UNIVERSITÀ
DI TRENTO

Department of Information Engineering and Computer Science

Master's Degree in
Artificial Intelligence Systems

Final Dissertation

A forecasting system for electricity production and customer demand

Supervisors

Elisa Ricci

Daniele Miorandi

Student

Samuel Bortolin

Academic year 2021/2022

Acknowledgements

Thanks to ...

Contents

Abstract

The abstract is a short summary of the work describing the target, the subject of the thesis, the methodology and the techniques, the data collection and elaboration, the explanation of the reached results and the conclusion. The abstract of the dissertation must have a maximum length of 3 pages and must include the following information:

- context and motivation
- short summary of the main problem you have dealt with
- developed and / or used techniques
- reached results, the personal contribution of the student has to be highlighted

Note: Please note that the approximate number of pages is 70. These 70 pages include:

- table of contents
- abstract
- chapters

Exclude:

- frontispiece (title page)
- acknowledgements
- bibliography
- attachments

1 Introduction

Brief introduction to the work ...

1.1 Problem statement

This is the problem ...

1.2 Approach to the problem

This is the approach ...

1.3 Outline

Here it is written how the thesis is organized ...

2 State of the Art

In this chapter, the current state of the art is analyzed in the context of electricity data representation and time series forecasting methods. In the first section, a brief introduction to the proposed standards for electricity data representation is presented. Subsequently, various technologies presented in the literature for time series forecasting are discussed. In particular, several implementations and use cases are presented. Additionally, an extensive analysis is conducted on two prominent subjects in the research community: Transformers and Automated Machine Learning (AutoML). These topics are thoroughly explored in dedicated subsections. Furthermore, the three use cases of interest, electricity demand forecasting, consumption baseline forecasting, and electricity production forecasting, are treated in more detail in dedicated sections. At the end of this chapter, it will be clear the context around which the proposed system is developed.

2.1 Electricity data representation

In this section, a discussion of electricity data representation and proposed standards is presented. Data issues and frameworks for handling this kind of data are also discussed.

One of the most popular standards for energy data is the Green Button Data[1], which is an industry initiative in response to the 2012 White House call-to-action to provide customers easy and secure access to their energy usage information in a both consumer-friendly and computer-friendly format. The data supported is not limited to electricity but may also include natural gas, and water usage. Customers using this service are able to securely and easily download their own detailed energy usage in a standard format. This possibility allows customers to choose to upload their own energy data to a third-party application or automate the secure transfer to authorized third parties, based on affirmative customer consent and control. It is a very powerful initiative in the U.S. that enables a variety of new services. Services like UtilityAPI are compatible with the Green Button standard providing support for APIs and XML schemas[2].

[78] provided a case study and explained the lessons learned through the roll-out of Green Button electricity, natural gas, and water data-access initiative, in order to make readily available energy and water consumption data for consumers and third-party companies, that can assist customers while ensuring security and privacy of their data. This paper presented a case study using the Green Button standard and the steps taken to ensure data security and privacy while enabling access to those consumption data by the consumer and third parties. Data security and privacy were achieved through the use of the Green Button standard and subsequent implementation by the Green Button Alliance of a compliance-testing program. Considerations and solutions were needed for data in transit, data at rest, and the authorization mechanisms for allowing unregulated third-party companies to interface directly with utilities on behalf of the consumer while ensuring the consumer maintains complete control of what is to be shared and the ability to revoke that sharing at any time.

In [20], Chen et al. studied the quality of electricity consumption data in a smart grid environment. They defined and classified which are common data quality issues that may arise in this field. In particular, the data quality issues related to electricity consumption are classified into three types: noisy data, incomplete data, and outlier data. These three types of data quality issues are discussed. The paper introduced the causes of electricity consumption outlier data and provided a review of the possible detection methods. This is a relevant study since most industrial studies in this field use real-world data that presents these issues.

[22] presented a big-data-based framework for dealing with electricity consumption behavior. It

[1] `https://www.greenbuttondata.org/`

[2] `https://utilityapi.com/docs/greenbutton`

conducted an analysis of the current state-of-the-art methodologies for the extraction of electro-information and it presented the drawbacks of existing modeling strategies for power consumption behavior. The paper proposed a way of integrating multiple-dimensional information into electricity consumption data, such as weather, holiday, and economic level. It dealt also with issues such as anonymization, abnormal detections, and meter failures. The final objective was to conduct an in-depth study for pattern identification, relational analysis, and understanding of the possible actions to perform based on electricity usage.

2.2 Time series forecasting

In this section, a brief introduction to forecasting competitions, the use of cross-validation for the evaluation of time series forecasting methods, and techniques for time series forecasting is presented. Time series forecasting is a classic and well-studied topic. The classical approaches, which are based on statistical methods such as Auto-Regressive Integrated Moving Average (ARIMA), have been joined in recent years by standard Machine Learning (ML) methods and recently also by Deep Learning (DL) methods with the use of neural networks and transformers.

A brief review of forecasting competitions was presented in [49]. These competitions were proposed to promote the development of new solutions and novel techniques in this field. Over time, these competitions have gained significant attention, resulting in a growing interest in time series forecasting and ultimately fostering the development of fascinating and impactful solutions. The first and most influential forecasting competitions were, and currently are, the M-competitions[3]. These competitions promoted the application of the most recent statistical approaches such as ARIMA, ML approaches such as Support Vector Regression (SVR), and DL approaches such as Long Short-Term Memory (LSTM) networks developed over time to be applied to the forecasting field. Issues like the statistical significance of the results, cheating using part of the test sets for training, and reproducibility of the results were addressed over the competitions. Also, other competitions were held such as Sante Fe competitions, the KDD cup, Neural network competitions, Kaggle time series competitions, and Global energy forecasting competitions.

[96] presented an interesting discussion on whether forecasting competition data are representative of reality. This is a very important point since the performance of new forecasting methods is typically evaluated by relying on data from past forecasting competitions. However, due to their many limitations, these datasets might not be indicative. Since obtaining a complete picture of the real world is impossible in practice, the paper proposed to use the M4 competition data as an indication of the real world. This is reasonable since this data set is composed of many series from the business world. The properties of this dataset were compared with past datasets, showing that many popular benchmarks may deviate from reality. The main differences observed were referred to the abnormality of the data, in fact, data from the real world presents relatively more skewed series with outliers, as well as their limited randomness/trend.

In [14], Bergmeir and Benítez presented a study on the use of cross-validation for time series forecasting methods evaluation. They aimed to combine the evaluation of traditional forecasting procedures, on the one hand, and the evaluation of ML techniques on the other hand. In fact, in traditional forecasting, a part from the end of each time series is reserved for testing, and the rest is used for training. Instead, when evaluating ML and other regression methods, often cross-validation is used in the evaluation process without paying much attention to the fact that there are theoretical problems concerning temporal evolutionary effects and dependencies within the data that invalidate the fundamental assumptions of cross-validation, such as to have i.i.d. (independent and identically distributed) data. They suggested that the use of a blocked form of cross-validation for time series evaluation should be the standard procedure, thus using all available information and circumventing the theoretical problems. They also affirmed that the use of cross-validation techniques, together with adequate control for stationarity, led to a more robust model selection.

In [19], Cerqueira et al. studied the application of performance estimation methods to time series forecasting. They stated that the dependency among observations in time series raises some caveats

[3] `https://en.wikipedia.org/wiki/Makridakis_Competitions`

about the most appropriate way to estimate models' performance since cross-validation cannot be applied to this type of data. Results of a comparative study of different performance estimation methods showed noticeable differences among them. In particular, their empirical experiments suggested that blocked cross-validation can be applied to stationary time series. However, when the time series are non-stationary, the most accurate performance estimation methods were out-of-sample methods and in particular the holdout approach repeated in multiple testing periods showed the most accurate estimates.

Subsequently, relevant studies within the context of time series forecasting are outlined, beginning with classical approaches and culminating in contemporary state-of-the-art approaches.

[26] is a paper published in 2006 and it reviewed the research into time series forecasting made from 1982 to 2005. Many relevant methods were presented such as exponential smoothing, ARIMA, state space models, structural models, Kalman filter, regime-switching models, functional-coefficient models, neural networks, and many others also involving the combination of approaches. There were also a lot of relevant studies that presented theoretical concepts such as Seasonality, Forecast evaluation and accuracy measures, and Prediction intervals and densities. The authors concluded by saying that enormous progress has been made in many areas, but that there were a large number of topics that need further development such as multivariate time series forecasting, forecasting methods based on nonlinear models, model selection procedures, robust statistical methods, and improved forecast intervals.

In [2], Nesreen et al. presented a large-scale comparison study for the major ML models adopted for time series forecasting. The models considered were Multi-Layer Perceptron (MLP), Bayesian neural networks, radial basis functions, Generalized Regression Neural Networks (GRNN), K-Nearest Neighbor (KNN) regression, Classification and Regression Trees (CART), SVR, and Gaussian processes. The study revealed significant differences between the different methods and proclaimed as the best two methods on the monthly M3 time series competition data the MLP and the Gaussian process regression.

[13] proposed a review and a comparison of different strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. They also considered the effects of deseasonalization, input variable selection, and combination on the strategies. From experimental results, they figured out that: (i) Multiple-Output strategies were the best-performing approaches, (ii) deseasonalization led to uniformly improved forecast accuracy, and (iii) input selection was more effective when performed in conjunction with deseasonalization.

In [100], Tealab studied the advances in time series forecasting models using Artificial Neural Network (ANN) methodologies. He took into consideration the papers published from 2006 to 2016. From the analysis of the papers, he concluded that, although many studies presented the application of neural network models, few of them proposed new neural network models for forecasting. He found out that these many studies had a similar hybrid methodology that consisted in adjusting a linear time series model, and subsequently using the residuals as the input variables of an ANN model.

[10] proposed a comparative study and analysis of various time series forecasting techniques such as linear regression model, ARIMA, LSTM, and many others. In particular, it explored their limitations and utility for different types of time series data across different domains.

[87] provided a comprehensive literature review of DL studies with a focus on financial time series forecasting implementation. They categorized the studies according to their intended forecasting implementation areas and grouped them based on their DL model choices, such as Convolutional Neural Networks (CNNs), Deep Belief Networks (DBNs), and LSTM.

