UNIVERSITY OF THE PELOPONNESE & NCSR "Demokritos"
MSC PROGRAMME IN DATA SCIENCE

Exploring Machine Learning for Electricity Price Forecasting

by

Anastasios Mastrapas

A thesis submitted in partial fulfillment
of the requirements for the MSc
in Data Science

Supervisor: Iraklis Klampanos
Postdoctoral Research Associate

Athens, July 2021

Exploring Machine Learning for Electricity Price Forecasting

Anastasios Mastrapas

MSc. Thesis, MSc. Programme in Data Science

University of the Peloponnese & NCSR "Demokritos", July 2021

UNIVERSITY OF THE PELOPONNESE & NCSR "Demokritos"
MSC PROGRAMME IN DATA SCIENCE

Exploring Machine Learning for Electricity Price Forecasting

by

Anastasios Mastrapas

A thesis submitted in partial fulfillment
of the requirements for the MSc
in Data Science

Supervisor: Iraklis Klampanos
Postdoctoral Research Associate

Approved by the examination committee on July, 2021.

Iraklis Klampanos Postdoctoral Research Associate	George Petasis Researcher (C)	Konstantinos Limniotis Collaborating Researcher

Athens, July 2021

UNIVERSITY OF THE PELOPONNESE & NCSR "Demokritos"
MSC PROGRAMME IN DATA SCIENCE

Declaration of Authorship

(1) I declare that this thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where states otherwise by reference or acknowledgment, the work presented is entirely my own.

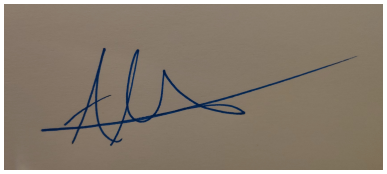
(2) I confirm that this thesis presented for the degree of Master of Science in Data Science, has

(i) been composed entirely by myself

(ii) been solely the result of my own work

(iii) not been submitted for any other degree or professional qualification

(3) I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or processional qualification except as specified.



Anastasios Mastrapas

Athens, July 2021

Acknowledgments

I would first like to thank my thesis advisor Dr. Iraklis-Angelos Klampanos of SKEL at NCSR-Demokritos. The door to Dr. Klampanos office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this paper to be my own work, but steered me in the right direction whenever he thought I needed it.

I would also like to thank all the teaching staff of Msc Programme in Data Science. In particular, I will never forget the passion and the dedication of Dr. George Petasis, Dr. Ioannis Moscholios, Dr. Christos Tryfonopoulos, Dr. George Giannakopoulos and Dr. Theodoros Giannakopoulos. They all contributed in their own way to introduce me to the world of data science.

This thesis was written at a period of grief for me. Hence, I must express my very profound gratitude to my family for providing me with unfailing support and continuous encouragement through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

To the unborn angel.

Abstract

The aim of this thesis is to explore the capabilities of Machine Learning algorithms in the task of electricity price forecasting. The focus is on the Hungarian wholesale electricity market (HUPX), which is considered a benchmark power exchange in the region of SE Europe. Taking advantage of the available domain expertise, a really extended dataset was built, consisting of 69 features. For the scope of this paper, several traditional machine learning algorithms as well as artificial neural networks were implemented, using some well-known python libraries such as `scikit-learn` and `keras`.

Moving from traditional to more sophisticated methods, it turns out that performance is constantly improving. Starting with a MAPE of 15% we managed to get down to the levels of 6% MAPE, thanks to the contribution of artificial neural networks, which proved their capabilities to effectively approximate a mapping function from input variables to output variable. In our effort to quantify the impact of domain expertise on the shaping of the results, a sensitivity analysis was performed, which confirmed the significant contribution of each feature category to improving the performance of the algorithms.

Finally, taking into account the results of other price forecasting studies in the Balkan markets, HUPX is concluded to be the most predictable power exchange, which is probably explained by the greater maturity of this power market.

Contents

List of Tables

List of Figures

List of Abbreviations

EPF	Electricity Price Forecasting
HUPX	Hungarian Power Exchange
ENTSO-E	European Network of Transmission System Operators for Electricity
IEA	International Energy Agency
CET	Central European Time
TTF	Title Transfer Facility
MWh	Megawatt hour
EUA	EU Allowances
TSO	Transmission System Operator
VRE	Variable Renewable Energy
CEE	Central Eastern Europe
SEE	Southern Eastern Europe
RoR	Run-of-River
SVM	Support Vector Machine
SVR	Support Vector Regressor
ARIMA	Auto-Regressive Integrated Moving Average

SOM	Self-Organised Map
RBFN	Radial Basis Function Network
SSA	Singular Spectrum Analysis
WNN	Wavelet Neural Network
CNN	Convolutional Neural Network
LSTM	Long Short Term Memory
DT	Decision Trees
RF	Random Forest
MAE	Mean Absolute Error
BPNN	Back-Propagation Neural-Network
DNN	Deep Neural Network
GRU	Gated Recurrent Unit
CV	Cross-validation
ANN	Artificial neural network
MLP	Multilayer Perceptron

Chapter 1

Introduction

Electricity price forecasting (EPF) is a branch of energy forecasting which focuses on predicting the spot and forward prices in wholesale electricity markets. Since the inception of competitive power markets two decades ago, electricity price forecasting has gradually become a fundamental process for energy companies' decision making mechanisms. Furthermore, the recent penetration of renewables has had the effect of increasing the uncertainty of future supply, demand and prices. All parties involved in the electric industry have come to understand that probabilistic electricity price forecasting is now more important for energy systems planning and operations than ever before. The field of machine learning has developed considerably in recent years, indicating able to bring satisfactory results in tasks such as electricity price forecasting.

1.1 Domain background

Since the early 1990's, the process of deregulation and the introduction of competitive markets have been reshaping the landscape of the traditionally monopolistic and government-controlled power sectors. In many countries worldwide, electricity is nowadays traded under market rules using spot and derivative contracts. However, electricity remains a very special commodity. It is economically non-storable, and power system stability requires a constant balance between production and consumption.

Figure 1.1: Power system stability requires a constant balance between production and consumption

At the same time, electricity demand depends on weather (temperature, wind speed, precipitation, etc.) and the intensity of business and everyday activities (on-peak vs. off-peak hours, weekdays vs. weekends, holidays and near-holidays, etc.). These unique and specific characteristics lead to price dynamics not observed in any other market, exhibiting seasonality at the daily, weekly and annual levels, and abrupt, short-lived and generally unanticipated price spikes. This has encouraged researchers to intensify their efforts in the development of better forecasting techniques.

At the corporate level, electricity price forecasts have become a fundamental input to energy companies' decision making mechanisms. As the California crisis of 2000–2001 [2] showed, electric utilities are the most vulnerable, since they generally cannot pass their costs on to the retail consumers. Extreme price volatility, which can be up to two orders of magnitude higher than that of any other commodity or financial asset, has forced market participants to hedge not only against volume risk but also a against price movements. Price forecasts from a few hours to a few months ahead have become of particular interest to power portfolio managers. A generator, utility company or large industrial consumer who is able to forecast the volatile wholesale prices with a reasonable level of accuracy can adjust its bidding

strategy and its own production or consumption schedule in order to reduce the risk or maximise the profits in day-ahead trading.

1.2 Problem description

Electricity price forecasting focuses on predicting the spot and forward prices in wholesale electricity markets. The problem to be studied in this thesis is to forecast the hourly day-ahead market clearing price of the Hungarian wholesale electricity market, which is announced every day at the Hungarian Power Exchange (HUPX) [3]. HUPX is considered a benchmark power market in the region of SE Europe. Being the oldest one, it is the most mature market providing participants with adequate liquidity on a daily basis. All active traders in the Balkan region try every day to predict the HUPX price in the best possible way.

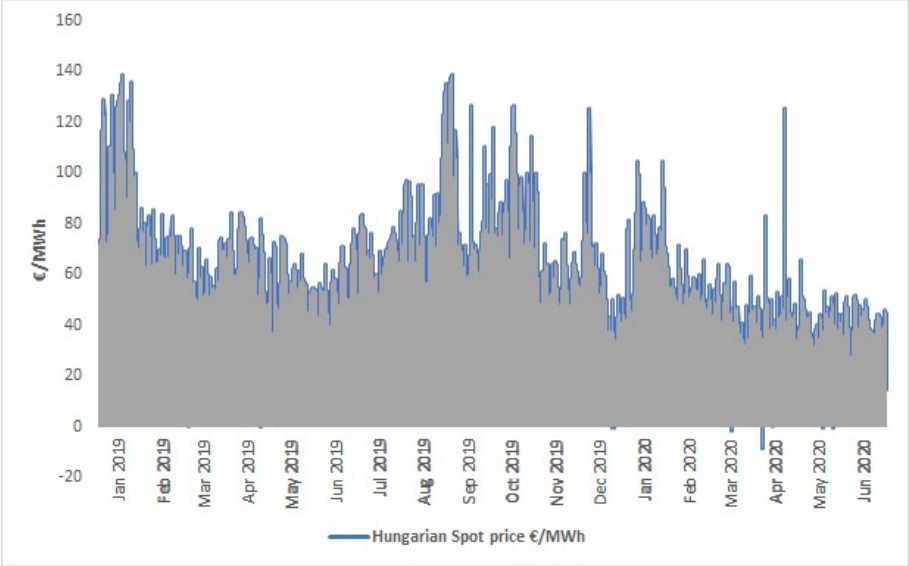


Figure 1.2: Hungarian hourly spot price in EUR/MWh

Both traditional machine learning (e.g. regression trees) and more sophisticated deep-learning algorithms are utilized to establish a forecasting model for wholesale electricity prices with respect to the Hungarian power market on an hourly basis. The study is interested in highlighting how prediction accuracy improves as we move from traditional algorithms to more complex ones and to inform and encourage future ML-based methods. As a policy tool, such models could be used by energy traders, transmission system operators and energy regulators for an enhanced

decision-making process.

1.3 Thesis structure

In the next chapter, after giving some machine learning background, we present the several methods that have been tried for EPF, while extensive reference is made to studies that explore the capabilities of machine learning algorithms in the task of electricity price forecasting.

In Chapter 3 the several dataset components are presented. Specific explanations are provided regarding the feature selection. More specifically 7 categories of features are used, which includes 69 features in total.

In the fourth chapter several experimental runs and the respective results are presented, after being applied to the HUPX electricity price forecasting task. It turns out that moving from traditional to more sophisticated methods, an increasingly better performance is achieved.

Chapter 5 includes on the one hand the conclusions of this work and on the other hand various thoughts for possible extensions.

Chapter 2

Electricity Price Forecasting

2.1 Machine Learning Background

The task of predicting electricity prices belongs to supervised learning, as the training set we feed into the algorithm includes the desired solutions, called labels. Our goal is to predict a numerical target value, such as the value of electricity in HUPX, given a set of features called predictors. This type of work is called regression. To train the system, we must give it many examples, including their predictors and labels.

In this chapter our objective is to present the several algorithms to be used for the task of electricity price forecasting. In section 2.1.1 we describe the characteristics of some of the most popular machine learning algorithms, which are considered sufficient to meet the requirements of a regression task. In section 2.1.2, the most powerful ensemble machine learning algorithms are presented, which are promising as they have been proven effective in a wide range of tasks. In section 2.1.3 the characteristics of Artificial neural networks (ANNs) are given, which are at the very core of Deep Learning. Being powerful they are good at scaling up with samples and at fitting non-linearities in the data.

2.1.1 Traditional Machine Learning Algorithms

Regression algorithms belong to the family of supervised machine learning algorithms which is a subset of machine learning algorithms. One of the main features of supervised learning algorithms is that they model the dependencies and relationships between the target output and the input features to predict the value for new data. Regression algorithms predict output values based on input characteristics from the data fed into the system. According to the prevailing methodology, a model is built by an algorithm using the training data features and then the model is used for predictions on unseen data.

Some of the popular types of regression algorithms are linear regression, decision trees, support vector machines and nearest neighbors.

2.1.1.1 Linear Models

The models presented in this section are a set of algorithms intended for regression in which the target variable is expected to be a linear combination of the features. The mathematical notation, if $\hat{y}$ is the predicted value, is:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + ... + w_p x_p$$

The vector $w = (w_1, ..., w_p)$ is designated as coefficient and w_0 as intercept.

Linear Regression

Linear regression is a statistical approach that enables studying the relationships between continuous variables. It is a linear model that assumes a linear relationship between the input variables (X) and the single output variable (y). Here y can be calculated from a linear combination of input variables (X). When there is an input variable (x), the method is called simple linear regression. When there are many input variables, the process is referred to as multiple linear regression.

In order to minimize the residual sum of squares between the observed targets in the dataset and the targets predicted by the linear approximation, linear regression

fits a linear model with coefficients $w = (w_1, ..., w_p)$. The mathematical expression of the problem to be solved is:

$$\min_w ||Xw - y||_2^2$$

Ridge Regression

Some of the problems of Ordinary Least Squares are addressed by using Ridge regression, which imposes a penalty on the coefficients size. The target is the minimization of a penalized residual sum of squares by the ridge coefficients:

$$\min_w ||Xw - y||_2^2 + \alpha||w||_2^2$$

The amount of shrinkage is controlled by the complexity parameter $\alpha \geq 0$: as the value of α gets larger, the amount of shrinkage gets greater as well and so the coefficients' robustness to collinearity increases.

