In: *Signal Processing* 82.12 (2002), pp. 1903 –1915. ISSN: 0165-1684. DOI: https://doi.org/10.1016/S0165-1684(02)00318-3. URL: http://www.sciencedirect.com/science/article/pii/S0165168402003183.

[7] Shyh Jier Huang and Kuang Rong Shih. "Short-term load forecasting via ARMA model identification including non-Gaussian process considerations". In: *IEEE Transactions on Power Systems* 18.2 (May 2003), pp. 673–679. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2003.811010.

[8] Rafal Weron. *Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach.* John Wiley & Sons, 2007.

[9] James Durbin and Siem Jan Koopman. *Time Series Analysis by State Space Methods.* Clarendon Press, 2001.

[10] D.C. Montgomery, C.L. Jennings, and M. Kulahci. *A regression-based approach to short-term system load forecasting*. Wiley Series in Probability and Statistics. Wiley, 2015. ISBN: 9781118745229. URL: https://books.google.be/books?id=pwzQBwAAQBAJ.

[11] R. Weron. *Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach*. The Wiley Finance Series. Wiley, 2007. ISBN: 9780470059999. URL: https://books.google.be/books?id=cXcWdMgovvoC.

[12] Alex D. Papalexopoulos and Timothy C. Hesterberg. *Introduction to Time Series Analysis and Forecasting*. IEEE, 1989. URL: https://ieeexplore.ieee.org/document/39025/.

[13] Kelly Kissock and John Seryak. "Understanding manufacturing energy use through statistical analysis". In: *Mechanical and Aerospace Engineering Faculty Publications* ().

[14] Anzar Mahmood Anila Kousar Muhammad Yamin Younis Bilal Akbar Ghulam Qadar Chaudhary Khuram Pervez Amber Muhammad Waqar Aslam and Syed Kashif Hussain. "Energy Consumption Forecasting for University Sector Buildings". In: *Energies* (2017). URL: http://www.mdpi.com/1996-1073/10/10/1579.

[15] Junwei Miao. "The Energy Consumption Forecasting in China Based on ARIMA Model". In: (). URL: https://www.researchgate.net/publication/301369875_The_Energy_Consumption_Forecasting_in_China_Based_on_ARIMA_Model.

[16] R.C. Souza P.M. Maçaira and F.L. Cyrino Oliveira. "Modelling and Forecasting the Residential Electricity Consumption in Brazil with Pegels Exponential Smoothing Techniques". In: *Procedia Computer Science* ().

[17] Kevin L. Priddy and Paul E. Keller. *Artificial Neural Networks: An Introduction*. SPIE Press, 2005.

[18] Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer Science & Business Media, 2008.

[19] Riccardo Bonetto and Michele Rossi. "Machine Learning Approaches to Energy Consumption Forecasting in Households". In: *CoRR* abs/1706.09648 (2017). URL: http://arxiv.org/abs/1706.09648.

[20] Krzysztof Gajowniczek and Tomasz Ząbkowski. "Short Term Electricity Forecasting Using Individual Smart Meter Data". In: *Procedia Computer Science* 35 (2014). Knowledge-Based and Intelligent Information & Engineering Systems 18th Annual Conference, KES-2014 Gdynia, Poland, September 2014 Proceedings, pp. 589 –597. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2014.08.140. URL: http://www.sciencedirect.com/science/article/pii/S1877050914011053.

[21] Z. Min and P. Qingle. "Very Short-Term Load Forecasting Based on Neural Network and Rough Set". In: *Intelligent Computation Technology and Automation, International Conference on(ICICTA)*. Vol. 03. May 2010, pp. 1132–1135. DOI: 10.1109/ICICTA.2010.38. URL: doi.ieeecomputersociety.org/10.1109/ICICTA.2010.38.

[22] Agoston E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. SpringerVerlag, 2003. ISBN: 3540401849.

[23] Martin Sewell. *Ensemble Learning*. Tech. rep. 2008.

[24] Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. 1st. Chapman & Hall CRC, 2012. ISBN: 1439830037, 9781439830031.

[25] Cha Zhang and Yunqian Ma. *Ensemble Machine Learning: Methods and Applications*. Springer Publishing Company, Incorporated, 2012. ISBN: 1441993258, 9781441993250.

