UNIVERSITY OF THE PELOPONNESE & NCSR “DEMOCRITOS”
MSC PROGRAMME IN DATA SCIENCE

Energy Price Forecasting in Italy

by

Galanis Panagiotis

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

Supervisor: Theodoros Giannakopoulos
Principal Researcher, NCSR Demokritos

Athens, June 2024

Galanis Panagiotis

MSc. Thesis, MSc. Programme in Data Science

University of the Peloponnese & NCSR "Democritos", June 2024

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

Energy Price Forecasting in Italy

by

Galanis Panagiotis

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

Supervisor: Theodoros Giannakopoulos
Principal Researcher, NCSR Demokritos

Approved by the examination committee on June, 2024.

(Signature)	(Signature)	(Signature)
....	....	....
Theodoros Giannakopoulos Principal Researcher NCSR Demokritos	Dr. Evangelos Karkaletsis Director NCSR Demokritos	Costas Vassilakis Professor University of Peloponnese

Athens, June 2024

UNIVERSITY OF THE PELOPONNESE & NCSR "DEMOCRITOS"
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 Bachelor of Science in Informatics and Telecommunications, 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.

(Signature)

. . . .

Galanis Panagiotis

Athens, June 2024

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Mr. Theodoros Giannakopoulos, for his unwavering support, patience and guidance throughout the course of this research. His expertise and insightful feedback have been invaluable in shaping and finalization of this thesis.

I am immensely thankful to my parents and to my siblings, whose love and encouragement have been my stronghold. Their belief in my abilities and constant support, both emotionally and intellectually, have been the pillars upon which I built my aspirations.

Lastly, this journey would not have been possible without the collective support of many individuals who, in one way or another, contributed to the success of this thesis.

To my family and friends.

Περίληψη

Σκοπός αυτής της εργασίας είναι να διερευνήσει την αποτελεσματικότητα των αλγορίθμων Μηχανικής Μάθησης στην πρόβλεψη των τιμών ηλεκτρικής ενέργειας, με ιδιαίτερη έμφαση στην Ιταλική χονδρική αγορά ηλεκτρικής ενέργειαςIPEX, ένα κύριο σημείο αναφοράς στη Νότια Ευρώπη. Αξιοποιώντας τη γνώση του τομέα, συντάχθηκε ένα εκτεταμένο σύνολο δεδομένων που περιλαμβάνει 168 μεταβλητές. Η μελέτη περιλαμβάνει την εφαρμογή διαφόρων παραδοσιακών τεχνικών μηχανικής μάθησης και νευρωνικών δικτύων, χρησιμοποιώντας σημαντικές βιβλιοθήκες Python όπως το scikit-learn και το keras.

Οι αγορές ηλεκτρικής ενέργειας αλλάζουν συνεχώς, γεγονός που καθιστά αναγκαία την αλλαγή των δεδομένων εκπαίδευσης των αλγορίθμων μας. Βάσει των πειραμάτων μας, διαπιστώσαμε ότι το ιδανικό διάστημα εκπαίδευσης θα πρέπει να περιλαμβάνει μόνο τις τελευταίες 15 ημέρες ενώ οι προβλέψεις θα πρέπει να γίνονται μόνο για την επόμενη ημέρα και όχι για μεγαλύτερη περίοδο λόγω αύξησης του σφάλματος. Επιπλέον, για την επικύρωση των αποτελεσμάτων, χρησιμοποιήθηκε η διαδικασία Nested Cross Validation αντί του απλού Cross Validation για την αποφυγή data leakage.

Καθώς προχωρούμε από βασικές σε πιο προηγμένες μεθοδολογίες, υπάρχει μια σαφής τάση βελτίωσης της απόδοσης. Παρατηρήσαμε μια μείωση στο Μέσο Απόλυτο Ποσοστιαίο Σφάλμα (MAPE) από περίπου 20% σε 5%, μια απόδειξη της δύναμης των νευρωνικών δικτύων στην ακριβή μοντελοποίηση της σχέσης μεταξύ των παραγόντων που επηρεάζουν τις τιμές και των προβλεπόμενων τιμών. Επιπλέον, διεξήχθη μια ανάλυση ευαισθησίας για να αξιολογηθεί η επιρροή της εξειδικευμένης γνώσηςτου τομέα στα αποτελέσματα, η οποία επισήμανε τον ζωτικό ρόλο που διαδραματίζει κάθε χαρακτηριστικό στην ενίσχυση της αποτελεσματικότητας των αλγορίθμων.

Abstract

The aim of this thesis is to investigate the effectiveness of Machine Learning algorithms in forecasting electricity prices, with a particular emphasis on the Italian wholesale electricity market (IPEX), a key reference point in South Europe. Utilizing domain knowledge, an extensive dataset comprising 168 variables was compiled. The study encompasses the application of various conventional machine learning techniques and artificial neural networks, employing prominent Python libraries like scikit-learn and keras.

Electricity markets are constantly changing, which necessitates the updating of our algorithms' training datasets. Based on our experiments, we found that the ideal training dataset should be rolling and include only the last 15 days. Predictions should be made only for the next day and not for a longer period, as the error increases with the length of the forecast interval. Additionally, for the validation of the results, nested cross-validation was used instead of simple cross-validation because we have time-series data and we want to avoid data leakage. The nested cross-validation procedure provides an almost unbiased estimate of the true error.

As we progress from basic to more advanced methodologies, there is a clear trend of enhanced performance. We observed a reduction in the Mean Absolute Percentage Error (MAPE) from around 20% to 5%, a testament to the power of artificial neural networks in accurately modeling the relationship between input factors and the predicted prices. Additionally, a sensitivity analysis was conducted to assess the influence of specialized knowledge on the results, which underscored the vital role that each feature plays in bolstering the algorithms' effectiveness.

Contents

List of Tables

List of Figures

List of Abbreviations

EPF	Electricity Price Forecasting
ENTSO-E	European Network of Transmission System Operators for Electricity
CET	Central European Time
TTF	Title Transfer Facility
MWh	Megawatt hour
EUA	EU Allowances
TSO	Transmission System Operator
SVM	Support Vector Machine
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
MAPE	Mean Absolute Percentage Error
sMAPE	Symmetric Mean Absolute Percentage Error
MASE	Mean Absolute Scaled Error
rMAE	Relative Mean Absolute Error
GME	Gestore Mercati Energetici
ETRM	Energy Trade and Risk Management

Enel	Ente Nazionale per l'Energia Elettrica
IEM	Italian Electricity Market
ARERA	The Italian Regulatory Authority for Energy, Networks and Environment
NECP	National Energy and Climate Plan
PCR	Price Coupling Regions
MGP	Mercato del Giorno Prima
EUPHEMIA	EU + Pan-european Hybrid Electricity Market Integration Algorithm)
MI	Mercato Infragiornaliero
MPEG	Mercato dei Prodotti Giornalieri
MSD	Mercato per il Servizio di Dispacciamento
MB	Mercato del Bilanciamento
GW	Gigawatt
ANNs	Artificial Neural Networks
LASSO	Least Absolute Selection Shrinkage Operator
SVM	Support Vector Machines
SVR	Support Vector Regression
GP	Gaussian Processe
MLP	Multilayer Perceptron
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
SFE	Supply Function Equilibrium

SPCM	Strategic Production-Cost Model
ACE	Agent-based Computational Economics
ARIMA	Autoregressive Integrated Moving Average
SOM	Self-Organizing Map
ERBFN	Enhanced Radial Basis Function Network
RBFN	Radial Basis Function Network
OED	Orthogonal Experimental Design
WNN	Wavelet Neural Network
SSA	Singular Spectrum Analysis
DT	Decision Trees
RF	Random Forests
DE	Differential evolution

Chapter 1

Introduction

1.1 Background of Electricity Markets

The concept of electricity markets is central to the distribution and regulation of electricity. Historically, electricity was supplied by vertically integrated monopolies that owned and controlled the entire energy value chain, including generation, transmission, and distribution systems. These entities were typically state-owned or heavily regulated to prevent the abuse of monopoly power. The inception of electricity markets can be traced back to the late 1980s and early 1990s, when a wave of **deregulation and privatization** swept through the electricity industry, particularly in the UK and Chile. The idea was to introduce competition into the electricity sector to increase efficiency and reduce costs. This led to the separation of the generation, transmission, and distribution segments of the industry, allowing multiple generators to compete in an open market.[1][2]

Electricity markets are typically divided into wholesale and retail segments. The **wholesale electricity market** involves the sale of electricity among generators, retailers, and other financial intermediaries. Prices in the wholesale market are determined by supply and demand dynamics and are influenced by fuel costs, plant availability, and consumer demand.The **retail electricity market**, on the other hand, is where electricity is sold to the end-users. In deregulated markets, consumers can choose their electricity provider from a pool of competitors. The retail market aims to provide consumers with competitive prices and innovative products and services.[3][4]

One of the key components of electricity markets is the **balancing mechanism**, which ensures that supply meets demand in real-time. This is crucial because electricity cannot be stored easily and must be produced as it is consumed. Grid operators use various tools, such as demand response and frequency regulation, to maintain the balance.[5]

The evolution of electricity markets has also been influenced by the increasing focus on sustainability and the transition to a low-carbon economy. The integration of renewable energy sources, such as wind and solar, poses new challenges due to their intermittent nature. Markets are adapting by developing new products and services that can accommodate the variability of renewable energy. Today, electricity markets are complex systems that involve a wide range of participants, including generators, transmission and distribution system operators, retailers, regulators, and consumers. They are governed by a set of rules and regulations that ensure fair competition and protect consumers' interests. As we move towards a future with a greater reliance on electricity for transportation and heating, and as we strive for net-zero emissions, the role of electricity markets will become even more significant. They will continue to evolve, incorporating advanced technologies like smart grids and energy storage, to meet the changing needs of society and the environment.From their regulated beginnings to the complex, competitive markets of today, electricity markets have come a long way and will continue to adapt to the needs of a modern, sustainable energy system.[6]

1.2 Overview of the Italian Electricity Market

1.2.1 Historical Context

The Italian Electricity Market plays a pivotal role in the energy sector, not only within Italy but also as a part of the broader European energy landscape. Understanding the intricacies of this market is crucial for stakeholders, ranging from policy-makers to consumers and investors.

The Italian electricity sector began with the establishment of small private and municipal companies in the late 19th century. However, the industry was nationalized in 1962 with the creation of the state-owned entity **Enel (Ente Nazionale per l'Energia Elettrica)**, which held a monopoly over the generation, transmission, and distribution of electricity in Italy. The liberalization process started in the 1990s, following the European Union's directives aimed at creating a single energy market. The Italian government implemented legislative decrees to end Enel's monopoly and open the market to competition. This led to the unbundling of generation, transmission, and distribution activities and the establishment of the **Italian Electricity Market (IEM)** in 1999.The Italian electricity market has undergone significant regulatory changes, particularly influenced by European Union directives aimed at promoting competition and sustainability. The regulatory framework, shaped by **ARERA**, ensures a level playing field for all market participants while fostering innovation and consumer benefits.[7][8]

Electricity prices in Italy have seen fluctuations due to various factors, including global energy crises, changes in supply and demand, and policy shifts. Historically, prices reached an all-time high of 815.57 EUR/MWh in September 2022. The market has also been impacted by the European Union's energy policies and Italy's own National Energy and Climate Plan (NECP), which sets targets for emissions reductions and energy efficiency.[9][10]

1.2.2 Revolution and Future Outlook

The Italian electricity market has seen several revolutions over the years.Italy has significantly increased its **renewable energy** capacity, particularly in solar and wind energy, leading to a more sustainable energy mix.Investments in **smart grid technology** have improved the efficiency and reliability of the electricity supply. The liberalization has **empowered consumers** with the freedom to choose their electricity suppliers based on competitive pricing and services.The market has adapted to these challenges by implementing advanced technologies and regulatory frameworks to ensure stability and promote sustainability[11][12].

The Role of Terna, the transmission system operator, plays a crucial role in managing the high-voltage grid and ensuring the continuous flow of electricity. Terna's responsibilities have expanded to include the integration of renewable sources and the promotion of energy efficiency[13].

The Italian Electricity Market has come a long way from its nationalized roots to a competitive and dynamic marketplace. It stands as a testament to Italy's commitment to energy liberalization, sustainability, and innovation. As the market moves forward, it will likely continue to face new challenges and opportunities, particularly in the areas of decarbonization and digital transformation[14].

1.2.3 Structure and Characteristics

The Italian electricity market is structured into several segments: generation, transmission, distribution, and retail. The generation sector is liberalized, with prices set by the market. Transmission is managed by Terna S.p.A., an independent entity, while distribution and supply are liberalized with a price cap established by the Italian Regulator. The market operator, Gestore dei Mercati Energetici (GME), manages the Pool system, collecting bids and determining the dispatch of electricity.

Price Coupling Regions (PCR) has played a crucial role to the structure of Italian and other European Electricity Markets. This project of European Power Exchanges develops a single price coupling solution to be used to calculate electricity prices across Europe respecting the capacity of the relevant network elements on a day-ahead basis. This is crucial in order to achieve the overall EU target of a harmonised European electricity market. The integrated European electricity market is expected to increase liquidity, efficiency and social welfare[15].

One crucial principle used to set up the price of power is the Merit Order Effect. It ranks power sources based on their marginal costs, with the cheapest sources being utilized first. As we see in Figure 1.1, renewable energy sources, like wind and solar, have very low marginal costs because they don't require fuel, which means they often come first in the merit order. As more renewable energy is supplied to the grid, it tends to lower the average price of electricity. This is because renewables can

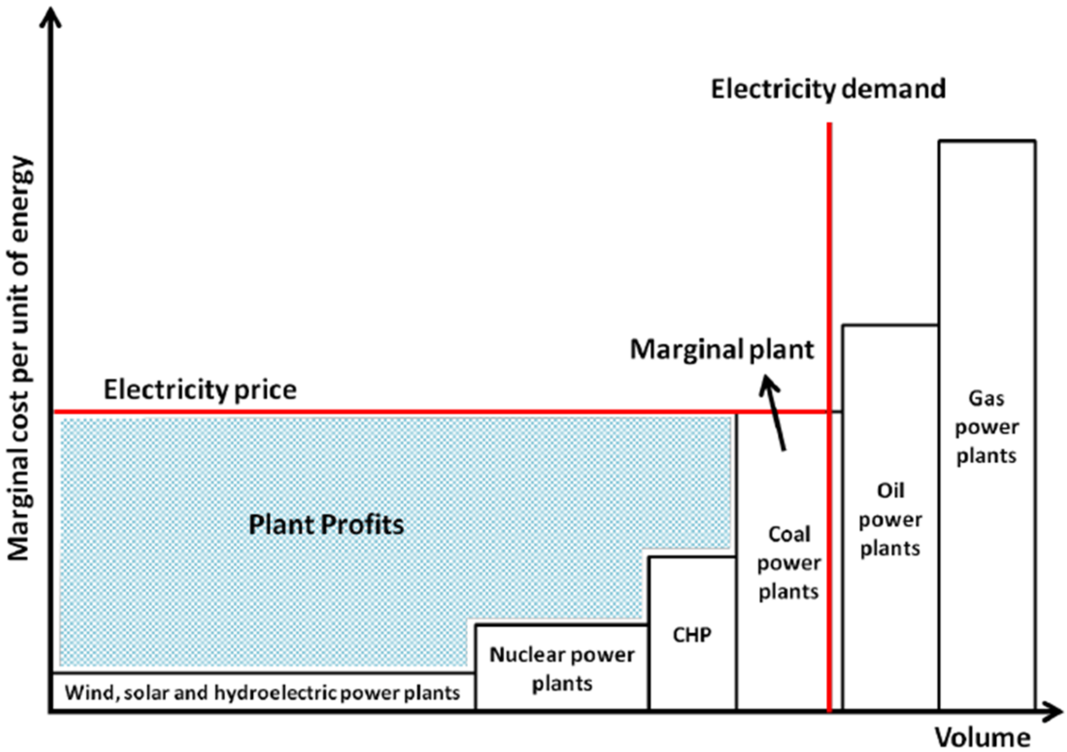

Figure 1.1: Merit Order Effect

offer their power at a lower price, pushing out more expensive, conventional power plants. The merit order effect can lead to significant savings in electricity costs for consumers and has been a key factor in promoting the adoption of renewable energy sources[16].

The Italian Electricity Market (IEM) Structure consists of several segments:

- **The Day-Ahead Market (MGP)**, where electricity is traded one day before the actual delivery. MGP hosts most of the electricity sale and purchase transactions. In the MGP, hourly energy blocks are traded for the next day. Participants submit bids/asks where they specify the quantity and the minimum/maximum price at which they are willing to sell/purchase.

  Algorithm EUPHEMIA (EU + Pan-european Hybrid Electricity Market Integration Algorithm) solves the market coupling problem on the PCR (Price Coupling Regions) perimeter, as it is shown in Figure 1.2. Euphimia maximizes the welfare of the solution, as most competitive prices arise and there is efficiency in the capacity allocation.

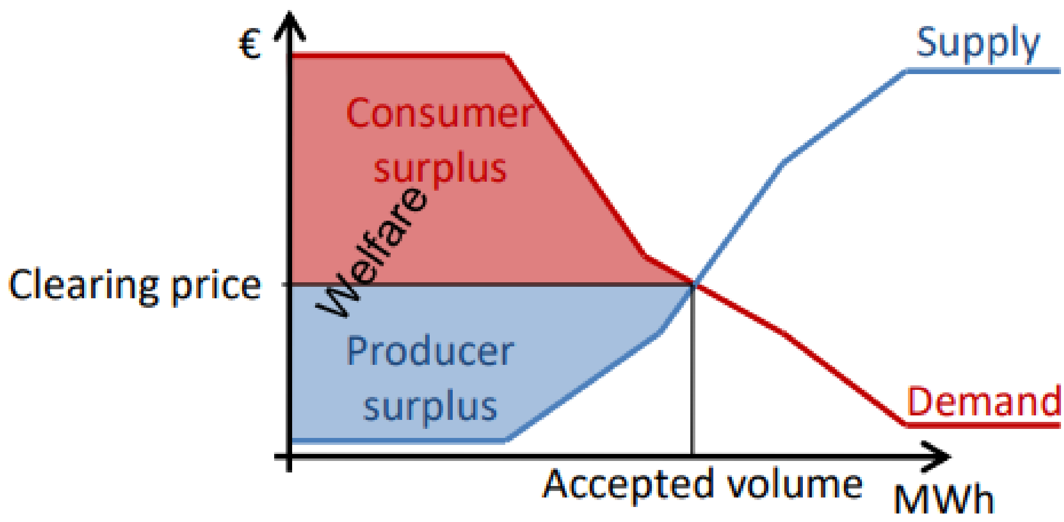

Figure 1.2: Price Coupling Solution

MGP is an auction market and not a continuous-trading market[17]. All the supply offers and the demand bids pertaining both to pumping units and consuming units belonging to foreign virtual zones that are accepted in the MGP are valued at the marginal clearing price of the zone to which they belong. This price is determined, for each hour, by the intersection of the demand and supply curves and is differentiated from zone to zone when transmission capacity limits are saturated.

The accepted demand bids pertaining to consuming units belonging to Italian geographical zones are valued at the "Prezzo Unico Nazionale" (PUN – national single price); this price is equal to the average of the prices of geographical zones, weighted for the quantities purchased in these zones[18].

- **The Intraday Market (MI)**, allows Market Participants to modify the schedules defined in the MGP by submitting additional supply offers or demand bids. It operates through three MI-A auction sessions and one MI-XBID continuous trading session. During the **MI-A** sessions, purchase and sale negotiations occur simultaneously with the allocation of intraday interconnection capacity across Italian market areas and other interconnected regions in Market Coupling. The **MI-XBID** session, split into three phases, similarly allocates intraday interconnection capacity among Italian market areas and other connected regions during the negotiation process within the XBID system[18].

- **The Daily Products Market (MPEG)** is the venue for trading energy products with a delivery obligation, operating in a continuous mode. It's open to all electricity market participants and offers two pricing options for trading: a "unit price differential" relative to the PUN (Price of Unbalance), and a "full unit price" which is the absolute value of the electricity traded. Tradable delivery profiles include Baseload for all days and Peak Load for weekdays. Market participants who are also part of the PCE can trade these products, with the net positions recorded in the PCE as per the Electricity Market Rules[18].

- **The Dispatching Services Market (MSD)** is crucial for maintaining real-time equilibrium between energy supply and demand. Operated by Terna S.p.A., the MSD acquires resources necessary for system relief from intra-zonal congestion, energy reserve creation, and real-time balancing. Terna serves as a central counterparty, compensating accepted offers based on their bid price. The MSD is structured into two main parts: the ex-ante MSD, which includes six scheduling substages (MSD1 to MSD6), and the Balancing Market (MB), both of which run in multiple sessions as per dispatching rules. The ex-ante MSD has a single bidding session, while the MB allows for continuous offer submissions, with hourly updates throughout the day[18].

1.3 Importance of electricity price forecasting

Electricity price forecasting (EPF) is a critical aspect of the energy sector, providing significant benefits for various stakeholders, including energy companies, consumers and regulators. Here are a few key points about its importance:

- **Decision-Making of Market Participants:** For those involved in the production, distribution, and sale of electricity, accurate price forecasting is essential for operational planning, risk management, and strategic decision-making.

It enables them to optimize their operations, manage the risks associated with price volatility, and plan their investments and maintenance schedules effectively[19].

- **Market Efficiency:** In deregulated markets, where electricity is traded using spot and derivative contracts, forecasting helps maintain market efficiency. It allows participants to hedge against volume and price risks due to electricity's unique characteristics, such as non-storability and the need for a constant balance between supply and demand[19].

- **Consumers:** For consumers, both residential and commercial, understanding future price trends is important for budgeting and energy usage decisions. Accurate forecasts can lead to more informed choices regarding energy consumption and investments in energy efficiency or renewable energy sources[20].

- **Economic Impact:** The price of electricity affects all levels of economic activity. Forecasting helps stakeholders in cash flow analysis, capital budgeting, and financial procurement. It also aids in regulatory rule-making and integrated resource planning[21].

- **Volatility Management:** The electricity market is known for its extreme price volatility. Accurate forecasting helps in managing this volatility and is particularly important for power portfolio managers who operate in day-ahead trading[22].

- **Cost Savings:** Even a 1% reduction in the mean absolute percentage error (MAPE) of short-term price forecasts can lead to significant savings. For instance, a utility with a 1GW peak load could save around $300,000 per year[21].

- **Operational Efficiency:** For grid operators, predicting electrical consumption ahead of time, from an hour to weeks or even longer, is crucial for maintaining operational efficiency and ensuring the stability of the power system[23].

- **Policymakers:** Policymakers rely on electricity price forecasts to develop regulations that ensure a stable and fair market. Forecasts help in setting policies that can balance the interests of producers, consumers, and the environment. They also provide insights into future market trends, which is crucial for planning national energy strategies and ensuring energy security[19].

1.4 Factors Influencing Electricity Prices

Electricity prices are a reflection of a complex interplay of various factors. These prices are not only crucial for household budgets but also for the economy at large. Understanding these factors can help consumers and policymakers make informed decisions. The most important factors are the followings:

- **Fuel costs** are a primary driver of electricity prices. The majority of electricity generation involves the burning of fossil fuels, such as coal, natural gas, and oil. When the prices of these fuels rise, so does the cost of electricity production. For instance, natural gas prices can fluctuate significantly due to market demand, supply constraints, or geopolitical tensions, directly impacting electricity prices[24].

- **Power Plant Cost**: The construction, maintenance, and operation of power plants represent significant capital and operational expenditures. These costs vary depending on the type of plant, with nuclear and renewable energy plants typically requiring higher initial investments compared to fossil fuel plants. The financing costs of these plants, along with their operational efficiency, contribute to the overall cost of electricity[24].

- **Demand and Supply Dynamics**: Electricity prices are highly sensitive to the balance of demand and supply. During peak demand periods, such as hot summer afternoons, prices tend to spike as more expensive and less efficient power plants are brought online to meet the increased load. On the supply side, disruptions in fuel supply or generation capacity can lead to price increases[24].

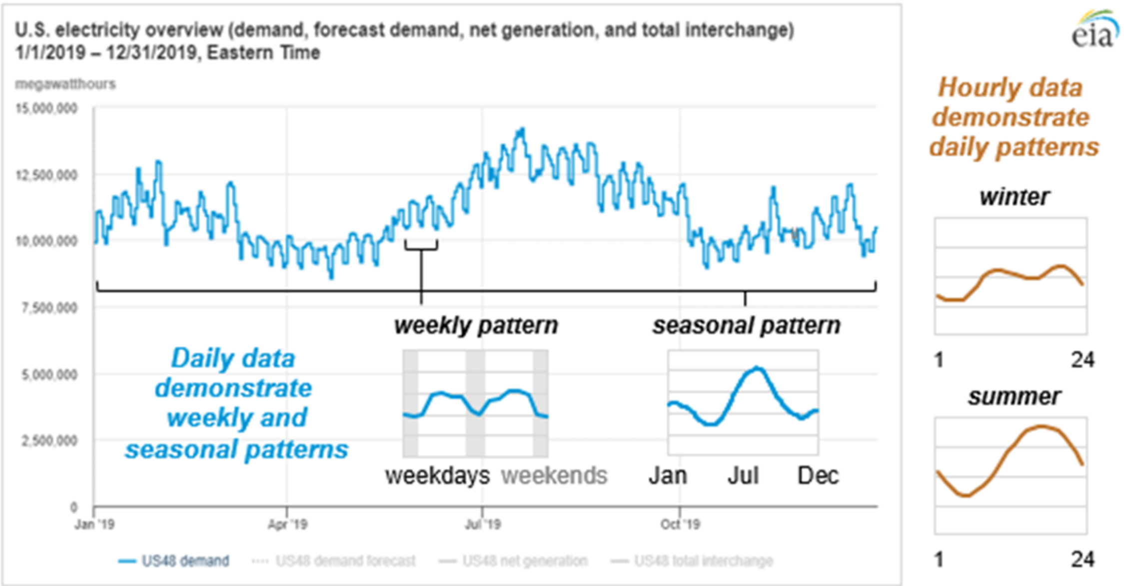

Figure 1.3: Seasonality of Electricity Prices

- **Weather Conditions**: Weather plays a significant role in electricity pricing. Extreme temperatures can lead to increased demand for heating or cooling, which in turn raises electricity prices. Conversely, favorable weather conditions can reduce costs; for example, ample rainfall can bolster hydropower generation, while strong winds can increase wind power output, both of which can lower electricity prices. This seasonal pattern is crucial for managing electricity systems and setting rates for consumers[24]. In Figure 1.3 the seasoniality of Electricity Prices is shown.

- **Renewable Energy Integration**: The integration of renewable energy sources like solar and wind into the electricity grid is increasingly influencing prices. While these sources can provide low-cost electricity, their intermittent nature requires grid operators to maintain backup generation capacity, which can be costly. Additionally, the need for grid upgrades to accommodate distributed generation sources can affect electricity prices[24].

- **Transmission and Distribution System Cost**: The electricity transmission and distribution systems—the grid—entail substantial costs. These include the construction and maintenance of power lines, substations, and other infrastructure necessary to deliver electricity from power plants to consumers.

Additionally, costs incurred from repairing damages caused by extreme weather or upgrading systems for better efficiency and cybersecurity are factored into electricity prices[24].

- **Regulatory Environment**: Regulations can have a profound impact on electricity prices. Governments may impose taxes, subsidies, or price controls that directly affect the cost of electricity. Environmental regulations, such as emissions trading schemes or renewable energy mandates, can also influence prices by affecting the types of fuels used for electricity generation[24].

- **Cross-Border Transactions**: Cross-border transactions enhance electricity market efficiency and competition, often reducing prices. They enable resource optimization and access to diverse, cheaper energy sources. This integration can stabilize prices and promote energy security. Ultimately, it supports a more economical and reliable electricity supply[25].

- **Market Deregulation and Competition**: In deregulated markets, competition among electricity providers can lead to more efficient operations and potentially lower prices. However, the transition from a regulated to a deregulated market can also lead to short-term price volatility as the market adjusts to new dynamics[26].

- **Technological Advancements**: Technological advancements in electricity generation, transmission, and distribution can lead to cost reductions. For example, improvements in turbine efficiency, energy storage solutions, and smart grid technologies can help lower the cost of electricity over time[27].

- **Global Energy Markets**: The global energy market is another factor that affects electricity prices. Events such as oil embargoes, natural disasters affecting key energy infrastructure, or shifts in global energy policy can have ripple effects on electricity prices worldwide[27].

1.5 Challenges in Electricity Price Forecasting

Energy price forecasting is a critical component of energy economics and policy-making. Accurate forecasts enable better planning, investment decisions, and policy formulation. However, the task is fraught with challenges due to the inherent volatility and unpredictability of energy markets.

- **Volatility of Prices** Electricity prices are highly volatile, influenced by a range of factors such as fuel costs, weather conditions, and market dynamics. This volatility makes it difficult to predict prices with high accuracy. Short-term prices can fluctuate wildly due to immediate supply and demand imbalances, while long-term prices are affected by structural changes in the market[28].

- **Impact of Unforeseen Events** Unpredictable events such as natural disasters, geopolitical tensions, or sudden regulatory changes can disrupt energy markets and lead to significant price swings. These events are difficult to predict and can have a profound impact on energy price forecasts[28].

- **Complexity of Energy Markets** Energy markets are complex systems with many interconnected components. The interaction between physical infrastructure, market mechanisms, and regulatory frameworks adds layers of complexity to price forecasting[29].

- **Dependence on External Factors** Energy prices are not only determined by market dynamics but also by external factors such as political stability, economic policies, and technological advancements. Changes in these areas can have unexpected effects on energy prices[28].

- **Data Availability and Quality** High-quality, timely data is essential for accurate forecasting. However, data may be incomplete, outdated, or inconsistent, which can lead to inaccurate forecasts[29].

- **Modeling Challenges** Forecasting models need to account for a wide range of variables and their interactions. However, models may become obsolete as market conditions change, and they may not capture the full complexity of the market. Moreover, the choice of model and its parameters can significantly influence the forecast outcome[29].

- **Technological Advancements** While technological advancements can improve forecasting accuracy, they also introduce new challenges. For example, the increasing integration of renewable energy sources, which have variable output, complicates the forecasting process[29].

- **Regulatory and Policy Uncertainty** Regulations and policies can change rapidly and have a direct impact on energy prices. Forecasters must anticipate potential policy shifts, which is inherently uncertain[28].

- **Market Participant Behavior** The behavior of market participants, including producers, consumers, and traders, can be unpredictable and can influence prices. Forecasting models must try to anticipate human behavior, which is not always rational or predictable[30].

- **Integration of Renewable Energy** The growing share of renewable energy in the energy mix introduces additional uncertainty. The variability of renewable energy production, dependent on weather conditions, poses a challenge for forecasters[29].