In [48], Hewamalage et al. presented an empirical study resulting in an open-source software framework for time series forecasting using Recurrent Neural Network (RNN) architectures. They concluded that RNN architectures were capable of directly modeling seasonality when the time series presented homogeneous seasonal patterns. If this is not the case, they recommend a deseasonalization step for achieving better results. They also provided some comparisons against exponential smoothing and ARIMA demonstrating that RNN models even were not perfect, they were good alternatives.

In [16], Lim and Zohren analyzed the main architecture used in both one-step-ahead and multi-horizon time series forecasting. They described how temporal information is incorporated into predic-

tions by each of the models, such as CNNs, RNNs, and networks with attention mechanisms. They also highlighted the recent developments in hybrid DL models, which combine statistical models with neural network components to improve performance.

[71] presented the recent ML advances for time series forecasting. They started by analyzing the linear methods, paying more attention to penalized regressions and ensembles of models. They continued presenting nonlinear methods, including tree-based methods, such as Random Forest and Gradient-Boosted Decision Trees (GBDT), and shallow and deep neural networks, in their feed-forward and recurrent versions. Finally, they also consider ensemble and hybrid models by combining different alternatives.

These reviews have aided in understanding the most commonly utilized methods in time series forecasting and identifying potentially effective solutions for general use cases. Furthermore, specific studies that present distinct methodologies are carefully examined and analyzed.

[120] presented a hybrid model combining ARIMA and ANN models. This was an innovative study of 2003 and was done to take advantage of both strengths of the ARIMA and the ANN models in linear and nonlinear modeling respectively. The experimental results indicated that the combination was an effective way to improve the forecasting accuracy achieved by both models used separately.

[18] proposed to use an approach including multiple Support Vector Machines (SVMs) for time series forecasting. The multiple SVMs that best fit the regions partitioned by a self-organizing feature map were constructed by finding the most appropriate kernel function and the optimal parameters of the different SVMs. Simulation results showed that the proposed multiple SVMs approach achieved significant improvement in the generalization performance in comparison with the single SVMs models. In addition, it was shown that the multiple SVMs were also able to converge faster and use fewer support vectors.

[115] attempted to develop an automatic ANN modeling scheme for time series forecasting. This scheme was based on the GRNN, a special type of neural network. By taking advantage of several GRNN properties (i.e., a single design parameter and fast learning) and by incorporating several design strategies (e.g., fusing multiple GRNNs), they were able to make the proposed modeling scheme to be effective for modeling large-scale business time series.

[81] described a new type of ensemble improving the predictive performance with respect to existing ensembles for time series forecasting. In particular, they proposed a new form of diversity generation that explores some specific properties of time series prediction tasks. Their experiments confirmed that the proposed method for generating diversity was able to improve the performance of the equivalent ensembles with standard diversity generation procedures.

In the study conducted by Sean and Letham [99], they introduced Prophet, a modular regression model that offers interpretable parameters. One of the notable advantages of Prophet is its flexibility in parameter adjustment based on domain knowledge, allowing for intuitive customization based on the characteristics of the time series being analyzed. Additionally, the researchers also outlined a crucial component that measures and tracks forecast accuracy. This feature serves to flag forecasts that should be manually reviewed, allowing incremental improvements in the forecasting process. This capability is vital as it helps in identifying when adjustments are required for the existing model or when an entirely different model may be more suitable for accurate predictions.

In [15], Borovykh et al. presented a method for conditional time series forecasting based on an adaptation of the recent deep convolutional WaveNet architecture. The proposed method took advantage of dilated convolutions for considering a broad history horizon when forecasting. The conditioning process in the study involved the application of multiple parallel convolutional filters to individual time series. This approach facilitated efficient data processing and allowed the exploitation of the correlation structures present among the multivariate time series. Through experimental evaluations, it was demonstrated that the proposed network was particularly suitable for regression-type problems. Notably, the network exhibited the capability to effectively learn dependencies within and between the time series, even in cases where extensive historical time series data was not available. Moreover, the performance of the network surpassed that of linear and recurrent models, highlighting its superior predictive capabilities.

[27] proposed a hybrid system that searches for a suitable function to combine the forecasts of

linear and nonlinear models. The proposed system performed linear and nonlinear modeling of the time series and a data-driven combination that searches for the most suitable function, between linear and nonlinear formalisms, and also the number of models that maximizes the performance of the combination. As a linear model, the ARIMA model is used and as nonlinear models, MLP and SVR were used. Experimental results showed that the proposed hybrid system attains superior performance when compared to both single and hybrid models previously reported in the literature.

In [89], Shen et al. proposed SeriesNet, which is a novel time series forecasting model able to learn features of time series data in different interval lengths. It is composed of a LSTM network and a dilated causal convolution network. The fact that the proposed model could learn multi-range and multi-level features from time series data led to a higher predictive accuracy compared to those models using fixed time intervals.

[92] presented the winning submission of the M4 forecasting competition. The winning approach employed a dynamic computational graph neural network system, which combined a standard exponential smoothing model with advanced LSTM networks within a unified framework. This novel hybrid and hierarchical forecasting method demonstrated superior performance compared to all other models submitted in the competition. The successful integration of these techniques resulted in enhanced forecasting capabilities, showcasing the effectiveness of the proposed approach in surpassing existing methodologies.

In the context of building better pricing modeling and forecasting frameworks to meet difficulties, [97] proposed to combine seasonal and trend decomposition utilizing LOESS (locally estimated scatterplot smoothing) and Prophet methodologies to perform a more accurate and resilient time series analysis of Italian electricity spot prices. The proposed method could assist in enhancing projections and providing a better understanding of the variables driving the data. Experimental results showed that the combination of approaches improved the forecast accuracy and lowered the Mean Absolute Percentage Error (MAPE) performance metric by 18% compared to the Prophet baseline model.

2.2.1 Attention and transformers

In this subsection, an overview of attention-based and transformer approaches is presented, with a focus on time series forecasting applications.

[80] provided an overview of the state-of-the-art attention models proposed. They classified existing attention models according to the following criteria: the softness of attention, forms of input features, input representation, and output representation. They also summarized the network architectures used in conjunction with the attention mechanism and described some applications where attention mechanisms are used to improve performance. Finally, they discussed the interpretability that attention mechanisms provide to the DL models, but also the challenges and prospects of attention models.

[45] presented a review of the developed attention mechanism with a focus on the neural machine translation task. They covered the most adopted attention models and their variants, such as self-attention, soft attention, hard attention, local attention, global attention, additive attention, and multiplicative attention.

[82] analyzed that attention mechanisms have raised significant interest in the research community since they promised relevant improvements in the performance of neural network architectures. In particular, since self-attention was proposed, it has been widely used in transformer-like architectures and has led to significant breakthroughs in many applications. In the work, Pedro and Oliveira performed an objective comparison of several different attention mechanisms for the classification of samples in the Skin Cancer MNIST dataset. The results showed that attention modules only sometimes improved the performance of CNN architectures, but also that this improvement was not consistent in different settings. On the other hand, the results obtained with self-attention mechanisms showed consistent and significant improvements, leading to the best results even in architectures with a reduced number of parameters.

The study presented in [61] introduced an evolutionary attention learning approach for enhancing LSTM models in multivariate time series prediction tasks. The researchers proposed a competitive random search method inspired by evolutionary computation to optimize the configuration of param-

eters within the attention layer. Experimental results demonstrated that the proposed model achieved competitive prediction performance when compared to other baseline methods. This highlights the effectiveness of the evolutionary attention learning approach in improving the predictive capabilities of LSTM models for multivariate time series forecasting.

In [90], Shih et al. analyzed that the standard attention mechanisms in multivariate time series forecasting reviewed just the information at each previous time step from which to select relevant information to generate the outputs. For this reason, these mechanisms were unable to effectively capture temporal patterns across multiple time steps. In the study, they proposed a set of filters designed to extract time-invariant temporal patterns. Additionally, they developed a novel attention mechanism that not only selects relevant time series but also utilizes frequency domain information for multivariate time series forecasting. Finally, they applied the proposed approach to many real-world tasks demonstrating that it was able to achieve state-of-the-art performance in most of them.

In [34], Du et al. proposed a novel temporal attention encoder-decoder model to successfully deal with multivariate time series forecasting. They developed an end-to-end DL structure that combined the classical encoder-decoder learning structure with a temporal attention mechanism. This integrated approach enabled the simultaneous learning of long-term temporal dependency and hidden non-linear correlation features within multivariate temporal data. Experimental results on five multivariate time series datasets showed that the proposed model had the best forecasting performance compared with baseline models, such as SVR, RNN, CNN, LSTM, and Gated Recurrent Unit (GRU).

[47] aimed to predict the energy use of solar-assisted water heating systems using a novel ML approach. They proposed to use a LSTM network enhanced by an attention mechanism and the decomposition of input data into sub-layers. They compared the performance of the proposed approach with a feed-forward neural network, a LSTM network, and an Attention-based LSTM neural network. The experimental results showed that the proposed model outperformed conventional models in this task.

Liu et al. analyzed that the current attention-based recurrent neural networks can effectively represent and learn the dynamic spatiotemporal relationships between exogenous series and target series, but they only perform well in one-step time prediction and short-term time prediction. In [65], they proposed dual-stage two-phase-based RNN (DSTP-RNN) for long-term time series prediction. The DSTP-based structure was designed to enhance spatial correlations between exogenous series. The first phase of the model generated violent but decentralized response weights, while the second phase led to stationary and concentrated response weights. Multiple head attention was then employed on the target series to amplify the long-term dependencies. Experimental results showcased the effectiveness of the proposed model across a range of applications, outperforming nine baseline methods on four datasets from the domains of energy, finance, environment, and medicine. These findings suggest that the DSTP-RNN model holds promise for developing accurate and robust prediction systems in diverse fields.

Qi et al. analyzed that with the development of attention mechanisms, the frequency and time attention information of audio can be fully exploited, and the amplitude properties of audio can also be better integrated with a good fusion module. In [85], they improved existing frequency-temporal attention by extracting the attention information with the frequency-temporal attention and performing an additive fusion of features. Then, they applied attentional feature fusion based on multi-scale channel attention, and finally, temporal dependencies are learned through a self-attention module. Experimental results on four datasets demonstrated that the proposed model outperformed existing state-of-the-art models. These findings highlight the effectiveness of the proposed approach in leveraging attention mechanisms and feature fusion for improved audio analysis and prediction tasks.