[26] Ian H. Witten and Eibe Frank. *DATA MINING, Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005.

[27] João Mendes-Moreira et al. "Ensemble Approaches for Regression: A Survey". In: *ACM Comput. Surv.* 45.1 (Dec. 2012), 10:1–10:40. ISSN: 0360-0300. DOI: 10.1145/2379776.2379786. URL: http://doi.acm.org/10.1145/2379776.2379786.

[28] Goméz-Vela Francisco García Torres Miguel Divina Federico Gilson Aude and Torres José F. "Stacking Ensemble Learning for Short-Term Electricity Consumption Forecasting". In: *Energies* 11.4 (2018).

[29] Richard Weber. *Time series*. Faculty of Mathematics. Last modified: 17 January 2007, Accessed: 2018-04-07. URL: http://www.statslab.cam.ac.uk/~rrw1/timeseries/t.pdf.

[30] Rob J Hyndman. *Cyclic and seasonal time series*. Accessed: 2018-04-07. URL: https://robjhyndman.com/hyndsight/cyclicts/.

[31] Francisco Martínez-Álvarez et al. "A Survey on Data Mining Techniques Applied to Electricity-Related Time Series Forecasting". In: *Energies* 8.11 (2015), pp. 13162–13193. ISSN: 1996-1073. DOI: 10.3390/en81112361. URL: http://www.mdpi.com/1996-1073/8/11/12361.

[32] Serkan Usanmaz Janset Kuvulmaz and Seref Naci Engin. "Time-Series Forecasting by Means of Linear and Nonlinear Models". In: *MICAI 2005: Advances in Artificial Intelligence* (2005), pp. 504–513.

[33] A. Galka. *Topics in Nonlinear Time Series Analysis: With Implications for EEG Analysis*. Advanced series in nonlinear dynamics. World Scientific, 2000. ISBN: 9789810241483. URL: https://books.google.be/books?id=JSfqwnIFXOEC.

[34] Junjing Yang Siew Eang Lee Chirag Deb Fan Zhang and Kwok Wei Shah. "A review on time series forecasting techniques for building energy consumption". In: *Renewable and Sustainable Energy Reviews* 74 (2017), pp. 902–924.

[35] Mohammad Azhar Mat Dauta et al. "Building electrical energy consumption forecasting analysis using conventional and artificial intelligence methods: A review". In: *Renewable and Sustainable Energy Reviews* 70 (2017), pp. 1108–1118.

[36] Peter J. Brockwell and Richard A. Davis. *Time Series: Theory and Methods*. Springer Science & Business Media, 2013.

[37] Wei-Yin Loh. "Classification and regression trees". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1.1 (2011), pp. 14–23. ISSN: 1942-4795. DOI: 10.1002/widm.8. URL: http://dx.doi.org/10.1002/widm.8.

[38] *scikit-learn - 1.10. Decision Trees*. http://scikit-learn.org/stable/modules/tree.html. Accessed: 2018-04-11.

[39] L. Breiman et al. *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.

[40] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN: 1-55860-238-0.

[41] Lars Kai Hansen and Peter Salamon. "Neural network ensembles". In: *IEEE transactions on pattern analysis and machine intelligence* 12.10 (1990), pp. 993–1001.

[42] Iris Ginzburg and David Horn. "Combined neural networks for time series analysis". In: *Advances in Neural Information Processing Systems*. 1994, pp. 224–231.

[43] Michael P Perrone and Leon N Cooper. "When networks disagree: Ensemble methods for hybrid neural networks". In: *How We Learn; How We Remember: Toward An Understanding Of Brain And Neural Systems: Selected Papers of Leon N Cooper*. World Scientific, 1995, pp. 342–358.

[44] Thomas G. Dietterich. "Ensemble Methods in Machine Learning". In: *Proceedings of the First International Workshop on Multiple Classifier Systems*. MCS '00. London, UK, UK: Springer-Verlag, 2000, pp. 1–15. ISBN: 3-540-67704-6. URL: http://dl.acm.org/citation.cfm?id=648054.743935.

[45] Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. CRC Press, 2012.

[46] Saurav Kaushik. *How to build Ensemble Models in machine learning? (with code in R)*. Accessed: 2018-04-20. URL: https://www.analyticsvidhya.com/blog/2017/02/introduction-to-ensembling-along-with-implementation-in-r/.