1.6 Thesis Structure

In the forthcoming chapter, we will first provide a foundation in machine learning. Following that, we will explore various approaches that have been applied to Electricity Price Forecasting (EPF). Additionally, we will delve into some studies that investigate the effectiveness of machine learning techniques in predicting electricity prices.

Chapter 3 introduces the various components of the dataset. It offers detailed explanations concerning the selection of features. To be more precise, it utilizes twelve distinct categories of features, encompassing a total of 168 features.

Chapter 4 details a series of experimental trials and their corresponding outcomes, which have been conducted for the MGP electricity price forecasting endeavor. The findings indicate that a transition from conventional to advanced methodologies results in a progressive improvement in performance.

Chapter 5 presents, firstly, the conclusions drawn from this study and, secondly, it discusses a range of ideas for potential future expansions.

Chapter 2

Machine Learning Algorithms

2.1 Machine Learning Background

The process of forecasting electricity prices falls under the category of supervised learning because the training data provided to the algorithm contains the expected outcomes, known as labels. The objective here is to forecast a quantitative target variable, like the price of electricity in the MGP market, using a collection of variables known as predictors. This task is referred to as regression. For the system to learn effectively, it requires numerous instances that encompass both the predictors and their corresponding labels.

This chapter aims to introduce various algorithms suitable for predicting electricity prices. **Section 2.1.1** outlines the features of widely recognized machine learning algorithms deemed adequate for regression tasks. **Section 2.1.2** discusses the most robust ensemble machine learning algorithms, which hold promise due to their proven effectiveness across diverse applications. Finally, **Section 2.1.3** details the properties of Artificial Neural Networks (ANNs), which form the foundation of Deep Learning. Their strength lies in their ability to handle large datasets and model complex, non-linear relationships within the data.

2.1.1 Traditional Machine Learning Algorithms

Regression algorithms are a category of supervised machine learning algorithms, which themselves are part of the broader set of machine learning techniques. The hallmark of supervised learning is its ability to establish and utilize the connections between input variables and the predicted output to estimate values for new, incoming data. Specifically, regression algorithms use the features of the input data to forecast numerical outcomes. The typical process involves constructing a model through an algorithm that learns from training data, and then applying this model to make predictions about new, unseen data. Common examples of regression algorithms include linear regression, decision trees, support vector machines, and nearest neighbor methods.

2.1.1.1 Linear Models

The algorithms described in this section constitute a collection designed for regression purposes, where it is anticipated that the target variable will be a linear combination of the input features.

The mathematical notation, if ŷ is the predicted value, is:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + ... + w_p x_p \tag{2.1}$$

The vector $w = (w_1, ..., w_p)$ is designed as a coefficient and w_0 as interpret.

Linear Regression

Linear regression is a statistical method used to explore the association between continuous variables. It is based on a linear model which posits that the dependent variable (y) is a linear function of the independent variables (X). In this model, y is derived as a linear combination of the X variables. If there is only one independent variable (x), this method is known as simple linear regression. Conversely, when multiple independent variables are involved, it is called multiple linear regression.

To reduce the sum of the squares of the residuals, which is the difference between the actual observed outcomes in the dataset and the outcomes predicted by the linear model, linear regression employs a model with coefficients $w = (w_1, ..., w_p)$. The mathematical formulation of the problem that needs to be addressed is as follows:

$$\min_{w} \|Xw - y\|_2^2 \tag{2.2}$$

Ridge Regression

Ridge regression tackles certain issues inherent in Ordinary Least Squares by applying a penalty to the magnitude of the coefficients. The goal is to minimize the sum of squared residuals, adjusted by a penalty on the ridge coefficients. This approach helps to prevent overfitting by shrinking the coefficients, leading to a more robust model:

$$\min_{w} \|Xw - y\|_2^2 + \alpha\|w\|_2^2 \tag{2.3}$$

The degree of shrinkage is governed by the complexity parameter, which is greater than or equal to zero ($a \geq 0$). As this parameter's value increases, the shrinkage becomes more pronounced, enhancing the coefficients' resistance to col-linearity.

Lasso

The LASSO algorithm, which stands for Least Absolute Selection Shrinkage Operator, applies a restriction on the parameters to determine the extent of shrinkage. It aims to identify a subset of predictors that minimizes the prediction error for a quantitative response variable by enforcing a limitation on the model parameters. Consequently, this leads to the regression coefficients of certain variables being reduced to zero.

Following the shrinkage procedure, any variables with a regression coefficient of zero are omitted from the model. The response variable ultimately has the strongest

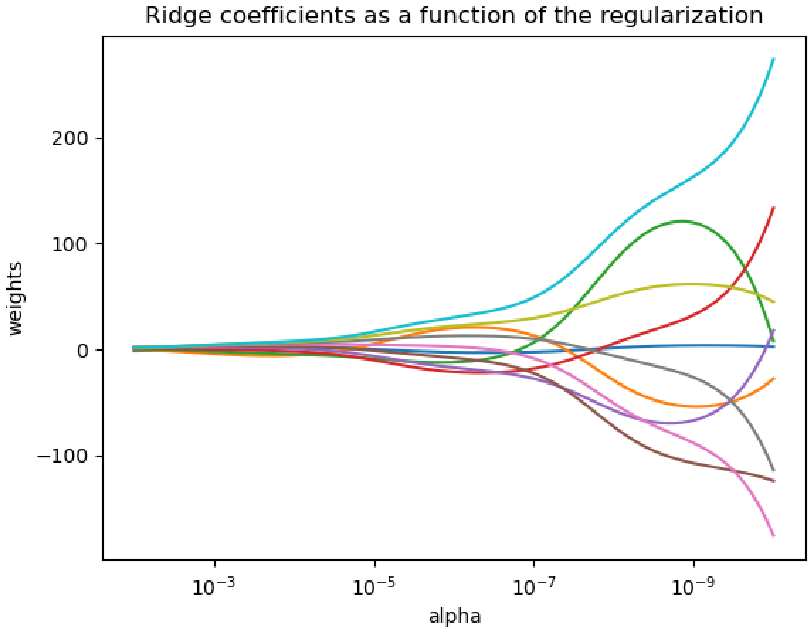

Figure 2.1: Ridge coefficients as a function of the regularization

association with variables that have non-zero regression coefficients. As a method of both shrinkage and variable selection, lasso regression analysis aids analysts in identifying the most significant predictors.

In mathematical terms, the LASSO model is formed by incorporating a regularization term into a linear model. The function that needs to be minimized, which is the objective function, is defined as follows:

$$\min_{w} \frac{1}{2n_{\text{samples}}} \|Xw - y\|_2^2 + \alpha\|w\|_1 \tag{2.4}$$

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[31].

ElasticNet

ElasticNet is a type of linear regression model that is trained using both l_1 and l_2-norm regularization on the coefficients. This leads to the creation of a sparse

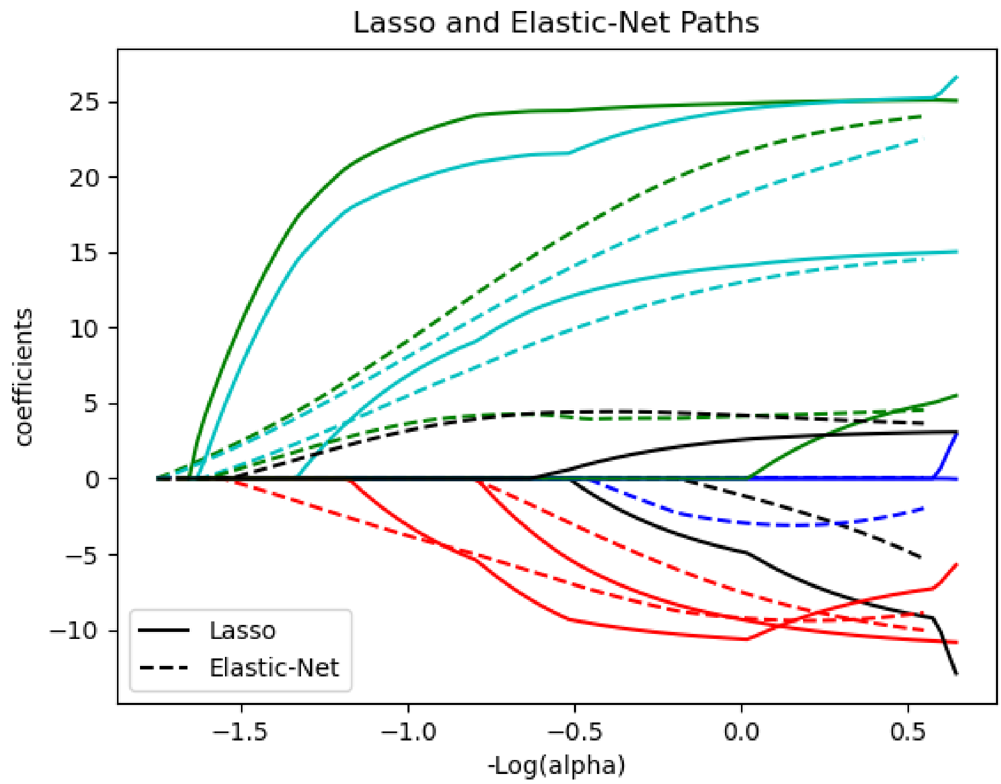

Figure 2.2: Lasso and ElasticNet paths

model, similar to Lasso, where only a small number of weights are non-zero, yet it retains the regularization characteristics of Ridge regression. The balance between l_1 and l_2 regularization is managed by adjusting the l_1 ratio parameter.

ElasticNet is particularly effective when dealing with multiple interrelated features. Unlike Lasso, which may randomly select one of these features, ElasticNet tends to include both. A benefit of balancing Ridge and Lasso is that ElasticNet can acquire some of Ridge's stability with respect to rotation.

The objective function to minimize is in this case:

$$\min_{w} \frac{1}{2n_{samples}} \|Xw - y\|_2^2 + \alpha\rho\|w\|_1 + \frac{1}{2}\alpha(1-\rho)\|w\|_2^2 \tag{2.5}$$

ElasticNet is generally favored over Lasso because Lasso can produce inconsistent results when multiple features have strong correlations or when there are more

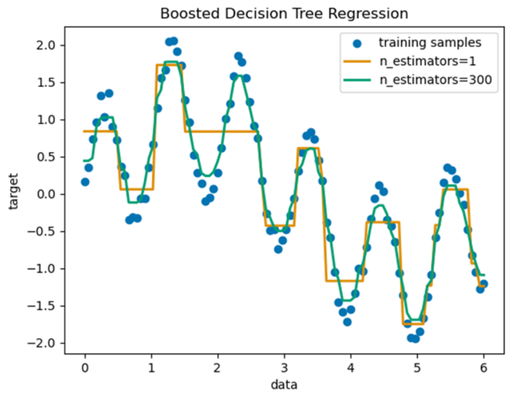

Figure 2.3: A decision tree is boosted using the AdaBoost

features than training samples. ElasticNet is considered more reliable in these situations.

2.1.1.2 Decision Trees

Primarily utilized in regression and classification, Decision Trees stand as a non-parametric supervised learning method. The objective is to develop a model capable of predicting the value of a target variable through the acquisition of straightforward decision rules inferred from data characteristics. With increased tree depth, these decision rules grow in complexity, resulting in a more refined fit of the model.

Decision trees offer a host of benefits. They stand out from other algorithms due to their minimal data preparation needs and their ease of comprehension and interpretation. Additionally, the computational cost of employing decision trees scales logarithmically with the quantity of data points used for training. These trees are adept at processing both numerical and categorical data and can effectively handle problems with multiple outputs. However, the risk of overfitting is present when trees become too complex and fail to generalize the data effectively. Pruning techniques, which limit the minimum number of samples at a leaf node or set a maximum depth for the tree, can mitigate this issue. Moreover, slight changes in the data can lead to the formation of an entirely different tree, and there's a risk of bias if one class dominates. Therefore, it's advisable to balance the dataset before fitting it to the decision tree.

2.1.1.3 Nearest Neighbors

The goal of nearest neighbor methods is to identify a set number of training examples that are nearest to a new data point and use their labels to make a prediction. The user can specify the number of neighbors to consider (as in k-nearest neighbor learning), or the number can adjust depending on the concentration of nearby points (as in radius-based neighbor learning). While any distance metric can be employed, the Euclidean distance is typically the default choice. Neighbors-based algorithms are characterized by their ability to "memorize" all training data, which is why they are categorized as non-generalizing machine learning techniques.

Scikit-learn implements two distinct types of neighbors regressors: The KNeighborsRegressor requires the user to determine a specific integer value for 'k', which then guides the model to learn from the 'k' closest neighbors to each point of inquiry. On the other hand, the RadiusNeighborsRegressor allows the user to assign a floating-point number to a variable 'r', setting the learning to consider only those neighbors that fall within a fixed radius 'r' from the point in question[32].

In the standard nearest neighbors regression, all points in the local vicinity are given equal importance when predicting the outcome for a new data point. However, it can sometimes be beneficial to assign greater significance to points that are closer to the query point, as they are likely more relevant. This weighting is controlled by the 'weights' parameter. By default, 'weights = "uniform"' means every point has the same influence. If 'weights = "distance"', the influence of each point is inversely proportional to its distance from the query point. Additionally, one can supply a custom function based on distance to determine the weights.

2.1.1.4 Support Vector Machines

A Support Vector Machine (SVM) is a versatile and robust machine learning approach, extensively utilized for linear or nonlinear classification, regression, and the identification of anomalies.

Employing Support Vector Machines offers numerous benefits. They are particularly adept at working within spaces with a high number of dimensions, even when

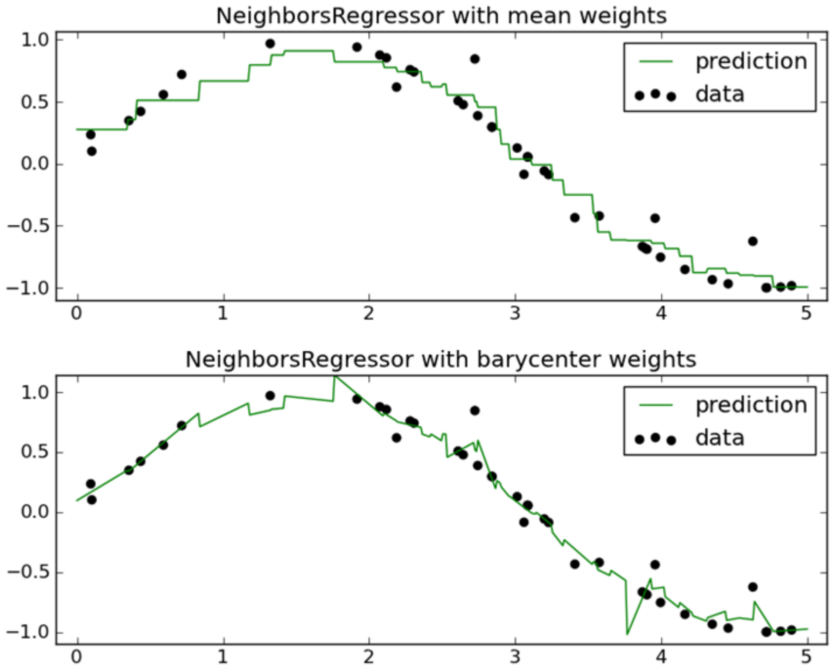

Figure 2.4: k-Nearest Neighbor and the interpolation of the target using both barycenter and constant weights

the dimensions outnumber the samples. SVMs utilize a select group of training data points, known as support vectors, in their decision-making process, which enhances memory efficiency. Additionally, their adaptability is notable, as they allow for the specification of various Kernel functions to tailor the decision function.

Support Vector Regression (SVR) maintains the key features of Support Vector Machines (SVMs) and seeks to establish a curve that fits the provided data points. Unlike classification tasks where the curve serves as a boundary for decision-making, in SVR, the focus is on aligning a vector with a specific point on the curve. Support vectors play a crucial role in accurately aligning the data points with the underlying function they signify. By maximizing the margin between the regression curve and the support vectors, the model approximates the true curve more closely, accounting for the inevitable noise in the data samples.

In our research, we utilize the SVR library from scikit-learn. The SVR model in scikit-learn includes an ‘epsilon’ parameter that governs the loss function. As per the documentation, this parameter defines the ‘epsilon-tube’, a range where predictions falling within ‘epsilon’ distance from the true values are not penalized in the training loss function.

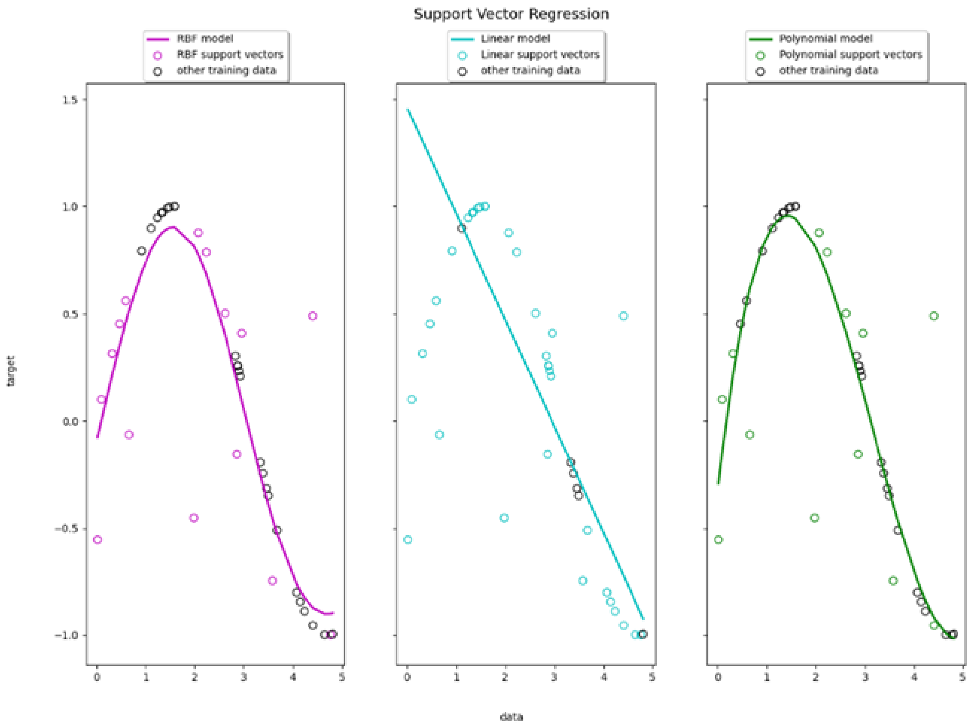

Figure 2.5: 1D regression using linear, polynomial and RBF kernels

2.1.1.5 Gaussian Processes

Gaussian Processes (GP), as a broad-based supervised learning approach, are crafted to tackle both regression and probabilistic classification tasks. The benefits of Gaussian Processes include:

- GPs provide a probabilistic framework that not only predicts the most likely outcome but also quantifies uncertainty around it. This is crucial in scenarios where understanding uncertainty is as important as making predictions.

- Unlike point estimates, GPs yield a distribution over functions. For instance, when predicting temperature patterns, GPs provide a range of possible temperatures along with probabilities, offering a comprehensive view of future possibilities.

- GPs adjust their complexity based on the data, making them suitable for both simple and intricate datasets. They capture complex patterns without rigidly predefined structures.

- Kernels (covariance functions) allow GPs to handle non-linearities, model complex relationships, and extrapolate and interpolate data from observed points. Proper kernel selection is essential for effective GP regression or classification[33].

Some of the disadvantages of GP are:

- GPs can be computationally expensive, especially for large datasets. Inverting the covariance matrix involves cubic time complexity.

- Storing the covariance matrix can be memory-intensive, limiting scalability.

- Selecting an appropriate kernel is crucial. Different kernels capture different structures, and choosing the wrong one may lead to suboptimal results.

- While GPs provide accurate predictions, interpreting the learned functions can be challenging due to their flexibility[34].

2.1.2 Ensemble Machine Learning Algorithms

Ensemble methods aim to enhance the generalizability and robustness of predictions by merging the outputs of multiple simple estimators created using a specific learning algorithm. This approach is designed to offer better performance than using a single estimator alone. Typically, ensemble methods can be categorized into two main types:

- **Averaging Methods:** The core idea behind averaging methods is to independently construct several estimators and then calculate the mean of their predictions. This process typically results in a reduction of the combined estimator's variance, leading to an overall improvement in performance over any individual base estimator. Examples of averaging methods include Random Forests and Bagging.

- **Boosting Methods:** Boosting methods involve incrementally building base estimators. Each new estimator focuses on decreasing the bias of the cumulative estimator. By following this strategy, an effective ensemble method is formed by amalgamating multiple simple models, often referred to as 'weak' models. AdaBoost and Gradient Tree Boosting are instances of boosting methods.

For the purposes of our exercise, we are using Random Forest Regressor, XGBoost Regressor, Bagging Regressor and Extra Trees Regressor.

A Random Forest Regressor is an advanced machine learning model that operates by constructing a multitude of decision trees during the training phase and outputting the average prediction of the individual trees. This ensemble method is particularly effective for regression tasks, where the goal is to predict continuous outcomes. It works by fitting numerous decision trees on different subsets of the dataset and then averaging their predictions to improve the predictive accuracy and control over-fitting. The random forest algorithm introduces randomness into the model building process by selecting different subsets of the data for each tree and by choosing a subset of features for splitting nodes. This randomness helps in making the model more robust and less prone to overfitting on the training data. As a result, random forest regressors often perform well on complex datasets with many features and can provide insights into the importance of each feature in predicting the target variable[35].

The XGBoost Regressor is a powerful machine learning algorithm that falls under the umbrella of ensemble methods, specifically the boosting category. It stands for eXtreme Gradient Boosting and is designed to be highly efficient, flexible, and portable. XGBoost operates on the principle of gradient boosting, where it builds a series of decision trees sequentially, with each tree attempting to correct the errors of its predecessor. This results in a model that combines the strengths of multiple weak learners to form a strong predictive model. XGBoost is particularly known for its speed and performance, which is achieved through optimizations such as parallel processing and tree pruning. The algorithm also offers built-in support for handling missing values and regularization, which helps prevent overfitting. It's widely used in various data science challenges and competitions for its ability to handle large datasets and produce accurate regression predictions[36].

The Bagging Regressor is a type of ensemble meta-estimator that enhances the stability and accuracy of machine learning algorithms. It operates by fitting multiple base regressors, each on random subsets of the original dataset, and then combines their individual predictions through voting or averaging to produce a final prediction. This technique effectively reduces the variance of the model, especially for complex models like decision trees, leading to improved performance. The term 'bagging' comes from Bootstrap Aggregating, reflecting the method's use of bootstrap samples to train each base estimator. By aggregating the predictions of diverse models trained on different data samples, the bagging regressor mitigates the risk of overfitting and is robust against the variability of the data[37].

The Extra Trees Regressor is an ensemble machine learning algorithm that is part of the tree-based modeling family. It stands for Extremely Randomized Trees Regressor and, like other random forest methods, it builds numerous decision trees to make predictions. However, the Extra Trees Regressor introduces even more randomness into the process: when splitting each node during the construction of the trees, it uses random thresholds for each feature rather than searching for the best possible thresholds. This method of randomization tends to make the model more robust against overfitting. The Extra Trees Regressor works by fitting a number of randomized decision trees on various sub-samples of the dataset and then using averaging to improve predictive accuracy and control over-fitting. It's particularly useful in scenarios where the data might contain a lot of noise[38].

2.1.3 Deep Learning Algorithms

Artificial Neural Networks (ANNs) form the foundation of Deep Learning. Due to their robustness, adaptability, and scalability, they are exceptionally suited for addressing substantial and intricate Machine Learning challenges. This includes tasks like categorizing vast numbers of images (such as those found on Google Images[39]), enabling voice recognition features (like Apple's Siri[40]), suggesting the most engaging videos to watch for countless users every day (such as on YouTube), or mastering complex games to the extent of outperforming the world champions, as seen with DeepMind's AlphaGo[41].

Deep learning neural networks are equipped with a range of remarkable abilities. They are capable of handling numerous inputs and outputs while autonomously discovering intricate mappings from the inputs to the outputs. These potent attributes hold significant promise for forecasting time series, particularly for assignments characterized by complex nonlinear relationships, multiple variables, and multi-step predictions. Coupled with the advanced functionalities of contemporary neural networks, there is considerable anticipation for their potential. This includes the inherent support for sequential data offered by recurrent neural networks and the autonomous extraction of features found in convolutional neural networks.

The Multilayer Perceptron (MLP) is a class of feedforward artificial neural network that has proven to be crucial for time series forecasting. Its importance lies in its ability to model complex and nonlinear relationships inherent in time series data. An MLP consists of multiple layers of nodes, each layer fully connected to the next, which allows it to learn from data in a deep and hierarchical manner. This structure enables the MLP to capture temporal dependencies and patterns that are not immediately apparent, making it highly effective for forecasting future values in a series based on past observations. The versatility of MLPs allows them to be applied to a wide range of time series forecasting problems, from univariate to multivariate and from single-step to multi-step forecasts. Their ability to learn arbitrary complex mappings from inputs to outputs is particularly beneficial for tasks with nonlinear dependencies, which are common in time series data[42]. The MLP's significance is further underscored by its successful application in various domains, such as financial market predictions, weather forecasting, and demand forecasting, where accurate time series forecasting is essential[43][44][45].

Long Short-Term Memory (LSTM) networks have emerged as a powerful tool in the realm of energy price forecasting due to their ability to capture long-term dependencies and patterns in time series data. LSTMs are a type of Recurrent Neural Network (RNN) specifically designed to avoid the long-term dependency problem, making them particularly suitable for predicting complex sequences such as electricity prices which are influenced by a myriad of factors over time. A study on the GitHub platform demonstrates the application of LSTM models, among others, for forecasting electricity demand and prices, highlighting their capability to integrate energy consumption and weather-related metrics from multiple cities in Spain, thus addressing a multivariate time series forecasting challenge[46].

In another innovative approach, a hybrid model combining LSTM with XGBoost, known as LSTM-XGBoost, has been introduced for load forecasting in energy communities based on smart meter data. This model separately forecasts the general load pattern and peak loads, which are then combined to form a comprehensive forecasting model. The hybrid model has shown to outperform traditional forecasting methods, indicating the potential of LSTM networks when fused with other machine learning techniques to enhance the accuracy of energy price predictions. Such advancements in LSTM-based forecasting models are pivotal for the energy sector, enabling more precise energy management and operation strategies[47].

2.2 Electricity Price Forecasting in Literature

Over the past 20 years, numerous techniques and concepts have been explored for forecasting electricity prices, achieving different levels of effectiveness. Generally, these can be categorized into six distinct groups.

2.2.1 Multi-agent models

Multi-agent models are designed to replicate the workings of a system comprising diverse entities such as power generators and retail companies. These entities engage in mutual interactions. The models tackle the pricing mechanism by aligning market supply with demand[48]. Included in this category are models based on equilibrium or game theory strategies (such as the Nash-Cournot framework and supply function equilibrium), as well as cost-driven and agent-based models. Typically, multi-agent models prioritize qualitative insights over quantitative outcomes. However, this can present challenges when precise quantitative predictions are required, particularly in the accurate forecasting of electricity prices.

Within the Nash-Cournot framework, electricity is regarded as a uniform product, and market equilibrium is achieved by the production capacity choices made by suppliers. However, a common issue with these models is that they often predict prices that exceed actual market prices. To remedy this, the notion of conjectural variations has been introduced. This concept seeks to reflect the reality that competitors tend to boost their output in response to elevated electricity prices. Studies have demonstrated that as the number of companies in the market diminishes, the anticipated price levels notably rise.

The supply function equilibrium (SFE) method models prices by establishing an equilibrium based on the supply and, potentially, demand curves submitted by companies into the wholesale market. To determine the SFE, a series of differential equations must be resolved. However, these models face considerable challenges in terms of numerical solvability. One approach to expedite calculations is to consolidate demand into segments. Yet, this technique might exclude extreme values from the analysis, which is deemed unacceptable for precise electricity price forecasting and risk management.

Furthermore, it's only when demand uncertainty or another form of uncertainty results in a pre-settlement equilibrium that is not predetermined, that the outcomes of supply curve bidding will deviate from the Nash-Cournot equilibrium. In the absence of such uncertainty, the supply curve bidding simplifies to a single point, aligning with the Nash-Cournot equilibrium. To address the computational complexity of general SFE models, linear SFE models have been introduced. These models presuppose linear demand and marginal costs, and the SFE is derived using either linear or affine supply functions. Every market participant is compensated at the marginal clearing price for their supply. Provided there's no transmission congestion, this market-clearing scenario ensures maximum social welfare, as the supply functions are non-decreasing and the clearing price remains consistent across all firms. Although this framework's application to electricity price forecasting is limited, it has been extensively applied in the analysis of bidding strategies, market design, market power, and congestion management[49].

The strategic production-cost model (SPCM) incorporates the bidding strategies of agents, which are influenced by their assumptions about market behavior, known as conjectural variation. Each producer aims to optimize its profits by considering both its cost structure and the anticipated actions of other market participants. This is represented by a parameter that indicates the slope of the residual demand function at various production levels. In the SPCM's simulation of the supply curve construction, it is presumed that an agent is only cognizant of its own costs and its beliefs regarding the change in its residual demand function. Unlike SFE models, there are no iterative processes, so firms cannot refine their bids or respond to their competitors' actions. The SPCM's primary benefit lies in its rapid computational ability, making it well-suited for real-time analysis and offering an advantage over the more computationally intensive Nash-Cournot and SFE models.

In the field of energy economics, agent-based computational economics (ACE) has emerged as a popular method for addressing a variety of practical and theoretical challenges in recent years. ACE utilizes a set of computational guidelines and frameworks to simulate the behavior and interactions of independent agents, which can be individuals or collective entities like groups or organizations. The primary objective of this approach is to evaluate how these agents' behaviors affect the overall system.

Multi-agent models are regarded as highly adaptable instruments for examining strategic actions within electricity markets. While this adaptability is a strength, it also presents a challenge, necessitating that the assumptions underpinning the simulations be substantiated with both empirical and theoretical evidence. Critical components such as the identity of the participants, their interaction methods, possible strategies, and the range of outcomes must be clearly established. Evidently, there exists a considerable risk associated with the modeling process.

2.2.2 Fundamental models

Fundamental methods are designed to understand the essential physical and economic interactions that underpin the trade and generation of electricity[50]. They establish functional connections between key factors such as demand, system characteristics, and meteorological conditions. These primary inputs are then independently defined and forecasted, typically using methods rooted in computational intelligence, statistics, or simplified forms. Broadly speaking, there are two types of fundamental models: those that are structurally simple and streamlined, and those that are detailed, incorporating a wealth of parameters related to demand and supply.