In [105], Vaswani et al. proposed the Transformer architecture, based solely on attention mechanisms and fully connected feed-forward networks, with the addition of layer normalization and residual connections. Experimental results on machine translation tasks showed that this groundbreaking model was superior in quality achieving better results than existing methods being more parallelizable and requiring significantly less time to train. Moreover, they also showed that the Transformer is able to generalize to other tasks like English constituency parsing.

In [114], Wu et al. presented a new approach to time series forecasting by developing a novel

method that employs Transformer-based models to forecast time series data. The approach leveraged self-attention mechanisms to effectively capture and understand complex patterns and dynamics from time series data. They created a generic framework that can be applied to univariate and multivariate time series data, as well as time series embeddings. The influenza-like illness (ILI) forecasting was used as a case study. They showed the effectiveness of the proposed approach and that the produced forecasting results are favorably comparable to the state-of-the-art.

[123] discussed the several severe issues of Transformer architecture that prevent it from being directly applicable to long sequence time series forecasting (LSTF), including quadratic time complexity, high memory usage, and the limitation of the encoder-decoder architecture. To address these issues, Zhou et al. designed an efficient transformer-based model for LSTF, named Informer. It had some distinctive characteristics, such as a ProbSparse self-attention mechanism, which achieves O(L log L) in time complexity and memory usage, the self-attention distilling highlights dominating attention by halving cascading layer input, the generative style decoder predicts the long time series sequences in one forward operation rather than a step-by-step way. Extensive experiments on four large-scale datasets demonstrated that Informer significantly outperformed existing methods and provided a new solution to the LSTF problem.

[44] analyzed the state-of-the-art models for multivariate time series forecasting. They discussed that sequence-to-sequence models rely on attention between timesteps, which allows for temporal learning but fails to consider distinct spatial relationships between variables. In contrast, graph neural network methods explicitly model variable relationships, however, often rely on predefined graphs and perform separate spatial and temporal updates without establishing direct connections between each variable at every timestep. The study addressed the presented problems by using a Transformer and translating the multivariate forecasting task into a spatiotemporal sequence formulation so that each Transformer input represented the value of a single variable at a given time. Using this formulation, they were able to let Long-Range Transformers jointly learn interactions between space, time, and value information. The proposed method, called Spacetimeformer, achieved competitive results on benchmarks of different domains while learning fully-connected spatiotemporal relationships purely from data.

In [63], Lim et al. introduced the Temporal Fusion Transformer (TFT), a novel attention-based architecture able to combine high-performance multi-horizon forecasting with interpretable insights into temporal dynamics. The presented TFT relied on recurrent layers for local processing and on interpretable self-attention layers for long-term dependencies. It was able to select relevant features and suppress unnecessary components through a series of gating layers. Experimental results on a variety of real-world datasets demonstrated significant performance improvements over existing benchmarks and they also highlighted some practical interpretability use cases enabled by the TFT architecture. TFT architecture was a breakthrough in time series forecasting and achieved state-of-the-art performance in several tasks. Hereafter, some studies using this architecture are presented.

In [83], the TFT was applied to forecast the trajectory of future vital signs based on time-varying measurements of past vital signs. This application holds significant importance since the deterioration of a patient's condition is usually preceded by several hours of abnormal physiology as indicated by the patient's vital signs. The model was developed using the Songklanagarind critical care dataset, which includes vital sign measurements from 140 patients. Experimental results showed that TFT was able to capture the temporal dynamics of vital signs and can potentially be used to detect irregular patterns in vital sign time series. This suggests that TFT has the potential to assist in the early detection of abnormal physiological patterns and contribute to improved patient care and monitoring in critical care settings.

In [121], a TFT was adopted to predict freeway speed with prediction horizons from 5 to 150 minutes. A traffic speed data set was employed to train and evaluate the TFT prediction model, showcasing the advantages offered by this approach. The TFT prediction performance was compared with several classic traffic speed prediction methods, and the results revealed that the TFT outperformed the other classic models when the prediction horizon is longer than 30 minutes. Moreover, the TFT is also more stable when the prediction horizon is 60 minutes or longer. These findings highlight the effectiveness of TFT in accurately predicting freeway speed and its superiority over traditional

methods, especially being stable also for longer prediction horizons.

[113] used a novel forecasting approach for interpretable wind speed prediction by incorporating variational mode decomposition (VMD), a TFT model, and an evolutionary algorithm. In the proposed approach, VMD was employed to break down the raw wind speed sequence into a set of intrinsic mode functions. Adaptive differential evolution was then used for optimizing several parameters of a TFT allowing it to achieve satisfactory forecasting performance. Empirical studies using real-world wind speed data sets demonstrated that the proposed model outperformed other comparable models in nearly all performance metrics. Moreover, TFT allowed gaining information about the importance ranking of the decomposed wind speed sub-sequences, meteorological data, and attention analysis of different step lengths. These findings underscore the effectiveness and interpretability of the proposed approach in wind speed prediction tasks.

2.2.2 Automated machine learning

In this subsection, an overview of the developed approaches in the AutoML field is presented, with a focus on time series forecasting applications.

[36] emphasized the necessity of automating the process of building good ML models, mainly due to the exponential growth in data volume, which exceeds the capacity of human data scientists. In this study, Elshawi et al. presented a comprehensive survey that focused on the Combined Algorithm Selection and Hyper-parameter tuning (CASH) problem in the ML domain. In addition, they highlighted the importance of automating other steps within the ML pipeline, ranging from data understanding to model deployment. They provided an overview of the state-of-the-art efforts, tools, and frameworks that have been proposed to address these challenges. Finally, the study discussed various research directions and open challenges that need to be tackled to realize the vision and goals of AutoML, including scalability, optimization techniques, time budget management, and data preparation.

[31] observed that data scientists cannot tackle the growing number of challenging tasks due to a lack of expertise and experience across all task domains. To help address this issue, the study provided a survey of ongoing research in the field of Meta-Learning and AutoML. It covered AutoML tools such as TransmogrifAI, Auto-Sklearn, AutoGluon, and NNI. It also summarized possible uses of Meta-Learning for DL, few-shot learning, and automating the ML process.

In [102], Truong et al. conducted a thorough investigation into the current landscape of AutoML tools designed to automate repetitive tasks within ML pipelines. These tasks include data preprocessing, feature engineering, model selection, hyperparameter optimization, and prediction result analysis. performed multiple evaluations of these tools using diverse datasets to assess their performance and compare their respective strengths and weaknesses. The study revealed that while most AutoML tools achieved reasonable results in terms of performance across a wide range of datasets, no single tool managed to outperform all others consistently in all tasks. Among the tools evaluated, H2O AutoML, Auto-Keras, and Auto-Sklearn demonstrated better performance than Ludwig, Darwin, TPOT, and Auto-ml across various evaluations and benchmarks.

[76] reviewed the various AutoML, hyperparameter tuning, and meta-learning approaches available in the literature and pointed out that most of them are neither properly documented nor very clear due to the differences in the approaches. The strengths and drawbacks of the various approaches and their reviews in terms of algorithms supported, features, and implementations are explored.

[46] presented a comprehensive review of the state-of-the-art in AutoML. He et al. introduced AutoML methods covering data preparation, feature engineering, hyperparameter optimization, and Neural Architecture Search (NAS) with a focus on automating the DL pipeline. They summarized the representative NAS algorithms' performance on the CIFAR-10 and ImageNet datasets and discussed relevant topics of the NAS methods such as one/two-stage NAS, one-shot NAS, joint hyperparameter and architecture optimization, and resource-aware NAS. Finally, they discussed some open problems related to the existing AutoML methods for future research such as flexible search space, areas exploration, interpretability, reproducibility, and robustness.

In [55], Karmaker et al. proposed a classification system for AutoML systems, using a tier schematic to distinguish systems based on their level of autonomy. They described what an end-to-end ML pipeline actually looks like and analyzed which subtasks have been automated and which are done

manually. In fact, most AutoML systems still require human involvement in some steps, including understanding the attributes of domain-specific data, defining prediction problems, creating a suitable training dataset, and selecting a promising ML technique. These steps often require a prolonged back-and-forth that makes the process inefficient and keeps AutoML systems from being truly automatic. They introduced the proposed level-based taxonomy for AutoML systems and defined each level according to the scope of automation support provided. Finally, they discussed the challenges that stand in the way of automating the whole end-to-end ML pipeline.

In [21], Chen et al. described AutoML as a bi-level optimization problem, where one problem is nested within another to search for the optimum in the search space. They reviewed the current developments of AutoML presenting the state-of-the-art techniques and frameworks in terms of three main categories: automated feature engineering (AutoFE), automated model and hyperparameter tuning (AutoMHT), and Automated Deep Learning (AutoDL). They concluded by presenting the open challenges of AutoML such as the lack of authoritative benchmarks, efficiency, design of search spaces, and interpretability.

In 2015, Feurer et al. stated that to be effective in practice, ML systems need to automatically choose a good algorithm and feature preprocessing steps for a new dataset at hand, and also set their respective hyperparameters. Starting from existing work on efficient Bayesian optimization methods, in [40] they presented a robust new AutoML system based on the scikit-learn framework: Auto-Sklearn. Auto-Sklearn incorporated 15 classifiers, 14 feature preprocessing methods, and 4 data preprocessing methods, resulting in a structured hypothesis space with 110 hyperparameters. One key improvement of Auto-Sklearn was its ability to leverage past performance on similar datasets, taking into account this information during the optimization process. Furthermore, the system constructed ensembles from the evaluated models. Experimental results on a wide range of more than 100 diverse datasets demonstrated that Auto-Sklearn substantially outperformed the previous state of the art in AutoML.

In [54], Jin et al. proposed a novel framework for efficient NAS enabling Bayesian optimization to guide the network morphism, which essentially keeps the functionality of a neural network while changing its neural architecture. The framework, called Auto-Keras, employed a neural network kernel and a tree-structured optimization algorithm to facilitate efficient exploration of the search space. Through extensive experiments conducted on real-world benchmark datasets, they demonstrated the superior performance of the developed framework over the state-of-the-art NAS methods.