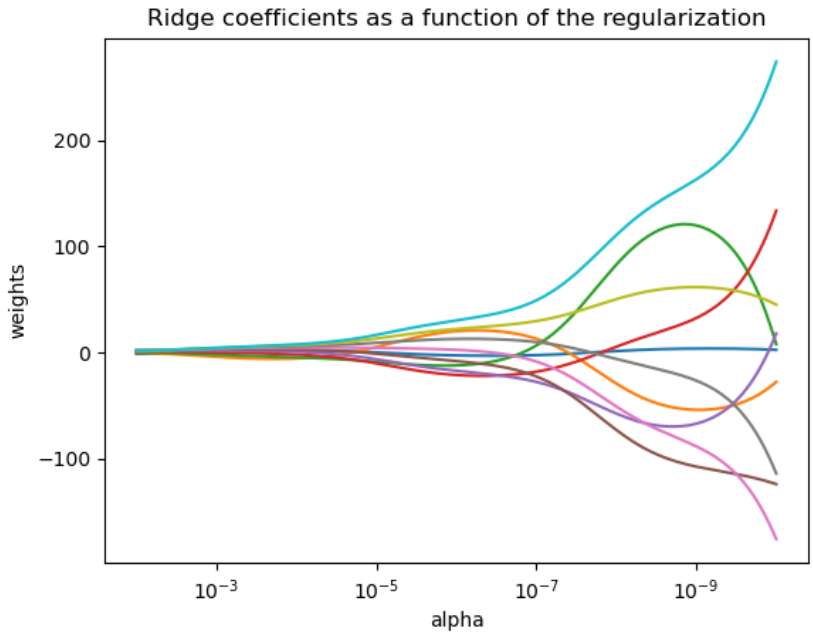


Figure 2.1: Ridge coefficients as a function of the regularization

Lasso

Least Absolute Selection Shrinkage Operator or LASSO algorithm has a constraint on parameters for defining the shrinkage. In order to obtain the subset of predictors that minimizes prediction error for a quantitative response variable, the

algorithm imposes a constraint on the model parameters. As a result, the regression coefficients for some variables shrink to zero.

After the shrinkage process, variables with a regression coefficient equal to zero are excluded from the model. The response variable is finally most strongly associated with Variables with non-zero regression coefficients. This lasso regression analysis, being a shrinkage and variable selection method, helps analysts to determine which of the predictors are most important.

Mathematically, a regularization term is added to a linear model to constitute LASSO model. The objective function for the minimization is as follows:

$$\min_w \frac{1}{2n_{\text{samples}}}||Xw - y||_2^2 + \alpha||w||_1$$

The lasso estimate thus solves the minimization of the least-squares penalty with $\alpha||w||_1$ added, where α is a constant and $||w||_1$ is the ℓ_1-norm of the coefficient vector [4].

Elastic-Net

ElasticNet belongs to linear regression models. Its training is performed with both ℓ_1 and ℓ_2-norm regularization of the coefficients. As a result, a sparse model is learned, where few of the weights are non-zero (like in Lasso case), while still maintaining the regularization properties of Ridge. The convex combination of ℓ_1 and ℓ_2 is controlled using the l1 ratio parameter.

In case of multiple features which are correlated each other, Elastic-net is proved really useful. While Lasso is likely to pick one of these at random, elastic-net will possibly pick both. An advantage of trading-off between Ridge and Lasso is that it allows Elastic-Net to inherit some of Ridge stability under rotation.

The objective function to minimize is in this case:

$$\min_w \frac{1}{2n_{\text{samples}}}||Xw - y||_2^2 + \alpha\rho||w||_1 + \frac{\alpha(1-\rho)}{2}||w||_2^2$$

Due to the fact that Lasso behaves often erratically when several features are strongly correlated or when the number of features is greater than the number of training examples, Elastic-net is preferred in general over Lasso.

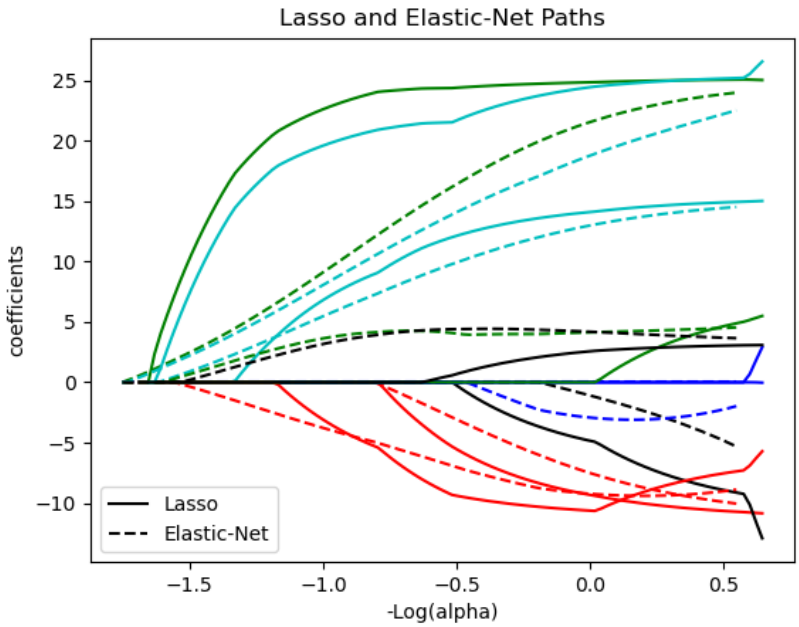


Figure 2.2: Lasso and Elastic-Net paths

2.1.1.2 Decision Trees

Used mainly for regression and classification, Decision Trees are a non-parametric supervised learning algorithm. The target is to build a model that make predictions of the value of a target variable by learning simple decision rules derived from the data features. As the tree gets deeper, the decision rules become more complex and the model better fits.

The use of decision trees implies several advantages. Compared to other algorithms decision trees require little data preparation, while it is considered relatively simple to understand and to interpret them. Furthermore, the cost of using the tree is logarithmic in the number of data points used to train the tree. Moreover, decision trees have capabilities not only to handle both numerical and categorical data but also to manage effectively multi-output problems. On the other hand, **overfitting** lurks in the corner when creating over-complex trees that do not generalise the data well. There are nevertheless solutions such as pruning, which sets either the minimum samples required at a leaf node or the maximum tree's depth in order to avoid this problem. Small variations in the data might result in the development of a completely different tree, while they may create biased trees especially in case of

presence of dominant classes. Hence, balancing the dataset prior to fitting with the decision tree is highly recommended.

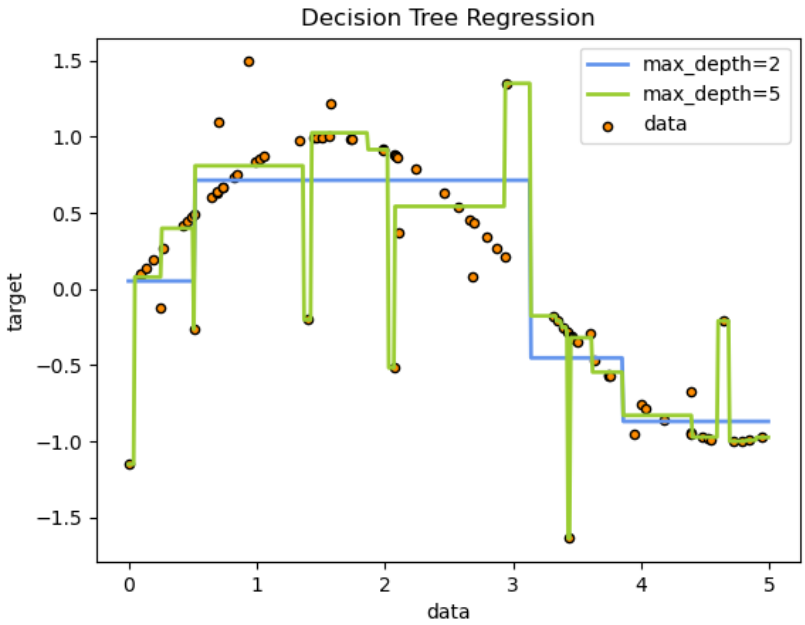


Figure 2.3: Decision Tree Regression

2.1.1.3 Nearest Neighbors

Nearest neighbor methods have the objective to find a predetermined number of training samples closer to the new point and predict the label from them. The number of samples can be defined by the user (k-nearest neighbor learning), or may vary based on the local density of points (radius-based neighbor learning). For defining the distance, any metric measure can be used, with standard Euclidean distance being the most common option. Neighbors-based algorithms simply "remember" all of its training instances, that is why they are well known as non-generalizing machine learning methods.

Two different neighbors regressors are being implemented by scikit-learn: Using **KNeighborsRegressor**, the user specifies an integer k and then the learning process is based on the k nearest neighbors of each query point. Using **RadiusNeighborsRegressor**, the user defines a floating-point value to a variable r and then learning is based on the nearest neighbors within a fixed radius r of the query

point.

Uniform weights are used by the basic nearest neighbors regression. This means that each point in the local neighbourhood contributes uniformly to the classification of a query point. In some cases, it may be advantageous to weight points such that nearby points contribute more to the regression than faraway points. This can be achieved through the **weights** keyword. The default value, **weights = "uniform"**, assigns equal weights to all points. **weights = "distance"** assigns weights depending on the inverse of the distance from the query point. Alternatively, a user-defined function of the distance may be provided, which will be used to calculate the weights.

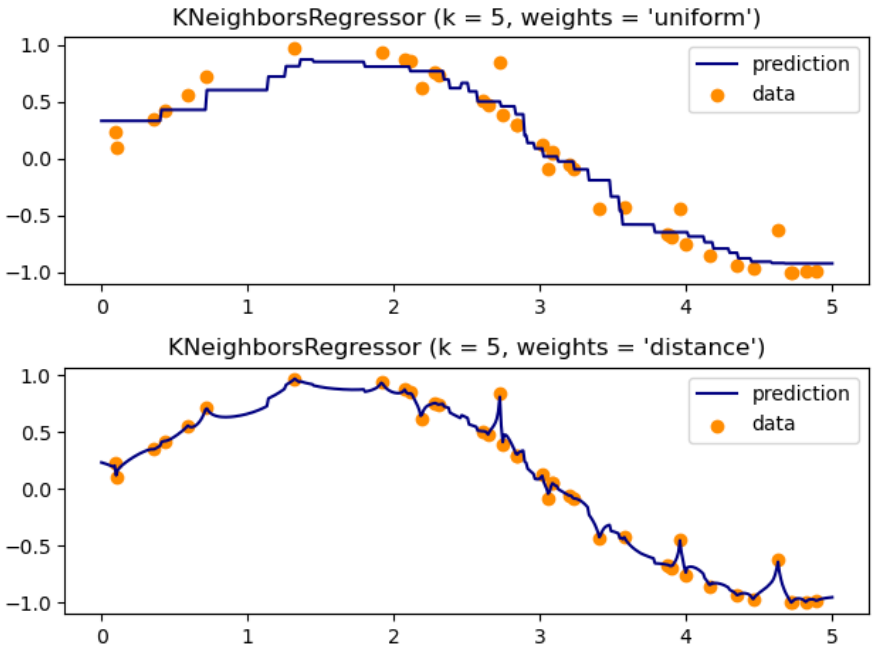


Figure 2.4: Changing value of the "weights" parameter in the KNeighborsRegressor

2.1.1.4 Support Vector Machines

A Support Vector Machine (SVM) is a powerful and flexible machine learning model. It is widely used for performing linear or nonlinear classification, regression and outlier detection.

Use of support vector machines has many advantages, since SVM are effective in high dimensional spaces and also effective in cases where number of dimensions is

greater than the number of samples. SVM method uses a subset of training points in the decision function (called support vectors), so it is also memory efficient. Moreover, they are versatile, since different Kernel functions can be specified for the decision function.

Support Vector Regression, retaining all the significant properties from Support Vector Machines, attempts to find a curve given data points. In SVR, a match is found between some vector and the position on the curve, unlike the classification problem where having the curve act as a decision boundary. Support vectors are also involved in finding the closest match between the data points and the actual function they represent. Intuitively, maximizing the distance between the regressed curve and the support vectors, we get closer to the real curve (we consider always some noise present in the statistical samples).

For the needs of our study, SVR library is used from scikit-learn. The SVR implementation in scikit-learn has a parameter, epsilon, that controls the loss function. Quoting from the documentation, "it specifies the epsilon-tube within which no penalty is associated in the training loss function with points predicted within a distance epsilon from the actual value."

2.1.1.5 Gaussian Processes

Gaussian Processes (GP), being a generic supervised learning method, have been designed to address regression and probabilistic classification problems.

Gaussian processes have the following advantages:

- The forecast is probabilistic (Gaussian) so that one can calculate empirical confidence intervals and make decisions based on those if one should redefine the prediction in some area of interest.
- The prediction interpolates the observations.
- They are versatile, given that different kernels can be specified.

The disadvantages of GPs are:

- They are not sparse, meaning that in order to perform the prediction they use the whole samples/features information.
- In high dimensional spaces they lose efficiency, mainly when the number of features exceeds a few dozen.

The GaussianProcessRegressor from scikit-learn implements Gaussian processes (GP) for regression purposes.

2.1.2 Ensemble Machine Learning Algorithms

Combining the predictions of several basic estimators which are built with a given learning algorithm, ensemble methods have the target to improve generalizability / robustness compared to a single estimator.

There are usually two sets of ensemble methods:

- In averaging methods, the basic principle is to create different estimators independently and then take the average of their predictions. It turns out that the variance of the combined estimator is reduced, hence its performance is usually better than any of the single base estimator. Forests of randomized trees and Bagging methods are examples of averaging methods.
- In boosting methods, a sequential building of the base estimators is followed. Each of them tries to reduce the bias of the combined estimator. According to this approach, a powerful ensemble method is produced by combining several weak models. AdaBoost and Gradient Tree Boosting are examples of boosting methods.

In our exercise, we make use of Random Forest, AdaBoost and Gradient Boosting.

A Random Forest, being a meta estimator, fits a number of decision trees in different sub-sets of the dataset and then uses averaging in order to control overfitting and improve the accuracy of the predictions. An AdaBoost regressor is a meta-estimator that firstly fits a regressor on the original dataset. As a second step,

AdaBoost fits additional copies of the regressor on the same dataset, in which the weights of instances are modified according to the current prediction error. Hence, next regressors focus more on difficult cases. Gradient Boosting, being an additive model in a forward stage-wise fashion, allows for the arbitrary differentiable loss functions to be optimized. Each stage includes a regression tree which is fit on the negative gradient of the given loss function.