In the realm of fundamental models, practitioners face two primary obstacles. The first is the issue of data availability. The extent and type of data accessible—such as information on plant capacities, costs, demand patterns, and transmission capabilities—vary across different markets. This variability directly influences the ability to construct accurate models. Fundamental models, due to their reliance on such data, are typically better suited for making medium-term forecasts rather than short-term predictions. This is also true for parsimonious structural models, which tend to be adjusted based on daily data, thus overlooking the intricate hourly dynamics.

The second challenge involves the integration of stochastic variations of key drivers into the models. Assumptions about the market's physical and economic interconnections are integral to model construction. Consequently, the resulting price forecasts are highly sensitive to any deviations from these assumptions. The complexity of a model also dictates the level of effort needed to fine-tune its parameters. Therefore, when employing fundamental approaches, there is a substantial risk associated with model construction and the potential need for frequent adjustments to maintain accuracy.

2.2.3 Reduced-form models

Reduced-form models are utilized to define the statistical characteristics of electricity prices over time. Their primary purpose is to facilitate risk management and the assessment of financial derivatives.

Finance-inspired reduced form models for price dynamics are primarily focused on capturing the overarching characteristics of daily electricity prices, rather than providing granular hourly price predictions. These models are instrumental in risk management and the valuation of derivatives, as they replicate key aspects like marginal distributions at future points, overall price movements, and the interrelation of various commodity prices[51].

However, the reliability of these models hinges on their ability to accurately reflect the principal attributes of electricity prices. If a model is overly complex, it can become computationally intensive, rendering it impractical for use within trading environments. The methodologies employed by these models are derived from those used for other energy commodities and interest rates, but they also integrate techniques from actuarial science and econometrics. Markov regime-switching models and jump-diffusion models exemplify a balance between capturing the distinct features of electricity prices and maintaining model simplicity.

Typically, reduced-form models are not designed to forecast electricity prices on an hourly basis with high precision. Instead, their strength lies in capturing the essential characteristics of spot electricity prices over a daily period. These models are frequently utilized in risk assessment and the pricing of derivatives, offering a streamlined yet fairly accurate representation of price fluctuations. Notably, when it comes to predicting volatility or sudden price jumps, reduced-form models have demonstrated commendable performance[52].

2.2.4 Statistical models

Statistical approaches utilize a mathematical blend of historical prices and external variables, like consumption levels, output, or meteorological conditions, to forecast current market prices[53]. These statistical models are beneficial for engineers and system managers as they provide a degree of physical context to the data components. Despite facing criticism for their limited ability to capture the typically non-linear patterns of electricity prices and associated key factors, these models have shown to deliver accuracy in real-world scenarios that is on par with their non-linear counterparts.

Additive and multiplicative models represent the two primary classifications in this context. For additive models, the forecasted price is determined by adding together various elements. Conversely, multiplicative models derive the anticipated price by multiplying several factors. Despite the prevalence of additive models, both categories share a close connection. This is because a multiplicative model used for pricing can be transformed into an additive model when applied to the logarithm of prices.

Statistical methods are diverse and include a range of models tailored to different forecasting needs. Among these are models that leverage historical data from similar days and apply exponential smoothing to predict future trends. Additionally, regression models and threshold autoregressive models are employed to understand and anticipate changes in price movements.

Further expanding the toolkit, autoregressive (AR) time series models and ARX time series models, which incorporate external factors, are utilized for their predictive capabilities. To address the complexities of market volatility, models designed to handle irregular variance, such as those accounting for heteroskedasticity, and the more sophisticated Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models, are also part of the statistical approach.

Some experts categorize statistical models as instruments of technical analysis. Technical analysts focus on discerning future asset performance by examining price charts for specific indicators and patterns, rather than calculating the asset's inherent or fundamental worth. While the efficacy of technical analysis is often debated in financial markets, it tends to be more applicable in electricity markets. This is attributed to the predictable seasonal patterns of electricity prices during stable periods without price spikes.

However, the ability of statistical methods to forecast sudden price surges is generally considered inadequate. This limitation is primarily associated with models that rely solely on price data, but it also extends to those incorporating fundamental factors. The academic discourse has yet to reach a consensus on whether statistical models should include price spikes in their estimation processes. Nonetheless, it is widely acknowledged that an appropriate stochastic model is necessary to accurately represent these price spikes.

2.2.5 Computational Intelligence (CI) models

Computational intelligence methods encompass a range of approaches that draw from artificial intelligence, machine learning, and both non-parametric and non-linear statistics. These methods are considered "intelligent" due to their capacity to adjust to the intricacies of dynamic systems through learning, fuzzy logic, and evolutionary computation. The primary tools in computational intelligence for predicting electricity prices include artificial neural networks, support vector machines (SVM), and fuzzy logic systems[54].

The primary advantage of computational intelligence (CI) methods is their adeptness at managing the non-linear and intricate aspects of electricity pricing. This makes CI methods superior to traditional statistical approaches for modeling such complex price behaviors. However, this very flexibility can also be a drawback. The capacity to adjust to non-linear and volatile price trends does not guarantee more accurate predictions. Furthermore, the vast array of CI tools available makes it challenging to pinpoint the best one. Comparing different CI techniques in detail is also problematic. Even when forecasting accuracies are documented for identical test periods and markets, the errors from each technique cannot be directly compared, making it impossible to draw broad conclusions about a method's overall efficacy. Instead, any conclusions must be specific to the particular implementation of a method, its chosen parameters, and the calibration dataset used. This critique applies to all forecasting methods, but it is particularly pertinent for CI methods due to their complex, multi-parameter configurations and inherent non-linearity.

2.2.6 Hybrid models

In the field of electricity price forecasting, it's quite usual to encounter hybrid models within scholarly works. These models are innovative in that they integrate elements from multiple forecasting methods. Classifying these hybrid solutions can be complex, and sometimes it may not even be feasible. A notable instance of such a hybrid

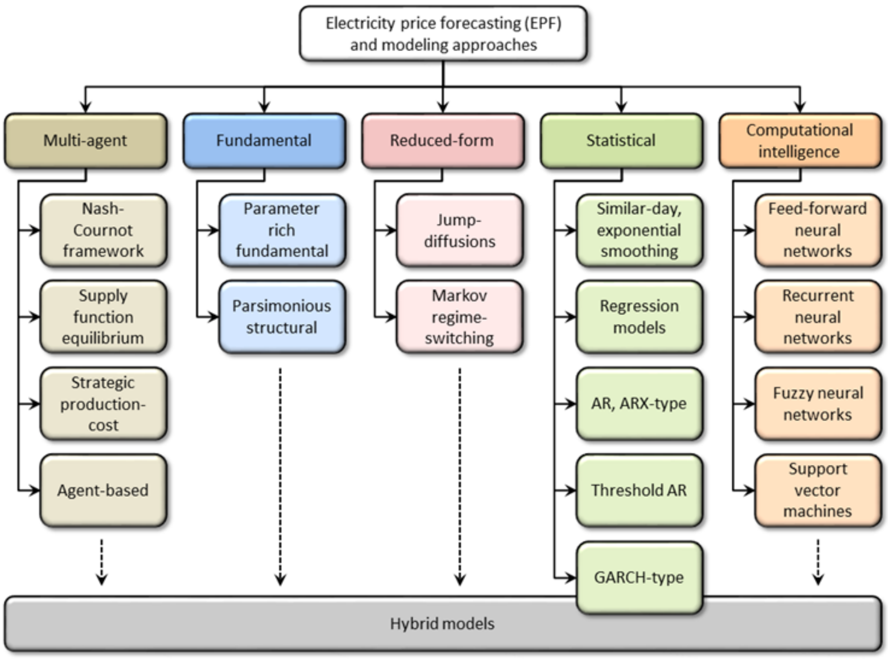

Figure 2.6: A taxonomy of electricity price forecasting (EPF) and modeling approaches according to Weron (2014)

model is the AleaModel[55], created by AleaSoft, which merges the capabilities of Neural Networks with Box-Jenkins models.

2.3 Machine Learning in Literature

Research indicates that both statistical and machine learning techniques[53] are highly effective for predicting electricity prices. However, statistical models, which typically employ linear forecasting, may struggle with high-frequency data[56], such as hourly electricity prices that fluctuate rapidly. On the other hand, these models tend to perform well with lower-frequency data, like weekly trends. To tackle the complexity of predicting nonlinear hourly price movements, various machine learning strategies have been developed. The scholarly community has amassed a broad spectrum of machine learning methods, ranging from conventional to advanced, and it's not uncommon to see a blend of these methods in a single hybrid model. Generally, more complex algorithms, like those used in deep learning, tend to outperform traditional machine learning techniques, which in turn surpass the capabilities of statistical models.

Support Vector Machines (SVMs) are a staple among traditional machine learning algorithms and are frequently assessed by researchers for their efficacy in predicting electricity prices. A hybrid model[57] that merges the auto-regressive integrated moving average (ARIMA) with Support Vector Regression (SVR) is highlighted in the literature. This model leverages the strengths of ARIMA in linear forecasting and SVR in capturing non-linear trends. The results from experiments indicate that this hybrid model surpasses conventional ARIMA models in accuracy, as measured by the mean absolute percentage error. Another innovative approach for short-term electricity price forecasting involves a two-tiered hybrid system combining SVM with a self-organizing map (SOM)[58]. Initially, the SOM network categorizes the input data into several groups autonomously. Subsequently, a set of SVMs is employed to tailor the training data for each group in a controlled manner. The efficacy of this model was validated using historical price data from the New England electricity market.

Artificial neural networks (ANNs) have gained significant attention in recent years for their potential to enhance electricity price forecasting. Researchers are combining various neural network architectures to optimize forecasting results. For instance, an Enhanced Radial Basis Function Network (ERBFN)[59] was introduced, integrating the Radial Basis Function Network (RBFN) with Orthogonal Experimental Design (OED), and applied in the PJM area of the United States. The application of OED to the ERBFN's learning rates has shown to decrease prediction errors, thereby increasing both the accuracy and reliability of the forecasts, including the ability to closely monitor price spikes.

Additionally, innovative hybrid models[60] have been developed, such as one that utilizes a modified wavelet neural network (WNN) coupled with singular spectrum analysis (SSA) to fine-tune initial weights and parameters for improved short-term electricity price forecasting. Another approach combines two deep neural networks—Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN)[61]—which has demonstrated superior performance over traditional machine learning methods like Support Vector Machines (SVM), Decision Trees (DT), and Random Forests (RF). Furthermore, an LSTM model[62] enhanced with the differential evolution (DE) algorithm, termed DE-LSTM, has been used to predict electricity prices with greater accuracy than both statistical and other ANN methods, as evidenced by lower mean absolute percentage error (MAPE) rates. Deep learning models[63], including DNN, LSTM, and Gated Recurrent Unit (GRU), have been proposed and shown to significantly outperform traditional models in predictive accuracy in the day-ahead market of Belgium, achieving MAPE rates of 12-13 % compared to the higher rates of statistical methods.

In Italy, there is a limited body of research investigating the performance of machine learning algorithms for forecasting electricity prices. Iftikhar in his paper[64], discusses a novel approach to predicting electricity prices. The technique involves several nonparametric regression methods and various time-series models to enhance the accuracy and efficiency of day-ahead electricity price forecasting. The method also includes dealing with extreme values, decomposing the price series into subseries, and then forecasting each subseries using different models. The performance of this method was tested using data from the Italian electricity market and he reached a MAPE of around 12%.

Chapter 3

Data Analysis and Feature Selection

3.1 Data Challenges

The foundation of any robust forecasting model lies in the quality of the data collected. Energy markets are dynamic, and the data reflects this volatility, often resulting in missing values or outliers. Missing values can occur due to various reasons, such as equipment malfunctions, transmission errors, or simply gaps in the data recording process. These missing entries pose a significant challenge as they can lead to biased forecasts if not handled appropriately[65].

Noise is another critical factor that complicates energy price forecasting. It can be introduced through measurement errors, data processing, or external factors that are not directly related to price movements. Distinguishing the signal from the noise is a complex task that requires sophisticated analytical techniques[19].

Outliers, on the other hand, can arise due to sudden market changes, policy shifts, or unusual consumption patterns. While outliers can sometimes provide valuable insights into anomalies or unique events, they more often skew the analysis and can lead to inaccurate predictions. Regulatory changes can lead to shifts in market dynamics, such as the introduction of subsidies or taxes, which can affect energy prices directly. These shifts must be reflected in the datasets to ensure accurate forecasting[19]. For instance, policy uncertainty can introduce volatility in energy commodity prices, necessitating datasets that capture these fluctuations for effective forecasting[28]. Accurate and up-to-date datasets that reflect the current regulatory environment are essential for reliable forecasts.

Electricity prices are characterized by their non-constant mean and variation, also known as heterogeneity of variance. This means that the average price level and the volatility of prices change over time, influenced by demand and supply dynamics, generation costs, and market regulations. Traditional linear models often assume constant variance, which is not the case with electricity prices, leading to suboptimal forecasts. The non-constant mean presents its own set of challenges. Energy prices can exhibit trends, seasonality, and cycles, all of which need to be accounted for in the forecasting model. Ignoring these elements can result in a model that is not sensitive to underlying patterns in the data, thus compromising the accuracy of the forecast[29].

To overcome these challenges, advanced statistical and machine learning methods are employed. Techniques such as imputation can address missing values, while robust statistical methods can mitigate the influence of outliers. Noise can be filtered using signal processing techniques, ensuring that the true underlying patterns are captured. For handling non-constant mean and variation, models such as ARIMA (Autoregressive Integrated Moving Average) and GARCH (Generalized Autoregressive Conditional Heteroskedasticity) are commonly used. These models are capable of capturing the dynamic nature of energy prices, accounting for trends, seasonality, and volatility clustering[66].

3.2 Data Sources, Acquisition and Data Explanation

Our goal is to construct an expansive dataset, leveraging the existing expertise in the domain. Our objective is to encompass all the key factors that influence the MGP electricity price formulation. Typically, day-ahead electricity prices are influenced by a variety of factors including fuel costs, consumption levels, renewable energy production (especially from photovoltaics, wind and hydro), the operational status of traditional power sources (such as nuclear facilities and CCGTs), the transmission capacity of cross-border interconnections, and the core attributes of adjacent electrical systems. Our comprehensive dataset is organized into 12 categories, encompassing a total of 168 distinct features. Commonly, studies of this nature focus primarily on data related to consumption and renewable energy source (RES) production. We anticipate that the intricate details incorporated into our dataset will significantly enhance the accuracy of our forecasting models.

All our data was acquired from the ENTSO-E[67]. ENTSO-E is the European Network of Transmission System Operators for Electricity and plays a pivotal role in the European energy market. It is responsible for the coordination of the transmission systems across Europe to ensure the seamless supply of electricity. ENTSO-E's comprehensive data, accessible through their API, is instrumental for energy price forecasting, providing valuable insights into energy consumption, production, and market dynamics.

3.2.1 Target Variable

The variable of interest in our study is the daily Italian electricity spot price, measured in euros per megawatt-hour (EUR/MWh), as reported by the Italian Power Exchange (GME – Gestore Mercati Energetici)[68]. The timeframe of our analysis extends from January 1, 2016, to November 30, 2022. It's important to note that significant changes occurred in the Greek energy market starting from December 1, 2022, due to the introduction of the continuous intraday XBID market. This development has led to a substantial transformation in market dynamics, particularly with the coupling of the Greek market with other European countries. Consequently, it was deemed prudent to limit the scope of our analysis to the period up to this date to ensure the relevance and accuracy of our study in light of these market changes[69]. The subsequent table (table 3.1) provides descriptive statistics which encapsulate the distribution's central tendency, variability, and form of our variable.

The boxplot below (Figure 3.1) provides additional statistical insights into the

Table 3.1: Target variable descriptive statistics

Count	60.624
Mean	94,48
Standard Deviation	104,86
Min	0,00
25th percentile	42,49
50th percentile	55,08
75th percentile	79,31
max	870,00

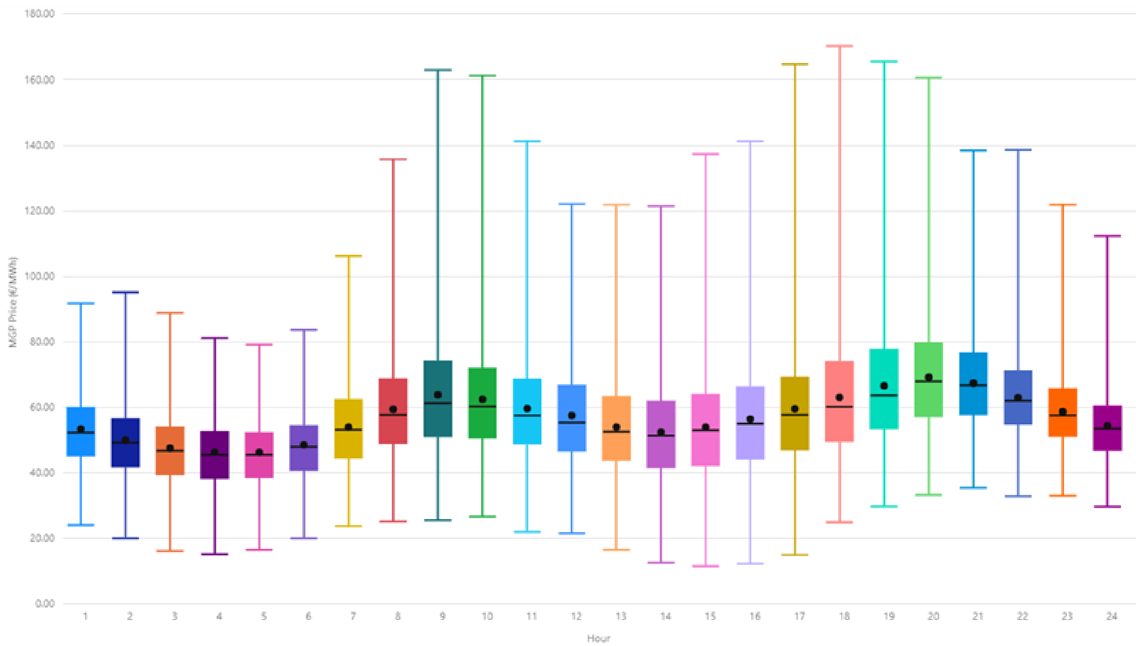

Figure 3.1: Boxplot: Italian spot price per hour of a day

Italian wholesale electricity price. On the horizontal axis, we have the hour of the day, while the vertical axis represents the electricity price in EUR/MWh. Notably, during off-peak hours (hours 1-7), prices tend to be lower. In contrast, during peak hours, prices are not only higher but also exhibit greater volatility.

In addition to their daily fluctuations, electricity prices are influenced by various factors, including electricity demand, which typically peaks during the summer and winter. The subsequent heatmap (Figure 3.2) illustrates the average hourly spot price in MGP across the 7-year duration of our study, commencing in January 2016.

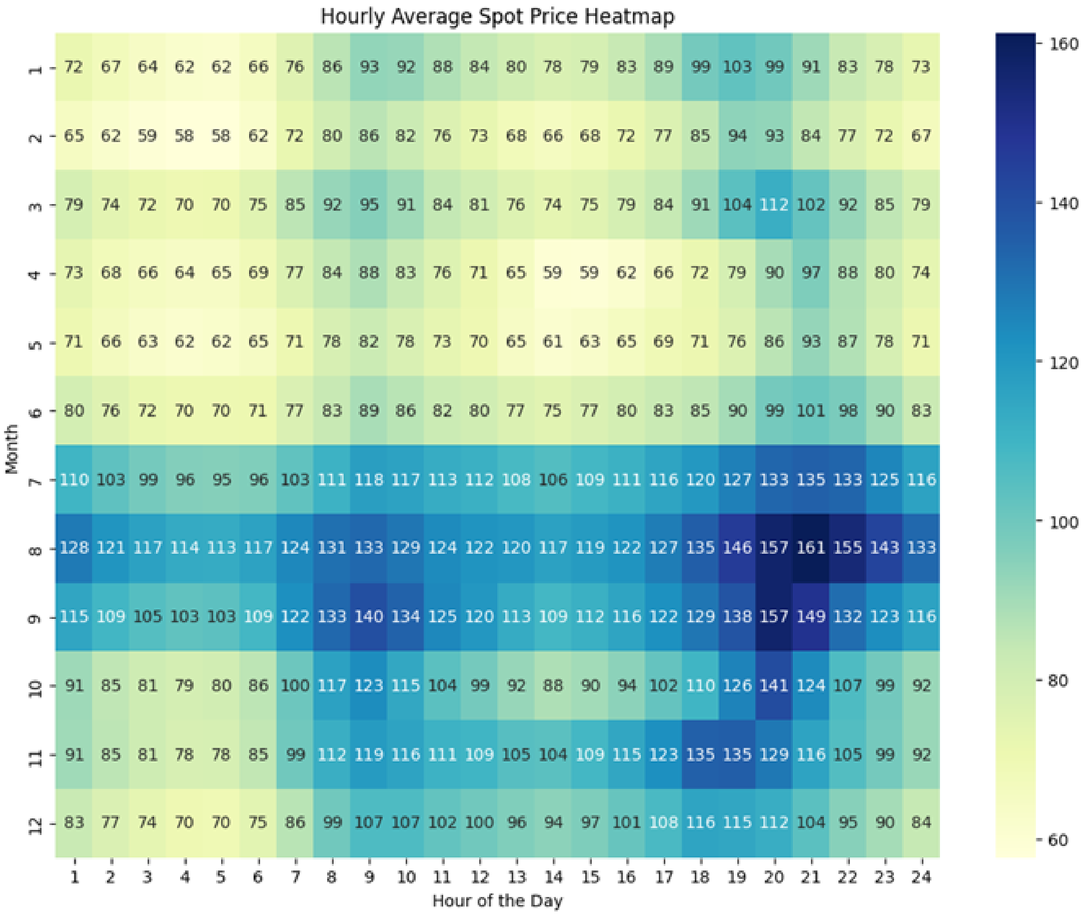

Figure 3.2: Heatmap: Hourly Average MGP spot price per month

3.2.2 Input Variables

Choosing the right features is crucial in forecasting electricity prices. Numerous elements can influence these prices to varying degrees. Factors such as the prices of other commodities (like gas and carbon), seasonal variations in consumption, the output from different energy sources (such as renewable energy sources and nuclear power), and the capacity of cross-border interconnections all play a role in determining the wholesale electricity prices.

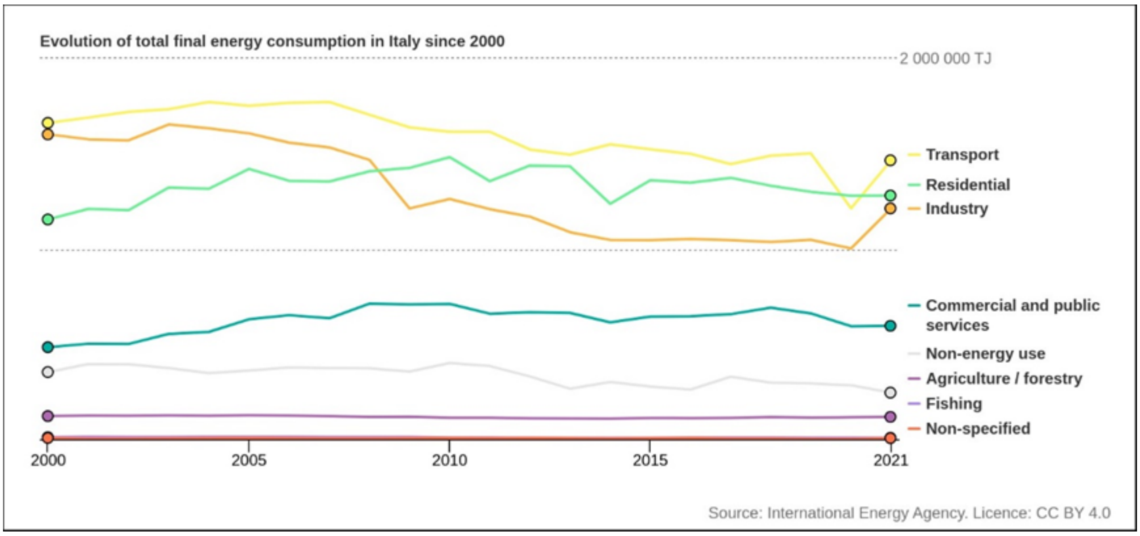

Figure 3.3: Evolution of electricity consumption in Italy since 2000 (Source: IEA)

The line graph (Figure 3.3) depicting the evolution of Italy's total final energy consumption from 2000 to 2021 reveals a comprehensive view of the country's energy usage across various sectors. Transport emerges as the most significant energy consumer, consistently topping the chart throughout the period. The residential and industrial sectors also show substantial energy consumption, with some fluctuations indicative of changing economic and social patterns. Notably, the commercial and public services sector exhibits a steady upward trend in energy use, reflecting growth and development in these areas. The sectors of non-energy use, agriculture/forestry, fishing, and non-specified maintain relatively lower consumption levels. Figure 3.3 underscores the diverse energy demands of Italy's economy and the importance of sector-specific analysis in understanding the nation's energy consumption landscape.

The energy landscape in Italy has undergone significant changes from 2000 to

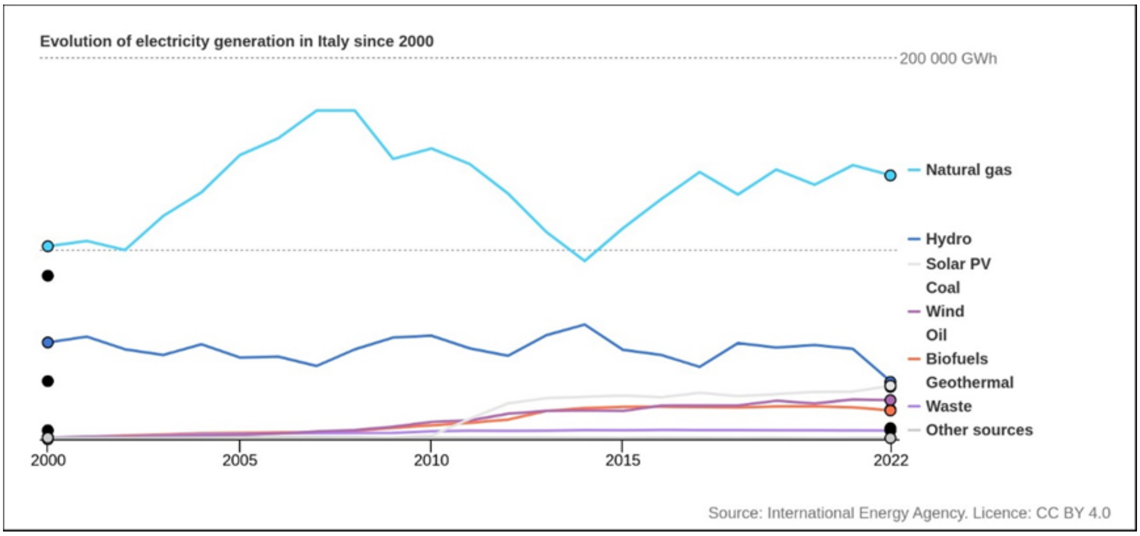

Figure 3.4: Evolution of electricity generation in Italy since 2000 (Source: IEA)

2022, as depicted in Figure 3.4. The graph illustrates a marked decline in the use of natural gas for electricity generation starting around 2010, despite it being the dominant source for many years. Hydroelectric power has shown remarkable stability as a contributor to Italy's energy mix. A notable surge in solar PV generation post-2010 reflects Italy's commitment to renewable energy. Coal usage, after a period of growth, began to decline around 2015, aligning with global trends towards decarbonization. Wind energy, while growing, has not matched the rapid expansion of solar power. The sharp decrease in oil usage after 2005 and the minor roles of biofuels and geothermal energy highlight the country's shift away from fossil fuels. Additionally, the slight increase in waste-to-energy generation indicates a move towards utilizing alternative energy sources. These trends underscore Italy's progressive transition towards a more sustainable and diversified energy portfolio, reducing reliance on fossil fuels and enhancing renewable energy utilization[70].

Italy's journey towards renewable energy since the turn of the millennium has been marked by significant strides and noteworthy trends. As seen in Figure 3.5, hydroelectric power has consistently been the backbone of Italy's renewable energy, despite experiencing some fluctuations, with peaks around 2005 and 2015. The emergence of solar PV generation began in earnest around 2005, and its steady ascent since then highlights Italy's increasing reliance on solar technology. Wind energy, too, has carved out a growing share of the energy mix, with its upward

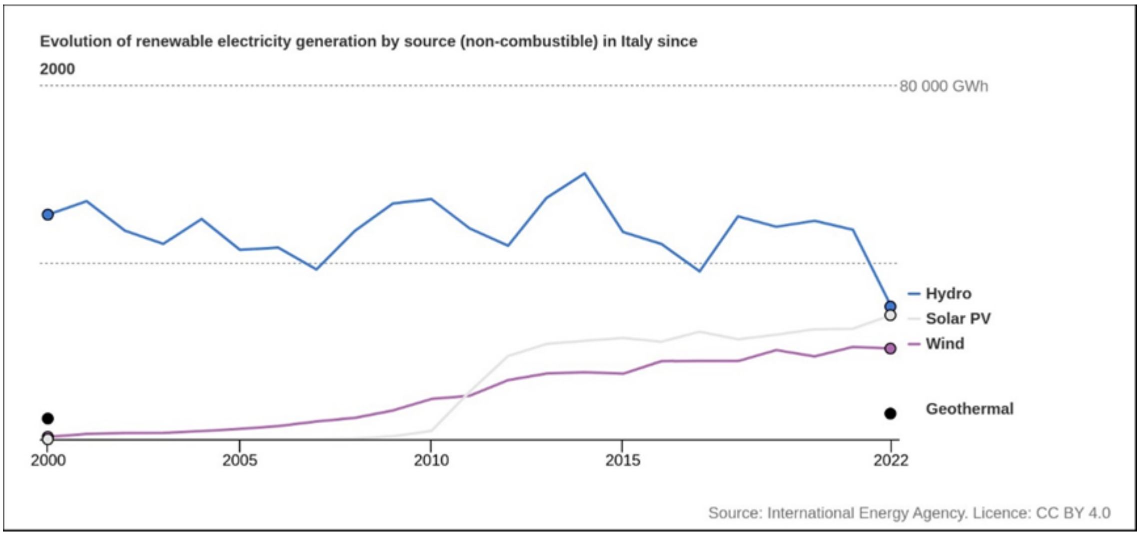

Figure 3.5: Evolution of renewable electricity generation by source (non-combustible) in Italy since 2000 (Source: IEA)

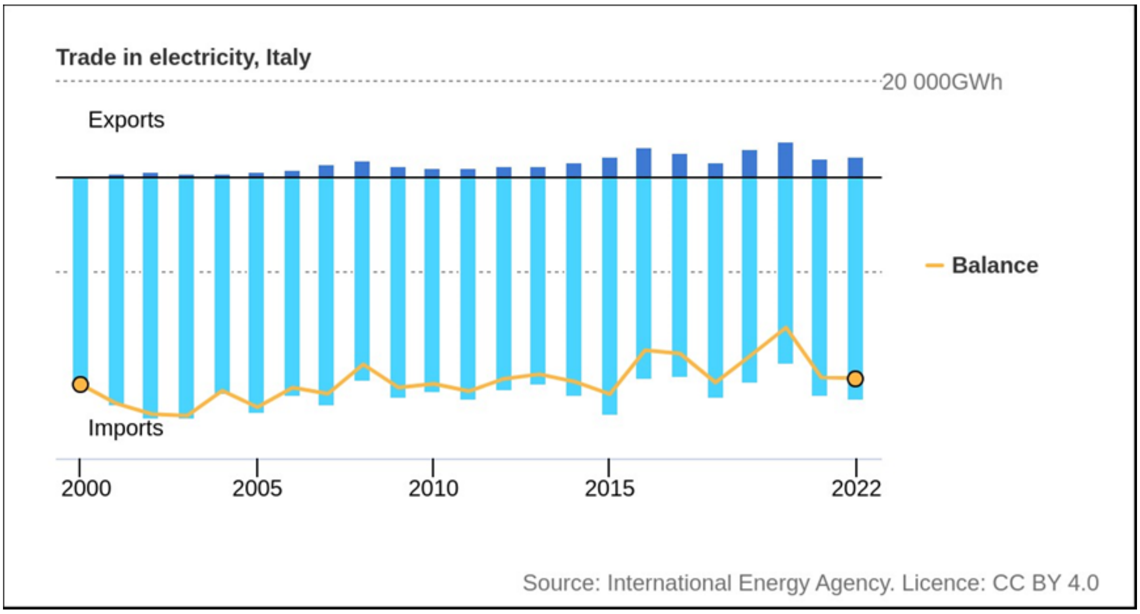

Figure 3.6: Trade in electricity, Italy (Source: IEA))

trajectory beginning concurrently with solar. In contrast, geothermal energy has maintained a steady course, contributing a stable output throughout the years. Figure 3.5 paints a picture of a nation committed to diversifying its renewable energy portfolio, increasingly harnessing the power of the sun and wind to complement its traditional hydroelectric sources.