In [35], Dyrmishi et al. presented a methodology and a framework for using Meta-Learning techniques to develop new methods that serve as effective decision support for the AutoML process. Meta-Learning allows learning from previous experience gained during applying various learning algorithms on different types of data and helps reduce the time needed to learn new tasks. In particular, they used Meta-Learning techniques to answer several crucial questions for the AutoML process such as which classifiers are expected to be the best performing on a given dataset, whether it is possible to predict the training time of a classifier, and which classifiers are worth investing a larger portion of the time budget to improve their performance by tuning them. In the Meta-Learning process, they used 200 datasets with different characteristics and 30 classifiers from Weka and Scikit-learn libraries. As a result, this method allowed Meta-Models to be obtained in a fully automated way.

[43] introduced an open-source benchmark framework for comparing different AutoML systems following best practices and avoiding common mistakes. The framework is extensible both in terms of AutoML frameworks and tasks. They used the framework to conduct a thorough comparison of 4 AutoML systems (Auto-WEKA, Auto-Sklearn, TPOT, and H2O AutoML) across 39 datasets. They analyzed the results and highlighted the need for further AutoML research. In fact, on some datasets, none of the frameworks outperformed a Random Forest within 4 hours, and high-dimensional or highly multi-class problems were often challenging for AutoML frameworks.

In [103], Vaccaro et al. reviewed some ML models and methods proposed in the literature to analyze their strengths and weaknesses. Then, they proposed their use, alone or in combination with other approaches, to provide possible valid AutoML solutions. They analyzed these solutions from a theoretical point of view and evaluated them empirically on three Atari games. The objective of the study was to identify what could be some promising ways to create effective AutoML frameworks able to replace the human expert as much as possible and thereby make the process of applying ML

approaches to typical problems of specific domains easier. The research provided valuable insights into potential directions for future work in the field of AutoML. The findings contribute to the ongoing efforts to develop robust and efficient AutoML frameworks, paving the way for advancements in automating machine learning processes and reducing the dependence on human expertise.

Nguyen et al. investigated that most AutoML-based Bayesian optimization approaches convert the AutoML optimization problem into a Hyperparameter Optimization (HPO) problem, i.e., by modeling the choice of algorithms as an additional categorical hyperparameter. They pointed out that using this approach algorithms and their local hyper-parameters are referred to at the same level, and this makes the initial sampling less robust. In [79], they attempted to formulate the AutoML optimization problem as a Bayesian optimization problem instead of transferring it into a HPO problem. They proposed a novel initial sampling approach to maximize the coverage of the AutoML search space to help Bayesian optimization construct a robust model. They tested this approach on 2 independent scenarios of AutoML with 2 operators and 6 operators over 117 benchmark datasets. Experimental results showed that the performance of Bayesian optimization was significantly improved by using the proposed sampling approach.

In [39], Feurer et al. introduced a new AutoML approach, named PoSH (Portfolio Successive Halving) Auto-Sklearn, which enabled AutoML systems to work well on large datasets under rigid time limits by using a meta-learning technique and a bandit strategy for budget allocation. They also studied how to let AutoML explore the design space and automatically select the best configuration by itself. These changes combined give rise to the next generation of their original Auto-Sklearn framework, called Auto-Sklearn 2.0. They verified the improvements of these additions in an extensive experimental study on 39 AutoML benchmark datasets and compared the results to other popular AutoML frameworks and Auto-Sklearn 1.0, showing that the relative error is reduced by up to a factor of 4.5, and yielding performance in 10 minutes that was better than what Auto-Sklearn 1.0 achieved within an hour.

In [125], Zimmer et al. presented Auto-PyTorch, a comprehensive framework that enables fully AutoDL by jointly optimizing network architecture, training pipelines, and hyperparameters. Auto-PyTorch utilizes multi-fidelity optimization techniques along with portfolio construction for warm starting and ensembling of Deep Neural Networks (DNNs), achieving state-of-the-art performance on many tabular benchmarks. One notable contribution of the study is the introduction of a new benchmark on learning curves for DNNs. This benchmark provides a valuable evaluation criterion for assessing the effectiveness of AutoDL frameworks. The authors conducted extensive experiments on typical AutoML benchmarks, demonstrating that Auto-PyTorch outperformed several state-of-the-art competitors in terms of performance.

In their work, the authors of [38] conducted a benchmark study to evaluate the performance of eight popular open-source AutoML tools: Auto-Keras, Auto-PyTorch, Auto-Sklearn, AutoGluon, H2O AutoML, rminer, TPOT, and TransmogrifAI. They also selected twelve widely used OpenML datasets to serve as benchmarks for different machine learning tasks, including regression, binary classification, and multi-class classification. The study focused on comparing the predictive scores and computational effort of the AutoML tools. The best predictive results achieved by these tools were compared with the best public results from OpenML. The comparison revealed that the AutoML tools consistently achieved competitive results, outperforming the best models from OpenML in five of the datasets. These findings highlight the potential of AutoML tools in fully automating the process of ML algorithm selection and tuning.

In the following, some studies with a focus on the utilization of AutoML for time series forecasting applications are presented.

In [52], a data augmentation method was presented to enhance the performance of neural networks in time series forecasting tasks, especially when limited data is available. The proposed method, known as Augmented-Neural-Network, incorporated forecasts from statistical models to achieve competitive results on intermediate-length time series. The study demonstrated that combining data augmentation with AutoML techniques like NAS can lead to the discovery of suitable neural architectures for specific time series. By applying this approach, significant improvements in forecasting accuracy were observed on a COVID-19 dataset, with an improvement of approximately 20% compared to neural networks

that did not utilize augmented data.

[6] conducted experiments on time series forecasting using ML, DL, and AutoML techniques. The datasets used in the experiments were quantitative data of the real prices of the currently most used cryptocurrencies. The results showed that AutoML for time series is still in the development stage and needs more study to be the main solution to adopt since it was unable to outperform manually designed ML and DL models.

In recent years, there has been significant improvement in the efficiency of AutoDL systems. However, there has been limited focus on AutoDL frameworks specifically designed for time series forecasting. In [29], Deng et al. proposed an efficient approach to jointly optimize the neural architecture and hyperparameters of the entire data processing pipeline for time series forecasting. Unlike traditional NAS search spaces, the authors designed a novel NAS space that encompasses various state-of-the-art architectures. This enables efficient macro-search over different DL approaches specifically tailored for time series forecasting. To efficiently explore this large configuration space, the researchers employed Bayesian optimization with multi-fidelity optimization techniques. The study conducted empirical investigations on various forecasting datasets using different budget types, which enabled efficient multi-fidelity optimization. Moreover, the proposed system, named Auto-PyTorch-TS, was compared against several established baselines. The results demonstrated that Auto-PyTorch-TS significantly outperformed the baselines across multiple datasets, highlighting its effectiveness in time series forecasting tasks.

2.3 Electricity demand forecasting

In this section, the techniques for electricity demand forecasting are presented. Most of the presented techniques are considered very simple nowadays and usually rely on large aggregated data, both on a high number of people considered (e.g., the consumption generated by an entire country) or on a very large temporal aggregation (up to 1 year aggregated data). Our use case is limited to the customers of a small company, from 2 to 4 thousand customers, and it requires forecasts for a one-month time horizon on an hourly basis.

In [91], a comprehensive review of electricity demand forecasting techniques was presented. The authors categorized electricity demand forecasting into three main categories: short-term forecasts (typically ranging from one hour to one week), medium-term forecasts (typically ranging from a week to a year), and long-term forecasts (exceeding a year). Based on the studies reviewed, electricity demand forecasting techniques were classified into three major groups: Traditional Forecasting techniques (such as regression methods, exponential smoothing, and iterative reweighted least-squares), Modified Traditional Techniques (including adaptive demand forecasting, Auto-Regressive (AR), Auto-Regressive Moving Average (ARMA), ARIMA, and SVM), and Soft Computing Techniques (such as Genetic Algorithms (GAs), fuzzy logic, neural networks, and knowledge-based expert systems). From the work, it can be inferred that demand forecasting techniques based on soft computing methods were gaining significant advantages thanks to their effectiveness. Moreover, there was a clear trend towards hybrid methods that combine two or more of these techniques to enhance forecasting accuracy and performance.

[98] investigated the use of weather ensemble predictions in electricity demand forecasting for a period ranging from 1 to 10 days ahead. They proposed a weather ensemble prediction by considering 51 scenarios for a weather variable. For each scenario, they produced a scenario for the weather-related component of electricity demand. The results showed that the average of the demand scenarios is a more accurate demand forecast than that produced using traditional weather forecasts. The mean of the 51 scenarios is mathematically equivalent to taking the expectation over the weather-related component of the demand probability density function. They also used the distribution of the demand scenarios to estimate the demand forecast uncertainty.

In [73], Mirasgedis et al. focused on incorporating weather data into models for mid-term electricity demand forecasting. They studied the daily and monthly electricity demand. They observed that the monthly model outperformed the daily model thanks to its higher level of aggregation. However, they also noted that the influence of weather on electricity demand is less effectively captured at a more granular level, as aggregated models may not fully account for the impact of unusual or extreme

weather conditions on electricity consumption. Through their analysis, the authors identified several key weather parameters that significantly affect electricity consumption in the Greek interconnected power system. These parameters include the temperature of the day being forecasted, the temperature of the two preceding days, and the relative humidity. These variables were found to be the most important factors in determining electricity demand variations. By incorporating weather data into their models, Mirasgedis et al. aimed to improve the accuracy of mid-term electricity demand forecasts, considering the influence of weather conditions on electricity consumption patterns.

[86] proposed a first use of a neural network with the backpropagation learning algorithm for Lao state yearly electricity demand forecasting. They compared it with a regression analysis model showing the higher effectiveness and the great potential of neural networks in this task.

[30] proposed two models for short-term Singapore electricity demand forecasting: the multiplicative decomposition model and the Seasonal ARIMA (SARIMA) model. Results showed that both models can accurately predict the short-term Singapore demand and that the Multiplicative decomposition model slightly outperformed the SARIMA model.

In [126], an empirical study was conducted to develop electricity demand forecasting models using publicly available data and three ML models. The study aimed to compare the performance of these models using different evaluation metrics. The data was composed of several measurements of the electricity market in Turkey from 2011 to 2016 and was available for different time granularities (from hourly to yearly aggregated). According to the best result in terms of MAPE, electricity demand was predicted with a 1.4% error using the Random Forest model. Overall, the study demonstrated the effectiveness of machine learning techniques, particularly the Random Forest model, in accurately predicting electricity demand.