2.1.3 Deep Learning Algorithms

Artificial Neural Networks (ANNs) are at the core of Deep Learning. Being powerful, versatile and scalable, they are ideal to tackle large and complex Machine Learning tasks such as classifying millions of images (e.g. Google Images [5]), providing speech recognition services (e.g. Apple's Siri [6]), recommending to hundreds of millions of users the best videos to watch on a daily basis (e.g. Youtube) or learning to beat the world champion in the Go game (DeepMind's AlphaGo [7]).

Deep learning neural networks have several interesting capabilities, since they can support multiple inputs and outputs and automatically learn arbitrary complex mappings from inputs to outputs. Such powerful features sounds a lot of promising for time series predictions, especially on tasks with complex nonlinear dependencies, multivariate inputs and forecasting with multiple steps. These features together with the capabilities of state-of-the-art neural networks may offer great expectations such as the native support for sequence data provided by recurrent neural networks and the automatic feature learning in convolutional neural networks.

The Multilayer Perceptron or MLP is an artificial neural network and its job is to approximate a mapping function from input variables to output variables. This general ability is valuable for time series tasks for several reasons:

- Robust to Noise: ANNs show robustness to noise in the mapping function and in input data and even in the presence of missing values they can support learning and prediction.
- Nonlinear: ANNs do not make strong assumptions about the mapping function and easily learn linear and non-linear relationships.

More specifically, ANNs can provide direct support to **multivariate inputs**, since they can be configured to effectively approximate an arbitrary defined but fixed number of inputs and outputs in the mapping function. Specifying an arbitrary number of input features, they provide direct support for multivariate forecasting. They also support **multi-step forecasts**. An arbitrary number of output values can be specified, providing direct support for multi-step and even multivariate forecasting. For all these capabilities, feedforward neural networks are useful for time-series forecasting.

2.2 Electricity Price Forecasting in Literature

A variety of methods and ideas have been tried for Electricity Price Forecasting over the last 15 years, with varying degrees of success. They can be broadly classified into six groups.

- **Multi-agent models:** Multi-agent models simulate the operation of a system with heterogeneous agents (generating units, retail companies), who interact with each other. These models approach the price process by matching the supply and demand in the market [8]. This class includes equilibrium or game theoretic approaches (like the Nash-Cournot framework, supply function equilibrium), cost-based models and agent-based models. In general, multi-agent models focus less on quantitative results rather than on qualitative issues. They pose problems if more quantitative conclusions have to be drawn, especially if electricity prices need to be predicted with a high degree of accuracy.

  In the **Nash-Cournot** framework, electricity is treated as a homogeneous commodity and the balance of the market is determined through the capacity determination decisions of the suppliers. Unfortunately, these models tend to provide prices higher than are actually observed. By introducing the concept of conjectural variations, researchers have addressed this problem. This idea aims to capture the fact that rivals react to high electricity prices by increasing their production. The results show that the expected values of prices increase by a significant amount, as the number of firms in the market decreases.

Supply function equilibrium method performs the price modelling as the equilibrium of companies bidding with supply curves (and possibly demand curves) into the wholesale market. A set of differential equations need to be solved for the calculation of the supply function equilibrium or SFE. Thus, these models have significant limitations regarding their numerical detectability. One way to make faster computations is to aggregate the demand into blocks. But this method would leave out of the analysis the extreme values, which is considered not acceptable when focusing on electricity price forecasting or risk management. Moreover, only if the demand uncertainty, or another source of uncertainty, leads to an ex-ante undetermined equilibrium, then the supply curve bidding will lead to results which differ from the Nash-Cournot equilibrium. Otherwise, the supply bidding collapses to a point, which corresponds to the Nash-Cournot equilibrium. Linear SFE models have been proposed so far, aiming to reduce the numerical complexity of the general SFE models. In such models, demand and marginal costs are expected to be *linear*, while the SFE can be obtained either in terms of linear or affine supply functions. All market participants receive the marginal clearing price for their supply. Under such market clearing conditions, the social welfare is maximized as long as there is no transmission congestion, since the supply functions are non-decreasing and the price that clears the market is identical for all companies. This framework has limited application to electricity price forecasting, but on the contrary it has been widely used for the analysis of bidding strategies, market design, market power and congestion management.

In case of **strategic production-cost model** or SPCM, agents' bidding strategies are taken into account based on conjectural variation. Each producer seeks to maximize its profits, taking into account on one hand the structure of its cost and on the other hand the expected behavior of the other market players, which is modelled through a parameter that represents the slope of the residual demand function for each level of the production. During the simulation of the supply curve building process, the SPCM makes the assumption that the agent is simply aware of its cost and its conjecture about the deriva-

tive of its residual demand function. As there are no iterations, companies do not have the opportunity to improve their bids and take into account the reactions of their competitors, as in the case of SFE models. SPCM is suitable for real-time analysis, because its main advantage is its computational speed, in comparison to the Nash-Cournot and SFE models.

In energy economics, **agent-based** computational economics (ACE) has proved to be a widely used approach to solving both practical and theoretical tasks over the last two decades. A class of computational rules and structures is the basic tool of ACE, so as the interactions of autonomous agents (either individuals or groups/organizations) to be simulated, with the ultimate goal being to assess their impact on the whole system.

Multi-agent models are considered, on the one hand, a class of extremely flexible tools for analyzing strategic behavior in electricity markets. This freedom is also a weakness, on the other hand, as it requires the simulation assumptions to be justified, both empirically and theoretically. Some elements need to be defined, such as the players, the way in which they interact, their potential strategies and the set of payoffs. Apparently, there is a significant risk of modelling.

- **Fundamental models:** Fundamental methods aim to capture the basic physical and economic relationships that exist in the trading and production of electricity [9]. Functional correlations among fundamental drivers (load, system parameters, weather data, etc.) are supported and fundamental inputs are formulated and predicted independently, often through computational intelligence, statistical or reduced-form techniques. In general, two subcategories of fundamental models can be identified: **parsimonious structural models and parameter rich models of demand and supply**.

  Two important challenges arise in the practical application of fundamental models. **Data availability** is the first challenge. Depending on the market, more or less information (e.g. plant capacities, costs, demand patterns, transmission capacities, etc.) is available to the researcher or professional to

build such a model. Due to the nature of fundamental data, pure fundamental models are more suitable for medium-term forecasts than short-term ones. This also applies to the parsimonious structural models, which are usually calibrated to daily data ignoring the fine relationships with hourly resolution. In general, their application is limited to derivatives pricing and risk management. **Incorporation of stochastic fluctuations of the fundamental drivers** is the second challenge. During the construction of the model, specific assumptions are made about the physical and economic relationships in the market, and therefore the price forecasts produced by the models are very sensitive to violations of these assumptions. In addition, the more detailed the model, the more effort is required to adjust the parameters. Consequently, in the application of the fundamental approach there is a significant modelling risk.

- **Reduced-form models:** The statistical properties of electricity prices over time are characterized by the reduced-form models, which have as ultimate goal the risk management and the valuation of derivatives. [10].

  A common characteristic of the finance-inspired reduced form models of price dynamics is that their main purpose is not to provide accurate price forecasts on a hourly basis, but rather to reproduce the main features of daily electricity prices, such as marginal distributions in future time points, price dynamics, and correlations between commodity prices. Such models are at the core of risk management and derivatives pricing systems. It is very possible to have unreliable model results, if the selected pricing procedure is not suitable for capturing the main features of electricity prices. At the same time, if the complexity of the model is too high, then the computational burden will inevitably prevent its use in trading departments. On the one hand, the tools used are rooted in methods developed to model other energy products or interest rates. On the other hand, they incorporate actuarial or econometric approaches. It can be said that the Markov regime-switching models and the jump-diffusion models are able to combine the two worlds: in order to capture the unique characteristics of electricity prices, they are trade-offs between model adequacy

and parsimony.

In general, reduced-form models are not expected to accurately predict hourly prices, but are expected to recover the main features of spot electricity prices, usually on a daily time scale. They are commonly used for risk analysis and derivatives pricing, providing a simplified but reasonably realistic picture of price dynamics. Interestingly, in terms of forecasts of volatility or price spikes, the reduced form models have been reported to perform quite well.

- **Statistical models:** Statistical methods predict the current prices using a mathematical combination of the previous prices and values of exogenous factors, such as consumption, production or weather data. [1]. Statistical models allow engineers and system operators to understand their behavior, because some physical interpretation can be attached to their components. Criticism is frequently exercised for their poor performance to adequately represent the usual non-linear behavior of electricity prices and related fundamental variables. However, in practical applications, their accuracy is comparable to that of non-linear alternatives.

  Additive and **multiplicative** models are the two most important categories. In case of additive models, the predicted price is the sum of a number of components, while multiplicative models calculate the predicted price as the product of some factors. Additive models are much more popular, although the two types of models are closely related, because a multiplicative model for prices can be converted to an additive model for log-prices.

  Statistical methods include **similar-day and exponential smoothing models**, **regression models**, **threshold autoregressive models**, **AR-type time series models**, **ARX-type time series models** and **heteroskedasticity and GARCH-type models**.

  Statistical models are often classified by some authors as technical analysis tools. Technical analysts do not try to estimate an asset's intrinsic or fundamental value. Instead, looking at price charts for indicators and patterns, they will determine the future performance of an asset. In financial markets, the

effectiveness and usefulness of technical analysis is frequently questioned, but in electricity markets is more likely to have a better chance, due to the seasonality of electricity price processes during normal, non-spiky periods. When trying to predict spikes, however, statistical methods have a rather poor performance. This applies mainly to price-only models, but even models with fundamental variables do not perform well. In the literature it is not clear if price spikes need to be part of the estimation procedure of statistical models, although there is no doubt that price spikes should be captured using an appropriate stochastic model.

- **Computational intelligence (CI) models:** Computational intelligence techniques include artificial intelligence-based, machine learning, non-parametric or non-linear statistical methods. These techniques, combining elements of learning, fuzziness and evolution, are regarded as "intelligent" because they are able to adapt to complex dynamic systems. Artificial neural networks [11], support vector machines (SVM) and fuzzy systems are undoubtedly the main categories of computational intelligence techniques in the task of electricity price forecasting.

  The ability of computational intelligence methods to handle non-linearity and complexity proves to be their major strength. Hence, CI methods are better than the statistical techniques at modelling these features of electricity prices. This flexibility is also, at the same time, their main weakness. Being able to adapt to nonlinear, spiky behaviors does not necessarily lead to better point forecasts. In addition, all the available CI tools are so diverse that it is difficult to find an optimal solution. Moreover, it is difficult to compare the various CI methods in detail. Even if the forecasting accuracy is reported for the same test period and the same market, the errors of the individual techniques are not really comparable, and therefore they cannot be used to make general statements about the effectiveness of a method. Instead, conclusions can only be drawn for the performance of a specific implementation of a method, with certain parameters and for a given calibration dataset. Although this criticism is not limited to CI methods, in their case it is especially true due to their

multiparameter specifications and their non-linearity.

- **Hybrid models:** It is very common in the literature to meet hybrid solutions trying to approach the task of electricity price forecasting. Such approaches combine techniques from two or more of the methods listed above, while their classification is non-trivial, if possible at all. As an example of hybrid model, AleaModel (developed by AleaSoft) combines Neural Networks and Box Jenkins models [12].

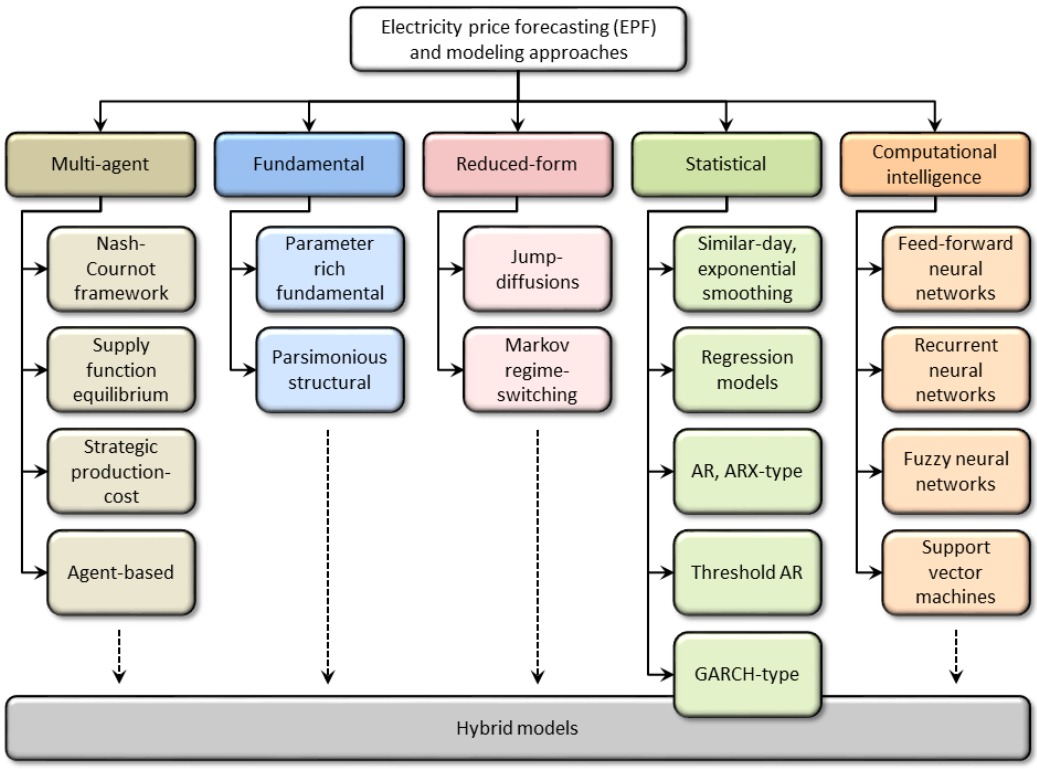


Figure 2.5: A taxonomy of electricity price forecasting and modeling approaches according to Weron [1]

2.3 ML in Electricity Price Forecasting

According to [1], statistical and machine learning methods have been shown to work best in the task of electricity price forecasting. A disadvantage of statistical models is that they are usually linear forecasters, and therefore, they may not perform well in data where the frequency is high, e.g. hourly data with rapid fluctuations. According to [13], while statistical techniques show good performance if

the data frequency is low, e.g. weekly patterns, the nonlinear behavior of hourly prices can become too complex to predict. Different machine learning methods have been proposed, in order to address this issue and predict the nonlinear behavior of hourly prices. The academic literature includes a large collection of machine learning approaches, from more traditional to more sophisticated techniques, while often combining two or more methods to produce a hybrid model. It turns out that more complicated algorithms (e.g. deep learning methods) achieve usually better performance compared to classic ML techniques, which in turn are proved more effective than the statistical methods.