Historically, Italy has been a net importer of energy, relying heavily on imports to meet its energy demands (Figure 3.6). However, recent trends indicate a shift towards a more balanced trade. The country's energy system has undergone substantial changes since 2010, with a notable increase in natural gas and renewable energies, while coal and oil have seen a decrease.

In this study, we have chosen 12 categories of features, each of which has several variables that can contribute to the forecast of the target variable. As mentioned before, all the data used in this master thesis are available on the website of the European Network of Transmission System Operators for Electricity (ENTSO-E)[67]. In table 3.2, the several types of features are presented.

Table 3.2: Feature Categories

Feature Category	Number of Features
General	2
Prices	5
Demand	12
RES (Wind and Solar) Production	26
Temperature	13
Commodities	2
Internal Exchanges	6
Cross-Border Italy	5
Cross-Border	20
RES Installed Capacity	26
Hydro (RoR and Reservoir) Production	24
Thermal Production (Coal-Biomass-Gas-Ligite)	26

3.2.2.1 General Feature Category

The General feature category includes 2 features:

- Hour: This attribute indicates the time of day, with values ranging from 1 to 24, corresponding to the hours in a day. It is based on the Central European Time (CET) zone.

- Weekday: This characteristic depicts the days of the week, assigning numerical values that start with 1 for Sunday and culminate with 7 for Saturday.

The figure 3.7 illustrates the variation in the MGP price, across different hours of the day and each weekday.

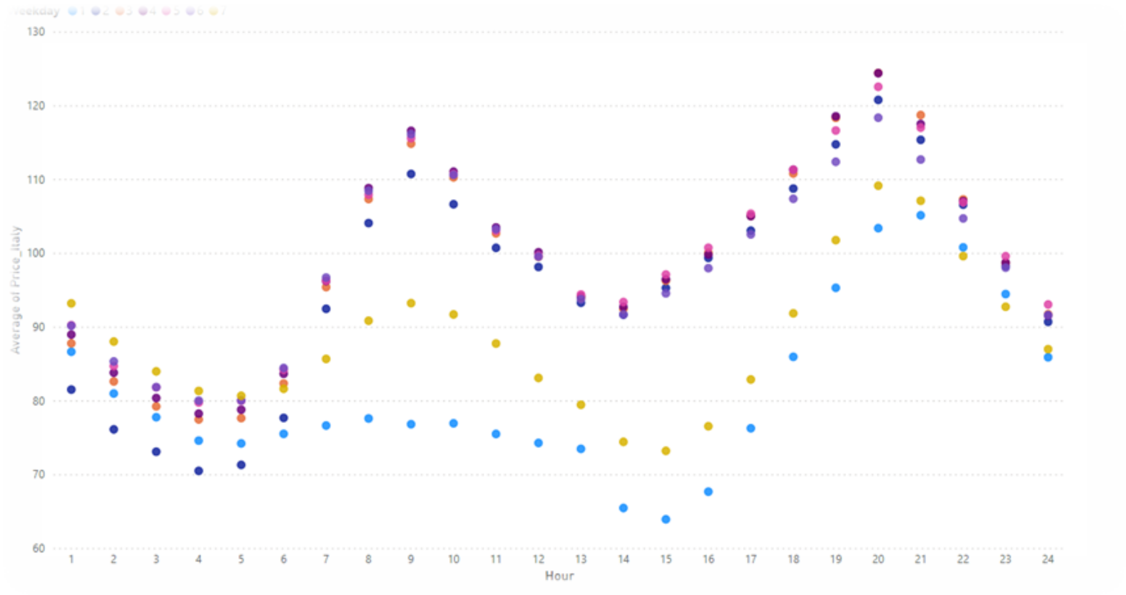

Figure 3.7: Average MGP spot price (EUR/MWh) per hour and per weekday

3.2.2.2 Commodities Feature Category

The commodities feature category includes 2 features:

TTF: The Title Transfer Facility (TTF) price is a crucial indicator within the European energy sector, representing the cost implications for gas-fired power plants throughout the continent. Situated in the Netherlands, the TTF is not a physical location but a virtual trading point that has risen to prominence due to the sheer volume of natural gas that is exchanged there. Its significance is underscored by its role as a benchmark hub, setting the standard for gas prices across Europe - a benchmark that even global markets observe attentively. The scale of trade conducted at TTF is staggering, with its volume being twice that of all other continental European gas trading platforms combined. This hub is a hive of activity, where every day, an assembly of more than 100 gas traders and financial institutions converge to buy and sell large volumes of gas. These transactions are pivotal for the energy markets, as they influence pricing and availability of natural gas for various sectors. Trading at TTF is executed in euros per megawatt-hour (MWh), offering a transparent and consistent pricing mechanism for participants. The data regarding these trades are meticulously recorded and made available on the website of the Intercontinental Exchange (ICE), providing insights into the market dynamics and aiding

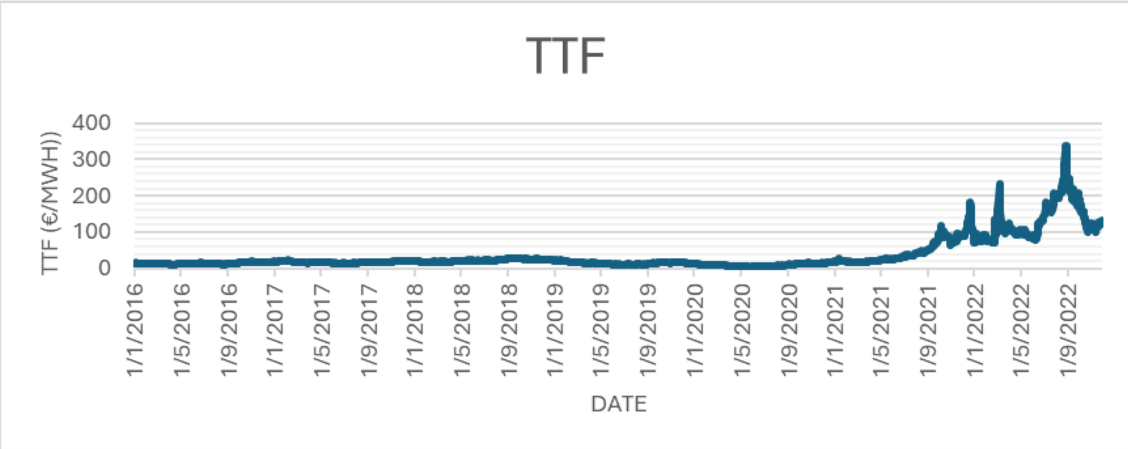

Figure 3.8: Evolution of TTF Price

stakeholders in making informed decisions. In essence, the TTF price is more than just a number - it is a barometer of the energy landscape in Europe, reflecting the interplay of supply and demand, geopolitical influences, and the economic health of the energy sector. In Figure 3.8 the evolution of TTF Price is shown. As Europe transitions towards more sustainable energy sources, the role of TTF and its pricing will continue to evolve, potentially serving as a bellwether for the broader shift in global energy consumption patterns[71].

EUA: EU Allowances, commonly known as EUAs, are essentially climate credits utilized within the framework of the European Union Emissions Trading Scheme (EU ETS)[72]. These allowances are allocated by the EU Member States and recorded in the respective Member State Registry accounts. As per the regulatory mandate, by the 30th of April each year, facility operators governed by the EU ETS are required to submit an EU Allowance for every metric ton of CO_2 they released in the preceding year. Defined under Article 3(a) of the EU ETS Directive, an emission allowance is the authorization to release one metric ton of carbon dioxide equivalent within a certain timeframe. These allowances are specifically designed to comply with the directive's stipulations and are transferable under its terms. EUAs play a pivotal role in influencing the operational costs of thermal power plants across the region, as they are traded in euros for each ton of CO_2 emitted. For those interested in the trading dynamics and current values of EUAs, comprehensive data is readily accessible on the ICE Exchange website. To enhance this explanation further:

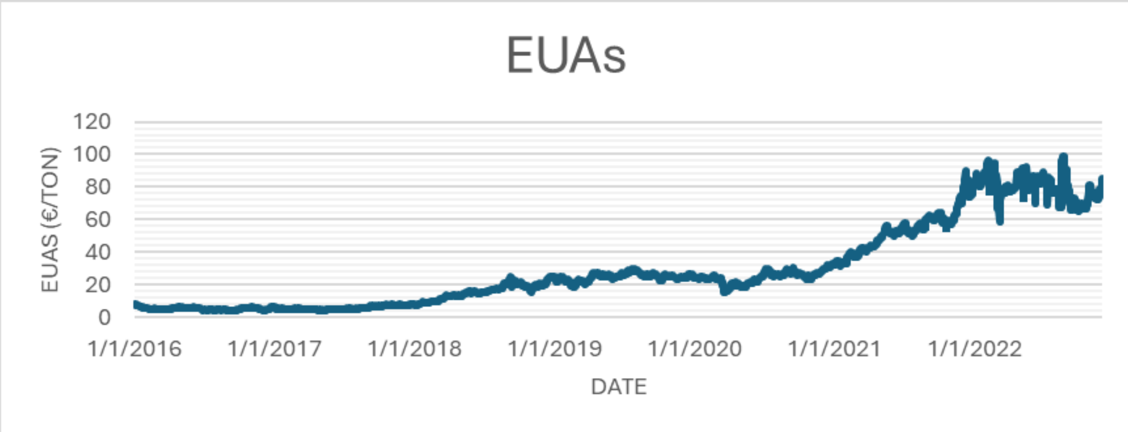

Figure 3.9: Evolution of EUA Price

EU Allowances (EUA) serve as the cornerstone of the European Union's strategy to combat climate change by regulating carbon emissions. Issued by individual EU Member States, these allowances are more than mere credits; they represent a commitment to a sustainable future. Each allowance permits the emission of one tonne of CO2 or its equivalent, ensuring that industrial players are accountable for their carbon footprint. The trading of EUAs is not just a financial transaction; it's a reflection of the environmental costs associated with power generation. As these allowances change hands, they signal the market's valuation of carbon emissions, influencing investment decisions and encouraging the adoption of cleaner technologies. The transparency of this system is maintained through the availability of transaction data on the ICE Exchange, providing a window into the market's pulse and the progress towards a greener economy[73]. In Figure 3.9 the evolution of the EUA Price is shown.

3.2.2.3 Prices Feature Category

The Italian energy exchange is deeply interconnected with other European energy markets, and the coupling of these exchanges plays a significant role in shaping Italy's energy landscape. Market coupling, a process that aims to harmonize electricity prices across different regions by linking control areas and market areas, has been instrumental in reducing price disparities and optimizing the use of cross-border electricity capacities. For example, the coupling of Italy's market with that

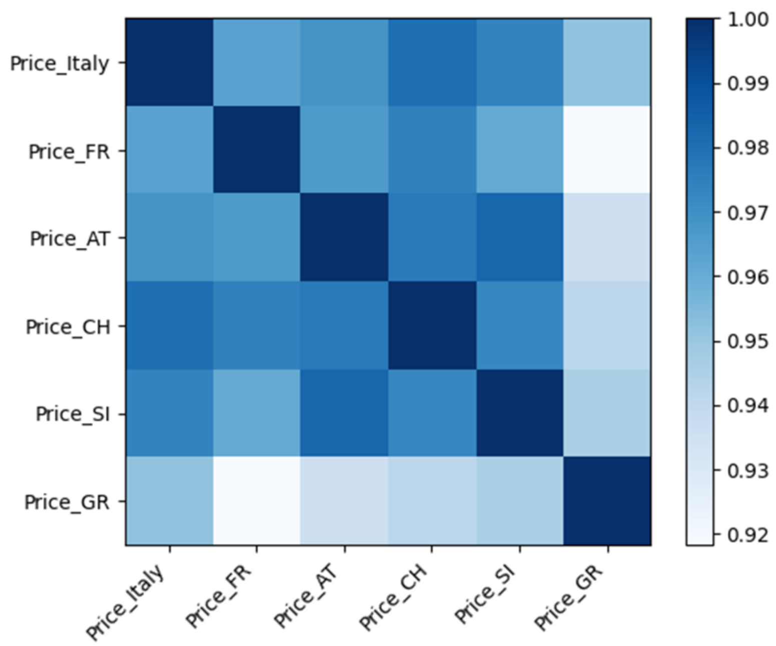

Figure 3.10: Correlation Matrix between the Italian and the other exchanges

of Slovenia has been shown to foster greater integration of power markets, which is a step towards the formation of a unified European exchange[74]. This integration is facilitated by mechanisms like the Price Coupling of Regions (PCR) and Flow-Based Market Coupling (FBMC), which help in maximizing the use of cross-border interconnection capacity. Such coupling arrangements ensure that electricity always takes the shortest route from producer to consumer, transcending market boundaries and leading to a more efficient electricity supply. Furthermore, the coupling of Italy with markets like Greece or France allows for a more competitive environment where Italian consumers can benefit from access to lower-cost generation options available in these countries. This not only impacts the pricing but also the security of supply and sustainability of the industry within Italy. This feature category includes 5 features that corresponds to the electricity energy prices of France (FR), Austria (AT), Switzerland (CH), Slovenia (SI) and Greece (GR). The correlation matrix (Figure 3.10) presented indicates a robust correlation between the energy exchanges of Switzerland and Italy, while the correlation between the energy exchanges of France and Italy is comparatively weaker[75].

3.2.2.4 Load and Temperature Feature Category

The load, or the total electricity demand, is a pivotal factor in the energy price forecasting of Italy. Accurate load forecasting is essential for energy providers to manage their resources efficiently and to ensure that they can meet the demand without overproducing, which can lead to increased costs. In Italy, the load influences the National Single Price of electricity (PUN), which is the average price at which electricity is sold and purchased on the Italian Power Exchange[76]. The load is affected by various factors, including weather conditions, economic activities, and consumer behavior. By predicting the load, energy providers can anticipate changes in energy prices and adjust their strategies accordingly. This is particularly important in Italy, where the energy market is liberalized, and prices are subject to fluctuations based on supply and demand dynamics[77]. Moreover, load forecasting helps in the management of financial risks associated with unpredictable electricity demand. It enables energy companies to develop more accurate budgeting, investment planning, and cash-flow analysis. For consumers, it translates into more stable and potentially lower energy prices, as it allows energy providers to optimize their generation capacity and reduce the need for expensive peak-time energy production.

In the routine operations of power markets, it is a standard procedure for Transmission System Operators (TSOs) to offer their forecast of the next day's hourly System Load (which is measured in MW). Market participants utilize this forecast as a critical input for their predictions concerning the upcoming day's electricity price levels. In this feature category, the hourly load of every zone in Italy is included as well as the load of France, Austria, Switzerland, Slovenia and Greece (Table 3.3). Load and electricity price have a positive correlation as shows Figure 3.11.

In the Italian energy system, the correlation between electricity demand and temperature exhibits a distinct V-shaped relationship. Studies have shown that up to a certain threshold, which is approximately 24.4°C, the electricity demand remains relatively stable. However, as temperatures rise beyond this point, there is a sharp increase in electricity usage, primarily due to the heightened demand for cooling[78] . This relationship is particularly pronounced during the summer months when hot

Table 3.3: Load Feature Category

Load_NORD	Load in the Italian zone NORD
Load_CNOR	Load in the Italian zone CNOR
Load_CSUD	Load in the Italian zone CSUD
Load_SUD	Load in the Italian zone SUD
Load_SARD	Load in the Italian zone SARD
Load_SICI	Load in the Italian zone SICI
Load_FR	Load in France
Load_AT	Load in Austria
Load_CH	Load in Switzerland
Load_SI	Load in Slovenia
Load_GR	Load in Greece

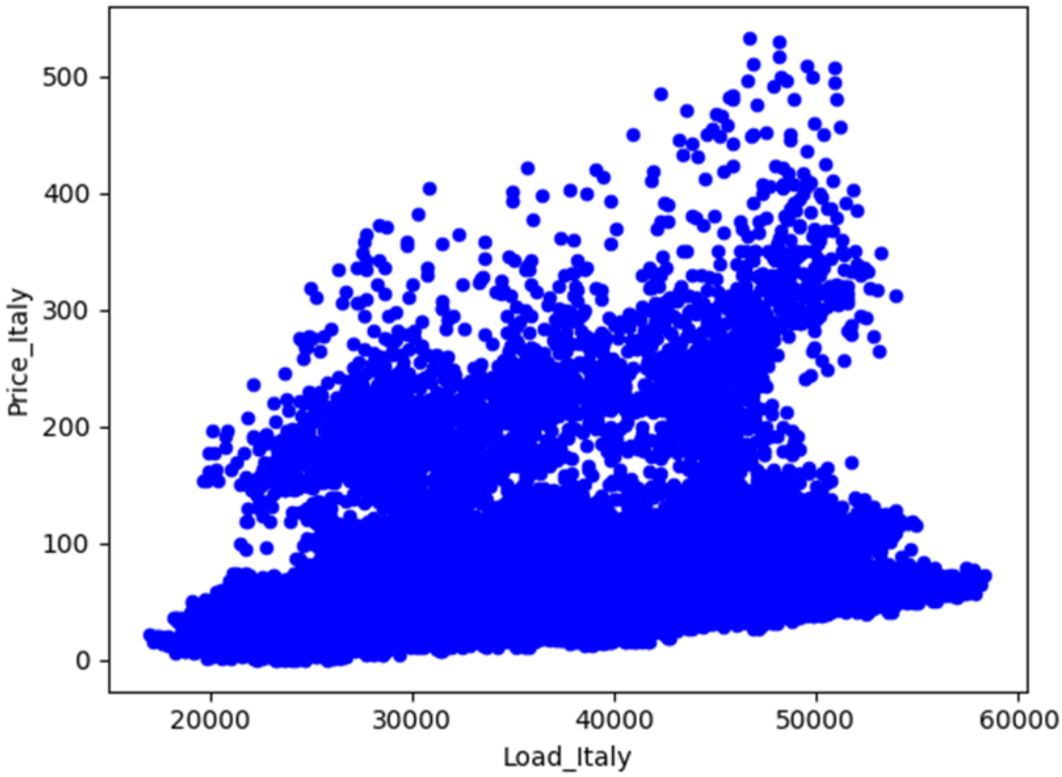

Figure 3.11: Correlation between MGP price and Load

Table 3.4: Temperature Feature Category

Temperature_NORD	Temperature in the Italian zone NORD
Temperature_CNOR	Temperature in the Italian zone CNOR
Temperature_CSUD	Temperature in the Italian zone CSUD
Temperature_SUD	Temperature in the Italian zone SUD
Temperature_SARD	Temperature in the Italian zone SARD
Temperature_SICI	Temperature in the Italian zone SICI
Temperature_FR	Temperature in France
Temperature_AT	Temperature in Austria
Temperature_CH	Temperature in Switzerland
Temperature_SI	Temperature in Slovenia
Temperature_GR	Temperature in Greece
Temperature_ME	Temperature in Montenegro

weather can lead to significant spikes in electricity consumption. On exceptionally hot days, temperature can account for up to 12% of hourly electricity use, underscoring the impact of climate on energy demand[78]. In this feature category, the hourly Temperature of every zone in Italy is included as well as the Temperature of France, Austria, Switzerland, Slovenia, Greece and Montenegro (Table 3.4).

3.2.2.5 RES Production Feature Category

Italy has undergone significant changes in its energy system over the past decade. The country's energy mix now includes more natural gas and renewables (RES), with reduced reliance on coal and oil. RES production affects energy prices through supply dynamics. High RES production can lower wholesale electricity prices, while low production may drive prices up. Accurate forecasting of RES generation is crucial for understanding pricing fluctuations[79]. In our study, RES production includes Wind and Solar Production from Italy and from neighboring countries (Table 3.5).

3.2.2.6 Internal and Cross - Border Exchanges Feature Category

The Scheduled Internal Exchange refers to the planned trade of electricity between Italian market areas (Table 3.6). Terna, the national transmission system operator,

Table 3.5: RES production Feature Category

Wind_NORD & Solar_NORD	Wind and Solar Production in NORD
Wind_CNOR & Solar_CNOR	Wind and Solar Production in CNOR
Wind_CSUD & Solar_CSUD	Wind and Solar Production in CSUD
Wind_SUD & Solar_SUD	Wind and Solar Production in SUD
Wind_SARD & Solar_SARD	Wind and Solar Production in SARD
Wind_SICI & Solar_SICI	Wind and Solar Production in SICI
Wind_FR & Solar_FR	Wind and Solar Production in France
Wind_AT & Solar_AT	Wind and Solar Production in Austria
Solar_CH	Solar Production in Switzerland
Wind_SI & Solar_SI	Wind and Solar Production in Slovenia
Wind_GR & Solar_GR	Wind and Solar Production in Greece
Wind_ME	Wind Production in Montenegro

Figure 3.12: A snapshot of the energy flow in the Italian transmission system

publishes the hourly trade program between these zones, shown in Figure 3.12. The direction of exchange flows is depicted through arrows, reflecting the movement of electricity from one area to another[80]. This exchange is crucial for balancing supply and demand, optimizing grid operation, and ensuring efficient utilization of resources. Factors such as transmission constraints, generation availability, and demand fluctuations influence the physical flow.

Electricity interconnections serve as tangible conduits enabling the transference

Table 3.6: Scheduled Internal flows in Italy

Scheduled_Internal_Exchange_NORD_CNOR
Scheduled_Internal_Exchange_SCUD_CNOR
Scheduled_Internal_Exchange_CSUD_SUD
Scheduled_Internal_Exchange_CSUD_SARD
Scheduled_Internal_Exchange_SUD_CALA
Scheduled_Internal_Exchange_SICI_CALA

of electrical power beyond national frontiers. The trading of power across borders facilitates the movement of electricity from nations where it is less costly to those where it is more expensive. Consequently, the capacity available within these interconnections is a critical factor in determining the prices of electricity at the wholesale level.

Italy has a lot of neighboring countries (France, Austria, Switzerland, Slovenia, Greece, Montenegro), hence it has a lot of electricity interconnections, as depicted in the following Figure. For the purposes of this thesis, it is crucial to take into account the available capacity of all interconnections in the Italian region in order to identify the 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 has to do only with the Italian borders. That means it includes features that refer to the hourly flow of the interconnections that connect Italy with the abovementioned countries. In Figure 3.13 a snapshot of the energy flow between Italy and neighbouring countries is shown.

The second cross-border feature category is related to the interconnections of the wider area, excluding the Italian ones. It includes information on the interconnections of the following borders:

- France - Belgium
- France - Germany
- France - Switzerland
- France - Spain

- France - United Kingdom
- Austria - Switzerland
- Austria - Germany
- Austria - Slovenia
- Austria - Czech Republic
- Austria - Hungary
- Switzerland - Germany
- Slovenia - Croatia
- Greece - Bulgaria
- Greece - North Macedonia

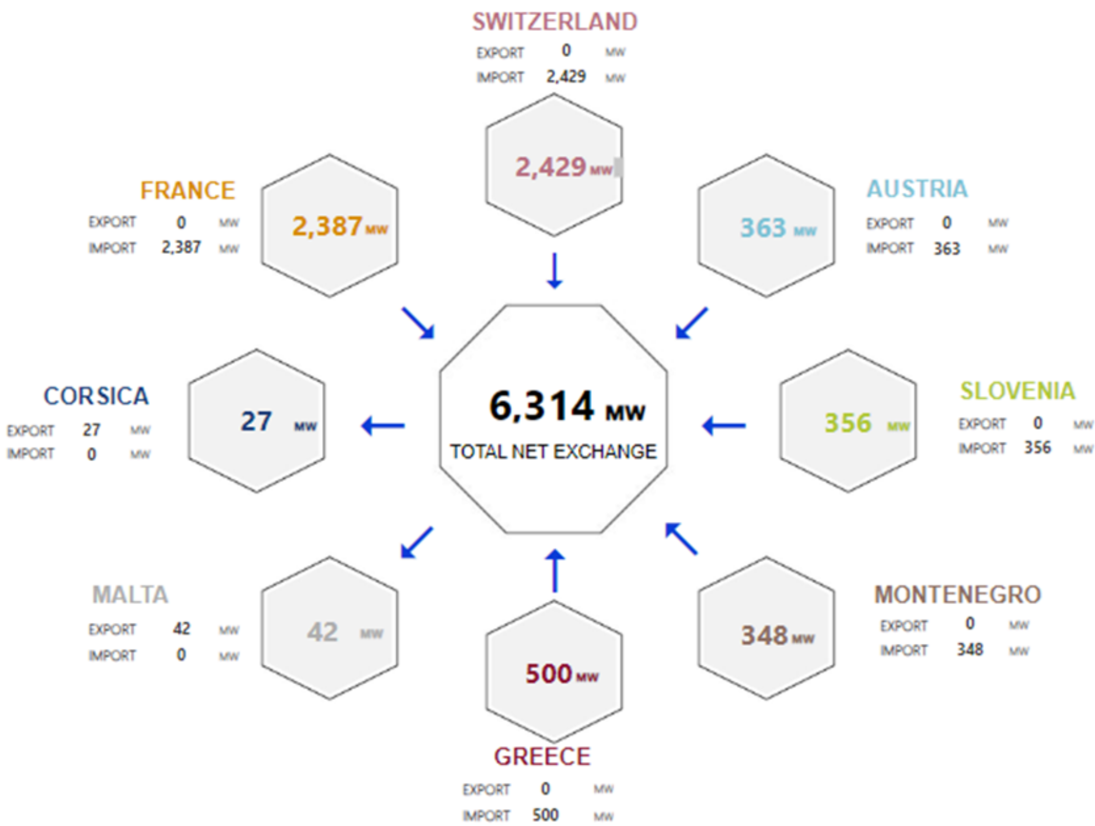

Figure 3.13: A snapshot of the energy flow in the Italian transmission system with the other countries

- Greece - Albania
- Greece - Turkey
- Montenegro - Bosnia & Herzegovina
- Montenegro - Serbia
- Montenegro - Albania
- Montenegro - XK

3.2.2.7 RES Installed Capacity Feature Category

Wind and solar energy have a significant impact on the energy prices in the Italian energy exchange, primarily through the Merit Order Effect as explained in Chapter 1. This effect occurs as wind and solar power, which have negligible marginal costs, are given priority in the energy dispatch order over more expensive fossil fuel-based power plants. As a result, the increased supply of renewable energy can lead to lower wholesale electricity prices. A study conducted on the Italian electricity market from 2015 to 2019 employed a multivariate regression model to assess this impact. The findings indicated that higher wind and solar generation correlates with a decrease in the daily zonal electricity price, confirming the Merit Order Effect in the Italian context.

The implications of these findings are significant for both energy policy and market planning. For instance, by promoting the development of wind and solar power, the Italian electricity market operator could potentially reduce the National Single Price, which would be beneficial for consumers. Additionally, the results of the study provide valuable insights for decision-makers and market planners, enabling them to predict future market structures and make informed decisions about the integration of renewable energy sources. This knowledge is particularly advantageous as Italy moves towards a more sustainable and less carbon-intensive energy system[81] .

The figure 3.14 shows the growth of Italy's renewable energy sector, with a clear upward trend in the installed capacity of both wind and solar power from January

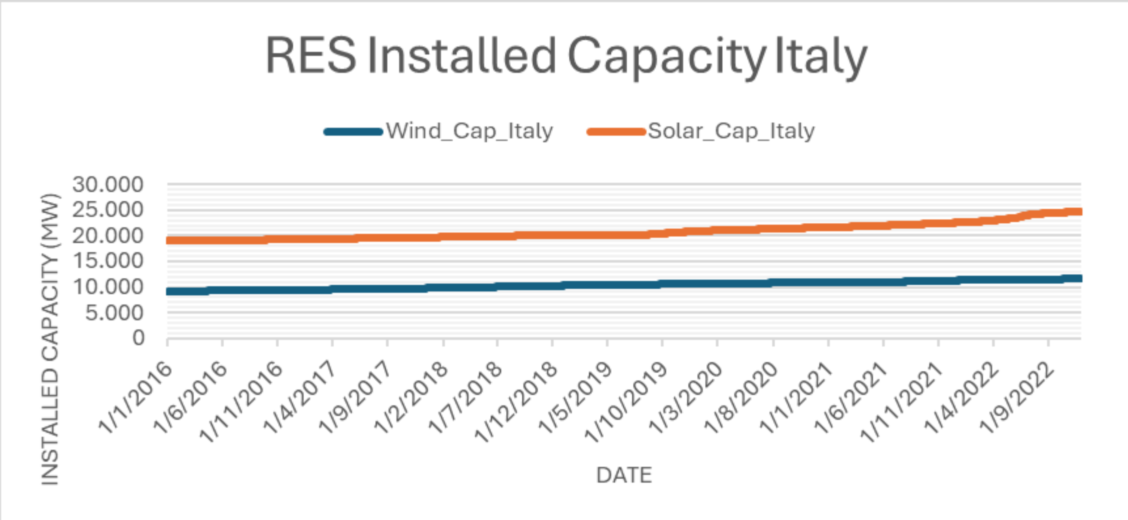

Figure 3.14: Wind and Solar Installed Capacity in Italy

2016 to November 2022. The line graph indicates that solar capacity has seen a significant increase, with Italy adding 1.6 GW of new solar PV capacity in 2022 alone, contributing to a total generation of 30.6 Terawatt hours (TWh). In addition, the installed wind power capacity stood at 11.5 GW and the wind farms produced 23.4 TWh of electricity. This consistent growth in renewable energy installations underscores Italy's commitment to reducing its reliance on fossil fuels and moving towards a more sustainable and environmentally friendly energy infrastructure. In this thesis, we have also included the installed capacity of solar and wind of the neighboring countries.