[47] Benoît Frénay. *Machine Learning: Lesson 9 - Ensemble Methods: from Single Models to Ensemble of Models*. Faculty of Computer Science.

[48] M. Ranjan and V. K. Jain. "Modelling of electrical energy consumption in Delhi". In: *Energy* 24.4 (1999), pp. 351–361.

[49] Javier Campillo et al. "Energy Demand Model Design For Forecasting Electricity Consumption And Simulating Demand Response Scenarios In Sweden". In: *International Conference on Applied Energy*. 2012.

[50] Ana Medina, Ángeles Cámara, and José-Ramón Monrobel. "Measuring the Socioeconomic and Environmental Effects of Energy Efficiency Investments for a More Sustainable Spanish Economy". In: *Sustainability* 8.10 (2016), pp. 1–21.

[51] Pasapitch Chujai, Nittaya Kerdprasop, and Kittisak Kerdprasop. "Time Series Analysis of Household Electric Consumption with ARIMA and ARMA Models". In: *International MultiConference of Engineers and Computer Scientists*. Vol. 1. 2013.

[52] Mohanad S.Al-Musaylhab et al. "Short-term electricity demand forecasting with MARS, SVR and ARIMA models using aggregated demand data in Queensland, Australia". In: *Advanced Engineering Informatics* 35 (2018), pp. 1–16.

[53] G. P Zhang. "Time series forecasting using a hybrid ARIMA and neural network model". In: *Neurocomputing* 50 (2003), pp. 159–175.

[54] D. W. Schrock and D. E. Claridge. "Predicting Energy Usage in a Supermarket". In: *6th Symposium on Improving Building Systems in Hot and Humid Climates*. Dalas, USA, 1989.

[55] J. C. Lam et al. "Multiple regression models for energy use in air-conditioned office buildings in different climates". In: *Energy Conversion and Management* 51.12 (2010), pp. 2692–2697.

[56] M. R. Braun, H. Altan, and S. B. M. Beck. "Using regression analysis to predict the future energy consumption of a supermarket in the UK". In: *Applied Energy* 130 (2014), pp. 305–313.

[57] Mohammad Mottahedia et al. "Multi-linear Regression Models to Predict the Annual Energy Consumption of an Office Building with Different Shapes". In: *Procedia Engineering* 118 (2015), pp. 622–629.

[58] J. Nizami and A. Z. Ai-Garni. "Forecasting electric energy consumption using neural networks". In: *Energy Policy* 23 (1995), pp. 1097–1104.

[59] S. Kelo and S. Dudul. "A wavelet Elman neural network for short-term electrical load prediction under the influence of temperature". In: *International Journal of Electrical Power & Energy Systems* 43 (2012), pp. 1063–1071.

[60] H. Chitsaz et al. "Short-term electricity load forecasting of buildings in microgrids". In: *Energy Build* 99 (2015), pp. 50–60.

[61] Huiting Zheng, Jiabin Yuan, and Long Chen. "Short-Term Load Forecasting Using EMD-LSTM Neural Networks with a Xgboost Algorithm for Feature Importance Evaluation". In: *Energies* 10.8 (2017), pp. 1168–1188.

[62] Ricardo L. Talavera-Llames et al. "A Nearest Neighbours-Based Algorithm for Big Time Series Data Forecasting". In: *Hybrid Artificial Intelligent Systems: 11th International Conference, HAIS 2016, Seville, Spain, April 18-20, 2016, Proceedings*. Ed. by Francisco Martínez-Álvarez et al. Cham: Springer International Publishing, 2016, pp. 174–185. ISBN: 978-3-319-32034-2. DOI: 10.1007/978-3-319-32034-2_15. URL: https://doi.org/10.1007/978-3-319-32034-2_15.

[63] J. F. Torres et al. "Deep Learning-Based Approach for Time Series Forecasting with Application to Electricity Load". In: *Biomedical Applications Based on Natural and Artificial Computing: International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2017, Corunna, Spain, June 19-23, 2017, Proceedings, Part II*. Ed. by José Manuel Ferrández Vicente et al. Cham: Springer International Publishing, 2017, pp. 203–212. ISBN: 978-3-319-59773-7. DOI: 10.1007/978-3-319-59773-7_21. URL: https://doi.org/10.1007/978-3-319-59773-7_21.