Support Vector Machines (SVMs) belong to traditional ML algorithms, which are quite popular to researchers, who often test their effectiveness in the task of electricity price forecasting. In [14], a hybrid model is proposed that combines both auto-regressive integrated moving average (ARIMA) and SVR models to take advantage of the unique power of ARIMA and SVR models in linear and nonlinear modelling. SVR is used to capture non-linear patterns. The experimental results show that the proposed model outperforms the traditional ARIMA models based on the mean absolute percentage error. [15] proposes a method for short-term electricity price forecasting based on a two-stage hybrid network of support-vector machine (SVM) and self-organised map (SOM). In the first step, the input data are grouped into multiple subsets in an unsupervised manner, using a SOM network. Then in the second stage, a group of SVMs is used to fit the training data of each subset in a supervised way. To confirm its effectiveness, the proposed model was trained and tested on historical energy price data from the New England electricity market.

Over the last few years, emphasis has been placed on the use of artificial neural networks (ANNs), exploring their potential for better electricity price forecasting. Researchers often combine different architectures in order to achieve the optimal result. [16] proposed an Enhanced Radial Basis Function Network (ERBFN), which combines the Radial Basis Function Network (RBFN) and Orthogonal Experimental Design (OED), applied in the PJM area of United States. By applying OED to ERBFN learning rates, the prediction error can be reduced during the training

process to improve both accuracy and reliability. This would mean that even the "spikes" could be closely monitored. The simulation results showed the effectiveness of the proposed ERBFN for providing quality information in a volatile price environment. In [17], a hybrid model for short-term electricity price forecasting is proposed, based on modified wavelet neural network (WNN) and singular spectrum analysis (SSA). The proposed algorithm optimizes the initial weights and the parameters of dilation and translation in WNN, while for the evaluation of the method case studies of half-hourly electricity price data are applied. [18] proposes an electricity price forecasting approach, combining two deep neural networks, the Long Short-Term Memory (LSTM) and the Convolutional Neural Network (CNN). The results of the experiments showed that compared to other traditional machine learning methods (e.g. SVM, DT, RF), the performance of the model's prediction proves to be better. More specifically, the proposed algorithm achieved MAE 8.85, while DT and RF achieved MAE 9.74 and 9.20 respectively. The paper uses PJM Zone data to perform the training of the model. In [19], LSTM with the differential evolution (DE) algorithm, which is designated as DE-LSTM, is used to predict electricity prices. DE is designed to identify suitable hyperparameters for LSTM. Experiments are performed to verify the performance of the DE–LSTM model, building the necessary datasets with the electricity prices in New South Wales, France. and Germany/Austria. Results indicate that the proposed DE–LSTM model outperforms both statistical and other ANN methods. The proposed algorithm achieves MAPE of 9.52%, while its competitors, BPNN and SVM, achieve MAPE of 14.94% and 12.09% respectively. In [20], different deep learning models are proposed for predicting electricity prices, leading to improvements in predictive accuracy. Deep learning forecasters include the deep neural network (DNN), the LSTM and the gated recurrent unit (GRU). All of them are shown to achieve a predictive accuracy that is statistically significantly better than traditional models. The empirical study is being conducted in the day-ahead market of Belgium. The proposed architectures outperform the statistical methods (MAPE>15%) achieving a MAPE of 12-13%.

In the region of SE Europe, there are not many studies that have been performed to explore the effectiveness of machine learning algorithms in the task of electricity

price forecasting. In [21], 24-hours ahead forecasting of the electricity price in the newly established power exchanges of SE Europe is performed, using simple artificial neural networks. The study focuses on the Bulgarian, Croatian and Serbian power markets. The results show that among the three neighboring markets, the price in Bulgarian power exchange is the most unpredictable one (MAPE 21%), while the price in the Serbian power exchange is the most predictable one (MAPE 9.28%). The price in the Croatian power exchange is in the middle, achieving a MAPE of 17%.

Chapter 3

Data Analysis and Input Variable Selection

Our goal is to build a really extended dataset, taking advantage of the available domain expertise. We aim to include every possible driver behind the formulation of the HUPX electricty price. In general, day-ahead electricity prices are affected by fuel prices, level of consumption, production level of renewables (mainly photovoltaics and wind energy), availability of conventional units (e.g. nuclear plants, CCGTs), available capacity in neighboring interconnections and fundamentals of nearby electric systems. The final dataset includes seven feature categories which have 69 features in total. The usual drivers that are used in similar studies include mainly consumption and RES production data. Therefore, we expect that the more sophisticated content of our dataset will contribute to the performance of the forecasts.

3.1 Target Variable

Our target variable is the Hungarian electricity spot price in euros per MWh (EUR/MWh), as announced on a daily basis from the Hungarian Power Exchange [3]. The study period is from January 1st, 2019 to June 30th, 2020. The following table give us some descriptive statistics, that summarize the central tendency, dispersion and shape of the target variable's distribution.

Count	13128
mean	44.77
standard deviation	19.45
min	-8.12
25th percentile	31.85
50th percentile	42.30
75th percentile	56.22
max	138.82

Table 3.1: Target variable descriptive statistics

The boxplot below give us some further statistical information regarding the Hungarian wholesale electricity price. The horizontal axis is the hour of the day, while the vertical one is the electricity price in EUR/MWh. As can be seen, during off-peak hours (hours 1-7) prices are usually lower, while during peak-hours prices are not only higher but also more volatile.

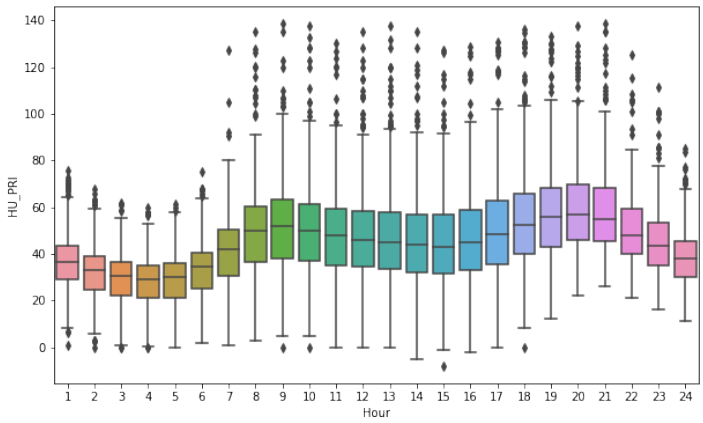


Figure 3.1: Boxplot: HUPX spot price per hour of a day

Apart from their volatility during the day, electricity prices are also sensitive to other factors, like the electricity demand, which is usually higher during summer and winter months. The heatmap that follows presents the hourly average spot price in HUPX for each of the 18 months of the study period, starting from January 2019.

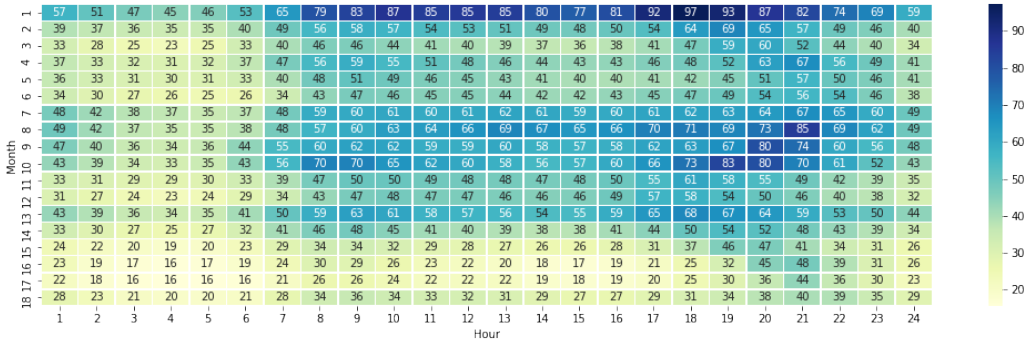


Figure 3.2: Heatmap: Hourly Average HUPX spot price per month

3.2 Input Variables

Variable selection is a very important topic in the electricity price forecasting. There are a lot of factors that have, more or less, impact on the electricity price. The wholesale price of electricity is affected by other prices (e.g. gas price, carbon price), the seasonality in consumption, the level of production of the several technologies (e.g. RES, nuclear) and the available capacity in the interconnections with the neighbouring countries.

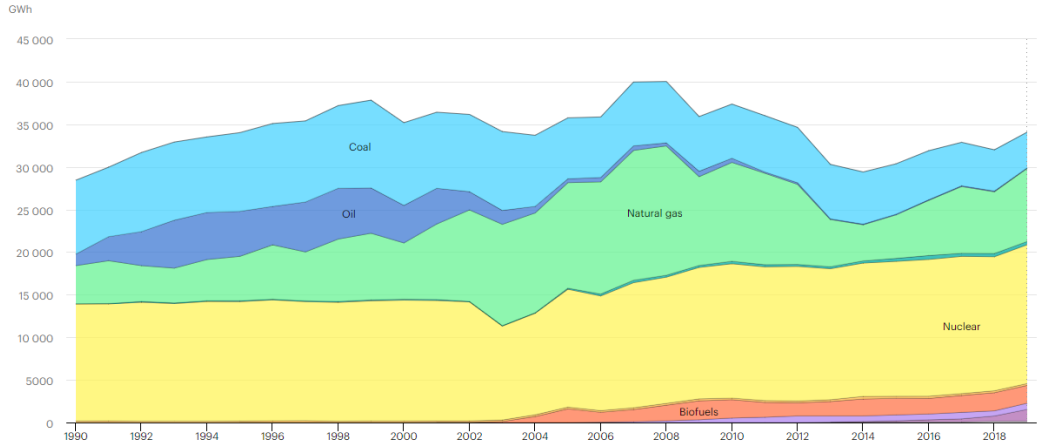


Figure 3.3: Electricity generation by source, Hungary 1990-2019 (Source: IEA)

In Hungary, the electricity production remained stable in recent years at around 30 TWh per year, mainly based on nuclear energy, which provides approximately half of the electricity produced in the country. The generation from gas and coal follows it in importance. In 2019, almost 90% of all the electricity generated was

produced by these three technologies.

On the other hand, despite representing only 2.5% of the total production of the year 2019, it is worth noting the rapid increase in recent years in solar energy genaration.

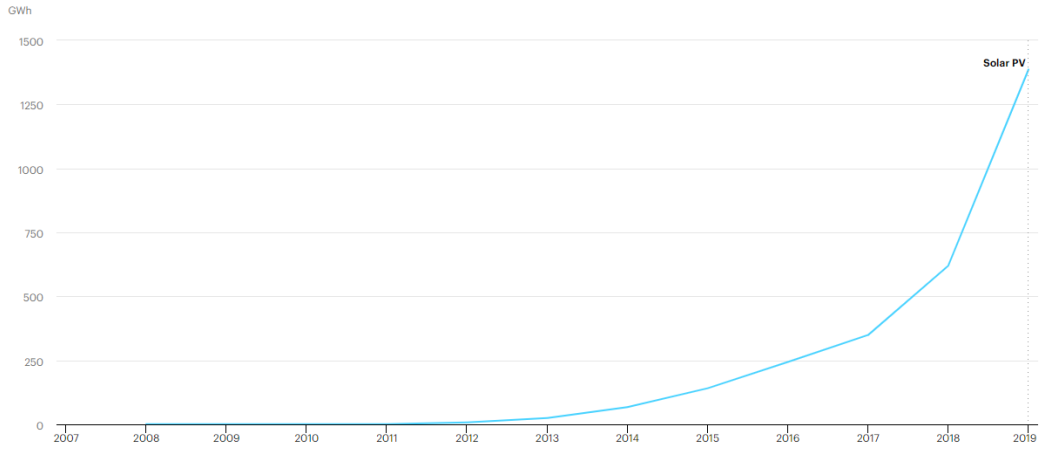


Figure 3.4: Solar PV electricity generation, Hungary 2007-2019 (Source: IEA)

Hungary's electricity demand increased by 4.0% in the last four years, from 41.8 TWh in 2016 to 43.5 TWh in 2019. However, the country's electricity production is enough to cover up to 75% of its electricity demand. This places Hungary as a major importer of electricity to cover a very significant part of its demand.

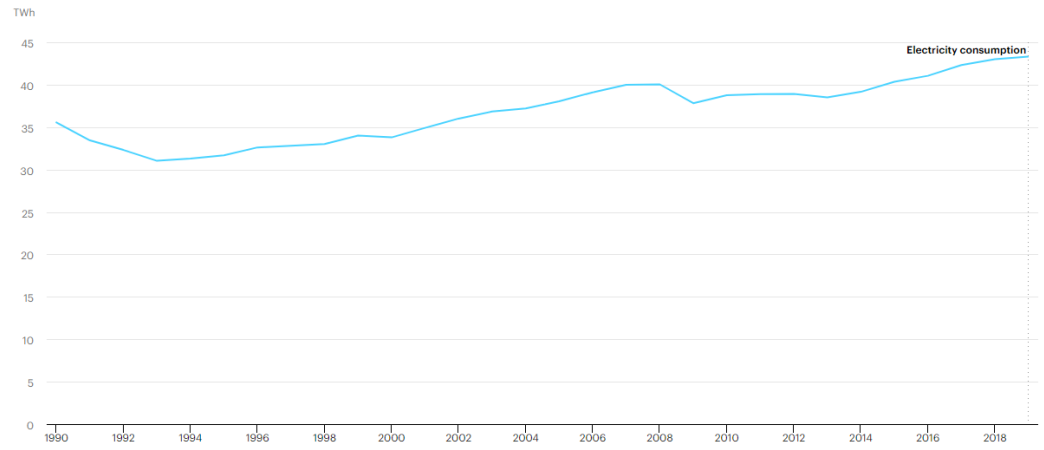


Figure 3.5: Electricity consumption, Hungary 1990-2019 (Source: IEA)

In 2019 Hungary imported 12.7 TWh more of electricity than it exported. The electricity imports of Hungary mainly came from Slovakia, Austria and Ukraine, which contributed 45%, 32% and 20%, respectively, to the 19.5 TWh of electricity

imported that year. In contrast, Croatia, Romania and Serbia received 61%, 28% and 11%, respectively, of Hungary's electricity exports.