3.2.2.8 Hydro Production Feature Category

The impact of hydroelectric production on energy prices in Italy, considering both run-of-river (ROR) and reservoir-based hydropower is huge. The recent drought in Italy has significantly affected hydroelectric generation, leading to implications for energy markets and agriculture. More specifically, hydroelectric power in Italy has experienced a sharp decline in 2022 due to a severe drought[82]. Hydropower facilities, primarily located in the mountainous regions of northern Italy, typically provide nearly one fifth of the country's energy demands. However, the lack of rainfall has caused problems, affecting water availability even at higher altitudes. From January to May 2022, hydro production fell by approximately 40% compared

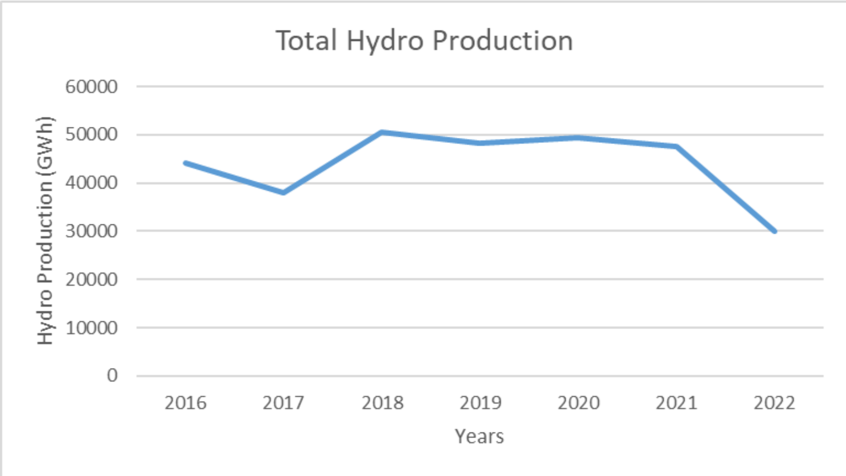

Figure 3.15: Total Hydro Production in Italy 2016-2022

to the same period in 2021[83] (Figure 3.15). Reduced hydroelectric production affects the overall energy supply mix, potentially impacting electricity prices. As hydropower capacity decreases, other energy sources (such as fossil fuels or imports) may need to compensate, potentially leading to price fluctuations. The situation underscores the importance of diversifying energy sources and investing in renewable alternatives to mitigate such risks.

In this feature category we include the production level of the Hydro RoR and Hydro Reservoirs of Italy and neighboring counties such as France, Austria, Switzerland, Slovenia and Greece.

3.2.2.9 Thermal Units Production Feature Category

Italy's energy system has changed notably since 2010, with a shift towards more natural gas and renewable energies and reduced reliance on coal and oil. The country's energy intensity (measured by the ratio of total final consumption to gross domestic product) declined by 15% between 2005 and 2021, reflecting a shift in the economic structure from industrial to the service sector combined with energy efficiency improvements[79]. Italy has 2 main types of thermal power generation. Natural gas which is the dominant fuel in Italy's electricity sector, accounting for 50% of generation. It is also a significant component of the overall energy supply. Coal, which although the need has diminished, it remains the second most frequently

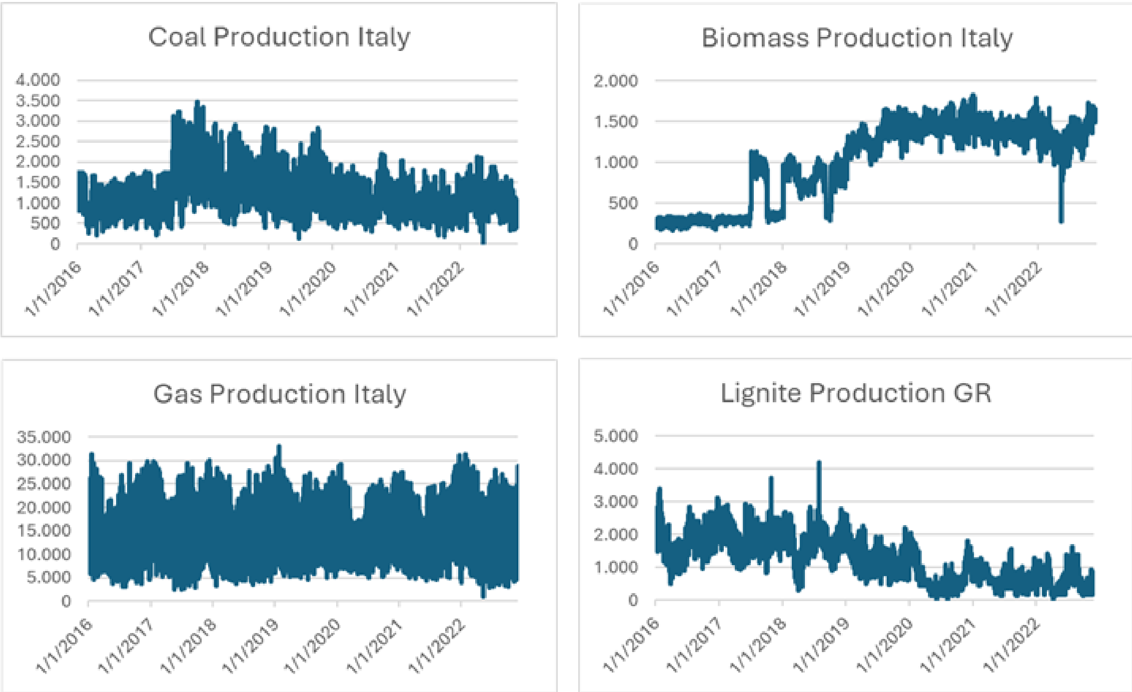

Figure 3.16: Electricity production of critical technologies in Italy and other countries

used fuel in Italy's electricity generation mix. Italy still operates 13 coal-fired power plants[84]. As can be seen in Figure 3.16, critical technologies of the wider area differ in both level of production and volatility. At this point, we would like to add that apart from the thermal production in Italy, we have included also different technologies from the neighboring countries such as lignite and biomass.

3.3 Data Preprocessing

Data preprocessing is a crucial step in the data analysis framework. It involves a series of operations to clean, organize, and transform raw data into a suitable format for further analysis, modeling, and machine learning. Real-world data is often messy, containing inconsistencies, inaccuracies, and missing values. Data preprocessing[85] rectifies these issues, improving data quality and making it more reliable for downstream tasks. Some key aspects of data preprocessing are:

Data Cleaning: This step involves identifying and correcting errors, inconsistencies, and missing values in the dataset. Techniques like imputation and outlier removal are commonly used.

Data Transformation: Data transformation includes scaling, normalization, and encoding categorical variables. Scaling ensures that numerical features have similar ranges, while normalization brings them to a common scale (e.g., between 0 and 1). One-hot encoding converts categorical variables into binary vectors.

Feature Extraction and Selection: Feature extraction aims to create new features from existing ones, while feature selection identifies relevant features for modeling. Techniques like principal component analysis (PCA) and recursive feature elimination (RFE) are used.

Our experiments have shown that 2.7% of our dataset rows have missing values. So, we had 2 options: Removing samples with missing values is a basic strategy that is sometimes used, but it comes with a cost of losing probable valuable data and the associated information or patterns. A better strategy is to impute the missing values. We have decided to impute those values using a K-Nearest Neighbors approach. After the data imputation, none of the values are missing.

Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to rescale the values of numeric columns in the dataset without distorting differences in the ranges of values or losing information. From our experiments, we have concluded that the Normalizer offers marginally better accuracy than the StandardScaler or the MinMaxScaler.

Chapter 4

Experimental Runs and Results

In this chapter, we delve into evaluating the performance of various machine learning approaches - ranging from traditional methods to ensemble techniques and deep learning - for forecasting electricity prices in the context of MGP electricity price forecasting. Additionally, we conduct a sensitivity analysis to understand how individual features impact the final prediction outcomes.

4.1 Implementation Issues

In our study, we extensively utilize scikit-learn for developing various models and implementing experiments. This Python library is highly valuable for machine learning, offering efficient tools for tasks such as classification, regression, clustering, and dimensionality reduction.

In Tables 4.1 & 4.2, you'll find the corresponding scikit-learn functions associated with various machine learning algorithms:

As for the ANN algorithms in our case, it was based on the TensorFlow 2.0 Framework. TensorFlow is an end to end powerful open-source machine learning framework developed by Google. It serves as a foundational layer for differentiable programming, integrating four essential capabilities:

Efficient Tensor Operations: TensorFlow efficiently executes low-level tensor operations on various hardware devices, including CPUs, GPUs, and TPUs.

Gradient Computation: It computes gradients for arbitrary differentiable expres-

Table 4.1: Scikit-learn functions for traditional ML algorithms

Algorithm	Function
LinearRegression	sklearn.linear_model.LinearRegression
Ridge	sklearn.linear_model.Ridge
Lasso	sklearn.linear_model.Lasso
Elastic net	sklearn.linear_model.ElasticNet
Lars	sklearn.linear_model.Lars
KNeighbors	sklearn.neighbors.KNeighborsRegressor
Decision Tree	sklearn.tree.DecisionTreeRegresso
HuberRegressor	sklearn.linear_model.HuberRegressor
SVR	sklearn.svm.SVR

Table 4.2: Scikit-learn functions for ensemble ML algorithms

Algorithm	Function
AdaBoostRegressor	sklearn.ensemble.AdaBoostRegressor
ExtraTreesRegressor	sklearn.ensemble.ExtraTreesRegressor
GradientBoostingRegressor	sklearn.ensemble.GradientBoostingRegressor
RandomForestRegressor	sklearn.ensemble.RandomForestRegressor

sions, enabling automatic differentiation during training.

Scalability Across Devices: TensorFlow scales computations to multiple devices, allowing distributed training and parallel processing.

Exporting to External Runtimes: Programs built with TensorFlow can be exported to external runtimes, such as servers, browsers, mobile devices, and embedded systems.

Keras, as the high-level API of TensorFlow 2, serves as an accessible and highly productive interface for solving machine learning (ML) problems, particularly in the realm of modern deep learning. It offers essential abstractions and building blocks, enabling developers to create and deploy ML solutions with rapid iteration velocity. The fundamental components in Keras are layers and models. Initially, Keras was developed as part of the research effort for the project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System)[86].

4.2 Univariate Approach

Univariate time series forecasting is a statistical method used to predict future values of a single variable based on its past values. This technique is particularly useful when historical data is available and the future values are assumed to be a function of time alone. The process involves identifying patterns in the historical data, such as trends and seasonality, and using them to forecast future data points.

One of the key concepts in univariate time series forecasting is the identification of a trend, which is the long-term movement or direction of the data. Trends can be upward, downward, or stationary, and they reflect the underlying pattern in the data over time. Seasonality is another important factor, referring to regular and predictable patterns that recur over specific periods, such as daily, monthly, or quarterly fluctuations. To ensure accurate predictions, it's essential to account for both trend and seasonality in the forecasting model[87][88].

Several techniques are employed in univariate time series analysis, with autoregression (AR) and moving average (MA) being two of the most common. AR models use the dependency between an observation and a number of lagged observations, while MA models focus on the relationship between an observation and a residual error from a moving average model applied to lagged observations. For more complex datasets, where deep learning models can automatically learn and extract features from raw data, techniques like neural networks, LSTMs, and CNNs are used to handle both trend and seasonal components without the need for pre-processing[89]

Choosing univariate time series forecasting over multivariate time series forecasting for energy price forecasting can be beneficial for several reasons:

- **Simplicity and Accessibility**: Univariate models are simpler as they focus on a single time-dependent variable. This simplicity makes the models more accessible, easier to understand, and quicker to implement, which can be particularly advantageous if you have limited resources or need to make fast decisions.

- **Data Availability**: In some cases, relevant data for multivariate analysis may not be available or may be incomplete. Univariate forecasting does not require multiple input variables, making it a practical choice when data is scarce or when the influence of other variables on energy prices is minimal or uncertain.

- **Modeling Efficiency**: Univariate models can be more computationally efficient because they involve fewer calculations. This efficiency can be crucial when dealing with large datasets or when computational resources are a constraint.

- **Focused Analysis**: If the goal is to understand the behavior of energy prices independently of other factors, univariate analysis allows for a focused study of the price variable itself, which can provide insights into its intrinsic patterns and cycles.

- **Good Performance in Certain Conditions**: Research has shown that univariate models can perform comparably to multivariate models under certain conditions. For example, a study found that univariate models performed better for the first half of the day, while multivariate models were better in the second half of the day in short-term electricity price forecasting. This indicates that univariate models can be quite effective, especially when combined with other forecasting methods[90].

It's important to note that the choice between univariate and multivariate forecasting should be based on the specific context of the energy market and the availability of data. Univariate models can offer a robust starting point for energy price forecasting, especially when simplicity and speed are required[91]. However, for a more comprehensive analysis that considers multiple influencing factors, multivariate models may be more appropriate. The key is to evaluate the trade-offs and choose the method that best suits your forecasting needs and constraints.

For the purposes of our study, we have chosen one of the most advanced algorithms used for univariate forecasting, Facebook's Prophet. Facebook's Prophet is an open-source forecasting tool designed to handle a wide array of time series data that is notable for its ease of use and ability to produce high-quality forecasts.

Developed by Facebook's Core Data Science team, Prophet is based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, as well as holiday effects. This makes it particularly effective for data with strong seasonal effects and several seasons of historical data. Prophet is robust to missing data, shifts in the trend, and typically handles outliers well, which is crucial for maintaining the accuracy of forecasts despite the common data irregularities[92].

One of the key advantages of Prophet is its accessibility to individuals who are not experts in time series forecasting. The tool is implemented in both R and Python, offering a flexible choice for users comfortable with either language. The forecasting process with Prophet is fully automatic, but it also allows for manual tuning, giving data scientists and analysts the ability to refine forecasts based on their domain knowledge. This combination of automation and customizability helps in creating forecasts that are both fast and accurate, catering to a wide range of business needs[93].

Prophet's design philosophy emphasizes simplicity and practicality, aiming to provide a powerful forecasting tool that can be used effectively across various industries. It is particularly well-suited for business applications where forecasting is critical, such as in stock level prediction, resource allocation, and market trend analysis. The tool's ability to incorporate holiday effects is especially useful for businesses that experience significant fluctuations during specific periods. As an open-source project, Prophet continues to evolve, with updates and improvements being made regularly by the community and the original developers[89].

As we observe from the two following graphs, when the algorithm tries to predict further into the future, the MAPE increases significantly. Also, the MAPE decreases as the training dataset increases. We observe that the univariate approach does not perform satisfactorily as energy prices are affected by many factors, for example, the cost of natural gas and greenhouse gases, as we have described in previous chapters.

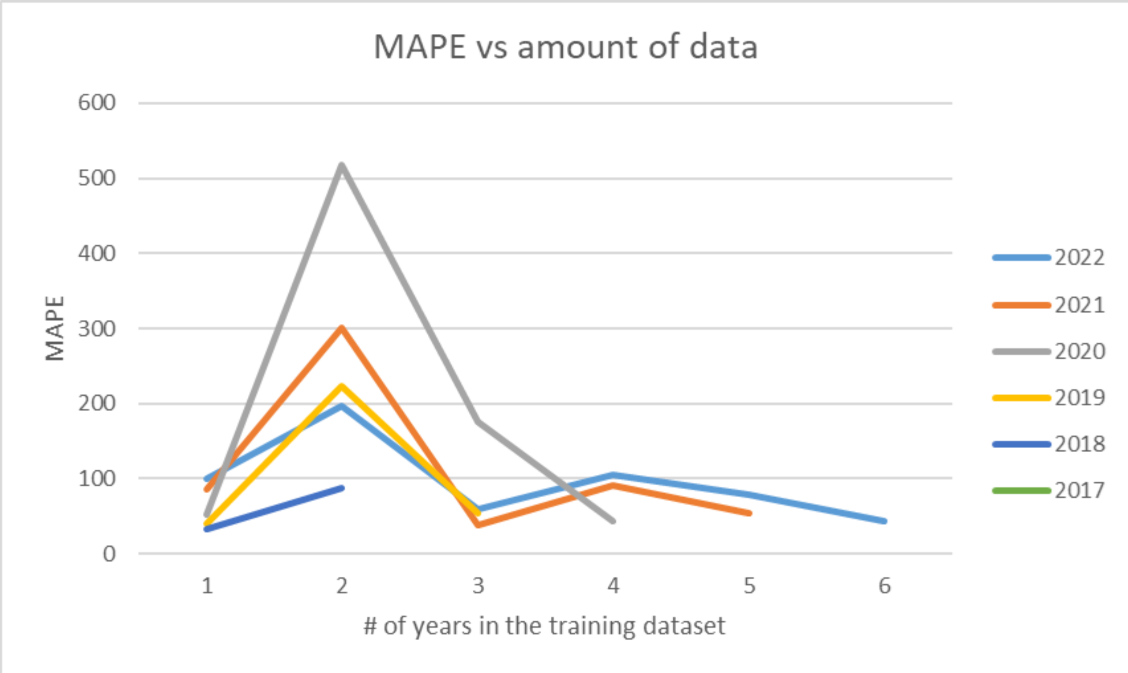

Figure 4.1: MAPE of the Facebook's Prophet when more data are added to the training dataset

Train	Predict	2017	2018	2019	2020	2021	2022
2016	2017-2022	21	33	41	53	86	99
2016-2017	2018-2022		88	223	518	301	197
2016-2018	2019-2022			55	175	38	59
2016-2019	2020-2022				44	90	106
2016-2020	2021-2022					54	79
2016-2021	2022						44

Figure 4.2: Prophet's Results as a table

4.3 Validation Process

In machine learning, we face the challenge of ensuring that our models generalize well to unseen data. Simply fitting a model on the training data is insufficient; we need to evaluate its performance on unseen data. This is where cross-validation comes into play.

4.3.1 Cross Validation

Cross-validation (CV) is a widely used technique for tuning hyperparameters and obtaining reliable estimates of model performance. Two common types of cross-validation are k-fold cross-validation and hold-out cross-validation.

Given the variations in terminology across different literature, we provide a clear definition of our cross-validation (CV) procedure. Here are the steps involved:

1. Data Splitting:
   - We begin by dividing the dataset into two subsets: the training set and the test set.
   - If any hyperparameters need tuning, we further split the training set into a training subset and a validation set.
2. Model Training and Parameter Tuning:
   - The model is trained on the training subset.
   - We explore different hyperparameters and select the ones that minimize the error on the validation set.
3. Final Model Training:
   - Using the chosen hyperparameters, we train the model on the full training set.
   - The model's performance is then evaluated on the test set, and the resulting error is recorded.

This systematic approach ensures that our model is well-tuned and provides reliable estimates of its performance on unseen data.

When working with time series data, it is advisable to avoid using traditional cross-validation methods (such as k-fold) for two main reasons:

- **Temporal Dependencies**: When handling time series data, it's crucial to split the dataset carefully to avoid data leakage. To faithfully replicate the "real-world forecasting scenario" where predictions are made for the future from the present standpoint (as described by Tashman in 2000)[94], it's essential to exclude any data about events that happen after the timeframe used to train the model. Therefore, instead of applying k-fold cross-validation, we employ hold-out cross-validation for time series data. In this method, a portion of the data is separated based on time and set aside to validate the model's performance. For instance, as depicted in the previous figure, the data in the test set is chronologically subsequent to the training set data. In a similar manner, the validation set follows after the training subset in chronological order.

- **Arbitrary Choice of Test Set**: The selection of the test set in figure 4.3 may appear somewhat arbitrary, and this choice could lead to our test set error being an inadequate estimate of the error on an independent test set. To address this, we use a method called Nested Cross-Validation.

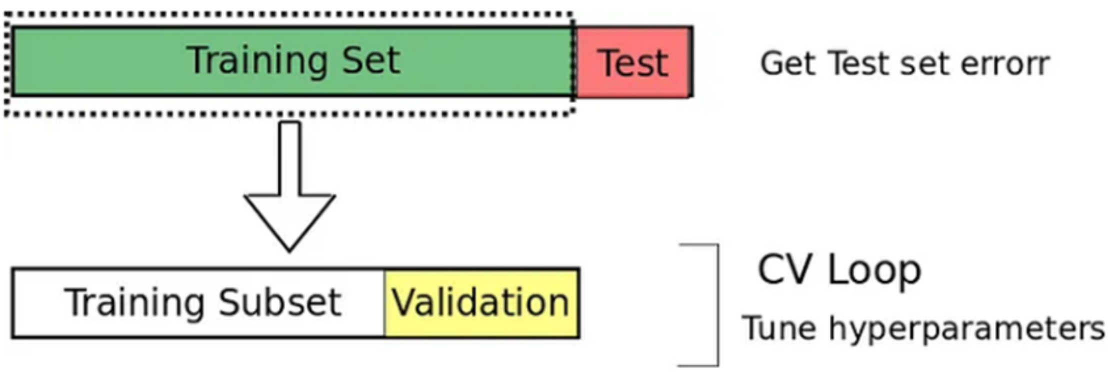

Figure 4.3: Example of hold-out cross-validation. The data is split into a training set and a testing set. Then the training set is split again to tune hyperparameters- one part is used to train the model (training subset) and the other part (validation set) is used to validate the model.

4.3.2 Nested Validation

Nested Cross-Validation (Nested CV) incorporates two levels of cross-validation: an outer loop for estimating model error and an inner loop for tuning hyperparameters. In the inner loop, the process remains the same as previously described: the training data is divided into a smaller training subset and a validation set. The model is then trained on this subset, and the best parameters are selected based on their performance on the validation set. The outer loop enhances this process by creating several distinct splits of the data into training and test sets. The model's error is calculated for each of these splits and then averaged to provide a more accurate and robust estimate of the model's error. This two-tiered approach ensures a thorough evaluation of the model's predictive capabilities. This is advantageous because a nested cross-validation procedure provides an almost unbiased estimate of the true error[95].

To deep dive into the Nested Cross-Validation, we have followed the Day Forward-

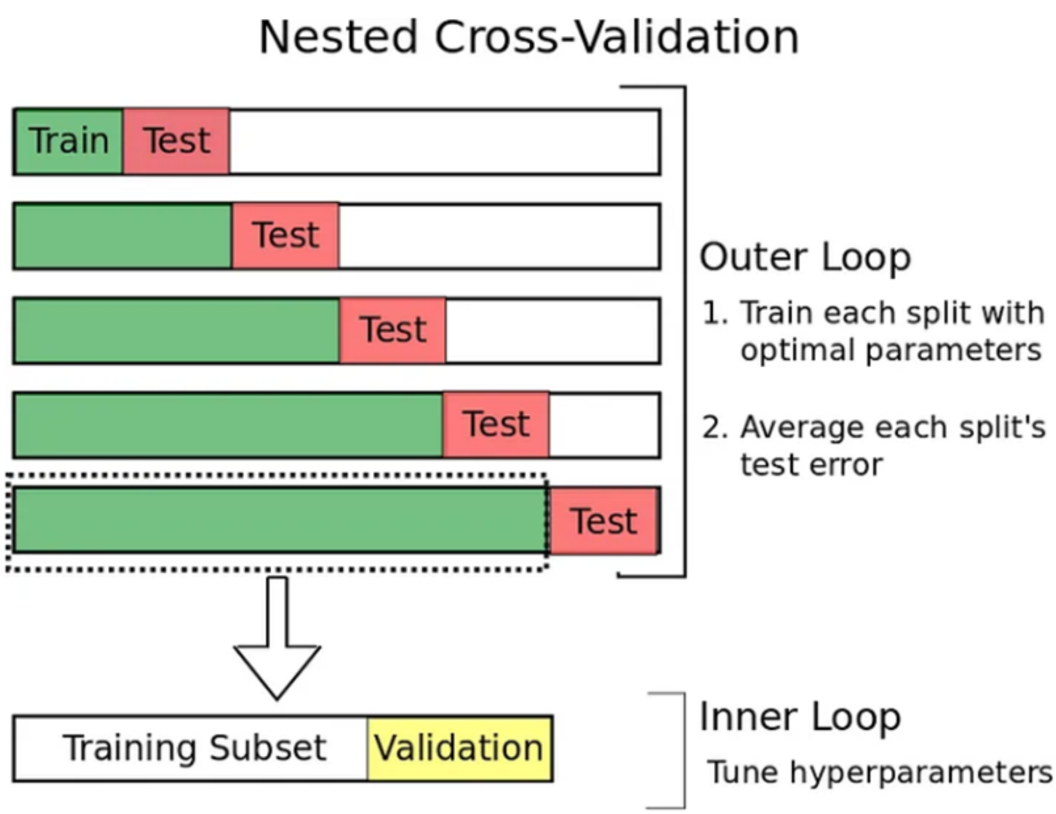

Figure 4.4: Nested CV Example

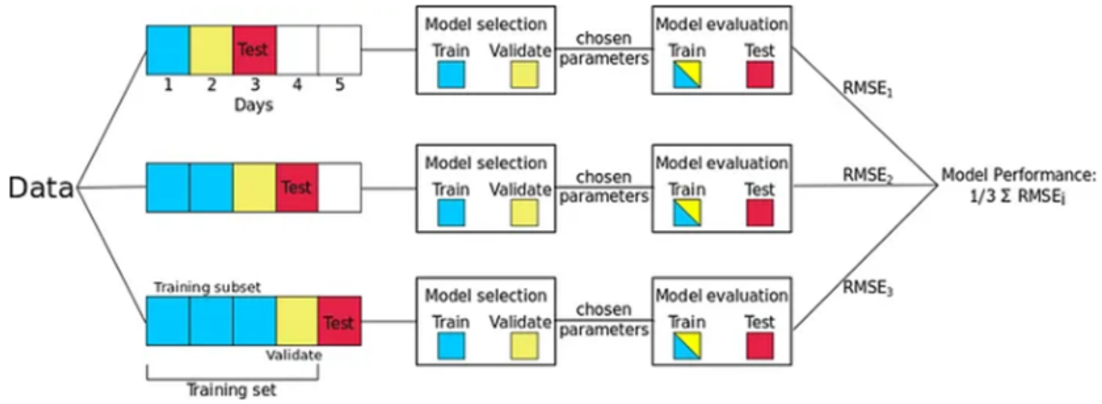

Figure 4.5: Day Forward-Chaining Nested Cross-Validation

Chaining approach. To obtain a more accurate prediction of the error in model forecasts, it's typical to divide the data into numerous training and testing segments and calculate the mean error across these divisions. We employ a strategy known as Day Forward-Chaining, which is derived from a technique called forward-chaining—also known in academic circles as rolling-origin evaluation (Tashman 2000)[94] and rolling-origin-recalibration evaluation (Bergmeir and Benítez 2012)[96]. This approach involves sequentially using each day's data as the test set, while incorporating all preceding data into the training set. For instance, with a dataset spanning five days, we would generate three distinct sets of training and testing data, as depicted in Figure 4.5. It's important to note that we have only three sets instead of five to guarantee the availability of at least one day's worth of training and validation data. By averaging the errors from each of these multiple training/testing splits, we can calculate a reliable estimate of the model's error rate.

4.4 Time Horizon with some baseline algorithms

Following the previous analysis about the validation process, we have decided to follow the Nested Cross - Validation process and especially the Day Forward-Chaining approach.

For the purposes of our study, we decided to use 3 variables. The first variable concerns the training period of our models (**training_var**) and was set at 180, 150, 120, 90, 60, 30, 25, 20, 15, 10, 7, and 1 days. The second variable concerns the sliding window (**sliding_var**) and it was decided to be 1 day so as not to lose data from our dataset. Furthermore, the third variable concerns how many days ahead we would predict with our algorithm (**forecast_var**) in order to see which gives us the smallest error. This variable could take values of 30, 20, 15, 10, 7, and 1. It is worth noting at this point that the **forecast_var** could not take larger values than the **training_var**. For this reason, when the training_var took values of 30, 25, 20, 15, 10, 7, and 1, the **forecast_var** could only take the value of 1.

Regarding the algorithms, we used some simple machine learning algorithms such as KNNRegressor, Linear Regressor, Ridge, Lasso, Multitask Lasso, ElasticNet, Huber Regressor, tree-based models like Decision Tree Regressor and Extra Trees, ensemble-based models like Random Forest Regressor, AdaBoost Regressor, Bagging Regressor, and support vector machine algorithms like Support Vector Regressor.

In addition, all the features of the dataset were used without removing any of them. Here are the results:

We have chosen the training_var = 60 days and we have plotted the mape of different algorithms. This line graph comparing 11 regression algorithms reveals that ensemble methods like Random Forest, AdaBoost, and Extra Trees Regressors show show consistent accuracy across all forecast periods. The K-NN and Ridge Regressors are top performers for short-term energy price forecasting, with low MAPE values. The Decision Tree Regressor maintains moderate accuracy throughout, making it a reliable choice for various forecasting windows. Overall, the graph (Figure 4.6) underscores the importance of algorithm selection based on the forecast horizon and desired balance between precision and computational efficiency.