[64] Jian Zheng et al. "Electric load forecasting in smart grids using Long-Short-Term-Memory based Recurrent Neural Network". In: *51st Annual Conference on Information Sciences and Systems (CISS)*. 2017.

[65] Antonio Galicia et al. "Scalable Forecasting Techniques Applied to Big Electricity Time Series". In: *Advances in Computational Intelligence: 14th International Work-Conference on Artificial Neural Networks, IWANN 2017, Cadiz, Spain, June 14-16, 2017, Proceedings, Part II*. Ed. by Ignacio Rojas, Gonzalo Joya, and Andreu Catala. Cham: Springer International Publishing, 2017, pp. 165–175. ISBN: 978-3-319-59147-6. DOI: 10.1007/978-3-319-59147-6_15. URL: https://doi.org/10.1007/978-3-319-59147-6_15.

[66] Mauro Castelli, Leonardo Vanneschi, and Matteo De Felice. "Forecasting short-term electricity consumption using a semantics-based genetic programming framework: The South Italy case". In: *Energy Economics* 47.C (2015), pp. 37–41. URL: https://EconPapers.repec.org/RePEc:eee:eneeco:v:47:y:2015:i:c:p:37-41.

[67] Mauro Castelli, Leonardo Trujillo, and Leonardo Vanneschi. "Energy Consumption Forecasting Using Semantic-based Genetic Programming with Local Search Optimizer". In: *Intell. Neuroscience* 2015 (Jan. 2015), 57:57–57:57. ISSN: 1687-5265. DOI: 10.1155/2015/971908. URL: https://doi.org/10.1155/2015/971908.

[68] E. M. Burger and S. J. Moura. "Gated ensemble learning method for demand-side electricity load forecasting". In: *Energy and Buildings* 109 (2016), pp. 23–34.

[69] S. Papadopoulos and I. Karakatsanis. "Short-term electricity load forecasting using time series and ensemble learning methods". In: *2015 IEEE Power and Energy Conference at Illinois (PECI)*. 2015, pp. 1–6.

[70] S. Li, L. Goel, and P. Wang. "An ensemble approach for short-term load forecasting by extreme learning machine". In: *Applied Energy* 170 (2016), pp. 22–29.

[71] Francisco Martínez-Álvarez et al. "A Survey on Data Mining Techniques Applied to Electricity-Related Time Series Forecasting". In: *Energies* 8 (2015), pp. 13162–13193.

[72] Mohammad Azhar Mat Daut et al. "Building electrical energy consumption forecasting analysis using conventional and artificial intelligence methods: A review". In: *Renewable and Sustainable Energy Reviews* 70 (2017), pp. 1108–1118.

[73] Heiko Hahn, Silja Meyer-Nieberg, and Stefan Pickl. "Electric load forecasting methods: Tools for decision making". In: *European Journal of Operational Research* 199.3 (2009), pp. 902 –907. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2009.01.062. URL: http://www.sciencedirect.com/science/article/pii/S0377221709002094.

[74] Hesham K. Alfares and Mohammad Nazeeruddin. "Electric load forecasting: literature survey and classification of methods". In: *International Journal of Systems Science* 33 (2002), pp. 23–34.

[75] Yi-Shian Lee and Lee-Ing Tong. "Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming". In: *Knowledge-Based Systems* 24.1 (2011), pp. 66 –72. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2010.07.006. URL: http://www.sciencedirect.com/science/article/pii/S0950705110001127.

[76] J.W. Marangon-Lima C. Unsihuay-Vila A.C. Zambroni de Souza and P.P. Balestrassi. "Electricity demand and spot price forecasting using evolutionary computation combined with chaotic nonlinear dynamic model". In: *International Journal of Electrical Power & Energy Systems* 32 (2010), pp. 108–116.

[77] A. Azadeh and S. Tarverdian. "Integration of genetic algorithm, computer simulation and design of experiments for forecasting electrical energy consumption". In: *Energy Policy* 35.10 (2007), pp. 5229 –5241. ISSN: 0301-4215. DOI: https://doi.org/10.1016/j.enpol.2007.04.020. URL: http://www.sciencedirect.com/science/article/pii/S0301421507001668.