In this study, we have chosen seven categories of features, each of which has several variables that can contribute to the forecast of the target variable. Most of the data are available on the website of the European Network of Transmission System Operators for Electricity (ENTSO-E)[22]. In the table 3.2, the several types of features are presented.

Feature category	Number of features
General	2
Prices	2
Consumption	9
Residual Load	6
Production	8
Cross-border HU	14
Cross-border	28

Table 3.2: Feature Categories

3.2.1 "General" feature category

The **"General"** feature category includes 2 features:

- Weekday: This feature represents the day of the week and takes values from 1 (for Sunday) to 7 (for Saturday).
- Hour: This feature represents the hour of the day and takes values from 1 to 24. The relative timezone is the Central European Time (CET).

The figure below shows that during Saturday and Sunday we usually have lower prices in the Hungarian wholesale electricity market.

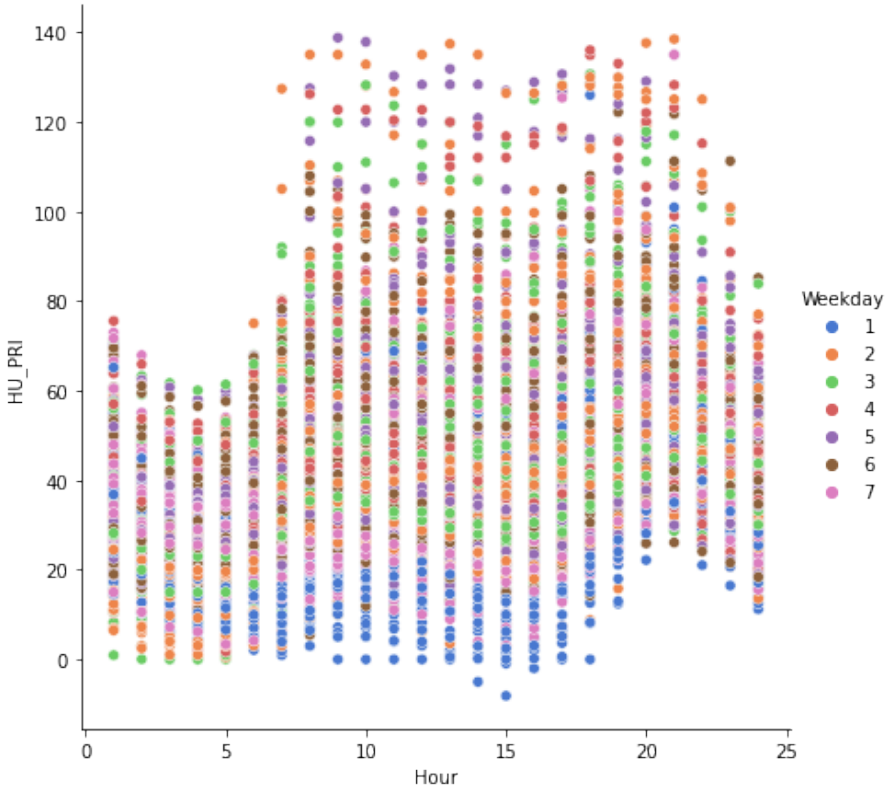


Figure 3.6: HUPX spot price (EUR/MWh) per hour and per weekday

3.2.2 "Prices" feature category

The **"Prices"** feature category includes 2 features:

- TTF: This feature represents the TTF price, which reflects the cost of the gas-fired power plants in the whole Europe. TTF is a virtual trading point

for natural gas in the Netherlands. Such an enormous volume of gas is traded using TTF that it has become a benchmark hub for gas prices in Europe. Even the rest of the world keeps a close eye on what is happening at TTF. Twice as much gas is traded through the TTF hub as on all other continental European gas trading platforms combined. Every single day, more than 100 gas traders and financial parties can be found buying and selling large volumes of gas on this virtual exchange. Gas at TTF trades in euros per megawatt hour, while data are available on the website of ICE Exchange[23].

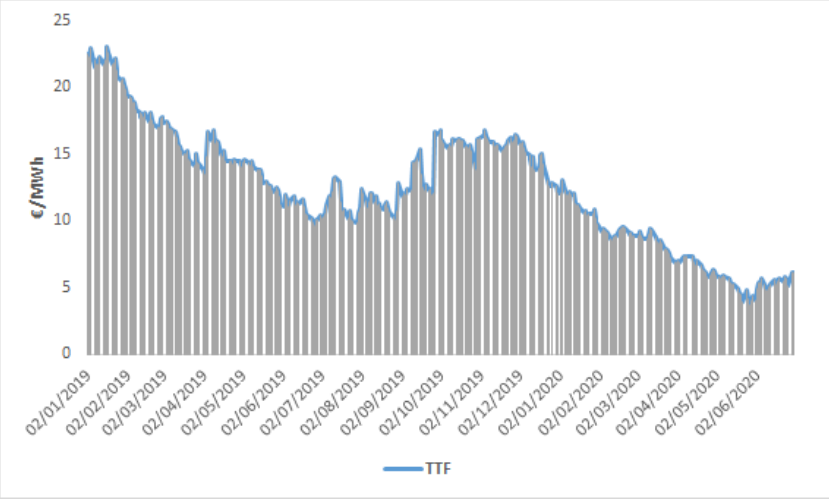


Figure 3.7: Evolution of TTF price

- EUA: EU Allowances (EUA) are climate credits (or carbon credits) used in the European Union Emissions Trading Scheme (EU ETS)[24]. EU Allowances are issued by the EU Member States into Member State Registry accounts. By April 30 of each year, operators of installations covered by the EU ETS must surrender an EU Allowance for each ton of CO2 emitted in the previous year. The emission allowance is defined in Article 3(a) of the EU ETS Directive as being "an allowance to emit one tonne of carbon dioxide equivalent during a specified period, which shall be valid only for the purposes of meeting the requirements of this Directive and shall be transferable in accordance with the provisions of this Directive". EUAs, that affect the cost of all the thermal power-plants, are traded in euros per ton of CO2, while data are available on the website of ICE Exchange[25].

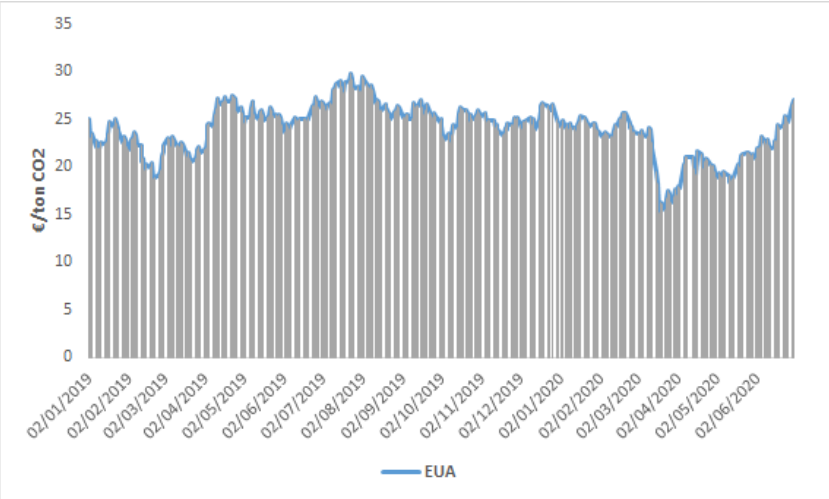


Figure 3.8: Evolution of EUA price

3.2.3 "Consumption" and "Residual Load" feature categories

The **"Consumption"** feature category is related to the electric load, which is considered among the main drivers that affect the formation of the electricity price.

Feature	Description
HU - CON	Hourly Load of Hungary
DE - CON	Hourly Load of Germany
CZ - CON	Hourly Load of Czech Republic
SK - CON	Hourly Load of Slovakia
AT - CON	Hourly Load of Austria
SI - CON	Hourly Load of Slovakia
HR - CON	Hourly Load of Croatia
RS - CON	Hourly Load of Serbia
RO - CON	Hourly Load of Romania

Table 3.3: "Consumption" feature category

This feature category includes 9 features, that corresponds to the consumption of Hungary, Germany, Czech Republic, Slovakia, Austria, Slovenia, Croatia, Serbia and Romania, as presented on Table 3.3.

It is considered as a common practice, in the daily business of the power markets, for the TSOs to provide their view of the hourly System Load of the next day. Market participants take this information into account in order to make their projections for the day-ahead level of electricity prices. Load and electricity price have a positive

correlation, as Figure 3.9 shows. Hourly Load is measured in MW, while all data for this feature category are available on ENTSO-E website.

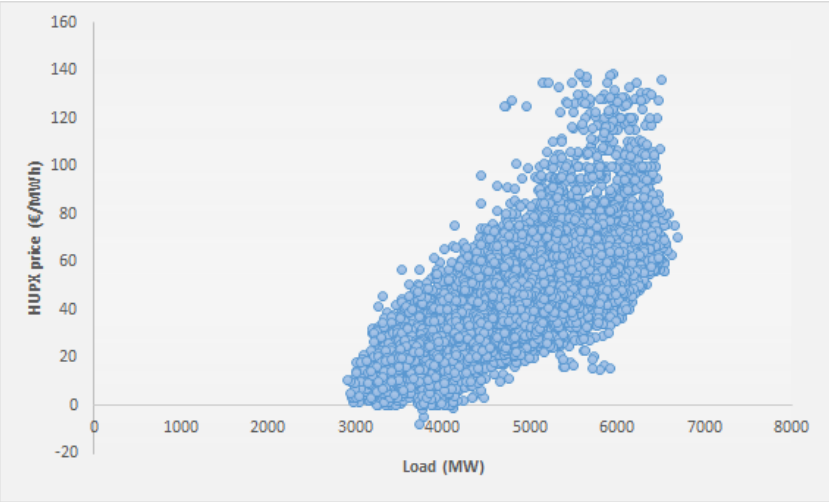


Figure 3.9: Correlation between HUPX price and Load

Residual load is an indicator in a power system. It shows how much capacity is left for conventional power plants to operate. Traditionally, when Variable Renewable Energy (VRE) sources are small in scale compared to the demand load, conventional power plants vary their power output in accordance with the demand load curve. As the capacity of VRE grows, its power output begins to affect the load balance of the power system. A new indicator was needed to describe the situation, giving birth to this terminology. The first use of the term "residual load" probably originates in a 2009 German study from Fraunhofer[26]. It used the German term "residuale Last". A common definition for residual load is *"what is left after substracting those generators who have to produce electricity (must run) and those that generate with (almost) no marginal costs (variable renewables like wind, solar and hydro)"*. Therefore, residual load (or residual demand) is defined as demand load minus renewable power output.

The **"Residual Load"** feature category includes 6 features, that corresponds to the residual demand of Germany, Czech Republic, Austria, Slovenia, Serbia and Romania, as presented on Table 3.4.

Feature	Description
DE - RDL	Residual Load of Germany
CZ - RDL	Residual Load of Czech Republic
AT - RDL	Residual Load of Austria
SI - RDL	Residual Load of Slovakia
RS - RDL	Residual Load of Serbia
RO - RDL	Residual Load of Romania

Table 3.4: "Residual Load" feature category

3.2.4 "Production" feature category

In this feature category we include the production level of the most significant types of power plants within the region of CEE and SEE. Their available capacity plays a critical role in the formation of the wholesale electricity prices, not only at HUPX but in the whole region.

The **"Production"** feature category includes 8 features, as presented on Table 3.5.

Feature	Description
HU - NUC	Nuclear generation in Hungary
HU - WND	Wind generation in Hungary
CZ - NUC	Nuclear generation in Czech Republic
AT - ROR	RoR generation in Austria
HR - ROR	RoR generation in Croatia
RS - ROR	RoR generation in Serbia
RO - NUC	Nuclear generation in Romania
RO - ROR	RoR generation in Romania

Table 3.5: "Production" feature category

As can be seen in figure 3.10, critical technologies of the wider area differ in both level of production and volatility. Data are available on the website of ENTSO-E[22].

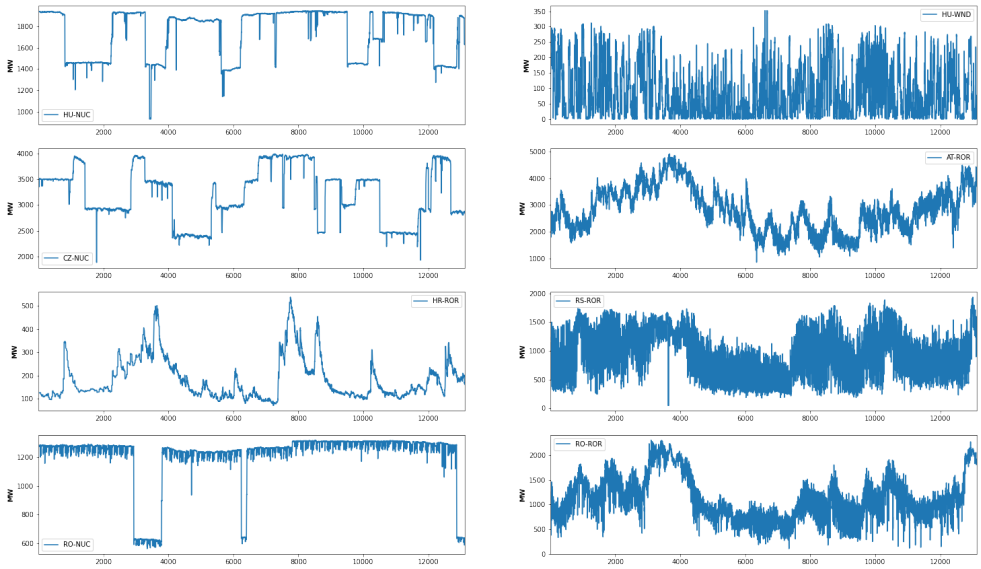


Figure 3.10: Electricity production of critical technologies in CEE and SEE

3.2.5 "Cross-border" features

Electricity interconnections are the physical links which allow the transfer of electricity across borders. Cross-border power trading is responsible for the flow of electricity from cheaper to more expensive countries. Therefore, the available capacity of the interconnections plays a significant role in the formation of the wholesale electricity prices.