Taking the K-NN algorithm, which gave us the best results as a reference, we will see which training period gives the smallest error. From the graph below (Figure 4.7), we observe that:

The graph shows that MAPE, when the training period is 180 days, starts from a high point and reaches a low point when the training period is 30 days. This indicates that the KNN algorithm improves in accuracy as the days of training are reduced. After the initial drop, the MAPE continues to decrease, albeit at a slower rate, suggesting that when the window of the training dataset is narrower, the

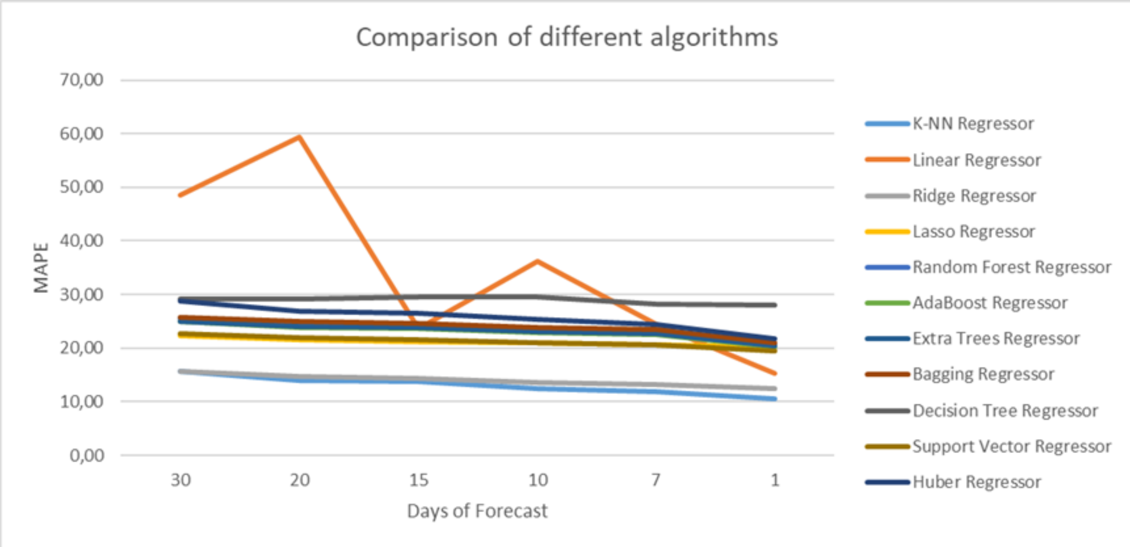

Figure 4.6: The efficiency of 11 different algorithms compared to the number of days forecasted

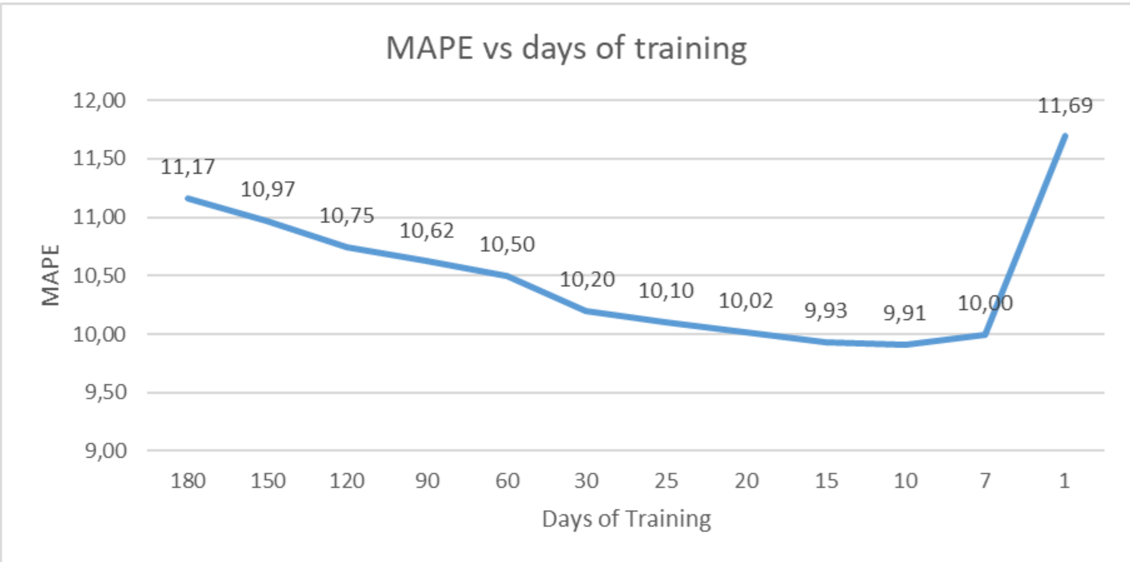

Figure 4.7: MAPE in comparison to the days of training of K-NN algorithm

algorithm operates better. The lowest MAPE value is observed when the training days are around 15, which suggests that this is the optimal training period for the KNN algorithm in this scenario. It implies that the model has learned well from the data without overfitting. Post training days 10, there is a noticeable increase in MAPE, peaking at training days 1.

Based on the above results, we conclude that the ideal days to train the model are 15 days, and we have the best results only when we predict the next day and not a longer period. This is happening because the optimal training period for machine learning models in energy price forecasting is influenced by several nuanced factors. Data quality and relevance play a crucial role; the most recent 15 days of data may be more indicative of current market conditions, rendering older data less pertinent for making accurate predictions. Model complexity is another significant factor; simpler models can reach their predictive potential with less data, and additional information might not enhance but rather confound their performance. The dynamic nature of the energy market also affects the training period; rapid changes in influencing factors mean that a shorter period of training data might better reflect the current trends without being overshadowed by outdated information. Lastly, the stationarity of the data is vital; the chosen training period should be sufficient to discern underlying patterns without being affected by non-stationary data, which could lead to poor generalization on future data points. These factors collectively suggest that a concise, focused dataset can often be more effective for training than an extensive one that may include irrelevant or misleading information.

4.5 Traditional ML Algorithms

Based on the above results, we will continue our experiments with the kNN Regressor algorithm, with the parameters **training_var=15** and **forecast_var=1** as we concluded previously. In this subsection, we will examine how our results change by altering some of the algorithm's parameters. We will also investigate the impact on the results of not using all the categories of features in the algorithm's training.

4.5.1 'n_neighbors' parameter

Initially, we will examine the **n_neighbors parameter**. The n_neighbors parameter in the KNN (K-Nearest Neighbors) Regressor algorithm is a critical hyperparameter that specifies the number of nearest neighbors to consider when making predictions. Essentially, it determines the size of the 'neighborhood' around a prediction point and influences how the algorithm weighs the importance of the closest training examples.

When n_neighbors is set to a low value, the model becomes more sensitive to noise in the training data, potentially leading to overfitting. This means the algorithm might perform well on the training data but poorly on unseen data, as it's too tailored to the specific examples it was trained on.

Conversely, a high value for n_neighbors leads to a smoother decision boundary which means the model is less sensitive to noise and may generalize better. However, if this value is too high, the model may underfit, failing to capture important patterns in the data.

The U-shaped curve (Figure 4.8) indicates that there is an optimal range for n_neighbors where the MAPE is minimized, suggesting the most accurate predictions occur within this range. Specifically, the MAPE decreases as n_neighbors increases

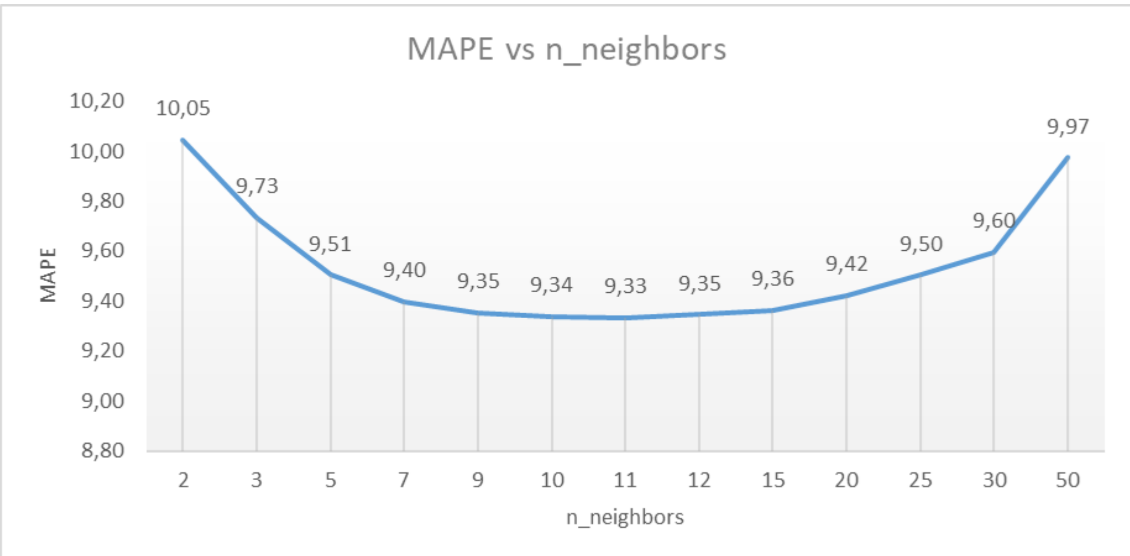

Figure 4.8: KNN performance with the adjustment of n_neighbors parameter

from 2, reaching its lowest point with n_neighbors values between 7 and 20, before slightly increasing again up to n_neighbors of 50.

This pattern reflects the balance between underfitting and overfitting. With too few neighbors, the model is overly complex and sensitive to noise in the training data, leading to overfitting and higher MAPE. As n_neighbors increases, the model averages more neighbors, becomes smoother, and generalizes better, reducing the MAPE. However, beyond the optimal range, the model starts to underfit; it becomes too generalized and loses the ability to capture the nuances in the data, causing the MAPE to rise again. The optimal price of n_neighbors is 11 and we will continue with this value parameter.

4.5.2 'metric' parameter

In the next experiment, we will examine the variable **metric**. This parameter determines the method used to measure the proximity between points, which directly affects the neighbors selected during the prediction process. Common metrics include the Euclidean distance, Manhattan distance, and Minkowski distance, each with its own way of quantifying the distance.

The impact of the metric parameter on the results can be profound. A metric that accurately reflects the structure of the data will lead to better identification of true nearest neighbors, resulting in more accurate predictions and lower errors. Conversely, an ill-suited metric can misguide the algorithm, leading to poor performance. Therefore, it's crucial to experiment with different metrics and evaluate their impact on the model's predictive accuracy, especially when dealing with complex or non-standard data distributions.

It's evident that the MAPE varies significantly across different metrics, highlighting the importance of metric selection in model performance. For instance, the 'hamming' metric shows a notably high MAPE of 18,59%, suggesting it may not be well-suited for this particular forecasting task. In contrast, metrics like 'braycurtis' and 'manhattan' yield the lowest MAPE values of 8,90%, indicating a better fit for our data.

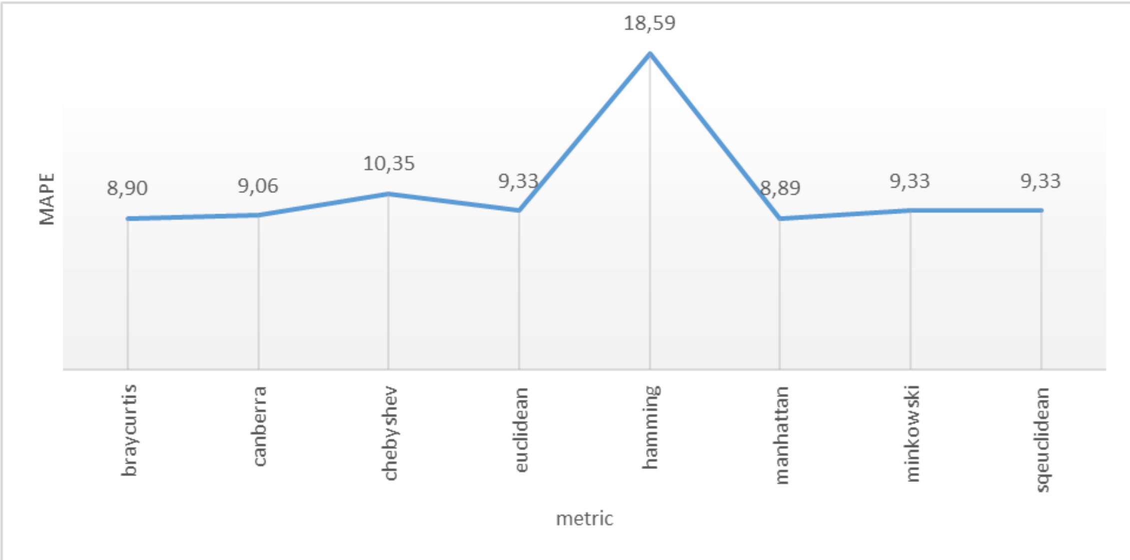

Figure 4.9: KNN performance with the adjustment of metric parameter

This variation in MAPE demonstrated by the graph (Figure 4.9) underscores the impact that the choice of metric has on the kNN algorithm's ability to accurately predict energy prices. Metrics that closely align with the true underlying patterns in the data will result in more accurate forecasts, as reflected by lower MAPE values.

4.5.3 'weights' parameter

The weights parameter in kNN Regressor algorithm determines how much influence each of the k-nearest neighbors has on the final prediction. There are typically two options for this parameter: **uniform** and **distance**.

With uniform weights, all neighbors contribute equally to the prediction, regardless of their distance from the query point. This approach assumes that all neighbors have the same level of importance and can be beneficial when the data distribution is uniform.

On the other hand, distance weights assign importance to neighbors based on their proximity to the query point. Closer neighbors have a greater influence on the prediction than those further away. This can be particularly useful in cases where closer data points are more likely to have similar target values and thus should have more say in the prediction.

Table 4.3: KNN performance with the adjustment of weights parameter

weights	MAPE
uniform	9,33%
distance	9,28%

The choice of weighting method can significantly alter the results of the KNN regressor. Using distance weights might help reduce the impact of outliers or irrelevant points by giving more weight to those points that are closer to the query point, potentially leading to more accurate predictions. However, it also makes the model more sensitive to the local structure of the data, which might not always be desirable. It's a balance between giving more importance to the most relevant points (distance) versus treating all neighbors equally (uniform)[97].

From the table 4.3, we can observe a slight decrease in MAPE when switching from uniform to distance weights, from 9,33% to 9,28%. This suggests that the distance weighting, which gives more influence to nearer neighbors, provides a marginally more accurate forecast than the uniform approach where all neighbors are considered equally, regardless of their distance to the query point.

This difference, albeit small, can be significant in the context of energy price forecasting where even minor improvements in accuracy can have substantial economic implications. The distance weighting method likely captures the nuances of the price fluctuations better by emphasizing the relevance of closer data points, which are presumably more similar to the query instance. It's a clear example of how fine-tuning the parameters of machine learning algorithms can yield tangible benefits in predictive tasks.

4.5.4 Feature categories

Additionally, we assess the model's effectiveness across multiple iterations, incorporating a new category of features in each run. This method aids in understanding the influence of various features on the model's performance. Specifically, we will conduct nine distinct trials as outlined subsequently in table 4.4.

At this point, it is worth noting that Run 8 was added retrospectively based on

Table 4.4: Runs with different feature categories

Runs	Features
1	Demand, RES Production and Prices feature
2	Run 1 + Commodities
3	Run 2 + Internal Exchanges
4	Run 3 + Cross-Border Italy + Cross-Border
5	Run 4 + Thermal Production
6	Run 5 + Hydro Production
7	Run 6 + RES Installed Capacity
8	Run 3 + Thermal production
9	All features included

the results of our experiments in the Ensemble ML Algorithms subsection and it will be explained in there. The Figure 4.10 illustrates the varying performance levels of the model across the different iterations:

The outcomes affirm the significance of the expanded dataset employed in training the model, concurrently demonstrating the impact of each category of features.

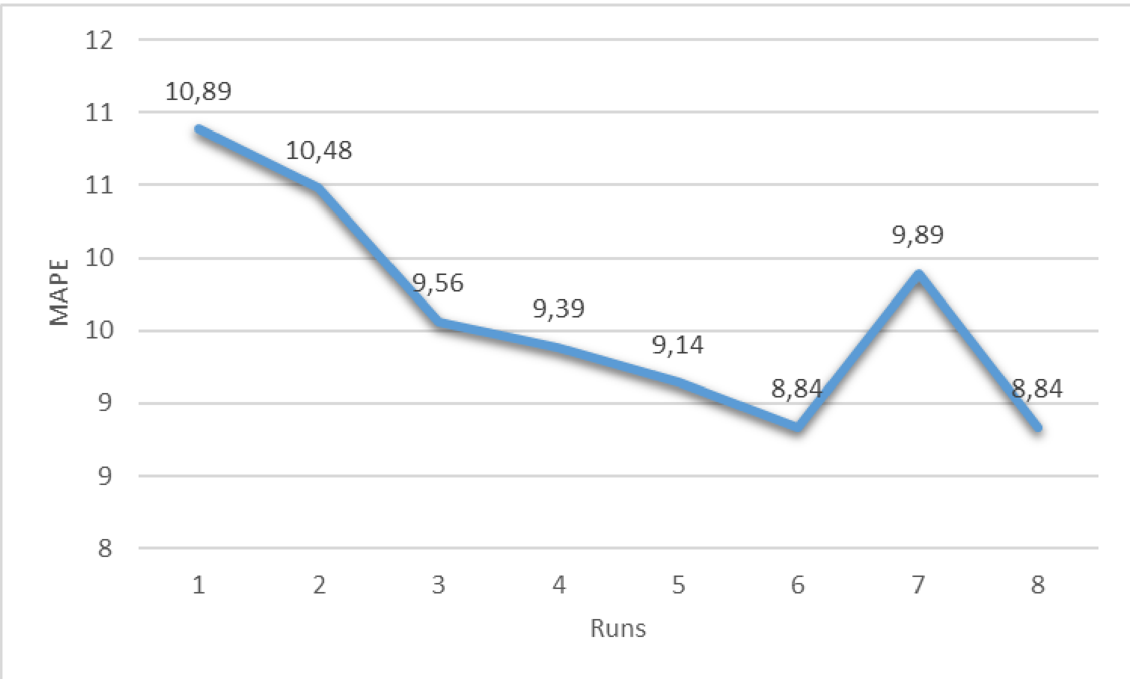

Figure 4.10: Model performance sensitivity when adding features

4.6 Ensemble ML Algorithms

Similarly to the traditional ML algorithms, a nested cross-validation procedure is followed to evaluate the ensemble machine learning algorithms. We keep the training period at 15 days, the forecast period at 1 day and for the evaluation metric MAPE, as in the case of classic ML algorithms. The table 4.5 presents the performance of the several ensemble ML algorithms in the nested cross-validation process.

The boxplot (Figure 4.11) summarizes the performance of the ensemble machine learning algorithms in the nested cross-validation procedure. Random Forest is shown to be marginally better than the other three algorithms for the particular task of MGP electricity price forecasting. That's why we will choose him to continue our experiments in the category of ensemble algorithms.

Table 4.5: Nested Cross-validation results of ensemble ML algorithms

Algorithm	MAPE - Mean	MAPE – std
Extra Tress Regressor	19,24	6,35
Bagging Regressor	19,34	6,70
XGBoost Regressor	19,24	6,89
Random Forest Regressor	19,16	6,43

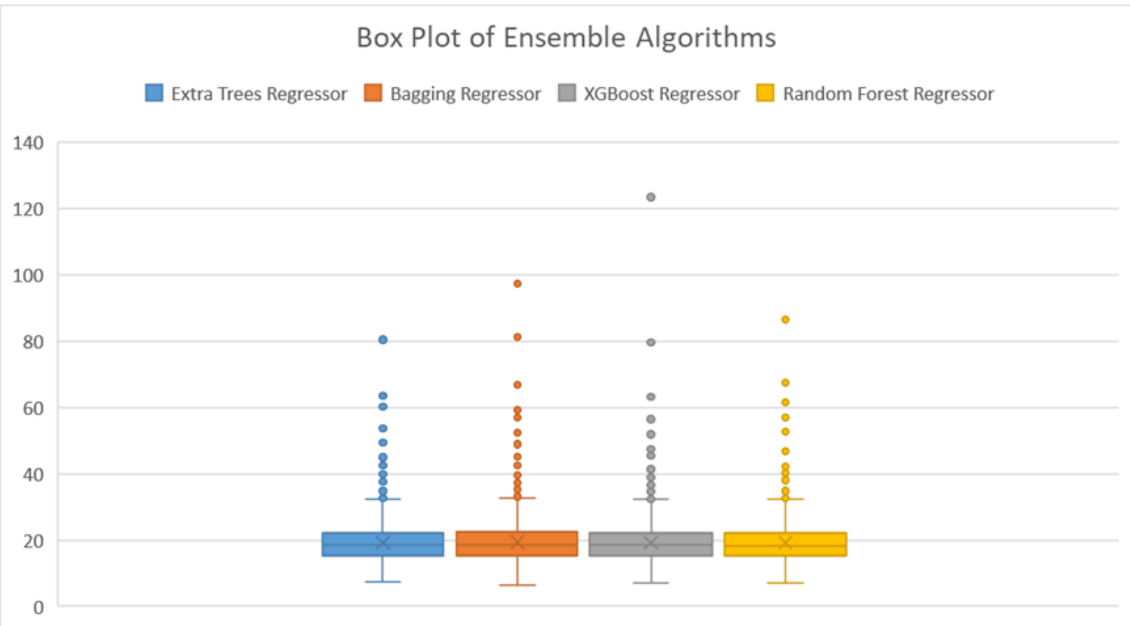

Figure 4.11: Boxplot: Cross-validation results of ensemble ML algorithms

4.6.1 'n_estimators' parameter

The n_estimators parameter in the Random Forest Regressor algorithm specifies the number of trees in the forest. It is a key hyperparameter that can significantly influence the model's performance and results. In essence, n_estimators controls the ensemble's size, with each tree contributing to the final prediction.

A higher number of trees generally improves the model's accuracy and robustness, as it reduces the variance part of the error, which is beneficial for preventing overfitting. However, beyond a certain point, the marginal gain in performance diminishes, and the computational cost becomes a factor to consider. More trees mean more computation, which can lead to longer training times and require more memory. Conversely, too few trees can lead to underfitting, where the model fails to capture the underlying patterns in the data adequately. Therefore, finding the right balance for n_estimators is crucial[98].

The graph (Figure 4.12) demonstrates the relationship between the n_estimators parameter of the Random Forest Regressor and the MAPE. The x-axis indicates the number of trees in the forest, ranging from 50 to 1000, while the y-axis shows the MAPE values.

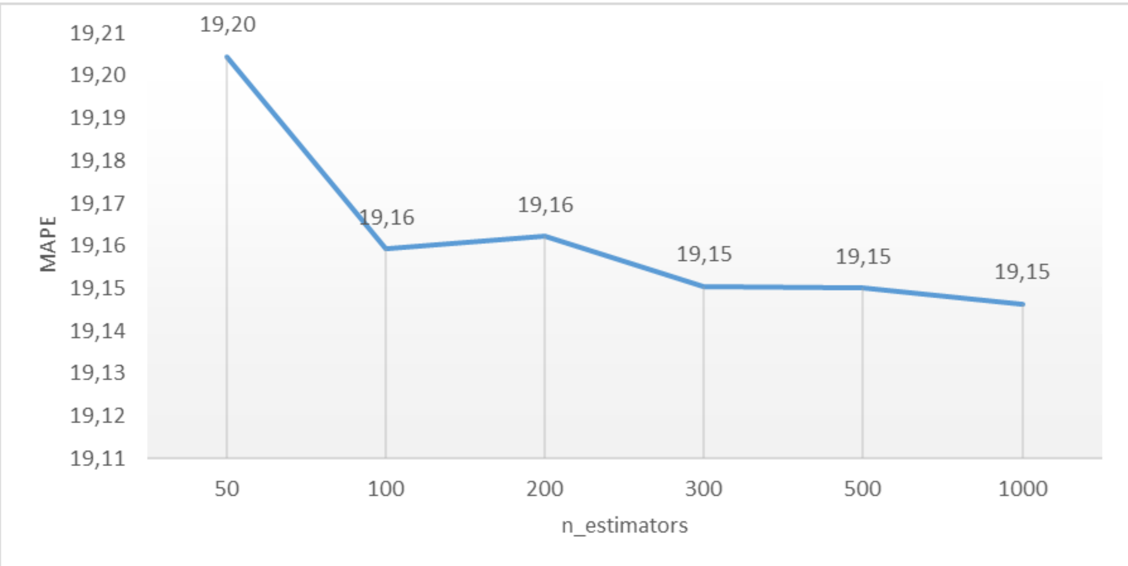

Figure 4.12: Random Forest Regressor performance with the adjustment of estimators parameter

As the number of estimators increases from 50 to 100, there is a noticeable decrease in MAPE, suggesting that the model's predictions become more accurate. This trend indicates that a larger number of trees can help improve the model's ability to capture the underlying patterns in the data. However, beyond 100 trees, the MAPE values plateau, indicating that adding more trees does not significantly enhance the forecasting accuracy. This plateau suggests that there is an optimal range for n_estimators where the model achieves a balance between accuracy and computational efficiency. Beyond this optimal point, the benefits of additional trees diminish, and the cost in terms of computation time and resources may not justify the marginal improvements in MAPE. For the purposes of our experiments, in order to balance the computational time and the error, we have decided to choose **500 estimators**.

4.6.2 'max_depth' parameter

The max_depth parameter is a hyperparameter that defines the maximum depth of each tree in the forest. Essentially, it sets a limit on how many nodes deep a tree can grow during the learning process. If max_depth is set to None, trees will grow until all leaves are pure or until all leaves contain fewer samples than min_samples_split. However, if an integer value is provided, it will restrict the growth of the trees to that depth.

Setting a deeper max_depth allows trees to learn more complex patterns by creating more decision nodes. This can lead to better performance on the training data, but it also increases the risk of overfitting. Conversely, a shallower max_depth can help prevent overfitting by simplifying the model, but it may also prevent the trees from capturing important patterns, leading to underfitting.

A well-chosen max_depth can help balance the bias-variance trade-off, leading to a model that generalizes well to new data[99].

A notable trend in the graph (Figure 4.13) is the decrease in MAPE as the max_depth decreases. This suggests that shallower trees, which are less complex, tend to perform better in this forecasting task. At a max_depth of 20, the MAPE

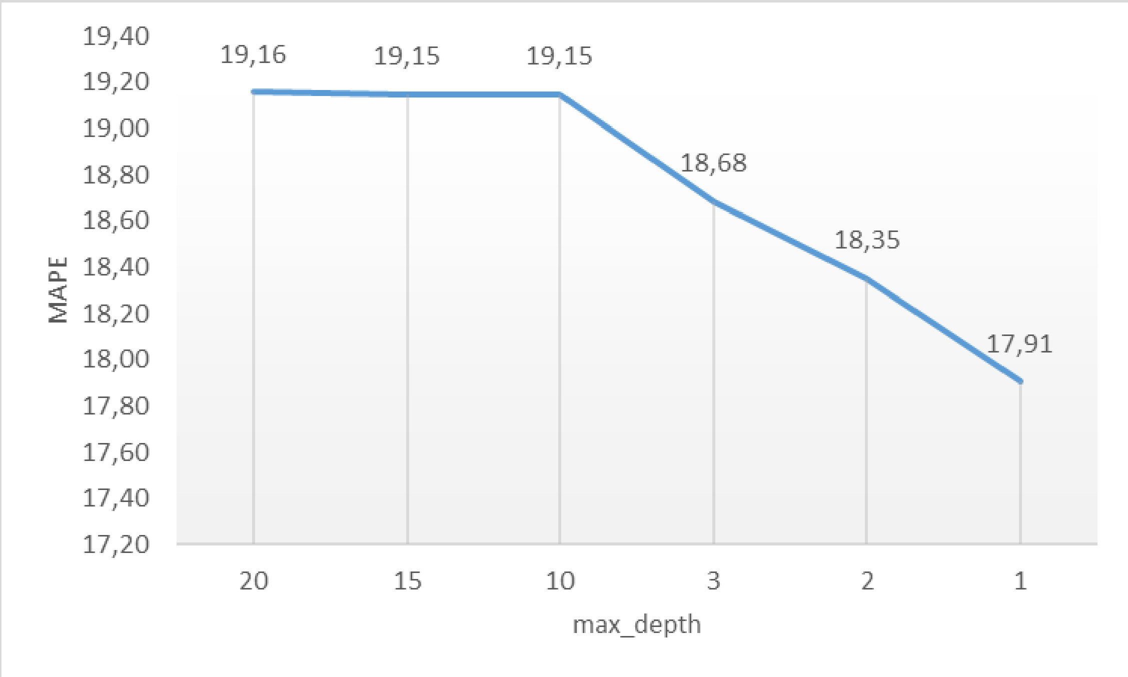

Figure 4.13: Model performance with the adjustment of maximum depth parameter

is higher, around 19,16, indicating less accurate predictions. As the max_depth is reduced, the MAPE improves, reaching its lowest at a max_depth of 1, with a MAPE of approximately 17,91.

4.6.3 'max_features' parameter

The max_features parameter a hyperparameter that dictates the number of features to consider when looking for the best split at each node of the trees. By default, for regression problems, it is often set to use all the features (n_features), which means that no feature subset selection is performed, and the algorithm behaves more like a bagged ensemble of trees rather than a true Random Forest.

If set to a lower value, such as sqrt(n_features) or a fraction of the total features, it introduces more randomness into the model, which can help in improving the generalization by reducing the correlation between individual trees. This can lead to a more robust model that performs better on unseen data. On the other hand, setting max_features too low might not allow the trees to consider enough information, potentially missing out on important features that could improve the model's performance.

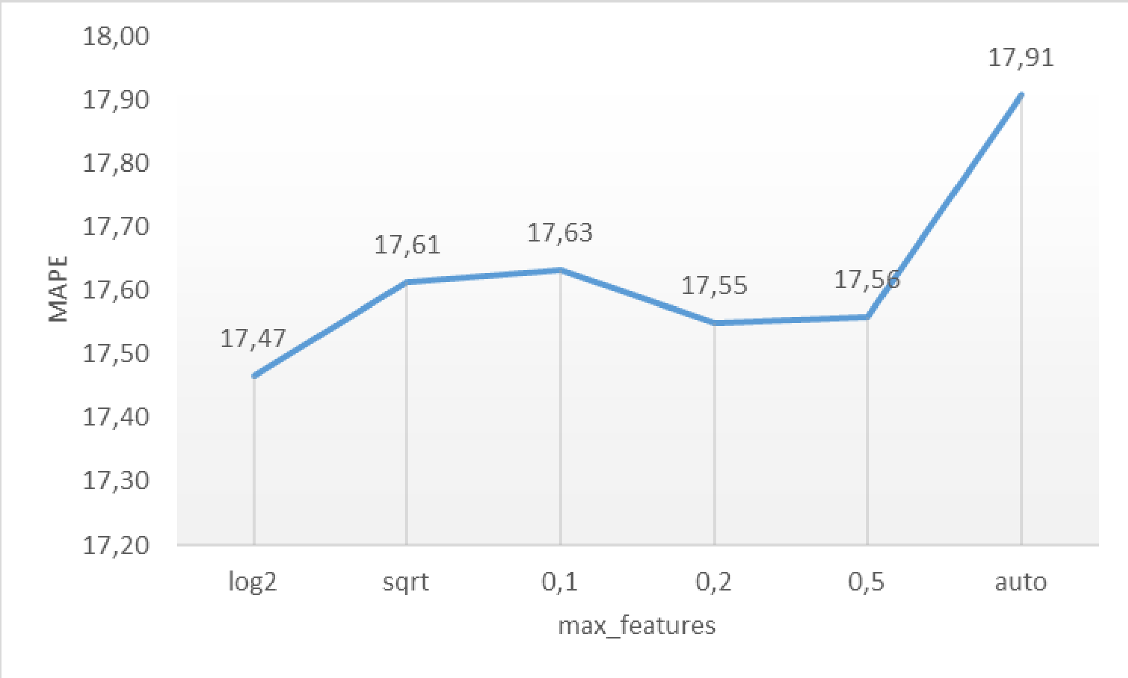

Figure 4.14: Model performance with the adjustment of maximum features parameter

In practice, tuning max_features is about finding the sweet spot where the model is complex enough to capture the underlying patterns in the data, but also simple enough to generalize well to new data.