[78] Harun Kemal Ozturk et al. "Electricity estimation using genetic algorithm approach: a case study of Turkey". In: *Energy* 30.7 (2005), pp. 1003 –1012. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2004.08.008. URL: http://www.sciencedirect.com/science/article/pii/S0360544204003391.

[79] Halim Ceylan and Harun Kemal Ozturk. "Estimating energy demand of Turkey based on economic indicators using genetic algorithm approach". In: *Energy Conversion and Management* 45.15 (2004), pp. 2525 –2537. ISSN: 0196-8904. DOI: https://doi.org/10.1016/j.enconman.2003.11.010. URL: http://www.sciencedirect.com/science/article/pii/S0196890403003479.

[80] Xueheng Qiu et al. "Ensemble deep learning for regression and time series forecasting". In: *Computational Intelligence in Ensemble Learning (CIEL), 2014 IEEE Symposium on*. IEEE. 2014, pp. 1–6.

[81] Lin-peng Jin and Jun Dong. "Ensemble deep learning for biomedical time series classification". In: *Computational intelligence and neuroscience* 2016 (2016).

[82] Ratnadip Adhikari. "A neural network based linear ensemble framework for time series forecasting". In: *Neurocomputing* 157 (2015), pp. 231–242.

[83] Anthony Bagnall et al. "Time-series classification with COTE: the collective of transformation-based ensembles". In: *IEEE Transactions on Knowledge and Data Engineering* 27.9 (2015), pp. 2522–2535.

[84] Snehamoy Chatterjee, Ansuman Dash, and Sukumar Bandopadhyay. "Ensemble support vector machine algorithm for reliability estimation of a mining machine". In: *Quality and Reliability Engineering International* 31.8 (2015), pp. 1503–1516.

[85] Zhi-Hua Zhou. "Ensemble learning". In: *Encyclopedia of biometrics* (2015), pp. 411–416.

[86] Rui Zhang et al. "Short-term load forecasting of Australian National Electricity Market by an ensemble model of extreme learning machine". In: *IET Generation, Transmission & Distribution* 7.4 (2013), pp. 391–397.

[87] Zhongfu Tan et al. "Day-ahead electricity price forecasting using wavelet transform combined with ARIMA and GARCH models". In: *Applied Energy* 87.11 (2010), pp. 3606–3610.

[88] Shu Fan et al. "Forecasting electricity demand by hybrid machine learning model". In: *Neural Information Processing*. Springer. 2006, pp. 952–963.

[89] Sumaira Tasnim et al. "Wind power prediction using cluster based ensemble regression". In: *International Journal of Computational Intelligence and Applications* (2017), p. 1750026.

[90] Jorjeta G Jetcheva, Mostafa Majidpour, and Wei-Peng Chen. "Neural network model ensembles for building-level electricity load forecasts". In: *Energy and Buildings* 84 (2014), pp. 214–223.

[91] Radiša Ž Jovanović, Aleksandra A Sretenović, and Branislav D Živković. "Ensemble of various neural networks for prediction of heating energy consumption". In: *Energy and Buildings* 94 (2015), pp. 189–199.

[92] *The R Project for Statistical Computing*. `https://www.r-project.org/`. Accessed: 2018-04-24.

[93] Vincent Goulet. *Introduction à la programmation en R*. École d'actuariat. Accessed: 2018-04-23. URL: `https://cran.r-project.org/doc/contrib/Goulet_introduction_programmation_R.pdf`.

[94] Ross Ihaka and Robert Gentleman. "R: A Language for Data Analysis and Graphics". In: *Journal of Computational and Graphical Statistics* 5.3 (1996), pp. 299–314. DOI: `10.1080/10618600.1996.10474713`. URL: `https://www.stat.auckland.ac.nz/~ihaka/downloads/R-paper.pdf`.

[95] B. Lantz. *Machine Learning with R*. Packt Publishing, 2015. ISBN: 9781784394523. URL: `https://books.google.be/books?id=ZaJNCgAAQBAJ`.

[96] S.V. Burger. *Introduction to Machine Learning with R: Rigorous Mathematical Analysis*. O'Reilly Media, 2018. ISBN: 9781491976395. URL: `https://books.google.be/books?id=buhPDwAAQBAJ`.