Hungary has a lot of neighbouring countries (Slovakia, Ukraine, Romania, Serbia, Croatia, Slovenia, Austria), hence it has a lot of electricity interconnections, as depicted in Figure 3.11. For the task of HUPX price forecasting, it is crucial to take into account the available capacity of all interconnections in the region of CEE and SEE, that we consider that has impact on the price. For the purposes of this work, we have considered two feature categories related to cross-border information.

The first cross-border feature category is called **"Cross-border HU"** and has to do only with the Hungarian borders. That means it includes features that refer to the hourly available capacity of the interconnections that connect Hungary with Austria, Slovakia, Croatia, Serbia and Romania.

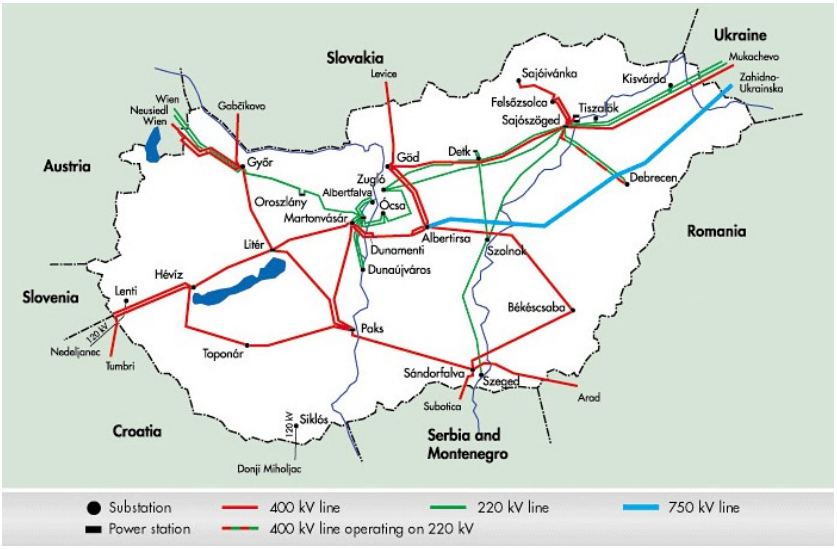


Figure 3.11: Electricity Transmission System of Hungary

The second cross-border feature category is called **"Cross-border"** and is related to the interconnections of the wider area, excluding the Hungarian ones. It includes information on the interconnections of the following borders:

- Germany - Czech Republic
- Austria - Czech Republic
- Poland - Czech Republic
- Slovakia - Czech Republic
- Germany - Austria
- Slovenia - Austria
- Italy - Austria
- Bulgaria - Serbia
- Bulgaria - Romania
- Croatia - Serbia
- Romania - Serbia

Chapter 4

Experimental Runs and Results

In this chapter we explore the performance of traditional machine learning, ensemble machine learning and deep learning algorithms in the task of HUPX electricity price forecasting. In addition, a sensitivity analysis is performed in order to examine the effect of individual features on the final result.

4.1 Implementation issues

Scikit-learn is used extensively in our study for the development of the several models and the implementation of the experiments. It is the most useful library for ML in Python, containing a lot of efficient tools for ML and statistical modeling including classification, regression, clustering and dimensionality reduction.

Algorithm	Function
Linear Regression	`sklearn.linear_model.LinearRegression`
Ridge	`sklearn.linear_model.Ridge`
Lasso	`sklearn.linear_model.Lasso`
Elastic net	`sklearn.linear_model.ElasticNet`
KNeighbors	`sklearn.neighbors.KNeighborsRegressor`
Decision Tree	`sklearn.tree.DecisionTreeRegressor`
SVR	`sklearn.svm.SVR`

Table 4.1: Scikit-learn functions for traditional ML algorithms

The functions in the above table are used for the implementation of the traditional ML algorithms. The respective scikit-learn functions for the ensemble ML

algorithms are presented in the table that follows.

Algorithm	Function
Random Forest	`sklearn.ensemble.RandomForestRegressor`
AdaBoost	`sklearn.ensemble.AdaBoostRegressor`
Gradient Boosting	`sklearn.ensemble.GradientBoostingRegressor`

Table 4.2: Scikit-learn functions for ensemble ML algorithms

The implementation of the ANN in our study, was based on `TensorFlow 2` and `Keras`. `TensorFlow 2` is an end-to-end, open-source ML platform. It is an infrastructure layer for differentiable programming, that combines four key abilities:

- Efficiently executing low-level tensor operations on CPU, GPU, or TPU.
- Computing the gradient of arbitrary differentiable expressions.
- Scaling computation to many devices.
- Exporting programs to external runtimes such as servers, browsers, mobile and embedded devices.

`Keras` is the high-level API of `TensorFlow 2`. It is an approachable, highly-productive interface for solving ML problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping ML solutions with high iteration velocity. The core data structures of Keras are layers and models. Keras was initially developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System[27]).

A cross-validation (CV) procedure is followed to evaluate the ML algorithms. `KFold` and `cross_val_score` functions are used from the `scikit-learn` library. According to the k-fold approach, the training set is split into k smaller sets. The following steps are followed for each of the k "folds":

- A model is trained using $k - 1$ of the folds as training data.
- The resulting model is validated on the remaining part of the data.

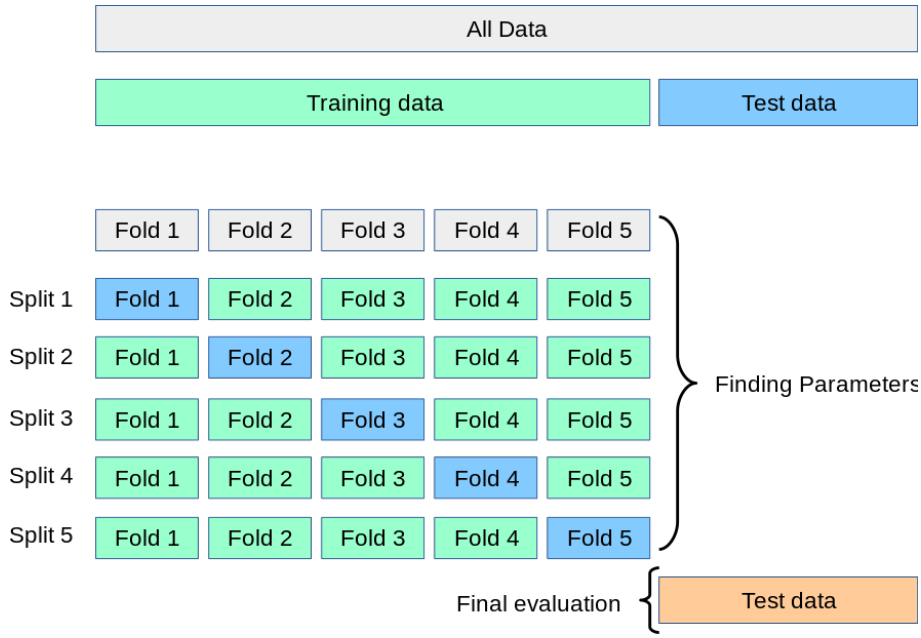


Figure 4.1: The cross-validation procedure

The performance measure reported by k-fold cross-validation is then the average of the values computed in the loop. In our experiments k is determined to 10 ($k = 10$). The scoring function selected for the cross-validation procedure is the **negative mean squared error**, provided by scikit-learn. After the algorithm selection, a hyperparameter optimization is performed applying the grid-search process. Grid-search defines a search space as a grid of hyperparameter values and evaluate every position in the grid. The `train_test_split` function is used subsequently for splitting the dataset for two different purposes: training and testing.

4.2 Traditional ML Algorithms: Experiments and Results

In this section, we examine the capabilities of classic machine learning algorithms in the task of electricity price forecasting. Several algorithms are compared and the one that prevails is used for the training of the model. An analysis is also performed of the features' contribution to the performance of the algorithm.

A cross-validation (CV) procedure is followed to evaluate the regression algorithms. The following table presents the performance of the several machine learning algorithms (Gaussian process was excluded from the experiments due to its bad performance in the cross-validation process):

Algorithm	CV results - mean	CV results - std
Linear Regression	-269.34	446.42
Ridge	-269.24	446.12
Lasso	-120.95	46.81
Elastic net	-118.56	41.41
KNeighbors	-181.78	113.55
Decision Tree	-214.92	95.98
SVR	-191.19	146.67

Table 4.3: Cross-validation results of traditional ML algorithms

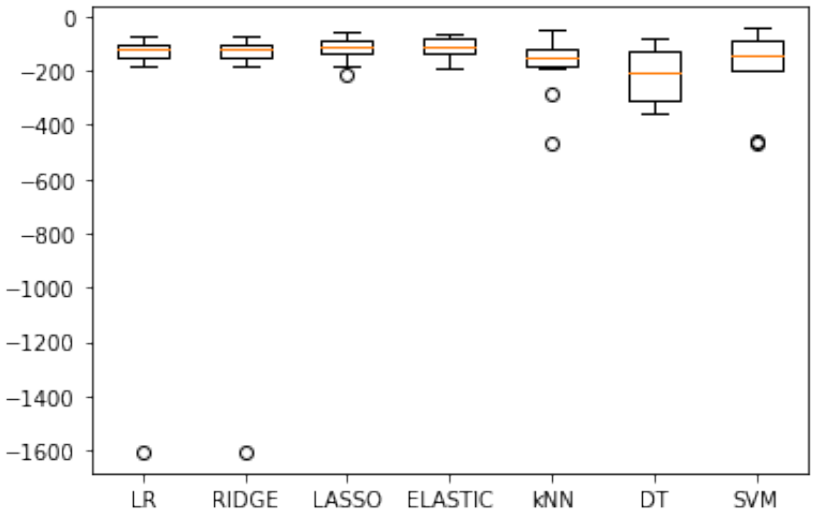


Figure 4.2: Boxplot: Cross-validation results of traditional ML algorithms

The boxplot above summarizes the performance of the machine learning algorithms in the cross-validation procedure.

Considering the results of the cross-validation procedure, the Elastic-net algorithm is chosen to proceed with the training of the model. Elastic-net's better performance leads us to the conclusion that our dataset possibly includes multiple features which are correlated with one another, which is a case in which the particular algorithm can give better results.

The basic parameter of the Elastic-net algorithm is the **alpha** parameter. It is a constant that multiplies the penalty terms. Its default value is 1.0. $alpha = 0$ is equivalent to an ordinary least square, solved by a linear regression algorithm. The Grid-search process is followed for the hyperparameter optimization. As figure 4.3 shows, the optimal value of the alpha parameter is 1.4.

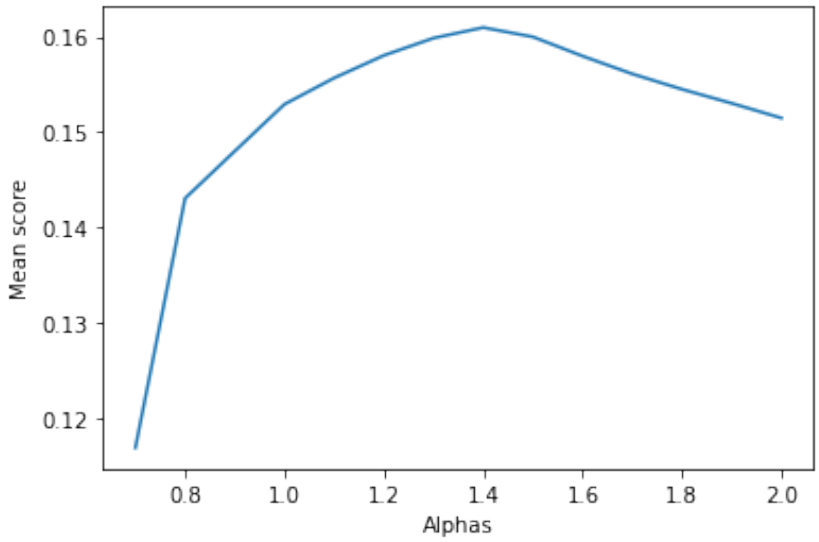


Figure 4.3: Hyperparameter optimization

The dataset is splitted subsequently for two different purposes: training and testing. The training subset is for building our model. The testing subset is for using the model on unknown data to evaluate the performance of the model. `Test_size = 0.2` is selected for splitting the dataset. Next steps include the training of the model using the training set and then the model evaluation using the test set.

For the training of the model, our first approach is to train using all the features of the dataset. In the next step we will examine the model's behavior with less

features during the training process. For the evaluation of the model, we take into account two different evaluation measures:

- Mean absolute error
- Mean absolute percentage error

The first measure computes the mean absolute error, which is a risk metric corresponding to the expected value of the absolute error loss or `l1-norm` loss. The second measure is a well-known evaluation metric for regression problems. The idea of this metric is to be sensitive to relative errors. It is for example not changed by a global scaling of the target variable. The table that follows presents the performance of the `Elastic-net` algorithm.

Algorithm	MAE	MAPE
Elastic-net	5.94	15.12

Table 4.4: Traditional ML algorithm performance

Figure 4.4 compares the target-variable of the test set with the predicted one. It turns out that it is difficult for a classic machine learning algorithm to predict well extreme prices.