From the graph (Figure 4.14), we can observe that the MAPE starts at approximately 17,47% with 'log2', slightly decreases to around 17,41% with 'sqrt', then increases with 0,1 and more significantly with 0,2, reaching about 17,55% before dropping again slightly at 'auto'. However, there's a sharp increase after 'auto' where MAPE peaks at approximately 17,91%. A smaller number of max_features tends to reduce overfitting by adding randomness to the model, which can be beneficial for generalization. However, the graph indicates that there is a threshold beyond which reducing max_features too much or increasing it can adversely affect the model's performance.

4.6.4 'min_samples_split' parameter

The min_samples_split parameter in the Random Forest Regressor algorithm specifies the minimum number of samples required to split an internal node. Its default value is typically set to 2, which means a node will be split if it contains 2 or more samples. This parameter is crucial for controlling the tree's growth and complexity within the forest.

A higher min_samples_split value increases the number of samples required to split a node, which can lead to simpler, more generalized trees that may prevent overfitting. This is because it enforces a larger group size for decision splits, thus reducing the model's sensitivity to noise in the training data. On the other hand, a lower value for min_samples_split allows the trees to grow deeper and more complex, which can capture more information about the data but also increases the risk of overfitting.

Therefore, adjusting min_samples_split can significantly alter the results of a Random Forest model. It's a balancing act: setting it too high might cause underfitting, where the model is too simple to capture underlying patterns, while setting it too low might lead to overfitting, where the model captures too much noise.

From the graph (Figure 4.15), it appears that as min_samples_split increases from 2 to 10, there is a peak in MAPE, suggesting that a smaller number of samples at each node (leading to more complex trees) does not necessarily result in better forecasting accuracy. Beyond the value of 10, the MAPE stabilizes, indicating that

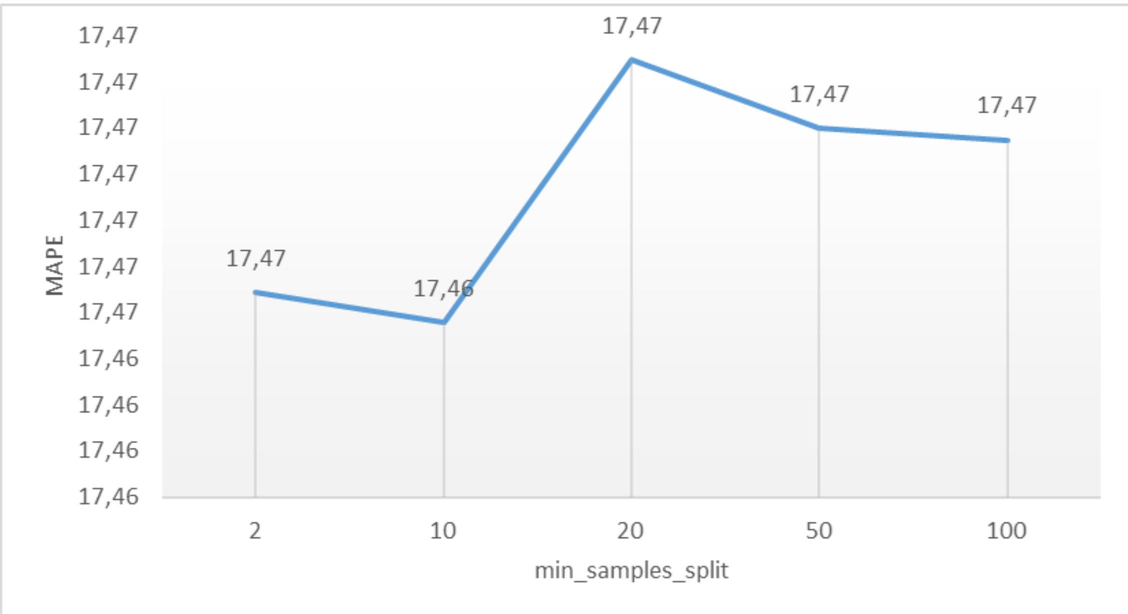

Figure 4.15: Model performance with the adjustment of minimum number of samples required to split an internal node (min_samples_split) parameter

further increasing the min_samples_split does not significantly improve the forecast accuracy. This could mean that beyond a certain point, allowing the trees to grow more complex does not yield better results and that a moderate level of complexity in the trees is optimal for this particular forecasting task. The changes in the MAPE are almost negligible, which is why we will not focus further on this parameter and will choose the default (min_samples_split=2) value.

4.6.5 'min_samples_leaf' parameter

The min_samples_leaf parameter is a measure that specifies the minimum number of samples required to form a leaf node. This parameter is instrumental in controlling the size of the tree and, consequently, the complexity of the model. A higher value for min_samples_leaf ensures that each leaf node has more observations, which can smooth the model and potentially improve its generalization capabilities by reducing overfitting.

When min_samples_leaf is set to a low value, trees in the forest may grow very deep with leaves representing very few samples, which can capture noise and specific patterns in the training data that do not generalize well to unseen data. This can lead to a model that overfits the training set. Conversely, setting a higher min_samples_leaf value can result in shallower trees that generalize better but may not capture all the nuances in the training data, potentially leading to underfitting if set too high[87].

From the graph (Figure 4.16), we can see that the MAPE starts at approximately 17,46% with a min_samples_leaf of 1 and remains relatively stable as it moves through values of 5 and 10. However, there is a noticeable increase in MAPE to around 17,50% at a min_samples_leaf of 50. This suggests that while small increases in min_samples_leaf do not significantly affect the model's accuracy, a larger min_samples_leaf value may lead to less precise predictions. It indicates that there is an optimal range for min_samples_leaf that minimizes prediction error without causing the model to become too generalized and lose predictive power. The changes in the MAPE are almost negligible, which is why we will not focus further on this parameter and will choose the default (min_samples_leaf=1) value.

4.6.6 Feature categories

Like with the traditional machine learning algorithms, we assess the model's effectiveness across multiple iterations, incorporating a new category of features in each run. 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:

As the graph (Figure 4.17) shows, as we add feature categories, the MAPE decreases. However, in run 6, it appears that the error increases sharply, which means that the categories that are being added, are crating noise in our model. At this point, it is worth explaining the creation of Run 8. The graph shows that, up to run 3, the error is continuously decreasing, in run 4 it increases, and in run 5 it decreases again. For this reason, we decided to create Run 8 where, having the feature categories from run 3, we add the feature categories from run 5 (namely thermal production). It is as if we are bypassing run 4.

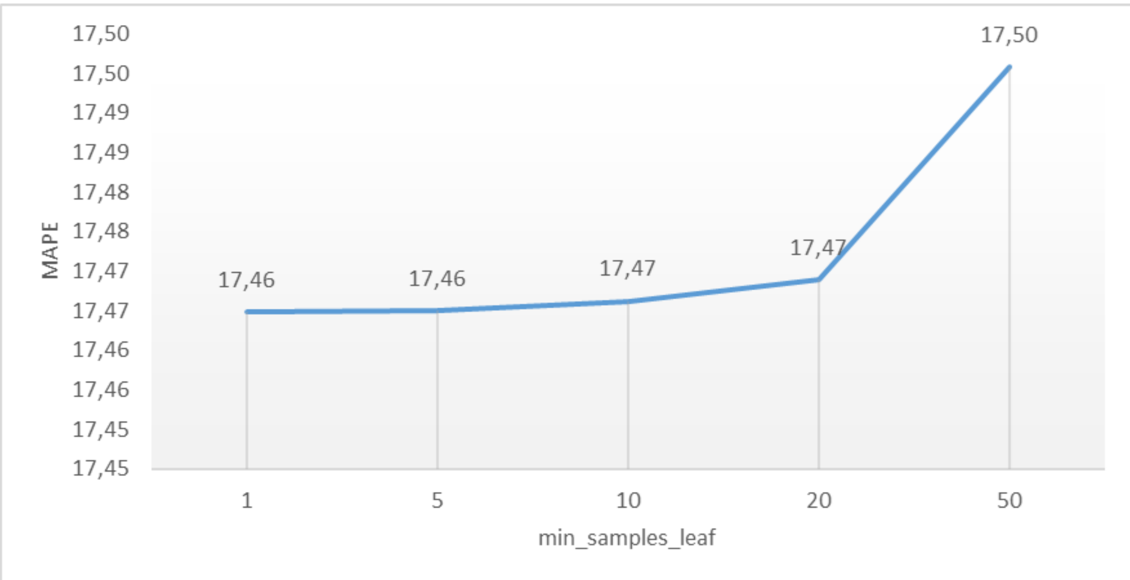

Figure 4.16: Model performance with the adjustment of minimum number of leaves

4.7 Deep Learning Algorithms

After exploring the realms of traditional machine learning (ML) and ensemble methods, one might wonder, “What’s next?” The answer lies in the powerful domain of deep learning (DL) algorithms. Deep learning represents the next leap in the evolution of artificial intelligence, offering a level of complexity and performance that traditional ML methods cannot match. The implementation of deep learning experiments requires a shift in approach. It involves setting up neural networks with multiple hidden layers, known as deep architectures. These networks are trained using large amounts of data and computational power, often leveraging GPUs for efficient processing. In our experiments, we have used a Multilayer Perceptron (MLP) and a Long Short Term Memory algorithm (LSTM). The construction of the Artificial Neural Network (ANN) utilized TensorFlow 2.0 and Keras as its foundational frameworks. In Keras, the fundamental building blocks are layers and models, which serve as the essential data structures for network architecture.

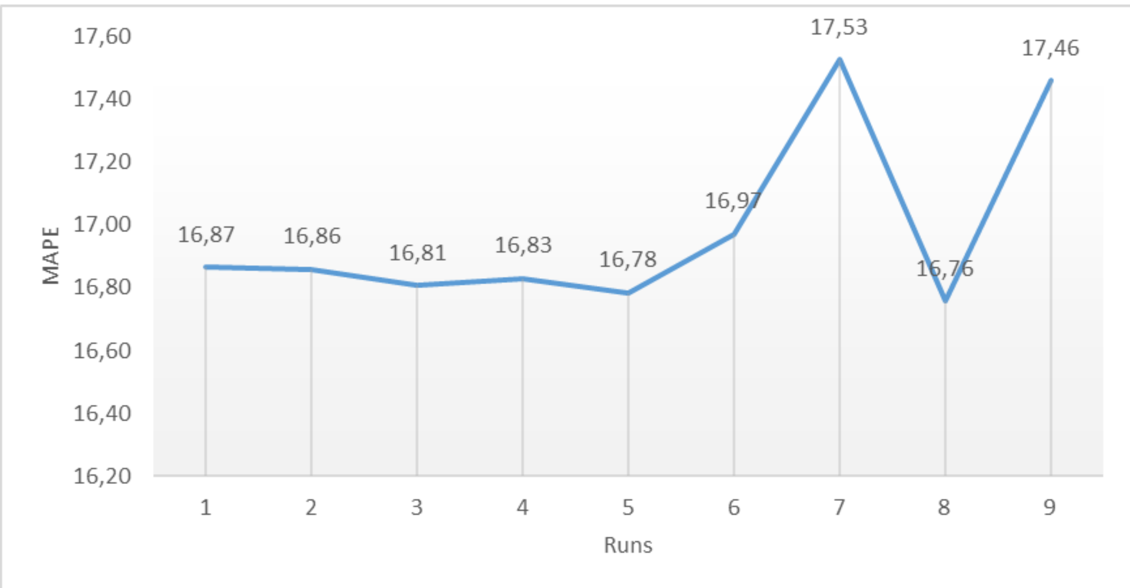

Figure 4.17: Model performance sensitivity when adding features

4.7.1 Multilayer Perceptron (MLP)

In this case, we employed the Sequential model, characterized by its linear arrangement of layers. This model is instantiated by supplying a sequence of layers to the Sequential constructor, with each layer potentially encompassing multiple 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, units per layer, activation layer, solver and learning rate.

We started the experiments using the following implementation: one input layer, one hidden layer with 100 units, and one output layer, 'relu' activation function, 'SGD' as a solver, Batch size set to 'auto' and learning rate as 'constant'.

4.7.1.1 'epochs' parameter

The epochs parameter plays a crucial role in determining the effectiveness of the model. An epoch refers to one complete pass through the entire training dataset. The number of epochs is a hyperparameter that defines the number times the learning algorithm will work through the entire training dataset. One epoch involves two main steps: the forward pass, where the current state of the model's weights is used to make predictions, and the backward pass, where the model's weights are updated in response to the error of those predictions. The choice of the number of epochs can significantly affect the model's performance. Too few epochs can result in an underfit model that does not learn enough from the training data, leading to poor generalization to new data. Conversely, too many epochs can cause the model to overfit the training data, capturing noise and details that do not generalize well[100].

At the start (Figure 4.18), the MAPE is relatively high, indicating that the model's predictions are not very accurate. As the number of epochs increases, the MAPE decreases, reaching its lowest point near 60 epochs. This suggests that the model has learned effectively from the training data, improving its predictive accuracy. Beyond this optimal point, the MAPE begins to rise, peaking around 200 epochs. This increase could be indicative of overfitting, where the model starts to memorize the training data, including noise, rather than learning to generalize from it. After the peak, the MAPE stabilizes slightly, then increases again towards 500 epochs, ending at a higher value than the optimal point. This further supports the possibility of overfitting, as the model's generalization to unseen data worsens with too many epochs. We will continue with our experiments with 60 epochs.

4.7.1.2 Number of layers and units per layer parameter

In the context of using a Multilayer Perceptron (MLP) for electricity price forecasting, the architecture of the network - specifically, the number of layers and the number of units per layer - is pivotal in determining the model's capacity to learn and generalize. The number of layers in an MLP adds depth to the model, enabling it to learn more complex representations of the data. However, adding too many layers can lead to overfitting, where the model learns the training data too well, including noise, which can degrade its performance on unseen data.

The units per layer, or neurons, represent the width of the layers. More neurons can allow the network to capture more information and create more intricate decision boundaries. Yet, similar to the number of layers, too many neurons can cause overfitting, making the model overly complex and sensitive to the training data's

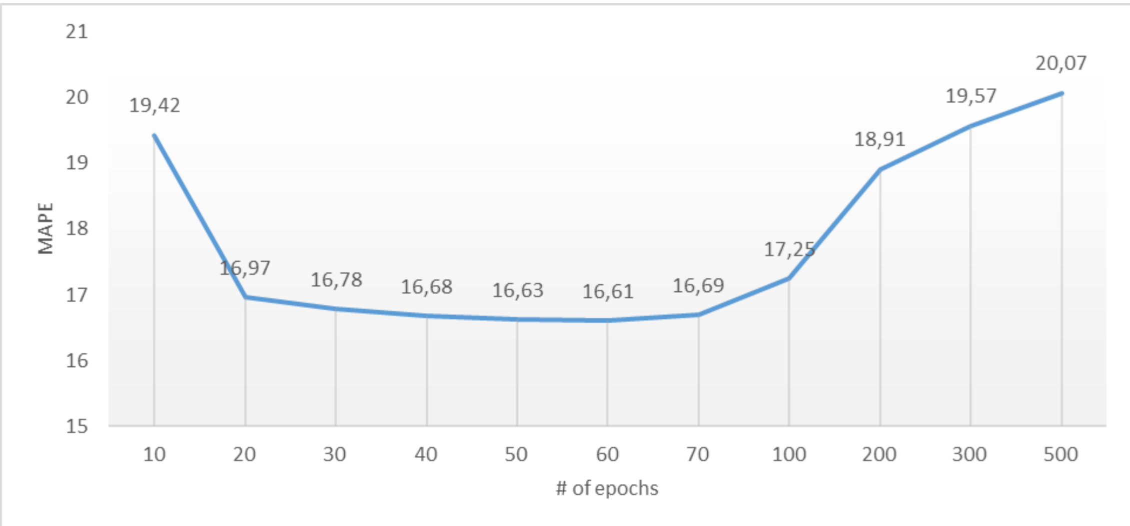

Figure 4.18: Model performance with different number of epochs

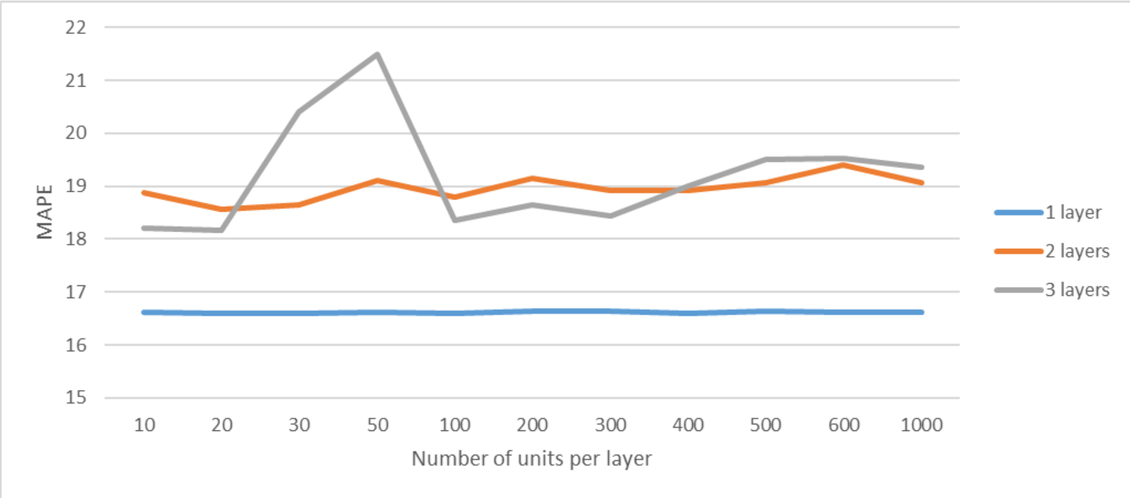

Figure 4.19: Model performance with different number of hidden layers and units per layer

idiosyncrasies.

Too few layers or neurons might result in underfitting, where the model is too simple to capture the underlying trends in electricity prices. Too many layers or neurons can make the model too complex and prone to overfitting[101]. In our case, we are using the one hidden layer that we mentioned before and we are adding two more hidden layers. We are also changing the number of units of each layer and see how this affects our results.

With just one hidden layer (Figure 4.19), the MAPE remains relatively stable across different numbers of units, suggesting that a single-layer network's capacity to model the complexity of electricity price data is limited. For the two hidden layer network, there's an initial decrease in MAPE as the number of units increases, indicating improved accuracy. However, after a certain point, the MAPE increases, suggesting overfitting due to too many units. The three hidden layer network starts with a high MAPE, then increses sharply until MAPE decrises at a lower level. This could indicate that deeper networks might require a more careful balance of units to avoid overfitting while still capturing the necessary level of detail in the data. Overall, the graph highlights the importance of tuning the number of layers and units in an MLP. It shows that there is an optimal range for the number of units per layer that minimizes the MAPE, and this optimal range can vary depending on the number of layers in the network. Too few units may not capture the complexity of the data, while too many can lead to overfitting, especially in networks with more layers. Based on the results we presented, we will continue our experiments using one hidden layer.

4.7.1.3 'activation function' parameter

The activation function in a Multilayer Perceptron (MLP) is critical as it introduces non-linearity into the network, enabling it to learn and perform more complex tasks than a linear model could. In the context of electricity price forecasting, the choice of activation function can significantly affect the model's accuracy and convergence speed.

Common activation functions include the sigmoid, logistic, identity, tanh, and ReLU. The sigmoid and tanh functions have historically been used in MLPs, but they can suffer from vanishing gradients, making training deep networks challenging. On the other hand, the ReLU function has become more popular due to its simplicity and efficiency, allowing models to train faster and perform better with fewer issues of vanishing gradients.

However, each activation function has its own characteristics that can influence the model's performance. For instance, ReLU can lead to dead neurons where some neurons can become inactive and stop contributing to the model's learning, which is known as the dying ReLU problem. To mitigate this, variants like Leaky ReLU or Parametric ReLU can be used[101].

The following graph (Figure 4.20) offers a comparative analysis of how different activation functions affect the MAPE with respect to the number of units per layer. The activation functions compared are ReLU, Tanh, Logistic, and Identity.

- **ReLU**: Starting with a low MAPE, it shows a stable behavior as the number of units increases.

- **Tanh**: It begins with a higher MAPE than ReLU but with an increase in units, it shows improved performance.

- **Logistic**: This function shows a improved performance until the 100 number of units and after that the there is a steep increase in MAPE

- **Identity**: It demonstrates almost the same stability as ReLU but at a higher MAPE level.

This graph highlights the importance of selecting the right activation function and tuning the number of units per layer in an MLP. While some functions like tanh show improved performance with more units, others like ReLU and Identity do not exhibit significant changes.

4.7.1.4 'solver' parameter

The solver is the optimization algorithm used to update the weights of the network during training. The choice of solver can significantly affect the convergence rate, the quality of the final model, and its ability to forecast electricity prices accurately.

Common solvers include Stochastic Gradient Descent (SGD), Adam, and L-BFGS. SGD is widely used due to its simplicity and effectiveness, especially when combined with momentum and learning rate schedules. However, it may require careful tuning of hyperparameters and can be sensitive to the choice of learning rate.

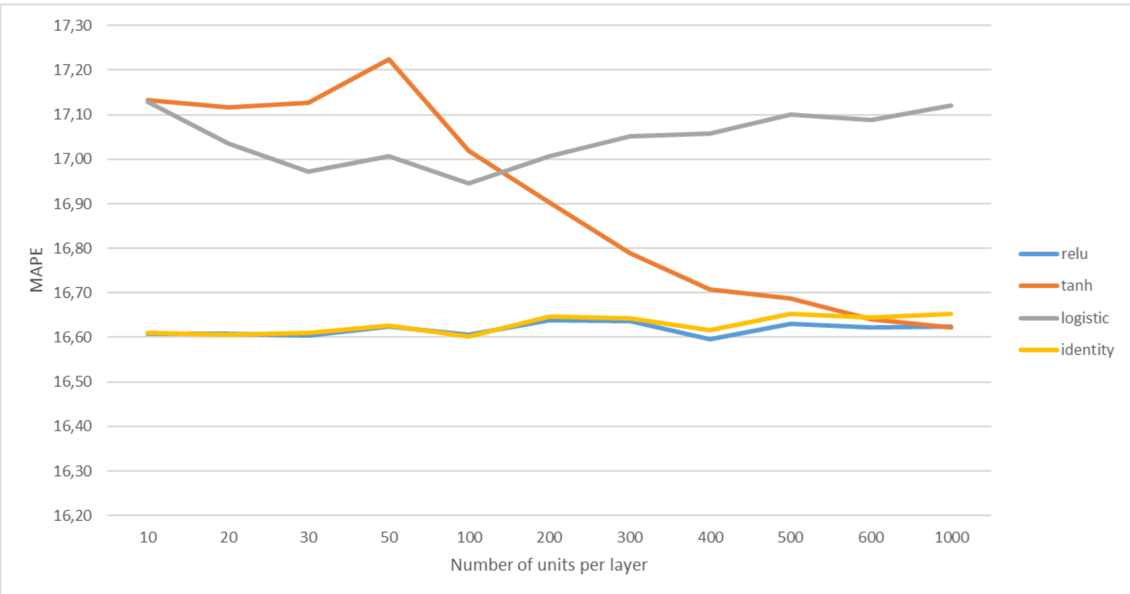

Figure 4.20: Model performance with different activation functions and units per layer

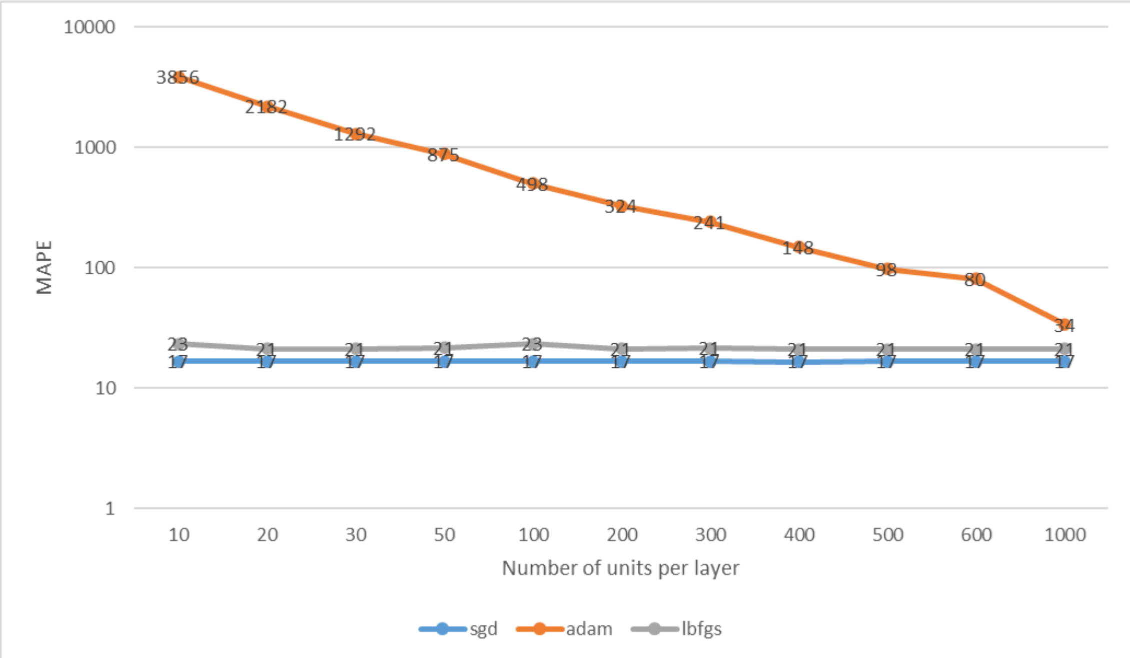

Figure 4.21: Model performance with different solvers and units per layer

Adam is an adaptive learning rate optimization algorithm that has been designed specifically for training deep neural networks. It combines the advantages of two other extensions of SGD, AdaGrad and RMSProp, and is often effective in practice with little hyperparameter tuning.

L-BFGS is an optimizer in the family of quasi-Newton methods. It's more suited for smaller datasets due to its memory requirements and can converge faster to a solution with less noise than SGD.

The choice of solver depends on the size of the dataset, the complexity of the problem, and the computational resources available. For electricity price forecasting, where the data can be noisy and non-stationary, a solver like SGD might be preferable for its robustness and ease of use.

The orange line in Figure 4.21 representing adam shows a significant decrease in MAPE as the number of units increases, indicating that the solver benefits from a larger network capacity. However, after a certain point, the improvements plateau, suggesting an optimal number of units for this solver. The blue line for SGD maintains a relatively consistent MAPE across the range of units, with slight fluctuations. This could indicate that Adam is less sensitive to changes in network size, possibly due to its adaptive learning rate. The grey line for L-BFGS also shows a consistent MAPE across different numbers of units, similar to SGD, which may be attributed to its ability to converge more rapidly on smaller datasets.

4.7.1.5 'learning rate' parameter

The learning rate is a hyperparameter that controls how much the weights of our network are adjusted with respect to the loss gradient. It has a significant impact on the training process and the performance of an MLP algorithm, especially in complex tasks like electricity price forecasting. A too high learning rate can cause the model to converge too quickly to a suboptimal solution, or it may cause the training process to be unstable. The model might overshoot the minimum of the loss function and fail to converge.

On the other hand, a too low learning rate makes the training process slower and can get stuck in local minima, leading to subpar results. It might also never converge to the global minimum within a reasonable time frame. An optimal learning rate is crucial for good performance. It ensures that the model learns efficiently and reaches the best possible set of weights. Techniques like learning rate schedules or adaptive learning rate algorithms (e.g., Adam, RMSprop) can be used to adjust the learning rate during training to improve convergence. In Figure 4.22 we can see the MAPE of different Learning Rates:

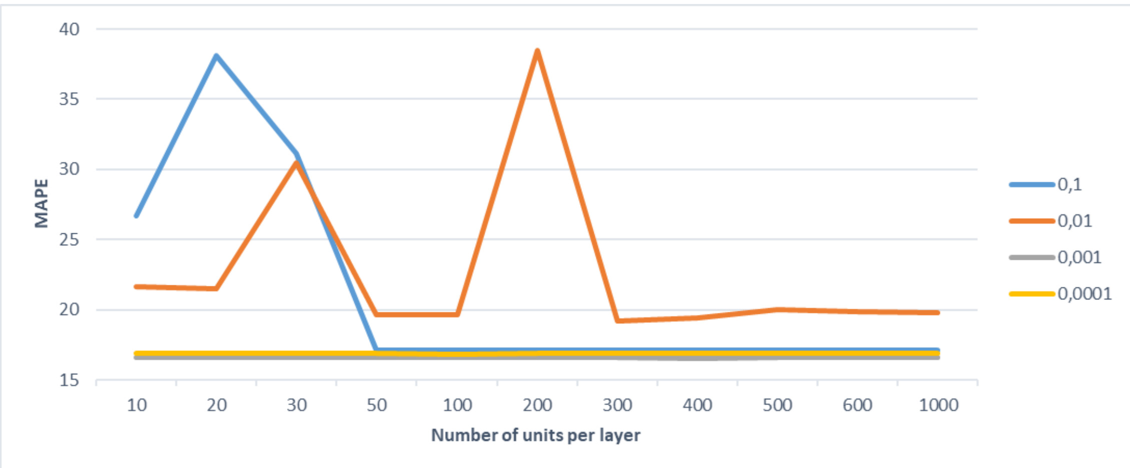

Figure 4.22: Model performance with different learning rates and units per layer

- **Learning Rate 0.1**: Starting near 22, this line shows a significant peak before stabilizing at a much lower level, which could mean that while the learning rate is generally effective, it may require careful tuning of other hyperparameters to maintain consistent performance.
- **Learning Rate 0,01**: This line exhibits high volatility, starting with a MAPE of around 20, peaking at nearly 40, and then decreasing sharply. This behavior suggests that a learning rate of 0.1 may be too high, causing the model to overshoot the optimal weights during training.
- **Learning Rate 0.001**: This line is relatively stable, indicating that a learning rate of 0.001 provides a good balance between convergence speed and stability, avoiding the overshooting seen with higher learning rates.
- **Learning Rate 0.0001**: With negligible variation in MAPE, the yellow line suggests that a learning rate of 0.0001 might be too low, leading to slow convergence and potentially getting stuck in local minima.