In [4], Al-Musaylh et al. conducted a study on short-term electricity demand forecasting using MARS (Multivariate Adaptive Regression Spline), SVR, and ARIMA models. The study utilized aggregated demand data from Queensland, Australia. The findings indicated that the MARS and SVR models were more suitable for short-term electricity demand forecasting compared to the ARIMA model. The ARIMA model, with its linear formulation, exhibited poorer performance across all forecasting horizons, resulting in high forecast errors. The study highlighted that the MARS models offered a powerful and efficient forecasting framework. These models demonstrated a balance between complexity and accuracy, outperforming the SVR models, suggesting that MARS can be considered an effective approach for short-term electricity demand forecasting.

In [69], the authors proposed a method for forecasting building energy consumption in some provinces of southern China using Support Vector Regression (SVR). The aim of the study was to enhance the reliability of SVR in predicting building energy consumption. To achieve this, the method incorporated multiple parameters as inputs, including weather data such as the yearly mean outdoor dry-bulb temperature, relative humidity, and global solar radiation. Additionally, economic factors such as the ratio of urbanization, gross domestic product, household consumption level, and total area of the structure were considered. By including these diverse parameters as inputs to the SVR model, the study demonstrated the presence of improvements in the accuracy and reliability of building energy consumption forecasts.

In [101], a RNN-based approach for forecasting Turkish electricity load was proposed. The study utilized different types of RNN architectures, including LSTM networks and GRU networks. The resulting 0.71% MAPE of their approach achieved better results than existing methods based on ARIMA and ANNs on Turkish electricity load forecasting which achieved 2.6% and 1.8% MAPE respectively. The findings suggest that the RNN-based approaches provide superior accuracy in predicting electricity load compared to traditional methods like ARIMA and ANNs.

In [33] a novel hybrid forecasting system was successfully developed. It was composed of four modules: data preprocessing module, optimization module, forecasting module, and evaluation module. In the data preprocessing module, a signal processing approach was employed to decompose, reconstruct, identify, and mine the primary characteristics of the electrical power system time series. Optimization algorithms were employed to optimize the parameters of these individual models in the optimization and forecasting modules. Experimental results showed that the hybrid system can be able to satisfactorily approximate the actual value of the electrical power system time series. This is

an interesting study from which to take inspiration for the system modeling structure, which can be intended for general-purpose and not just for the electrical power use case.

[57] proposed a novel model called recurrent inception CNN (RICNN) that combines RNN and 1-dimensional CNN (1-D CNN). The RICNN model utilizes a 1-D convolution inception module to calibrate the prediction time and the hidden state vector values obtained from nearby time steps. This module optimizes the network by incorporating the prediction time generated by the RNN and the nearby hidden state vectors. To evaluate the effectiveness of the RICNN model, the authors conducted experiments using power usage data from three large distribution complexes in South Korea. The results showed that the RICNN model outperformed benchmark models such as MLP, RNN, and 1-D CNN in daily electric load forecasting, specifically for 48-time steps with a 30-minute interval. The findings suggest that the RICNN model might be suitable for our specific use case. However, its effectiveness on a wider time horizon would require further investigation.

In [12], Bedi and Toshniwal introduced a DL-based framework, called D-FED, to address the challenge of forecasting electricity demand while capturing long-term historical dependencies. Existing methods typically focused on short-term dependencies and failed to consider long-term patterns. The D-FED approach utilizes a LSTM network combined with a moving window-based multi-input multi-output mapping approach of active learning. The study applied the D-FED framework to the electricity consumption data of Union Territory Chandigarh in India. To evaluate its performance, the predictions generated by the D-FED model were compared with those of other models, including ANN, RNN, and SVR. By employing the D-FED framework, the authors demonstrate improved forecasting accuracy compared to the other models examined.

In [75], Muzaffar and Afshari focused on electricity load forecasting by incorporating exogenous variables such as temperature, humidity, and wind speed. They employed a LSTM network to model the load data and its dependencies on these external factors. The study demonstrated that LSTM outperformed the other traditional methods such as ARMA, SARIMA, and ARMA with exogenous inputs (ARMAX) reducing the percentage of errors in forecasting the load time series. One notable advantage of LSTM highlighted in the study is its ability to learn and capture seasonality patterns and trends from the data, without the need for explicitly extracting these features beforehand. This capability of LSTM contributes to its improved forecasting accuracy compared to traditional methods.

In [112], Wen et al. presented a DL model for forecasting the load demand of aggregated residential buildings at an hourly resolution, taking into account the complexity and variability of the load. The study utilized hourly-measured residential load data from Austin, Texas, to evaluate the effectiveness of the proposed model. The forecasting error was quantitatively assessed using various metrics. The proposed model, called DRNN-GRU, was employed in the study. This model incorporates GRU within a deep RNN architecture. Notably, the model assumes access to future weather data to make accurate forecasts. However, it is important to consider that weather conditions can introduce uncertainty, especially over short to medium-term periods. The results demonstrated that the proposed DRNN-GRU model achieved higher accuracy in forecasting both the aggregated and disaggregated load demand of residential buildings compared to conventional methods. The study highlighted the improved performance of the DL model in capturing the complexities and variabilities inherent in residential load demand.

[23] presented a robust framework for short-term electrical load forecasting that is capable of capturing variations in building operation, irrespective of building type and location. The researchers explored nine different hybrids of RNNs and clustering techniques. The framework was tested on five commercial buildings representing different types: academic, research laboratory, office, school, and grocery store. The load forecasting results demonstrated that the DL algorithms implemented in the study achieved a 20-45% improvement in load forecasting performance compared to the current state-of-the-art methods for both hour-ahead and 24-ahead load forecasting. Several important findings emerged from the study. Firstly, the use of hybrid DL algorithms required as little as one month of data to deliver satisfactory hour-ahead load prediction. Secondly, when 15-minute resolution data was available, it led to a 30% improvement in hour-ahead load forecasting accuracy. Lastly, the formulated methods were found to be robust against weather forecasting errors, indicating that they can effectively handle uncertainties associated with weather predictions. The study provides valuable

insights regarding the quantity and granularity of data required for accurate short-term load prediction within the specified time range. However, it should be noted that these findings may not necessarily generalize to longer prediction ranges beyond the scope of the study.

In [107], Wang et al. proposed a novel approach for predicting periodic energy consumption using a LSTM network. The authors highlighted that the approach was unique because most general forecasting methods did not explicitly consider the periodic nature of energy consumption. The proposed method leverages the autocorrelation graph of real industrial data to extract hidden features. Experiments were conducted using a cooling system, specifically focusing on one-step-ahead forecasting. The performance of the LSTM model was compared against several traditional forecasting methods, including the ARMA model, the Auto-Regressive Fractional Integrated Moving Average (ARFIMA) model, and the backpropagation neural network (BPNN). The results showed that the LSTM model outperformed the traditional methods in terms of Root Mean Square Error (RMSE). Specifically, the RMSE of the LSTM model was found to be 19.7%, 54.85%, and 64.59% lower than that of the BPNN, ARMA, and ARFIMA models, respectively, on the test data. Additionally, the authors demonstrated that the proposed algorithm exhibited the highest generalization capability among the tested models.

In [110], Wang et al. proposed a stacking model for electricity load forecasting that combines the strengths of various basic prediction algorithms. The goal of the stacking model is to leverage different perspectives and approaches to improve the accuracy and generalization of the final prediction model. The study utilized load data obtained from two educational buildings located in Tianjin, China. These buildings consisted of classrooms for students and offices for university staff. The dataset was used as a case study to evaluate the performance of the proposed stacking model. Experimental results showed that the stacking method outperformed other tested machine learning models, including Random Forest, GBDT, Extreme Gradient Boosting (XGBoost), SVM, and KNN. The stacking model exhibited superior accuracy, generalization, and robustness in comparison to these alternative models.

In [95], Somu et al. introduced a deep learning framework called kCNN-LSTM for accurate building energy consumption forecasting. The framework incorporates three key components for leveraging the energy consumption data:

1. k-means clustering, this component performs cluster analysis on the energy consumption data to identify patterns and trends;

2. CNN, this component is used to extract complex features from the energy consumption data. It captures non-linear interactions and dependencies that influence energy consumption;

3. LSTM network, this component is responsible for handling long-term dependencies in the time series data.

The performance of the kCNN-LSTM model was evaluated by comparing it with the k-means variant of state-of-the-art energy demand forecast models. Evaluation metrics such as Mean Square Error (MSE), RMSE, Mean Absolute Error (MAE), and MAPE were used to assess the accuracy of the energy consumption demand forecasts. The experimental results demonstrated that the kCNN-LSTM model outperformed other models providing more accurate energy consumption demand forecasts.

In [62], a TFT model was proposed to forecast short-term loads. The TFT included a sequence-to-sequence model, which processes the historical and future covariates to enhance the forecasting performance. A Gated Residual Network (GRN) is applied to drop out unnecessary information and improve the efficiency of the model. The proposed method is tested on anonymized data from a university campus with a time resolution of 30 minutes. The anomalies and missing data (around 8.85% of the total data) are imputed with the KNN imputation method. The testing results demonstrate the effectiveness of the proposed method achieving a MAPE of less than 5%. Overall, the study presents an interesting work on a TFT model, which should be extended to obtain great performance also with higher prediction horizons in order to be applicable to the specific use case treated in this thesis.

[59] proposed an improved version of the TFT model, called ITFT, aiming to enhance the accuracy and comprehensiveness of forecasts for hourly load time series. The approach involved reconstructing the hourly load data into multiple day-to-day load time series at different hours. To improve the model's efficiency in capturing long-term dependencies, the ITFT model replaced the LSTM component with a GRU. Additionally, the authors incorporated quantile constraints and prediction interval

penalty terms into the original quantile loss function to prevent quantile crossover and generate more compact prediction intervals. The study demonstrated that the proposed ITFT method provided a comprehensive explanation and substantially enhanced the reliability and compactness of probabilistic load forecasting compared to other widely used methods such as Quantile Regression Neural Network (QRNN) and Temporal Convolutional Network (TCN).

2.4 Consumption baseline forecasting

In this section, the techniques for consumption baseline forecasting are presented. As demonstrated by most of the following papers analyzed, computing forecasts for a single customer is a more complicated use case compared to energy demand forecast over a customer base. Single customer data is more noisy and very few works consider a single habitation as the source of data in their studies, most of them consider commercial buildings, offices, or schools which present data with a reduced level of noise.