[97] Thomas Grubinger, Achim Zeileis, and Karl-Peter Pfeiffer. "evtree: Evolutionary Learning of Globally Optimal Classification and Regression Trees in R". In: *Journal of Statistical Software* 61.1 (2014), pp. 1–29. URL: `http://www.jstatsoft.org/v61/i01/`.

[98] R. E. Uhrig. "Introduction to artificial neural networks". In: *Industrial Electronics, Control, and Instrumentation, 1995., Proceedings of the 1995 IEEE IECON 21st International Conference on*. Vol. 1. 1995, 33–37 vol.1. DOI: 10.1109/IECON.1995.483329.

[99] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Fourth. New York: Springer, 2002.

[100] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32. ISSN: 0885-6125. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

[101] Ramón Díaz-Uriarte and Sara Alvarez de Andrés. "Gene selection and classification of microarray data using random forest". In: *BMC Bioinformatics* 7.1 (2006), p. 3.

[102] Pall Oskar Gislason, Jon Atli Benediktsson, and Johannes R Sveinsson. "Random forests for land cover classification". In: *Pattern Recognition Letters* 27.4 (2006), pp. 294–300.

[103] Andy Liaw and Matthew Wiener. "Classification and Regression by randomForest". In: *R News* 2.3 (2002), pp. 18–22. URL: http://CRAN.R-project.org/doc/Rnews/.

[104] Jerome H. Friedman. "Greedy Function Approximation: A Gradient Boosting Machine". In: *Annals of Statistics* 29 (2000), pp. 1189–1232.

[105] Jerome H. Friedman. "Stochastic gradient boosting". In: *Computational Statistics & Data Analysis* 38.4 (2002), pp. 367–378.

[106] C. Sammut and G.I. Webb. *Encyclopedia of Machine Learning*. Encyclopedia of Machine Learning. page 331. Springer US, 2011. ISBN: 9780387307688. URL: https://books.google.be/books?id=i8hQhp1a62UC.

[107] R.F. Schaefer et al. *Information Theoretic Security and Privacy of Information Systems*. Definition page 87. Cambridge University Press, 2017. ISBN: 9781107132269. URL: https://books.google.be/books?id=t5okDwAAQBAJ.

[108] Jason Brownlee. *Overfitting and Underfitting With Machine Learning Algorithms*. Accessed: 2018-04-24. URL: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/.

[109] Greg Ridgeway. *Generalized boosted models: A guide to the gbm package*. 2005.

[110] Ian Goodfellow et al. *Deep learning*. Vol. 1. MIT press Cambridge, 2016.