4.7.2 LSTM

Beyond the implementation with the MLP Regressor, we decided to use a state-of-the-art algorithm that is used for energy price forecasting. Its implementation is described below: The target column from the dataset is extracted and reshaped to fit the model's input requirements. The data is then normalized using MinMaxScaler to ensure that the LSTM model receives input values within a scaled range of 0 to 1, which is crucial for the model's performance. The model is fed sequences of data points to predict the next time step. This sequence length is a hyperparameter that will be tuned based on the periodicity and patterns in the data. The LSTM model is constructed using Keras, with two LSTM layers followed by a dense layer that outputs the predicted value. The first LSTM layer returns sequences to provide a three-dimensional array as input to the next LSTM layer, which is necessary for stacking LSTMs. The model is compiled with Stochastic Gradient Descent (SGD) as the optimizer and mean absolute error as the loss function. Additional metrics like MAPE are tracked.

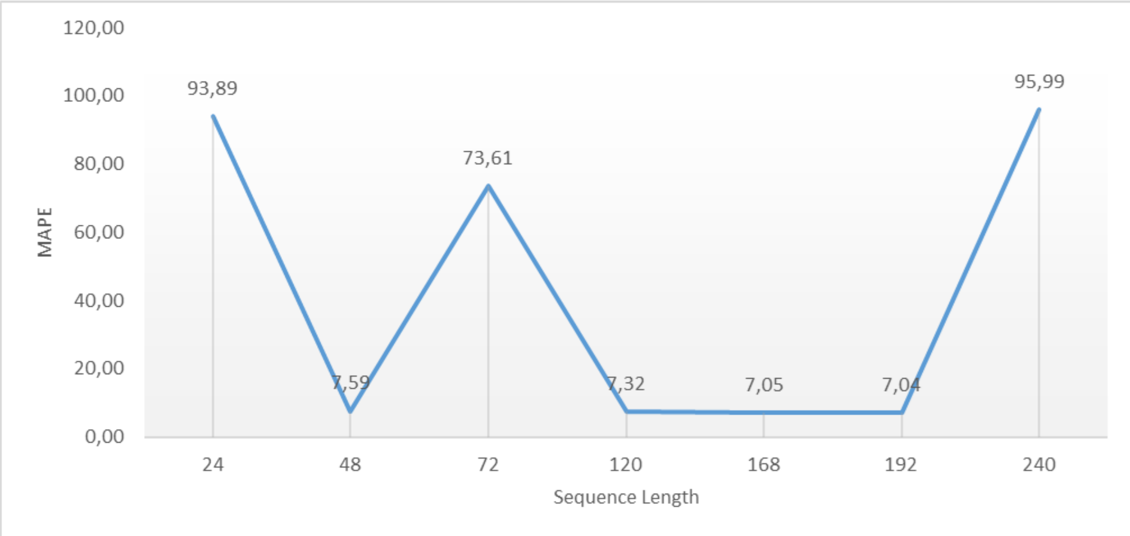

Figure 4.23: LSTM performance with different sequence lengths

4.7.2.1 'sequence length' parameter

In an LSTM for electricity price forecasting, the sequence length is a critical hyperparameter that defines the number of previous time steps used to predict future prices. It essentially determines the amount of historical data the model considers when making a prediction. A short sequence length may not provide the LSTM with enough context to capture the underlying patterns in the data, especially if the price movements are influenced by long-term trends or seasonal effects. This can lead to poor model performance as it might miss out on important information that affects price movements. Conversely, a long sequence length can provide more historical context, allowing the model to learn from more complex patterns and dependencies. However, it also increases the computational complexity and the risk of overfitting, as the model may start to learn irrelevant details or noise in the data.

This graph (Figure 4.23) highlights the importance of carefully selecting the sequence length for an LSTM model. The optimal sequence length appears to be around 72 (3 days of data) and between 168 (7 days) to 192 (8 days), where the model achieves the lowest MAPE, suggesting the most accurate predictions. Based on these results, we'll continue our runs with sequence length of 192.

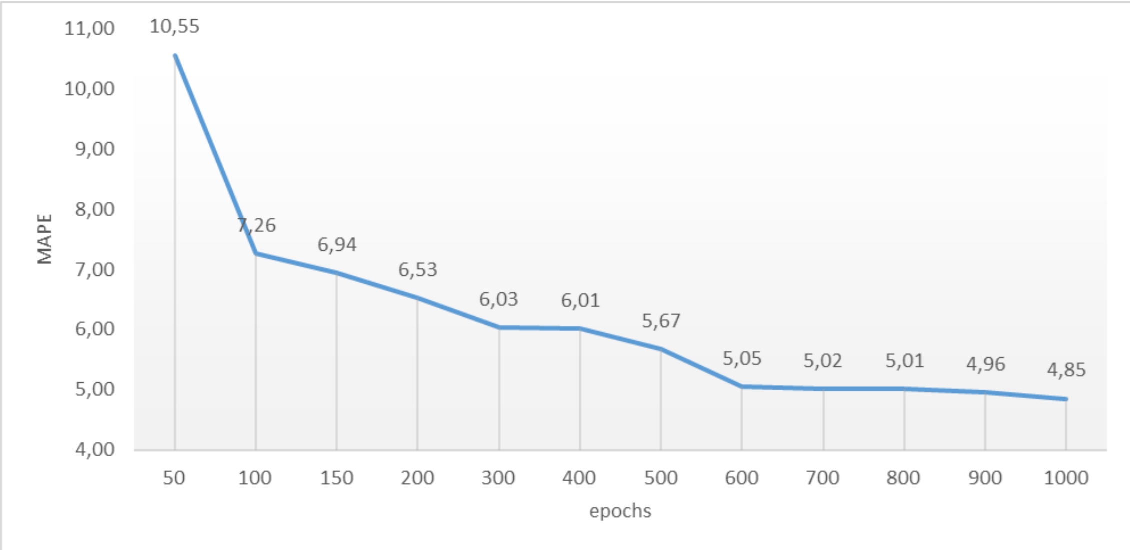

Figure 4.24: LSTM performance with different epochs

4.7.2.2 'epochs' parameter

In the context of training an LSTM network for electricity price forecasting, epochs refer to the number of times the entire dataset is passed forward and backward through the neural network. A small number of epochs might not be sufficient for the network to learn the complex patterns in the data, leading to underfitting. The model may not converge to a low error rate, resulting in poor forecasting accuracy.On the other hand, training with a large number of epochs allows the model more opportunities to learn and adjust its weights. However, too many epochs can lead to overfitting, where the model learns the noise in the training data rather than the underlying trend, negatively impacting its generalization to unseen data.

At the start (Figure 4.24), with fewer epochs, the MAPE is relatively high, indicating that the model has not yet learned enough from the data to make accurate predictions. As the number of epochs increases, there's a noticeable decrease in MAPE, suggesting that the model is learning and improving its forecasting accuracy. After a certain number of epochs, the MAPE levels off, indicating that additional training does not significantly improve the model's performance. This could be the point where the model has learned the patterns in the data effectively. The optimal range of epochs appears to be where the MAPE is lowest and before it plateaus, which is around 600 epochs based on the graph. Training beyond this point may

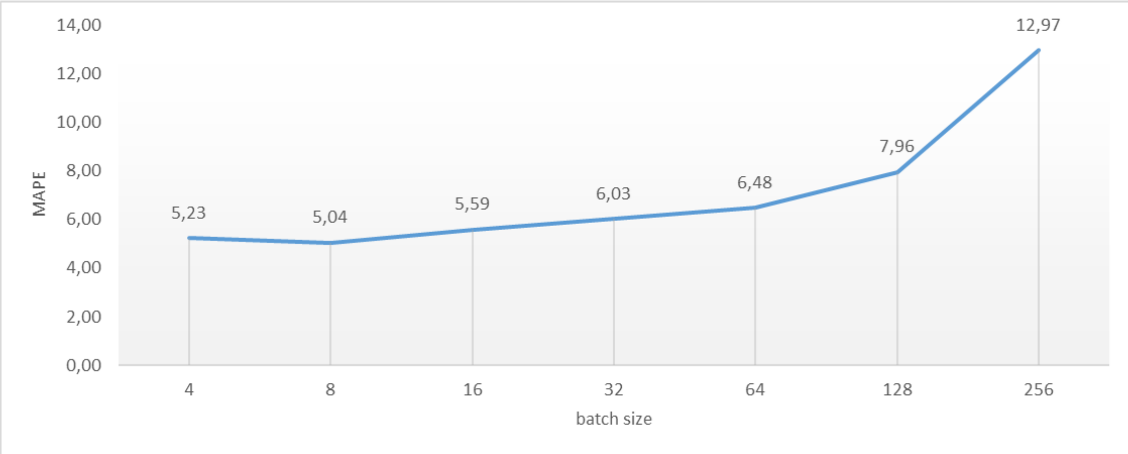

Figure 4.25: LSTM performance with different batch sizes

not yield significant improvements and could lead to overfitting.

4.7.2.3 'batch size' parameter

The batch size is a hyperparameter that defines the number of samples to work through before updating the internal model parameters. A small batch size often provides the benefit of faster convergence and can lead to a more robust model by providing a regularizing effect and reducing the likelihood of overfitting. However, smaller batches may result in a noisier gradient signal, which can make the training process more challenging and potentially lead to convergence on local minima. A large batch size can provide a more accurate estimate of the gradient, but it may require more memory and computational power. Additionally, larger batches can lead to poorer generalization on unseen data.

Starting with a batch size of 4 (Figure 4.25), the MAPE is around 5,23%, indicating a relatively low error rate. As the batch size increases to 8, there's a slight decrease in MAPE to approximately 5,04%, suggesting that small batch sizes are effective for this model. As the batch size continues to grow to 16 and beyond, the MAPE increases significantly, reaching around 6,03% for a batch size of 16 and peaking at approximately 12,97% for a batch size of 256.

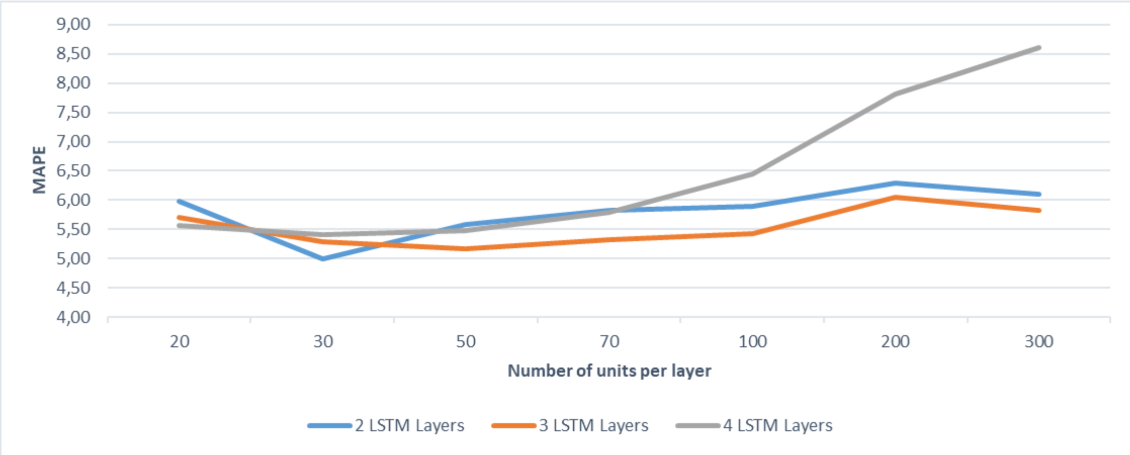

Figure 4.26: LSTM performance with different number of layers and units per layer

4.7.2.4 Number of layers and units per layer parameters

Multiple LSTM layers can capture deeper levels of temporal abstraction, which may improve the model's ability to understand long-term dependencies and intricate patterns in the data. However, adding more layers also increases the risk of overfitting, especially if the dataset is not large enough to support such complexity. The number of units in each layer determines the capacity of the model to learn. More units can allow the LSTM to capture more information and create a more detailed internal representation of the input data. Yet, similar to the number of layers, too many units can cause the model to overfit and also increase the computational burden.

As seen from the graph (Figure 4.26), starting with 2 LSTM layers, the MAPE for 20 units begins at a value close to 6%, drops to 5% at 30 units, and then starts to increase when the units are 50 and above. Similarly, when we add another layer and reach 3 LSTM layers, the MAPE starts at a value of 5.7% for 20 units, reaches a minimum of 5.16% at 50 units, and from there on, the MAPE begins to increase again. However, we observe that with an extra layer, the MAPE is overall smaller compared to the fewer layers. Finally, reaching 4 LSTM layers, the algorithm seems to overfit, and the MAPE increases across all units.

To conclude, we will encapsulate the outcomes derived from the various algorithms applied to the Italian electricity price forecasting task as presented in this thesis (Table 4.6).

Table 4.6: Algorithms performance

Algorithm	MAPE (%)
Univariate Algorithm (Facebook's Prophet)	21,29
Classic ML Algorithm (kNN Regressor)	8,84
Ensemble ML Algorithm (Random Forest Regressor)	16,76
Deep Learning Algorithm (MLP)	16,61
Deep Learning Algorithm (LSTM)	5,04

Chapter 5

Conclusions and Future Work

5.1 Conclusion

This thesis aims to examine how algorithms improve as we transition from conventional to advanced techniques in predicting electricity prices. We will concentrate on the day-ahead wholesale electricity prices in Italy, a key reference point for South Europe and especially for countries like Greece and Bulgaria. Additionally, we intend to measure how much specialized knowledge influences the outcomes.

The primary metric used for assessment was the Mean Absolute Percentage Error (MAPE), a widely recognized standard for evaluating time-series predictions. This study confirmed the superior ability of deep learning techniques to more accurately predict the relationship between input and output variables. A traditional machine learning approach yielded a MAPE of 8.84%, while the implementation of a LSTM model significantly improved performance, reducing the MAPE to 5.04%. An unexpected finding was the ineffectiveness of ensemble machine learning methods like Random Forest Regressor and Multilayer Perceptron, which demonstrated weak competitiveness in forecasting electricity prices with a MAPE of 16.76% and 16,61% respectively. Also, the use of a univariate algorithm such as Facebook's Prophet for energy price forecasting results in a large error (lowest MAPE that was achived was 21,29%) due to its dependence on other factors.

The expertise in the domain enabled the creation of an extensive dataset with

168 features, encompassing aspects such as load demand, wind and solar production, fuel costs, internal and external exchanges, RES installed capacity and hydro and thermal production, which theoretically influence electricity price trends. A sensitivity analysis validated the significance of these feature groups in enhancing algorithmic performance. Sequentially adding each category of features group, led to a progressive increase in algorithm efficiency, culminating in a MAPE of 10,89% to 8,84% for traditional algorithms. The same applies to the categories of ensemble machine learning and deep learning algorithms; it was evident that due to their complexity, they require all the available data.

This thesis's findings, when compared to those in [102], indicate that the Italian power exchange is one of the most forecastable in South Europe, as evidenced by the achieved MAPE of 5.04%. Reference [103] also employs Multilayer Perceptrons (MLPs) to predict day-ahead electricity prices and identifies the Serbian and Hungarian markets as one of the most predictables with a MAPE of 9.28% and 8,64%. Following are the Croatian and Bulgarian power exchanges with MAPEs of 17% and 21%, respectively. The predictability of these exchanges may be linked to their developmental stages, with the MGP being the most established and the IBEX (Bulgarian power exchange) being the least developed in the Southeast European region.

5.2 Future Work

Future work could include hybrid models, particularly those employing reinforcement learning or transfer learning, represent another frontier for advancing energy price forecasting methodologies. Hybrid models can harness the strengths of different algorithms, mitigating individual weaknesses and improving overall performance. Reinforcement learning, with its ability to learn optimal actions through trial and error, could optimize decision-making processes in response to market dynamics. Transfer learning, on the other hand, offers the potential to apply knowledge gained from one domain to another, enabling models to draw on a wider range of experiences. For instance, a model trained on another European energy market could be fine-tuned to predict Italian energy prices, accelerating the learning process and potentially uncovering universal patterns that transcend regional boundaries. Together, these advanced techniques could pave the way for more resilient, accurate, and adaptable forecasting systems that are well-suited to the complexities of the energy sector.

References

[1] Chun-Ping Chang Yin Chu. Vertical separation of transmission control and market efficiency in the wholesale electricity market. *The North American Journal of Economics and Finance*, 51 (C), 01 2020.

[2] Ronan Bolton. *Making Energy Markets*. 02 2022.

[3] Diversegy. Diversegy, https://diversegy.com/.

[4] Peter Cappers Sean Murphy. Unpacking the disconnect between wholesale and retail electric rates. 08 2019.

[5] ESO. Eso, https://www.nationalgrideso.com/.

[6] Oludolapo Akanni Olanrewaju Moses Jeremiah Barasa Kabeyi. Sustainable energy transition for renewable and low carbon grid electricity generation and supply, 03 2022.

[7] Luigi Ferraris. Terna: Challenges for the tso. from liberalisation to energy transition. 2, 01 2020.

[8] TERNA Driving Energy. Efficient territorial planning.

[9] Andrea Gilardoni. *The Italian Utilities Industry*. 01 2020.

[10] Trading Economics. Italy electricity price.

[11] enel.com. Introducing enel's new strategic plan: a roadmap for a net zero world, https://corporate.enelx.com/en/stories/2024/01/enel-new-roadmap-for-net-zero-world.

[12] statista.com. Renewable energy in italy - statistics facts, https://www.statista.com/topics/9224/renewable-energy-in-italy/topicoverview.

[13] terna.it. Efficient territorial planning, https://www.terna.it/en/electric-system/efficient-territorial-planning.

[14] Library Of Congress. Italy: New legislation liberalizes domestic electricity market, https://www.loc.gov/item/global-legal-monitor/2022-01-24/italy-new-legislation-liberalizes-domestic-electricity-market/.

[15] GME Gestore Mercati Energetici. price coupling of regions.

[16] Kerstine Appunn. Setting the power price: the merit order effect, https://www.cleanenergywire.org/factsheets/setting-power-price-merit-order-effect.

[17] mercatoelettrico.org. Euphemia: Description and functioning, https://www.mercatoelettrico.org/en/menubiblioteca/documenti/20181212euphemiadescriprio

[18] mercatoelettrico.org. Spot electricity market (mpe) - mgp, mi, mpeg, msd, https://www.mercatoelettrico.org/en/mercati/mercatoelettrico/mpe.aspx.

[19] Bompard Ettore Colella Pietro Huang Tao Imani, Mahmood Hosseini. Data analytics in the electricity market: a systematic literature review. *Energy Systems*, 33, 04 2023.

[20] Alexandra Jonker Amanda McGrath. What is load forecasting?

[21] Rafal Weron Katarzyna Maciejowska, Bartosz Uniejewski. Forecasting electricity prices. 2, 07 2023.

[22] Rafal Weron. *Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach*. 01 2007.

[23] IEEE Spectrum. Novel approaches for forecasting electricity demand researchers offer ways to make more accurate predictions post-covid.

[24] U.S Energy Information Administration. Electricity explained - factors affecting electricity prices.

[25] MILES MCKENNA MICHAEL POLLITT. Power pools: How cross-border trade in electricity can help meet development goals.

[26] www.eia.gov. Electricity explained factors affecting electricity prices.

[27] electric choice. 8 key factors that impact electricity prices.

[28] Tedeschi M. Mohammed K.S. et al. Rao, A. Role of economic policy uncertainty in energy commodities prices forecasting: Evidence from a hybrid deep learning approach. *Computational Economics*, 33, 02 2024.

[29] Shah M. Patel, H. Energy consumption and price forecasting through data-driven analysis methods: A review. *SN Computer Science*, 2, 06 2021.

[30] altexsoft Software rd engineering. What is price forecasting and how is it done.

[31] Seung-Jean Kim, K. Koh, M. Lustig, Stephen Boyd, and Dimitry Gorinevsky. An interior-point method for large-scale l1-regularized least squares. *Selected Topics in Signal Processing, IEEE Journal of*, 1:606 – 617, 01 2008.

[32] scikit learn.org. The k-nearest neighbors algorithms, https://scikit-learn.org/stable/api/sklearn.neighbors.html.

[33] Geegforgeeks. Gaussian processes in machine learning, https://www.geeksforgeeks.org/gaussian-processes-in-machine-learning/.

[34] Carl Edward Rasmussen. *Gaussian Processes in Machine Learning*, pages 63–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[35] scikit learn.org. Randomforestregressor, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.randomforestregressor.html.

[36] XGBoost. Xgboost documentation, https://xgboost.readthedocs.io/en/stable/index.html.

[37] scikit learn.org. Bagging regressor, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.baggingregressor.html.

[38] scikit learn.org. Extra trees regressor, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.extratreesregressor.html.

[39] Google. Images, https://images.google.com/.

[40] Apple. Siri, https://www.apple.com/siri/.

[41] Google. Deepmind, https://deepmind.google/technologies/alphago/.

[42] Machine Learning Mastery. How to develop multilayer perceptron models for time series forecasting, https://machinelearningmastery.com/how-to-develop-multilayer-perceptron-models-for-time-series-forecasting/.

[43] Yuntong Liu, Chunna Zhao, and Yaqun Huang. A combined model for multivariate time series forecasting based on mlp-feedforward attention-lstm. *IEEE Access*, 10:88644–88654, 2022.

[44] Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O. Arik, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecasting, 2023.

[45] Durrani Tariq S Namdari, Alireza. A multilayer feedforward perceptron model in neural networks for predicting stock market short-term trends, 2021.

[46] Jyothi Varanasi and Madanmohan Tripathi. *Electricity Price Forecasting Using LSTM Network and K-Means Clustering by Considering the Effect of Wind Power Generation*, pages 29–41. 01 2022.

[47] Leo Semmelmann, Sarah Henni, and Christof Weinhardt. Load forecasting for energy communities: a novel lstm-xgboost hybrid model based on smart meter data. *Energy Informatics*, 5, 09 2022.

[48] Mariano Ventosa, Álvaro Baıllo, Andres Ramos, and Michel Rivier. Electricity market modeling trends. *Energy Policy*, 33:897–913, 02 2005.

[49] YuanzhangXiao, ChaithanyaBandi, and Ermin Wei. Supply function equilibrium in networked electricity markets, 08 2023.

[50] Markus Burger, Bernhard Graeber, and Gero Schindlmayr. *Managing Energy Risk: An Integrated View on Power and Other Energy Markets*. 01 2007.

[51] Rabin Jana, Shawkat Hammoudeh, and Claudiu Albulescu. Financial modeling, risk management of energy and environmental instruments and derivatives: past, present, and future. *Annals of Operations Research*, 313, 06 2022.

[52] Rene Carmona, Michael Coulon, and Daniel Schwarz. Electricity price modeling and asset valuation: A multi-fuel structural approach. *Mathematics and Financial Economics*, 7:167–202, 05 2012.

[53] Rafał Weron. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International Journal of Forecasting*, 30, 10 2014.

[54] Dogan Keles, Jonathan Scelle, Florentina Paraschiv, and Wolf Fichtner. Extended forecast methods for day-ahead electricity prices applying artificial neural networks. *Applied Energy*, SCI:218–, 01 2016.

[55] AleaSoft. Aleamodel, https://aleasoft.com/aleamodel/.

[56] Nima Amjady and Meysam Hemmati. Energy price forecasting: Problems and proposals for such predictions. *Power and Energy Magazine, IEEE*, 4:20 – 29, 04 2006.

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

[58] Shu Fan, Chuanbo Mao, and Luonan Chen. Next-day electricity-price forecasting using a hybrid network. *Generation, Transmission Distribution, IET*, 1:176 – 182, 02 2007.

[59] Whei-Min Lin, Hong-Jey Gow, and Ming-Tang Tsai. An enhanced radial basis function network for short-term electricity price forecasting. *Applied Energy*, 87:3226–3234, 10 2010.

[60] Liye Xiao, Wei Shao, Meng-Xia Yu, Jing Ma, and Congjun Jin. Research and application of a hybrid wavelet neural network model with the improved cuckoo search algorithm for electrical power system forecasting. *Applied Energy*, 198:203–222, 07 2017.

[61] Ping-Huan Kuo and Chiou-Jye Huang. An electricity price forecasting model by hybrid structured deep neural networks. *Sustainability*, 10:1280, 04 2018.

[62] Lu Peng, Shan Liu, Rui Liu, and Lin Wang. Effective long short-term memory with differential evolution algorithm for electricity price prediction. *Energy*, 162:1301–1314, 11 2018.

[63] Jesus Lago, Fjo De Ridder, and Bart De Schutter. Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. *Applied Energy*, 221:386–405, 07 2018.

[64] Hasnain Iftikhar, Josue Turpo, Paulo Rodrigues, and Javier Linkolk López-Gonzales. Forecasting day-ahead electricity prices for the italian electricity

market using a new decomposition-combination technique. *Energies*, 16:6669, 09 2023.

[65] Manan Shah Harsh Patel. Energy consumption and price forecasting through data-driven analysis methods: A review. *SN Computer Science*, 06 2021.

[66] Tao Hong, Pierre Pinson, Yi Wang, Rafał Weron, Dazhi Yang, and Hamidreza Zareipour. Energy forecasting: A review and outlook. *IEEE Open Access Journal of Power and Energy*, 7:376–388, 2020.

[67] entsoe. entsoe transparency platform, https://transparency.entsoe.eu/.

[68] GME. Italian power exchange, https://gme.mercatoelettrico.org/en-us/home/results/electricity/mgp/results/pun.

[69] Energy Press. Greek italian market coupling, https://energypress.eu/greek-italian-slovenian-intraday-market-coupling-planned-for-sep-21/.

[70] iea.org. Italian energy mix, https://www.iea.org/countries/italy/energy-mix.

[71] ice.com. Dutch ttf natural gas futures, https://www.ice.com/products/27996665/dutch-ttf-gas-futures.

[72] European Commision. Eu emissions trading system (eu ets), https://climate.ec.europa.eu/eu-action/eu-emissions-trading-system-eu-ets$_en$.

[73] ice.com. Eua futures, https://www.ice.com/products/197/eua-futures.

[74] Pelagatti M. Parisio, L. Energy consumption and price forecasting through data-driven analysis methods: A review. *Economia Politica*, 36, 07 2019.

[75] next.com. What is market coupling?, https://www.next-kraftwerke.com/knowledge/market-coupling.

[76] AIEE (ASSOCIAZIONE ITALIANA ECONOMISTI DELL' ENERGIA). The energy price forecast, https://www.aiee.it/the-energy-price-forecast/?lang=en.

[77] Paulo Canas Rodriguesand Javier Linkolk López-Gonzales Hasnain Iftikhar, Josue E. Turpo-Chaparro. Forecasting day-ahead electricity prices for the italian electricity market using a new decomposition—combination technique. *Energies*, 09 2023.

[78] Chang Shen Anna Alberini, Giuseppe Prettico and Jacopo Torriti. Hot weather and hourly electricity demand in italy, 08 2018.

[79] iea.org. Executive summary , https://www.iea.org/reports/italy-2023/executive-summary.

[80] terna.it. Scheduled internal exchange , https://www.terna.it/en/electric-system/transparency-report/scheduled-internal-exchange.

[81] Pietro Colella Mahmood Hosseini Imani, Ettore Bompard and Tao Huang. Impact of wind and solar generation on the italian zonal electricity price. *Energies*, 09 2021.

[82] phys.org. Drought hits italy's hydroelectric plants , https://phys.org/news/2022-06-drought-italy-hydroelectric.html.

[83] Elizabeth Ingram. Hydroelectric generation in italy decreased 37.7https://www.hydroreview.com/business-finance/business/hydroelectric-generation-in-italy-decreased-37-7-in-2022/.

[84] irena.org. Energy profile italy , https://www.irena.org/-/media/files/irena/agency/statistics/statistical$_{p}rofiles/europe/italy_{e}urope_{r}e_{s}p.pdf$.

[85] Mustafa YILDIZ, Kadir Doğanşahin, and Bedri Kekezoglu. Data preprocessing in electrical energy consumption profile clustering studies. 8:1–13, 04 2023.

[86] keras.io. About keras 3, https://keras.io/about/.

[87] Saura. Hyperparameters of random forest classifier, https://www.geeksforgeeks.org/hyperparameters-of-random-forest-classifier/.

[88] Evangelos Spiliotis Fotios Petropoulos. The wisdom of the data: Getting the most out of univariate time series forecasting. *Feature Papers of Forecasting 2021*, pages 478–497, 06 2021.

[89] Jason Brownlee. Deep learning models for univariate time series forecasting, https://machinelearningmastery.com/how-to-develop-deep-learning-models-for-univariate-time-series-forecasting/.

[90] Rafał Weron Florian Ziel. Day-ahead electricity price forecasting with high-dimensional structures: Univariate vs. multivariate modeling frameworks. *Energy Economics*, 02 2018.

[91] Solís-García J. Nepomuceno-Chamorro I.A. Rubio-Escudero C. Vega-Márquez, B. An extensive comparative between univariate and multivariate deep learning models

in day-ahead electricity price forecasting. *16th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2021)*, 09 2021.

[92] facebook.github.io. Forecasting at scale, https://facebook.github.io/prophet/.

[93] facebook.github.io. Prophet: Automatic forecasting procedure, https://github.com/facebook/prophet.

[94] Leonard J. Tashman. Out-of-sample tests of forecasting accuracy: an analysis and review author abstract. *International Journal of Forecasting*, 16 (4):437–450, 2000.

[95] Richard Simon Sudhir Varma. Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7:1471–2105, 2 2000.

[96] José M. Bergmeir, Christoph ; Benítez. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213, 5 2012.

[97] Lakshmi Teja Illuri. Understanding weighted k-nearest neighbors (k-nn) algorithm, https://medium.com/@lakshmiteja.ip/understanding-weighted-k-nearest-neighbors-k-nn-algorithm-3485001611ce.

[98] scikit learn. Randomforestregressor, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.randomforestregressor.html.

[99] analyticsvidhya. A beginner's guide to random forest hyperparameter tuning, https://www.analyticsvidhya.com/blog/2020/03/beginners-guide-random-forest-hyperparameter-tuning/.

[100] Mauro Castelli, Aleš Groznik, and Aleš Popovič. Forecasting electricity prices: A machine learning approach. *Algorithms*, 13:119, 05 2020.

[101] Aurora Poggi, Luca Di Persio, and Matthias Ehrhardt. Electricity price forecasting via statistical and deep learning approaches: The german case. *AppliedMath*, 3:316–342, 04 2023.

[102] G P Girish, Rahul Bhagat, Preeti Honnamane, and Sweta Singh. Ai models for spot electricity price forecasting—a review. pages 97–103, 12 2023.

[103] Aleksandar Dedinec Aleksandra Dedinec. Electricity price forecasting of the south east european power exchanges. *7th International Conference on Information Society and Technology ICIST 2017*, 2017.