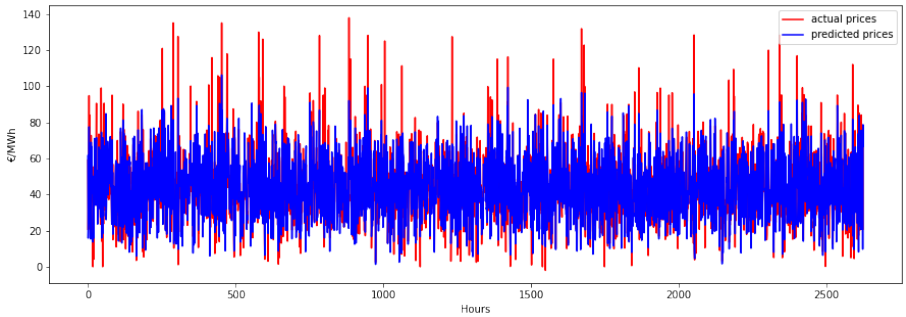


Figure 4.4: Test set versus predictions in traditional ML algorithm

Furthermore, we examine the performance of the model in several runs, removing each time a feature category. This procedure help us understand the contribution of the features to the efficiency of the model. More specifically, 6 different runs will be performed as described below:

- Run 1: All features are included

- Run 2: Cross-border features are excluded (*Cross-border HU* and *Cross-border* feature categories)
- Run 3: In addition to 2nd Run, *Prices* feature category is excluded
- Run 4: In addition to 3rd Run, *Production* feature category is excluded
- Run 5: In addition to 4th Run, *General* feature category is excluded
- Run 6: In addition to 5th Run, *Residual Load* feature category is excluded

The following graph presents the model's performance in the different runs. The results prove the value of the extended dataset used in the training of the model, showing at the same time the contribution of each feature category.

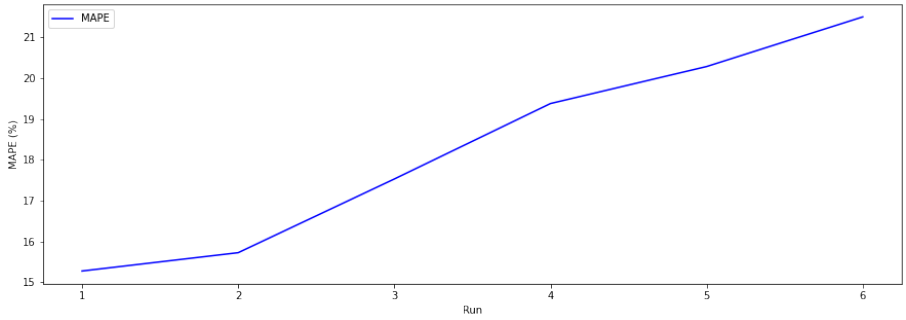


Figure 4.5: Model performance sensitivity removing features

4.3 Ensemble ML Algorithms: Experiments and Results

A cross-validation (CV) procedure is followed to evaluate the ensemble machine learning algorithms. We keep both $k = 10$ and `scoring = neg_mean_squared_error`, as in the case of classic ML algorithms. The table that follows presents the performance of the several ensemble ML algorithms in the cross-validation process.

Algorithm	CV results - mean	CV results - std
Gradient Boosting	-125.98	69.47
Random Forest	-111.62	54.58
AdaBoost	-204.75	89.31

Table 4.5: Cross-validation results of ensemble ML algorithms

The following boxplot summarizes the performance of the ensemble machine learning algorithms in the cross-validation procedure.

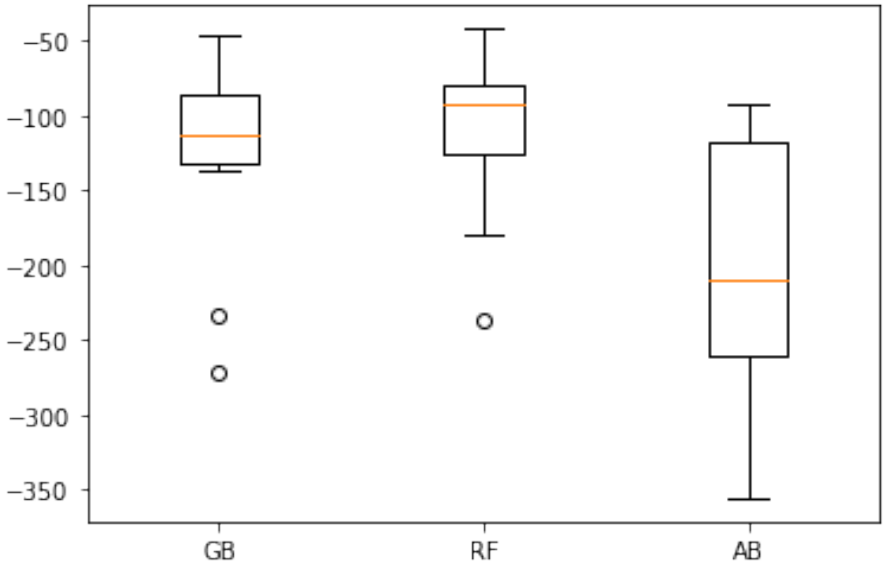


Figure 4.6: Boxplot: Cross-validation results of ensemble ML algorithms

Random Forest is shown to be the best algorithm among the three ensemble algorithms for the particular task of HUPX electricity price forecasting. A Grid-search procedure is then implemented over the `n_estimators` parameter of the `RandomForestRegressor`. This hyperparameter represents the number of trees in

the forest. As the following graph shows, the optimal value is `n_estimators = 110`.

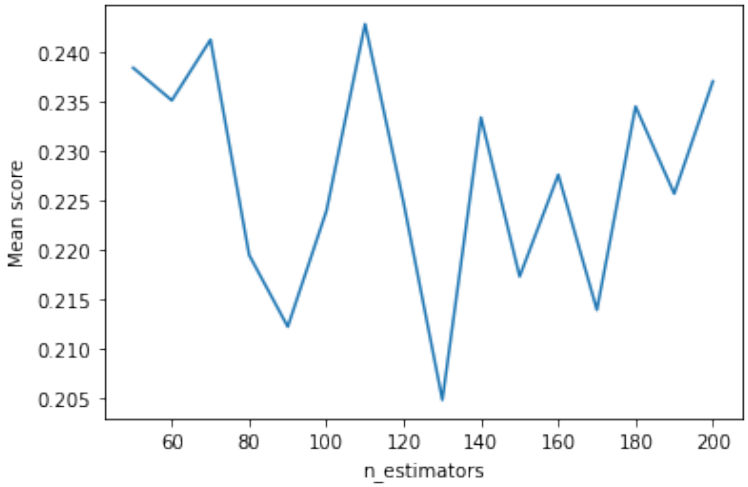


Figure 4.7: Hyperparameter optimization on Random Forest

Using the same metrics as in the case of classic ML algorithms, we proceed with the performance evaluation of Random Forest algorithm. The table that follows presents the performance of Random Forest algorithm.

Algorithm	MAE	MAPE
Random Forest	3.33	8.06

Table 4.6: Ensemble ML algorithm performance

The figure 4.8 compares the target-variable of the test set with the predictions of the `RandomForestRegressor`. It turns out that, besides the overall better performance of an ensemble ML algorithm compared to a traditional one, an ensemble method can approach more effectively the extreme electricity prices of the dataset.

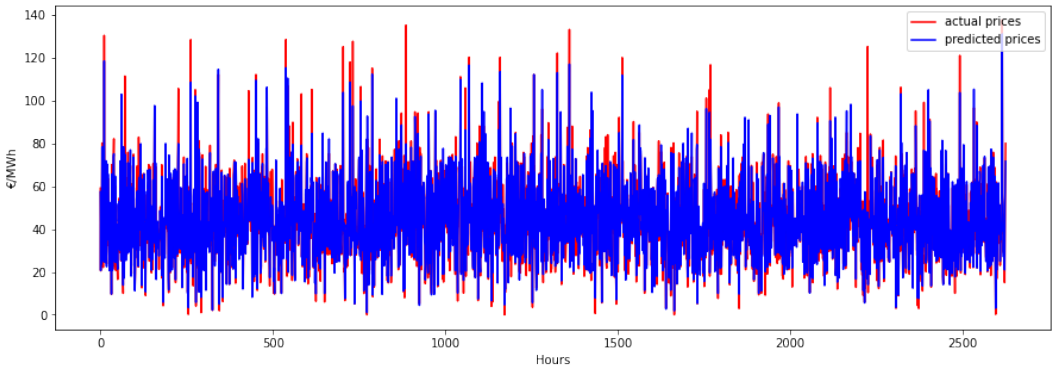


Figure 4.8: Test set versus predictions in ensemble ML algorithm

Furthermore, we are interested in exploring the sensitivity of the model's performance changing the maximum depth of the tree. `Max_depth` is one of the basic parameters of the `RandomForestRegressor`. If its value is `None`, then nodes are expanded until all leaves are pure. The figure that follows captures the course of model's performance as the maximum depth of the tree changes:

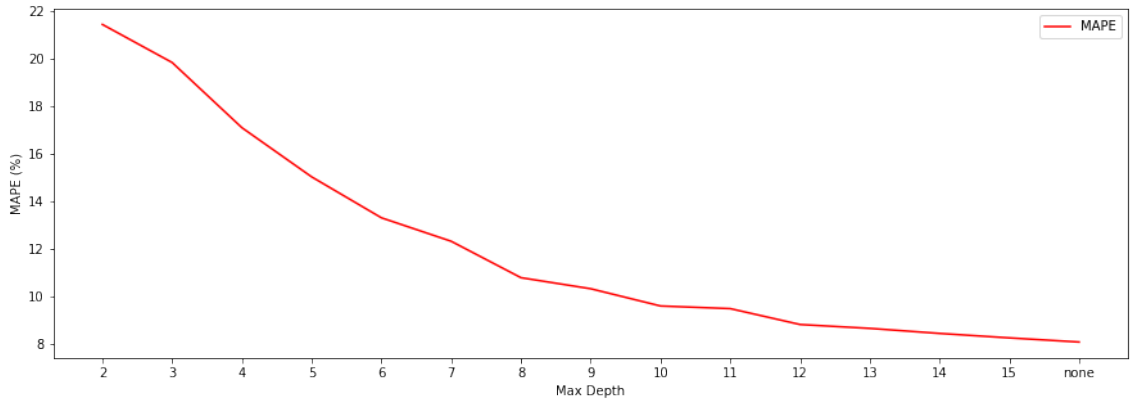


Figure 4.9: Model performance sensitivity changing max depth

Moreover, as in the case of traditional ML algorithms, we examine the performance of the model in several runs, removing each time a feature category. Such process highlights the contribution of the different features to the model effectiveness. Keeping the same structure of the experimental runs, we get the following results:

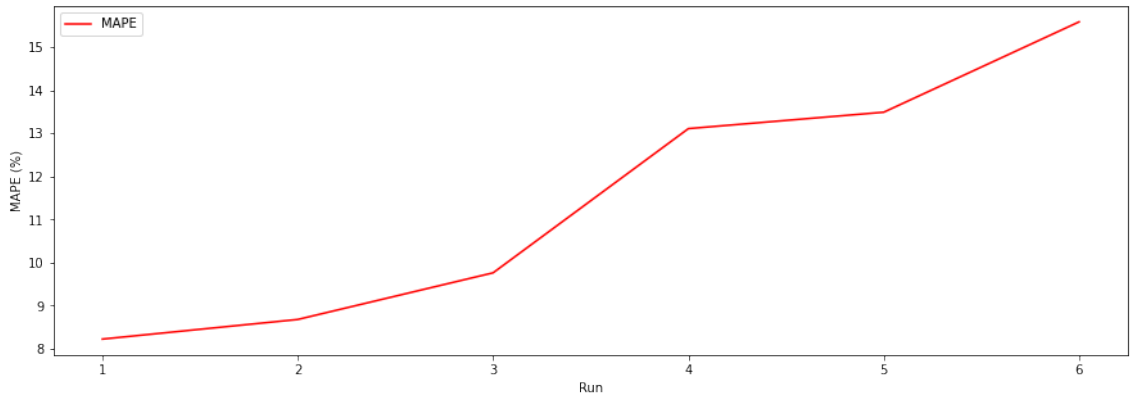


Figure 4.10: Random Forest performance sensitivity removing features

The results show the meaning of the existence of the different feature categories in the structure of the dataset, proving that when domain expertise is taking into account we can achieve better performance.

4.4 Deep Learning Algorithms: Experiments and Results

The implementation of the Artificial Neural Network (ANN) was based on `Tensor Flow 2` and `Keras`. The core data structures of Keras are layers and models. In our case, the `Sequential model` was used, which is a linear stack of layers. A `Sequential model` is created by passing a list of layers to the Sequential constructor. Each layer may have a number of units.