The S3C EU research project developed and tested different guidelines and tools, of particular interest for this use case is the guideline on how to create a consumption baseline[4]. In particular, they defined the baseline as the reference used to assess the effects of the demand response of a given consumer or set of consumers. The demand response effect is defined as the difference between the metered consumption and the baseline calculation. They explained that the baseline calculation method consists of three criteria: i) data selection method, ii) estimation method, and iii) result adjustment. They pointed out that the combination of these criteria depends on user consumption, weather dependency (including seasonal behavior), and load behavior and should all together fit the user consumption pattern.

In [28], Deb et al. compared multiple time series forecasting techniques for building energy consumption. The evaluated methods included ANN, ARIMA, SVM, Case-Based Reasoning (CBR), Fuzzy time series, Grey prediction model, Moving average and exponential smoothing (MA & ES), and KNN prediction method. The researchers also examined hybrid models, which involve combining two or more forecasting techniques. They found that the hybrid models exhibited the highest effectiveness in time series energy forecasting for individual buildings.

[7] aimed to compare the prediction capabilities of five different intelligent system techniques in forecasting the electricity consumption of an administration building. These five techniques were Multiple Regression (MR), Genetic Programming (GP), ANN, DNN, and SVM. The prediction models were developed based on five years of observed data and additional parameters such as solar radiation, temperature, wind speed, humidity, and weekday index. The weekday index was demonstrated as an important parameter that made it possible to differentiate between working and non-working days. Experimental results suggested that ANN performed better than all other four techniques with a MAPE of 6% whereas MR, GP, SVM, and DNN had a MAPE of 8.5%, 8.7%, 9%, and 11%, respectively.

[37] investigated the performance of different strategies for multi-step ahead building energy predictions. The results of the study showed the potential of recurrent models in short-term building energy forecasting. This study provided useful references for developing advanced DL models for practical applications.

In [66], Lusis et al. investigated the impact of calendar effects, forecasting granularity, and the length of the training set on the accuracy of day-ahead load forecasts for residential customers. The study compared the performance of regression trees, neural networks, and SVR techniques. The results showed that regression trees, neural networks, and SVR produced similar average RMSE values. However, statistical analysis revealed that the regression trees technique was significantly superior in terms of accuracy. The inclusion of historical load profiles with daily and weekly seasonality, along with weather data, reduced the importance of explicit calendar effects in the load forecast. The study also found that using one year of historical data was sufficient to develop an effective load forecast model for residential customers. Further expansion of the training dataset did not yield significant

[4] https://www.smartgrid-engagement-toolkit.eu/fileadmin/s3ctoolkit/user/guidelines/GUIDELINE_HOW_TO_CREATE_A_CONSUMPTION_BASELINE.pdf

improvements in forecast accuracy. Additionally, the study demonstrated that forecast errors could be reduced by employing a coarser forecast granularity. That is expected since aggregating data over longer time intervals, such as daily forecasts instead of hourly forecasts, reduces data variability and improves the quality of the results.

In [84], Platon et al. developed predictive models by using ANN and CBR for producing hourly electricity consumption forecasts for a building. CRB is based on the concept that the current trend of the building's electrical use can be approximated using past trends occurring at similar conditions. The experimental results of the study indicated that ANN outperformed CBR in electricity consumption forecasting. This finding suggests that CBR alone is not sufficient for accurate and reliable electricity consumption predictions. The superiority of ANN over CBR highlights the importance of using more advanced techniques like ANN to achieve better forecasting performance in this context.

In [53], Jie et al. proposed a baseline load forecasting and optimization method based on non-demand-response factors, considering the effects of non-demand-response factors on customer load characteristics and customer baseline load (CBL) forecasting. The proposed method combines non-demand-response factors mining, similar days selecting, and CBL calculating. A combined calculation model is adopted to predict the CBL. The case study reveals the greater accuracy of this method compared to average, linear regression, and neural network methods.

Forecast at the household level is also getting more and more popular in smart building control and demand response programs. This popularity inspired Dong et al. to develop in [32] a hybrid model to address the problem of residential hour and day-ahead load forecasting through the integration of data-driven techniques with forward physics-based models. They evaluated five different ML algorithms: ANN, SVR, least-square SVM (LS-SVM), Gaussian process regression (GPR), and Gaussian mixture model (GMM). They applied these models to four residential data sets obtained from smart meters. A subdivision of air conditioning (AC) consumption and not-AC was possible and this led to better results with respect to considering the total consumption. The final results showed that the hybrid model led to improvements compared to the other ML algorithms for both hour-ahead and 24-h ahead predictions.

In [74], Mocanu et al. focused on two newly developed stochastic models, namely Conditional Restricted Boltzmann Machine (CRBM) and Factored Conditional Restricted Boltzmann Machine (FCRBM), for time series prediction of energy consumption. The assessment of the two models was conducted on a benchmark dataset comprising nearly four years of one-minute resolution electric power consumption data collected from an individual residential customer. The authors compared the performance of CRBMs and FCRBMs with that of ANN. The results showed that as the prediction horizon increased, FCRBMs and CRBMs demonstrated greater robustness, with their prediction errors typically being half that of ANN. Additionally, the experiments revealed that all methods performed better when predicting the aggregated active power consumption, as opposed to predicting the demand of intermittent appliances (e.g., electric water heater) recorded from sub-meterings.

In [5], a robust ensemble model was proposed to forecast day-ahead mean daily electricity consumption on the household level. The proposed ensemble learning strategy utilized a two-stage resampling plan, which generated diversity-controlled but random resamples that were used to train individual ANN members. Experimental results on a case study showed that the proposed ensemble was able to generate better forecasts compared to ANN models and the Bagging ensemble.

To counter the high nonlinearity between inputs and outputs of building energy consumption prediction models, in [122] a novel vector field-based SVR method was proposed. The proposed method utilized multi-distortions applied to the sample data space or high-dimensional feature space, which was mapped by a vector field. By exploring these distortions, the optimal feature space was identified, where the high nonlinearity between inputs and outputs can be effectively approximated by linearity. To evaluate the effectiveness of the method, a large office building located in a coastal town in China was selected as a case study. The summer hourly cooling load data of the building were used as the energy consumption data for the analysis. The experimental results demonstrated that the proposed vector field-based SVR method achieved high accuracy, generalization ability, and robustness in predicting building energy consumption. By effectively addressing the nonlinearity between inputs and outputs, the method provided a reliable approach for accurate and robust energy consumption

prediction in buildings.

In [88], Shao et al. studied and analyzed the energy consumption of hotel buildings by developing a SVM model. The SVM model took as input variables the weather parameters and operating parameters of the hotel air-conditioning system. They selected as the kernel function the RBF (Radial Basis Function) kernel function and optimized the parameters of the kernel for finding the best accuracy for the model predictions. The R^2 (coefficient of determination) of the final model prediction in the case study was 0.94. This use case is different from the standard single-building forecasts since a hotel includes different rooms and aggregates the consumption over them, then it is more influenced by periods of the year where there could be more or fewer customers that require different levels of demand. Moreover, in this case, hotel parameters like the air conditioning system are available and help in reducing the overall forecast error.

[111] introduced a probabilistic load forecasting method designed to address the variability and uncertainty associated with future load profiles for individual consumers. The proposed method utilizes a Pinball loss-guided LSTM network, which is capable of capturing both long-term and short-term dependencies within the load profiles. The performance of the proposed method is evaluated on load forecasting tasks for both residential and commercial consumers. Comparative experiments are conducted, comparing the proposed method against traditional forecasting methods such as QRNN, GBDT, and traditional LSTM. The experimental results across different customers demonstrate that the proposed method outperforms the traditional methods in terms of forecasting accuracy and uncertainty estimation.

[17] aimed to use DL-based techniques for day-ahead multi-step load forecasting in commercial buildings. The authors proposed and formulated RNN and CNN models in both recursive and direct multi-step manners. To evaluate the performance of the DL models, they were compared with the SARIMA model with exogenous inputs (SARIMAX), which is a traditional forecasting method commonly used in this context. Among the DL models tested, the gated 24-hour CNN model, implemented in a direct multi-step manner, achieved the best performance. It significantly improved the forecasting accuracy by 22.6% compared to the SARIMAX model. This demonstrated the superiority of DL models in day-ahead multi-step load forecasting for commercial buildings.

In [58], Kim and Cho proposed a CNN-LSTM neural network architecture for predicting housing energy consumption. This architecture combines the strengths of CNNs and LSTM networks to effectively extract spatial and temporal features. The CNN layer in the proposed architecture is capable of capturing the relationships between various variables that influence energy consumption. It extracts relevant features by considering the spatial context of these variables. On the other hand, the LSTM layer captures the temporal dynamics and irregular trends present in the time series data. This combination allows the model to capture both the spatial and temporal aspects of housing energy consumption. The CNN-LSTM method demonstrated excellent prediction performance for housing energy consumption, outperforming conventional forecasting methods. It achieved the lowest RMSE compared to other methods when applied to the dataset on individual household power consumption. The model was able to effectively predict complex patterns of electric energy consumption at different time resolutions, including minute, hour, day, and week intervals. It consistently outperformed other methods in all these cases. The authors also mentioned that incorporating household characteristics such as occupancy and resident behavior could potentially have a significant impact on predicting electric energy consumption. Considering these factors could further improve the performance of the model and enhance its accuracy in predicting energy consumption for individual households.

In [94], Somu et al. introduced a hybrid model, called eDemand, for accurate and robust building energy consumption forecasting. The proposed model combines LSTM networks with an improved sine cosine optimization algorithm. To validate the effectiveness of the model, the authors collected live energy consumption data from an academic building, specifically the Indian Institute of Technology in Bombay. The data was utilized to forecast energy consumption over short-term, mid-term, and long-term intervals. The experimental results demonstrated that the eDemand model surpassed the performance of existing energy consumption forecast models, as evaluated using various metrics. The proposed model exhibited superior accuracy and robustness in predicting building energy consumption.