A | Code source of the EA method

The function takes two parameters: the problem number corresponding to the column to be predicted and w i.e. the history window

```
#library use
library(evtree)
library(rpart.plot)
require(reshape)

#Retrieving arguments that is the problem number and the
    historical window
args = commandArgs(trailingOnly=TRUE)

if (length(args)==0) {
  msg <- "At least two argument must be supplied : the first is
      the number of the problem the second the historical window
      .\n"
  stop(msg, call.=FALSE)
}

numProb = as.integer(args[1])
print(numProb)

w = as.integer(args[2])
h = 24
print(w)

#get the matrix corresponding to the historical window
nameData <- paste("./dataHW",w,"PW",h,".csv",sep = "")
matrix <- read.csv(file=nameData,head=TRUE,sep=",",
    stringsAsFactors=F)
matrix$X<- NULL

#Split the matrix to data_training and data_test
split = 0.7
corte = floor(split*nrow(matrix))
data_evtree_training = matrix[1:corte,]
data_evtree_test = matrix[(corte+1):nrow(matrix),]
```

```
data_evtree_training <- as.data.frame(data_evtree_training)
data_evtree_test <- as.data.frame(data_evtree_test)

#Make sure we only use the historical window
data_p_training <- data_evtree_training[,1:w]
data_p_test <- data_evtree_test [,1:w]


####define the column to predict####

numPredic <- w+numProb
colPredic <- paste("X",numPredic, sep ="")
data_p_training$X169 <- data_evtree_training[[colPredic]]
data_p_test$X169 <- data_evtree_test[[colPredic]]

####Evtree treatment####
#Define the parameter for evtree

minbucket = 8L
minsplit = 100L
maxdepth = 15L
ntrees = 300L
controlEv <- evtree.control(minbucket = minbucket, minsplit =
    minsplit, maxdepth = maxdepth,niterations = 1000L, ntrees =
    ntrees, alpha =0.25, operatorprob = list(pmutatemajor = 0.2,
     pmutateminor = 0.2, pcrossover = 0.8, psplit= 0.2, pprune =
     0.4), seed = NULL)
res <- c()
res <- c(format(Sys.time(), "%d-%b-%Y %H.%M"))

#create the model on the training data (predict X169 in function
     of the historical window)
Evtree_model_p <- evtree(X169 ~.,data = data_p_training, control
    = controlEv)

res <- c(res,(format(Sys.time(), "%d-%b-%Y %H.%M")))


#use the model to predict X169 on the test data
Evtree_prediction_p <- predict(Evtree_model_p,data_p_test)
nbRow <- nrow(data_p_test)
```

```
#Compute the mean relative error(mre)
mreEVTREE_p <- (sum(abs(data_p_test$X169 - Evtree_prediction_p)
    / data_p_test$X169)) / nrow(data_p_test)*100

#save the evtree model, the prediction and the mre in a R object
nameModel <- paste("./modelEvtree_HW",w,"PW",h,"_p",numProb,".
    rda", sep ="")
save(Evtree_model_p, Evtree_prediction_p,mreEVTREE_p, file=
    nameModel)

#print the time of start and end of the creation of evtree model
     and the mre
res <- c(res,mreEVTREE_p)
nameResult <- paste("./result_HW",w,"PW",h,"_p",numProb,".txt",
    sep = "")
write.table(res, sep =" ", eol="\n", nameResult)

#Do the plot real and predict data
require(ggplot2)
x_evtreeV3<-c(1:nbRow)
y1_evtreeV3 <- data_p_test$X169
y2_evtreeV3 <- Evtree_prediction_p

dfV3<-data.frame(x_evtreeV3,y1_evtreeV3,y2_evtreeV3)

g_evtreeV3 <- ggplot(dfV3, aes(x_evtreeV3))
g_evtreeV3  <- g_evtreeV3  + geom_line(aes(y=y1_evtreeV3),
    colour="blue")
g_evtreeV3  <- g_evtreeV3  + geom_line(aes(y=y2_evtreeV3),
    colour="green",alpha=0.8)

#Save the plot
nameplot <- paste("./evtreePlot_HW",w,"PW",h,"_p",numProb,".pdf"
    ,sep = "")
pdf(nameplot)
plot(g_evtreeV3)
dev.off()
```