First steps of the implementation included data preparation (by turning it into NumPy arrays) and data preprocessing (e.g. feature normalization). In order to approach the optimal model architecture, several experiments were performed for different number of epochs, layers and units per layer. Keeping the number of epochs constant (`epochs = 500`), we obtain the following results:

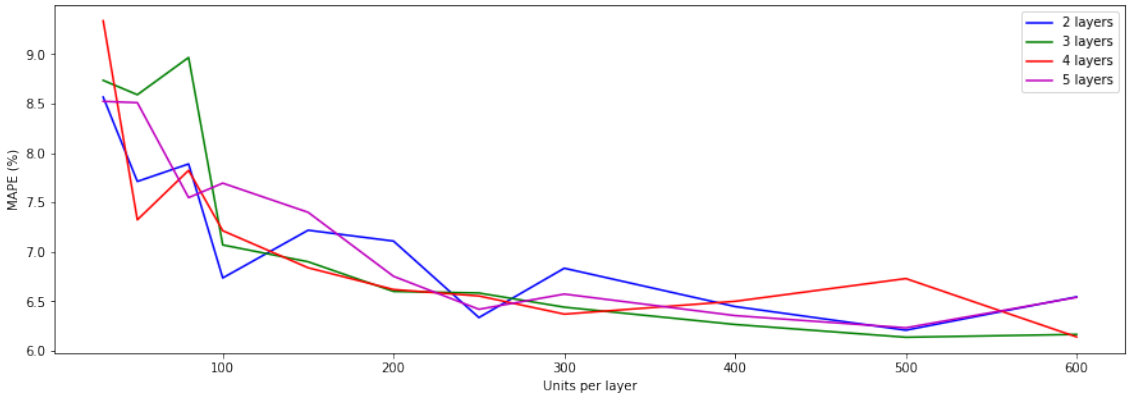


Figure 4.11: MLP results for several number of layers

According to the evaluation of the projections, MAPE was ranged between 6.1% and 9.4%. In most of the cases, the smallest error was recorded for `units = 500`. It is obvious that increasing the number of units per layer, a better performance is achieved. The same is not clearly true when increasing the number of layers. The addition of an extra layer to the model architecture did not have a clear positive effect on the model performance. Another observation is that the training time increases exponentially as further units are added to the model, as depicted in the following graph:

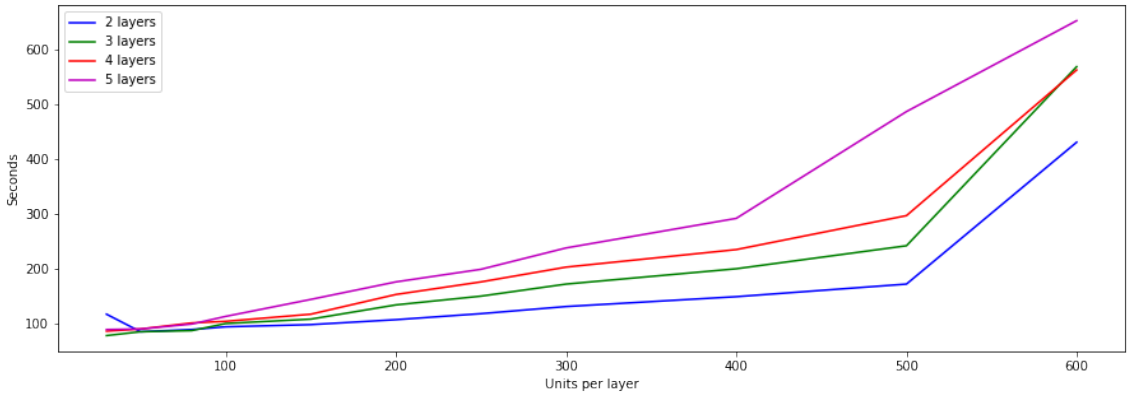


Figure 4.12: Training time results for several number of layers and units

Taking into account both error (MAPE in %) and training time, the best performance is achieved for the combination of 3 layers, each of which has 500 units. Keeping always the number of epochs at 500, the table that follows summarizes the performance results of a 3-layer MLP, for several number of layer inputs:

Experiment	1st layer units	2nd layer units	3rd layer units	Training time (sec)	MAPE (%)
1	30	30	30	78	8.73
2	50	50	50	85	8.59
3	80	80	80	87	8.97
4	100	100	100	100	7.07
5	150	150	150	108	6.90
6	200	200	200	134	6.60
7	250	250	250	150	6.58
8	300	300	300	172	6.44
9	400	400	400	200	6.27
10	500	500	500	242	6.14
11	600	600	600	569	6.17

Table 4.7: 3-layer MLP performance results

The final architecture includes a sequential model which includes **3 layers**, each of which contains **500 units**. **ReLU** is the activation function that was selected, because of its simplicity and its computational efficiency. For compiling the model the **SGD** optimizer was selected. Keras provides the SGD class that implements the stochastic gradient descent optimizer.

The following figure compares the target-variable of the test set with the pre-

dictions. The resulting picture is even better than ensemble ML algorithms' one, proving the capabilities of MPLs to effectively approximate a mapping function from input variables to output variables.

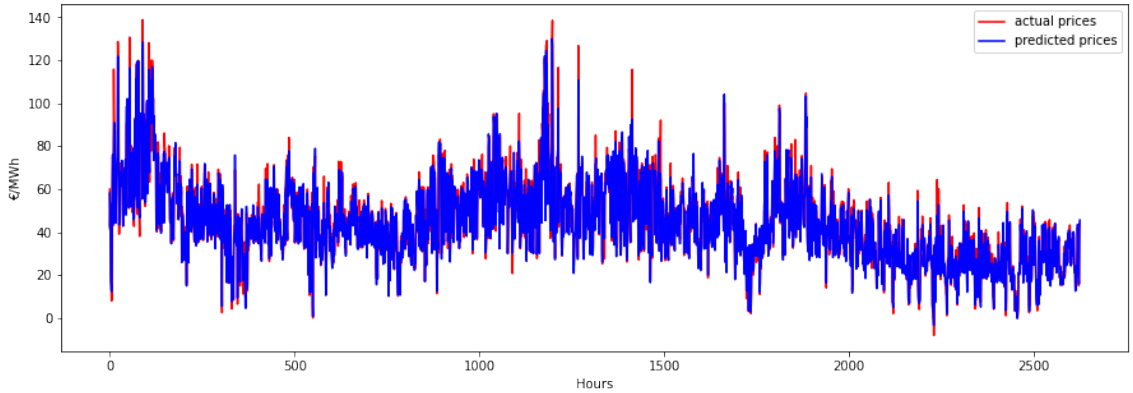


Figure 4.13: Test set versus predictions in MLP

Furhermore, as in the case of machine learning algorithms previously, we will explore the sensitivity of MLP performance in several runs, removing each time a feature category. Such process highlights the contribution of the different features to the model effectiveness. Keeping the same structure of the experimental runs, as described in chapter 4.2, we get the results that are presented in 4.14. Removing each time the several feature categories, the forecasting error increases gradually from 6% to 13%.

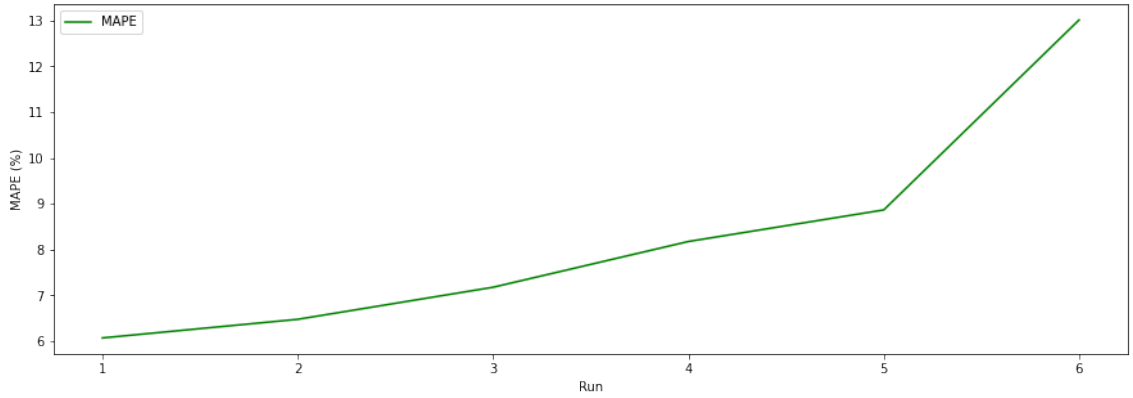


Figure 4.14: MPL performance sensitivity removing features

Finally, let us summarize with the performance achieved from all algorithms used in the task of Hungarian electricity price forecasting in this thesis.

Algorithm	MAPE (%)
Classic ML Algorithm (`Elastic-net`)	15.12
Ensemble ML Algorithm (`Random Forest`)	8.06
Deep Learning Algorithm (`Multilayer Perceptron`)	6.14

Table 4.8: Algorithms performance

Chapter 5

Conclusions and Future Work

The goal of this thesis is to explore the course of the algorithms' performance as we move from traditional to more sophisticated methods for the task of electricity price forecasting. Our focus is on the Hungarian wholesale electricity day-ahead price, which is considered a benchmark index in the region of SE Europe. We are also interested in quantifying the impact of domain expertise on the shaping of the results.

Mean absolute percentage error (MAPE) was mainly used as evaluation metric, which is a well-known metric for time-series forecasting tasks. What was confirmed throughout this study are the capabilities of deep learning methods to more effectively approximate a mapping function from input variables to output variable. From a MAPE of 15.12% achieved by a classic machine learning algorithm, we reached a performance of 6.14% using a Multilayer Perceptron model. Pleasant surprise was the behavior of the ensemble machine learning methods, which are proved to be very competitive on the task of electricity price forecasting, achieving a MAPE of 8.06%.

Domain expertise helped us to build a really extended dataset consisting of 69 features. These features are related to production, consumption, fuel cost and cross-border interconnections data and in theory they affect the formation of electricity prices. Performing a sensitivity analysis, the contribution of these feature categories to the good performance of the algorithms was confirmed. Removing one category

at a time, the performance of the algorithms was getting worse and worse, reaching a MAPE of 21.50%, 15.59% and 13% for classic, ensemble and deep machine learning algorithms respectively.

Comparing the results of this thesis with those of [21], it turns out that the Hungarian power exchange is the most predictable one in the region of SE Europe, judging from the performance of 6.14% that has been achieved. In [21], MLPs are also used to forecast the day-ahead electricity price, ranking the Serbian market as the most predictable one with a MAPE of 9.28%. The Croatian and the Bulgarian power exchanges follow with a MAPE of 17% and 21% respectively. Predictability is probably related to the maturity of the power exchanges, since it is widely known that HUPX (Hungarian power exchange) is the most mature and IBEX (Bulgarian power exchange) is the most shallow market in the region of SE Europe.

Future work could include the use of recurrent neural networks in the task of electricity price forecasting. Recurrent neural networks like the Long Short-Term Memory network (or LSTM) add the explicit handling of order between observations when learning a mapping function from inputs to outputs, not offered by MLPs. They are a type of neural network that adds native support for input data comprised of sequences of observations. Instead of mapping inputs to outputs alone, the network is capable of learning a mapping function for the inputs over time to an output. LSTM is able to solve many time series tasks unsolvable by feedforward networks using fixed size time windows. In addition to the general benefits of using neural networks for time series forecasting, recurrent neural networks can also automatically learn the temporal dependence from the data. Learning temporal dependence means that the most relevant context of input observations to the expected output is learned and can change dynamically. In the simplest case, the network is shown one observation at a time from a sequence and can learn what observations it has seen previously are relevant and how they are relevant to forecasting. The model both learns a mapping from inputs to outputs and learns what context from the input sequence is useful for the mapping, and can dynamically change this context as needed.

References

[1] Rafal Weron. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International Journal of Forecasting*, 30(4):1030–1081, 2014.

[2] Wikipedia. California electricity crisis. `https://en.wikipedia.org/wiki/2000%E2%80%9301_California_electricity_crisis`.

[3] Hungarian power exchange. `https://hupx.hu/en/`.

[4] S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l1-regularized least squares. *IEEE Journal of Selected Topics in Signal Processing*, 2007.

[5] Google images. `https://images.google.com/`.

[6] Apple siri. `https://www.apple.com/siri/`.

[7] Alphago. `https://deepmind.com/research/case-studies/alphago-the-story-so-far`.

[8] Mariano Ventosa, Álvaro Baıllo, Andrés Ramos, and Michel Rivier. Electricity market modeling trends. *Energy Policy*, 33(7):897–913, 2005.

[9] M. Burger, B. Graeber, and G. Schindlmayr. *Managing Energy Risk: An Integrated View on Power and Other Energy Markets*. Wiley, 2007.

[10] Derek Bunn. *Modelling Prices in Competitive Electricity Markets*. Wiley, 2004.

[11] Dogan Keles, Jonathan Scelle, Florentina Paraschiv, and Wolf Fichtner. Extended forecast methods for day-ahead electricity spot prices applying artificial neural networks. *Applied Energy*, 162:218–230, 2016.

[12] Aleamodel, aleasoft. `https://aleasoft.com/aleamodel/`.

[13] M. Hemmati N. Amjady. Energy price forecasting - problems and proposals for such predictions. *IEEE Power and Energy Magazine*, 4(2):20–29, 2006.

[14] J. Wang J. Che. Short-term electricity prices forecasting based on support vector regression and auto-regressive integrated moving average modeling. *Energy Conversion and Management*, 51(10):1911–1917, 2010.

[15] L. Chen S. Fan, C. Mao. Next-day electricity-price forecasting using a hybrid network. *IET Generation, Transmission and Distribution*, 1(1):176–182, 2007.

[16] M-T. Tsai W-M. Lin, H-J. Gow. An enhanced radial basis function network for short-term electricity price forecasting. *Applied Energy*, 87(10):3226–3234, 2010.

[17] M. Yu J. Ma C. Jin L. Xiao, W. Shao. Research and application of a hybrid wavelet neural network model with the improved cuckoo search algorithm for electrical power system forecasting. *Applied Energy*, 198(15):203–222, 2017.

[18] C-J. Huang P-H. Kuo. An electricity price forecasting model by hybrid structured deep neural networks. *Sustainability*, 10(1280), 2018.

[19] R. Liu L. Wang L. Peng, S. Liu. Effective long short-term memory with differential evolution algorithm for electricity price prediction. *Energy*, 10(52), 2018.

[20] B. De Schutter J. Lago, F. De Ridder. Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. *Applied Energy*, 221:386–405, 2018.

[21] A. Dedinec. Electricity price forecasting of the south-east european power exchanges. *7th International Conference on Information Society and Technology ICIST 2017*.

[22] Entso-e website. `https://www.entsoe.eu/`.

[23] Ice - ttf futures. `https://www.theice.com/products/27996665/Dutch-TTF-Gas-Futures`.

[24] Eu ets. `https://ec.europa.eu/clima/policies/ets_en`.

[25] Ice - eua futures. `https://www.theice.com/products/197/EUA-Futures`.

[26] Dynamische simulation der stromversorgung in deutschland nach dem ausbauszenario der erneuerbaren-energien-branche. https://www.bee-ev.de/fileadmin/Publikationen/Studien/100119_BEE_IWES-Simulation_Stromversorgung2020_Endbericht.pdf.

[27] Oneiros. https://en.wikipedia.org/wiki/Keras#cite_note-5.