In [64], the application of Deep Reinforcement Learning (DRL) techniques for forecasting building

energy consumption was investigated. This area has received limited attention in the literature. The study focused on an office building and investigated three commonly-used DRL techniques: Asynchronous Advantage Actor-Critic (A3C), Deep Deterministic Policy Gradient (DDPG), and Recurrent Deterministic Policy Gradient (RDPG). The objective of the study was to assess the potential of DRL techniques in building energy consumption prediction and compare their performance against commonly-used supervised models. The authors conducted comprehensive experiments and evaluations to evaluate the effectiveness of the proposed DRL models. The experimental results revealed that DDPG outperformed the supervised models in both single-step ahead prediction and multi-step ahead prediction tasks. It demonstrated superior accuracy and performance compared to the other techniques. On the other hand, while the RDPG model did not exhibit advantages over DDPG in single-step ahead prediction, it showed notable improvements in accuracy for multi-step ahead prediction. In contrast, the A3C technique performed poorly in both single-step ahead and multi-step ahead prediction tasks, suggesting its inadequacy for forecasting building energy consumption.

Developing individual models for each customer is not a scalable solution due to its high computational complexity. Instead, an ideal approach for an energy company would be to employ a model that utilizes customer-specific information as features and can generalize across the entire customer base. Recent research efforts have been focusing on exploring innovative and advanced techniques to adopt this approach. In the following are reported a few recent works that adopt this approach.

A novel deep ensemble learning-based probabilistic load forecasting framework is proposed in [116] to quantify the load uncertainties of individual customers. The presented framework employed the profiles of different customer groups and integrated them into the understanding of the task. Specifically, customers were clustered into separate groups based on their profiles and a multitask representation learning approach was employed on these groups simultaneously. This technique led to better feature learning across groups and it was particularly useful to improve the performance of predicting residential demand response and managing home energy in smart grids. However, this technique in order to be applicable needs customers' personal information and a large customer base to be effective.

[119] introduced a novel approach for day-ahead residential load forecasting. The method involved feature engineering, pooling, and a hybrid DL model. The feature engineering process comprised two-stage preprocessing, which included decomposition and multi-source input dimension reconstruction of the data from each user. Subsequently, a pooling operation was applied to merge the data from the target user and its interconnected users based on mutual information in descending order. Lastly, a hybrid model with two input channels was developed by combining a LSTM network with a self-attention mechanism. The evaluation was conducted on a dataset consisting of multiple residential users. The proposed load forecasting method demonstrated the best performance when using a four-user data pool, 49 time steps, and 24 feature dimensions. The achieved performance measures were 15.33% for MAPE, 56.86 kW for MAE, and 82.50 kW for RMSE. The results showcased that the proposed method is an effective choice for day-ahead residential load forecasting.

[77] introduced a daily, weekly, and monthly energy consumption prediction model for individual customers using a TFT approach. The study employed a TFT model that incorporated both primary and valuable data sources, along with batch training techniques. The dataset consisted of 169 customers, but the TFT model was specifically tested on data from a single customer to showcase the effectiveness of the customer-based predictive model. The performance of the proposed TFT model was compared to LSTM, interpretable LSTM, and TCN models. The overall symmetric MAPE (sMAPE) values for LSTM, interpretable LSTM, TCN, and the proposed TFT model were found to be 29.78%, 31.10%, 36.42%, and 26.46% respectively. The sMAPE results indicated that the TFT model outperformed the other DL models, demonstrating its superior performance. This recent work highlights the intelligent use of time series data from different customers, which improves the forecast quality compared to other models when predicting energy consumption for a single customer.

2.5 Electricity production forecasting

Electricity production forecasting is a classic problem in the time series field. In particular, recently the importance of renewable energy sources has been raised due to the clear effects of climate change.

Understanding how much energy the systems based on renewable energy sources can produce with respect to their maximum production capacity is crucial. However, this is quite difficult to forecast since it depends on future natural phenomena. Different studies in the literature distinguish between different renewable energy sources, such as photovoltaic (PV), wind, geothermal, biomass, and hydropower. In this section, the techniques for PV electricity production forecasting are presented.

PVGIS[5] is a tool of the EU-Joint Research Center that provides information about solar radiation and PV system performance for any location in Europe and Africa, as well as a large part of Asia and America. PVGIS uses high-quality solar radiation data obtained from satellite images, as well as ambient temperature and wind speed from climate reanalysis models. It is a free tool that allows specifying the details of a PV plant to obtain its potential generation. It is useful to get an indication of the potential generation but it is not accurate as the methods presented in the literature. In fact, specific methods can focus on specific plants taking into account historical data from which to extract the specific production performance and consider the most recent weather forecasts available.

[51] reviewed the theory behind the forecasting methodologies, and presented several successful applications of solar forecasting methods for both the solar resource and the power output of solar plants at the utility-scale level. Some examples of the presented approaches are Regressive methods, ANNs, Numerical Weather Prediction (NWP), and hybrid methods incorporating two or more techniques.

Zamo et al. in 2014 presented a pair of articles proposing a benchmark of statistical regression methods for short-term forecasting of PV electricity production. The first one treated deterministic forecasts of hourly production [117], and the second one probabilistic forecasts of daily production [118]. The proposed benchmark designated Random Forest as the best forecast model for hourly PV production with a short lead time (from 28 to 45 h). Their results also suggested that the RMSE can be reduced to about 5.8% by first forecasting the production for each individual power plant and then summing these forecasts up. For probabilistic forecasts of 2 days ahead daily production, quantile regression (QR) based forecasts performed significantly better than the climatology, with a CRPS (continuous ranked probability score) lowered by up to 50%. For most power plants, a QR-based forecast performs better than the others. But the most accurate forecast may vary from one power plant to another and with the number of forecast quantiles.

[8] presented a review of PV power forecasting up to 2016. Forecasting techniques such as regressive methods, ANN, KNN, SVM, Random Forest, and hybrid methods are presented. Most recent papers used ML techniques, due to the ease of modeling without the need of knowing PV plant characteristics, since these approaches are able to learn them from data. Also, spatial and temporal horizons and performance metrics are discussed. The forecast horizon where most research has been done is the day ahead. The reason is that most of the energy is traded in day-ahead markets when planning and unit commitment takes place. They discussed that spatial averaging is very useful since it reduces the variability of the solar resource and generates regional forecasts that are more reliable than single-site ones. This is caused by the smoothing effect, which cancels errors with opposite signs in different PV plants.

Barbieri et al. in [11] found out that ANNs and SVM are appropriate approaches for short-term horizons and NWP are better suited for longer horizons. In fact, while a probabilistic method based on historical data may be valuable for very long-term forecasts, such an approach cannot take into consideration the complex variations of the cloud cover causing short-term sunlight disruptions. Only a deterministic atmospheric modeling approach can deal with the stochastic changes of solar radiance during the day. Within this type of model, NWP data-based models are well adapted for day-ahead forecasts but suffer from a too coarse temporal resolution. Sky imagers are a precious tool to identify cloud types and anticipate the impact of shading on PV power generation. They concluded by introducing some future works such as the elaboration of algorithms that can calculate cloud cover and classify clouds using online data and a fine sampling period. In addition, measuring precisely the effects of each type of cloud on solar irradiance could greatly help in improving the results.

In [24], Das et al. based on the studies dated up to 2018 found out that ANN and SVM-based forecasting models performed well under rapid and varying environmental conditions. In addition, most of the studies adopted numerous techniques to develop their forecasting model to obtain bet-

[5] `https://joint-research-centre.ec.europa.eu/pvgis-online-tool_en`

ter accuracy. Moreover, a considerable number of studies classified the forecasted day into different categories based on the weather conditions using several techniques and then developed the forecasting model. However, the range of the observed error was remarkably high due to different weather conditions. For performing good predictions and minimizing errors, the separate sub-model for each weather condition has to perform well.

[93] classified solar PV forecasting methods into three major categories, that are time series statistical, physical, and ensemble methods. Among the most used methods, there are ANN and SVM thanks to their ability in solving complex and non-linear forecasting models. The metrics assessment shows that Artificial Intelligence (AI) models could decrease the error compared to other statistical approaches. The ensemble methods have been introduced recently and thanks to their ability to merge linear and non-linear techniques enhanced the accuracy and performance in comparison with individual models. The standard metrics that are used for evaluating solar prediction accuracy were presented for specific applications to allow the selection of the appropriate solar forecasting approaches to ensure better performance.

[25] presented a literature review on big data models for solar PV electricity generation forecasts, aiming to evaluate the most applicable and accurate state-of-the-art techniques to the problem. They also included the motivation behind each project proposal, and the characteristics and quality of data used to address the problem, among other issues. They affirmed that the prediction of solar electricity generation is currently an ongoing academic research question. ML is widely used, and approaches based on neural networks are considered the most accurate. Extreme learning machine (ELM) was also a great addition and it has reduced training time and raised precision.

[1] investigated the accuracy, stability, and computational cost of Random Forest and Extra Trees techniques for predicting the hourly PV generation output. They compared the performance of the proposed methods with SVR. They proved that all developed models have comparable predictive power and are equally applicable for predicting hourly PV output. Despite their comparable predictive power, Extra Trees outperformed Random Forest and SVR in terms of computational cost. They concluded by affirming that the stability and algorithmic efficiency of Extra Trees makes this technique an ideal candidate for wider adoption in PV output forecasting.

In [3], Ahmed et al. reviewed and evaluated contemporary PV solar power forecasting techniques. They noticed through correlation analysis that solar irradiance is the most correlated feature with PV output and consequently weather classification and cloud motion study are crucial operations for optimal results. In addition, they stated that the best data-cleaning processes are normalization and wavelet transforms, and that augmentation using generative adversarial networks is recommended for network training and forecasting. Furthermore, they also analyzed that the optimization of inputs and network parameters can be done by using GAs and particle swarm optimization. They determined that ensembles of ANNs are the best approach for forecasting short-term PV power.

[56] examined the performance of the LSTM method in Turkey's electricity production estimation and determined the optimization technique that provides the best performance in the LSTM estimation method. It was observed that the energy production estimation of LSTM and Adam optimization technique achieved successful results.

In [42], Gellert et al. proposed and evaluated a context-based technique to anticipate electricity production and consumption in buildings. They focused on a household with PVs and an energy storage system. They analyze the efficiency of Markov chains, stride predictors, and also their combination into a hybrid predictor in modeling the evolution of electricity production and consumption. Experimental results showed that the best predictor is the Markov chain configured with an electric power history of 100 values, a context of one electric power value, and an interval size of 1.