B | Code source of the ensemble method

```
library(readr)
library(evtree)
library(rpart)
library(ggplot2)
library("mlbench")
library("randomForest")
library("nnet")
library('caret')
library("gbm")

#Retrieving arguments that is the problem number and the
    historical window
args = commandArgs(trailingOnly=TRUE)
if (length(args)==0) {
  msg <- "At least two argument must be supplied : the first is
      the number of the problem the second the historical window
      .\n"
  stop(msg, call.=FALSE)
}


numProb = as.integer(args[1])
print(numProb)

w = as.integer(args[2])
h = 24
print(w)


#get the matrix corresponding to the historical window
nameData <- paste("./dataHW",w,"PW",h,".csv",sep = "")
matrix <- read.csv(file=nameData,head=TRUE,sep=",",
    stringsAsFactors=F)
matrix$X<- NULL
#Split the matrix to data_training and data_test
split = 0.7
```

```
corte = floor(split*nrow(matrix))
data_evtree_training = matrix[1:corte,]
data_evtree_test = matrix[(corte+1):nrow(matrix),]

data_evtree_training <- as.data.frame(data_evtree_training)
data_evtree_test <- as.data.frame(data_evtree_test)

#Make sure we only use the historical window
data_p_training <- data_evtree_training[,1:w]
data_p_test <- data_evtree_test [,1:w]

#saving the predictors to be used (needed for NN)
predictors = names(data_p_training)
####define the column to predict####

numPredic <- w+numProb
colPredic <- paste("X",numPredic, sep ="")
data_p_training$X169 <- data_evtree_training[[colPredic]]
data_p_test$X169 <- data_evtree_test[[colPredic]]

####load EVtree model####
#format is like this: modelEvtree_HW24PW24_p5
EVTreeModelName = paste("modelEvtree_HW",w,"PW24_p",numProb,".
    rda",sep = "")
print(EVTreeModelName)
objectsEVTree=load(EVTreeModelName)
modelEvTree = Evtree_model_p

res <- c()
res <- c(format(Sys.time(), "%d-%b-%Y %H.%M"))
#predict on the training and test sets
nbRowTest <- nrow(data_p_test)
nbRowTrain = nrow(data_p_training)
#use the model to predict X169 on the test data
#Evtree_prediction_p <- predict(Evtree_model_p,data_p_test)
nbRow <- nrow(data_p_test)
predictTrain_evtree = predict(modelEvTree, data_p_training)
predictTest_evtree <- predict(modelEvTree,data_p_test)
MRE_EvTree <- mreEVTREE_p



####RANDOM FOREST PART ####
```

```
#save the RF model, the prediction and the mre in a R object
modelRF = randomForest(X169 ~., data = data_p_training, ntree
    =100,maxnodes = 100, prox=TRUE)
predictTest_RF <- predict(modelRF,data_p_test)
mreRF_p <- (sum(abs(data_p_test$X169 - predictTest_RF) / data_p_
    test$X169)) / nrow(data_p_test)*100

#prediction on training
predictTrainRF = predict(modelRF, data_p_training)

nameModel <- paste("./modelRF_HW",w,"PW",h,"_p",numProb,".rda",
    sep ="")
save(modelRF, predictTest_RF, mreRF_p, file=nameModel)



####NN part####
f <- as.formula(paste("X169 ~", paste(predictors, collapse = " +
    ")))

modelNN <- nnet(f, data_p_training,size=10,linout=TRUE, skip=
    TRUE, MaxNWts=10000, trace=FALSE, maxit=1000)
predictTestNN = predict(modelNN,data_p_test)
mreNN_p <- (sum(abs(data_p_test$X169 - predictTestNN) / data_p_
    test$X169)) / nrow(data_p_test)*100

#prediction on training
predictTrainNN = predict(modelNN, data_p_training)

nameModel <- paste("./modelNN_HW",w,"PW",h,"_p",numProb,".rda",
    sep ="")
save(modelNN, predictTestNN, mreNN_p, file=nameModel)
####ADD PREDICTION TO TEST AND TRAINING####
#add prediction to test data
data_p_test$Pred_EV = predictTest_evtree
data_p_test$Pred_RF = predictTest_RF
data_p_test$Pred_NN = predictTestNN


#add prediction to training and test sets for later steps
data_p_training$Pred_EV = predictTrain_evtree
data_p_training$Pred_RF = predictTrainRF
data_p_training$Pred_NN = predictTrainNN
```

```
####TOP LAYER OF ENSEMBLE####
#use lm to build emsamble, check it out, it's too simple now
#model_lm = lm(X169 ~ Pred_EV + Pred_RF + Pred_NN, data=data_p_
    training, na.action = NULL)
model_gbm = gbm(X169 ~ Pred_EV + Pred_RF + Pred_NN, data = data_
    p_training,distribution = "gaussian", n.trees = 5,
    interaction.depth = 8, shrinkage = 1, n.minobsinnode = 1)
#predict
predictTest_gbm <- predict(model_gbm,data_p_test, n.trees = 5)
mreGBM_p <- (sum(abs(data_p_test$X169 - predictTest_gbm) / data_
    p_test$X169)) / nrow(data_p_test)*100

nameModel <- paste("./modelGBM_HW",w,"PW",h,"_p",numProb,".rda",
     sep ="")
save(model_gbm, predictTest_gbm, mreGBM_p, file=nameModel)

#saving time
res <- c(res,(format(Sys.time(), "%d-%b-%Y %H.%M")))

name <- c("evtree","rf","GBM", "nnet")
MRE <- c(MRE_EvTree,mreRF_p,mreGBM_p,mreNN_p)
ch <- paste(name, " : ", MRE, sep ="")
res <- c(res,ch)

nameResult <- paste("./result_HW",w,"PW",h,"_p",numProb,".txt",
    sep = "")
write.table(res, sep =" ", eol="\n", nameResult)
print(mreGBM_p)
print(mreNN_p)
print(mreRF_p)
```

<fe