A GA-based SVM (GASVM) model for short-term power forecasting of residential-scale PV systems is proposed in [104]. The GASVM model classified the historical weather data using an SVM classifier initially and later it is optimized by the GA using an ensemble technique. Experimental results demonstrated that the proposed GASVM model outperformed the conventional SVM model by a difference of about 669.624W in the RMSE value and 98.7648% of the MAPE error.

In [124], a hybrid model (SDA-GA-ELM) based on ELM, GA, and customized similar day analysis (SDA) has been developed to predict hourly PV power output. In the SDA, the Pearson correlation

coefficient is employed to measure the similarity between different days based on five meteorological factors, and the data samples similar to those from the target forecast day are selected as the training set of ELM. In the ELM, the optimal values of the parameters are searched by the GA to improve the prediction accuracy. The results show that the SDA-GA-ELM model has higher accuracy and stability than other tested approaches in day-ahead PV power prediction, such as ELM, SVM, SDA-ELM, and SVM-ELM.

[67] presented case studies on forecasting PV power production and electricity demand in Portugal. They studied an ensemble of different ML methods (SVM, Random Forest, LSTM, and ARIMA) to exploit the growing collection of energy supply and demand records. The ensemble used only electricity data to forecast since only this data is available online for any forecasting horizon. The ensemble method was based on offline training and online forecasting, by applying the most recent power measurements to trained models. The different ML methods performed different non-linear transformations to the same electricity data, thus introducing diversity in the ensemble. To assess the forecasting performance of the system, they considered two forecasting horizons relevant to the Internal Electricity Market, namely 36 hours ahead, relevant to the single day-ahead coupling, and 2 hours ahead, relevant to the single intraday coupling. The forecasting performance using only electricity data is compared with state-of-the-art models and improves the reference accuracy in their case studies. Since the ensemble relies only on energy data, the results showed that ML methods are useful to exploit energy big data for efficient energy forecasting systems. As demonstrated by other papers, weather information such as solar irradiance has a high correlation with energy production, and when studying a few plants, instead of all the ones in Portugal, this information has a relevant impact on performance.

A novel hybrid method for deterministic PV power forecasting based on wavelet transform (WT) and CNN is proposed in [106]. WT is used to decompose the original signal into different frequency series. CNN is employed to extract the nonlinear features and invariant structures exhibited in each frequency. A probabilistic PV power forecasting model that combines the proposed deterministic method and spine quantile regression is developed to statistically evaluate the probabilistic information in PV power data. Statistical results showed that the average MAPE, RMSE, and MAE of the proposed deterministic model outperformed the compared benchmarks in terms of seasons, forecasting horizons, and PV power locations.

Day-ahead power output time series forecasting methods are proposed in [41], in which the ideal weather type (sunny day) and non-ideal weather types (rainy days, windy days, and foggy days) have been separately discussed. For ideal weather conditions, a forecasting method is proposed based on meteorology data of the next day using LSTM networks. For non-ideal weather conditions, time series relevance and specific non-ideal weather type characteristics are considered in the LSTM model by introducing adjacent day time series and typical weather type information. Specifically, daily total power, which is obtained by the discrete grey model (DGM), is regarded as input variables and applied to correct power output time series prediction. Prediction performance comparison between proposed methods with traditional algorithms revealed that the RMSE accuracy of forecasting methods based on LSTM networks can reach 4.62% for ideal weather conditions. For non-ideal weather conditions, the dynamic characteristic is effectively described by proposed methods and the proposed methods obtained superior prediction accuracy. This is an interesting study on how to treat the problem in two subcases, however, for real applications, a single network able to adapt to weather conditions by itself is needed.

In [108], a CNN, a LSTM network, and a hybrid model based on CNN and LSTM models were proposed by Wang et al and they are applied to the data of the DKASC (Dattajirao Kadam Arts, Science and Commerce) college PV system. The results showed that when the input sequence is increased, the accuracy of the model is also improved, and the prediction effect of the hybrid model is the best, followed by that of the CNN. While the LSTM network had the worst prediction effect, the training time was the shortest.

In [109], a hybrid DL model (LSTM-CNN) is proposed and applied to PV power prediction. In the proposed hybrid prediction model, the temporal features of the data are extracted first by the LSTM network, and then the spatial features of the data are extracted by the CNN model. The results

showed that the hybrid prediction model had a better prediction effect than the single prediction models in isolation, and the proposed hybrid model is also better than the CNN-LSTM (extract the spatial characteristics of data first, and then extract the temporal characteristics of data).

A hybrid DL model combining wavelet packet decomposition (WPD) and LSTM networks is proposed in [60]. The hybrid DL model is utilized for one-hour-ahead PV power forecasting with five-minute intervals. WPD is first used to decompose the original PV power series into sub-series. After the decomposition, four independent LSTM networks are developed for the sub-series. Finally, the results predicted by each LSTM network are reconstructed and a linear weighting method is employed to obtain the final forecasting results. Experimental results show that the proposed hybrid DL model exhibits superior performance in both forecasting accuracy and stability with respect to LSTM, RNN, GRU, and MLP.

In [72], different kinds of neural networks for short-term output PV power forecasting have been developed and compared: LSTM, Bidirectional-LSTM (BiLSTM), GRU, Bidirectional-GRU (BiGRU), One-Dimension CNN (CNN1D), as well as other hybrid configurations such as CNN1D-LSTM and CNN1D-GRU. A database of the PV power produced by the microgrid installed at the University of Trieste (Italy) is used to train and comparatively test the neural networks. The performance has been evaluated over four different time horizons, for one-step and multi-step ahead. The results show that the investigated neural networks provide very good accuracy, particularly in the case of a 1-minute time horizon with one-step ahead (correlation coefficient is close to 1), while for the case of multi-step ahead (up to 8 steps ahead) the results are found to be acceptable (correlation coefficient ranges between 96.9% and 98%). The new advanced neural network algorithms are able to lead to acceptable accuracy in the case of cloudy days. However, even though the models have great potential for fine time granularity, these time ranges are quite limited and they should be proved to be effective also on longer ranges to be used by PV plant owners.

[68] aimed to predict hourly day-ahead PV power generation by applying the TFT model. TFT incorporates an interpretable explanation of temporal dynamics and high-performance forecasting over multiple horizons. The proposed forecasting model has been trained and tested using data from six different facilities located in Germany and Australia. The results have been compared with other algorithms like ARIMA, LSTM, MLP, and XGBoost. The use of TFT has been shown to be more accurate than the rest of the algorithms in forecasting PV generation in the tested different facilities. The importance of the decoder and encoder variables has been also calculated, revealing that solar horizontal irradiation and the zenith angle are the key variables for the model. This model was proved to be effective also in this task, showing the superiority of the TFTs over standard DL models, even though it is quite more complex and requires a larger amount of data to achieve good prediction results.

3 System Model

In this chapter, the model of the proposed system is presented. In the first section, an in-depth presentation of the designed system architecture is provided. The different components of the system with their functionalities and interactions are described. Tthe overall architecture was designed in order to build a Software as a Service (SaaS) capable of satisfying different use cases and being used directly by energy retailers. Lastly, the modules focused on the three use cases of interest are treated more in detail in dedicated sections. After this chapter, it will be clear what the main parts of the system are and how they cooperate to accomplish the various use cases of interest.

3.1 System architecture

The use case diagram for the proposed system is presented in figure 3.1. Users can interact with the system via dedicated APIs. To be authorized, a user must be authenticated and have an account with an active subscription. Users can request the following asynchronous operations:

- Load new data with a create or update logic specifying the type of data;
- Train new models based on available data for a specific use case specifying the time granularity (hourly or daily). The supported specific use cases for the models that will be used in the dedicated forecast requests are electricity demand forecasting, consumption baseline forecasting, and electricity production forecasting;
- Forecast future data for a specific use case using a specified model for a certain time granularity, for a certain starting date and time horizon.

In addition, a task scheduler periodically triggers the performance evaluation of the available models and if needed triggers the models re-training for having up-to-date models with respect to the user data so that they might perform better on future forecasts.

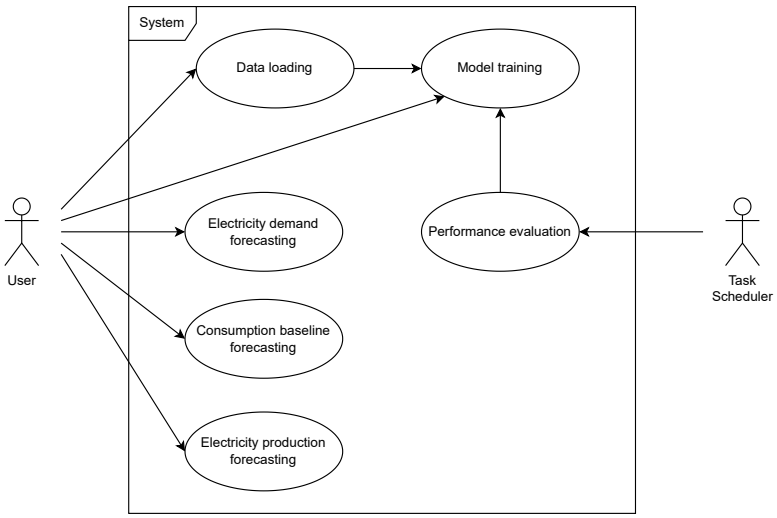


Figure 3.1: The use case diagram for the proposed system.

For achieving the presented use cases the system architecture shown in figure 3.2 was designed. It is composed of the following components:

- Authentication layer: manages the authentication process and verifies user permissions;

- API layer: manages the user interactions with the system;
- Task manager: coordinates the execution of the tasks;
- Data loading task: parses and stores the users' loaded data;
- ML models interface: handles the interactions with the ML models storage where the trained ML models are stored;
- Model training task: trains new models based on available data;
- Forecasting task: forecasts future data using a specified model;
- Task scheduler: manages the periodic scheduled tasks;
- Performance evaluation task: evaluates the performance of the available models and if needed triggers the models re-training.

The system interacts with the following external components:

- Users data storage: used to store and retrieve user data, including account data;
- Data storage: used to store and retrieve the users' loaded data;
- Weather data storage: used to retrieve the weather data;
- ML models storage: used to store and retrieve the trained models;
- Forecasts storage: used to store and retrieve the computed forecasts.

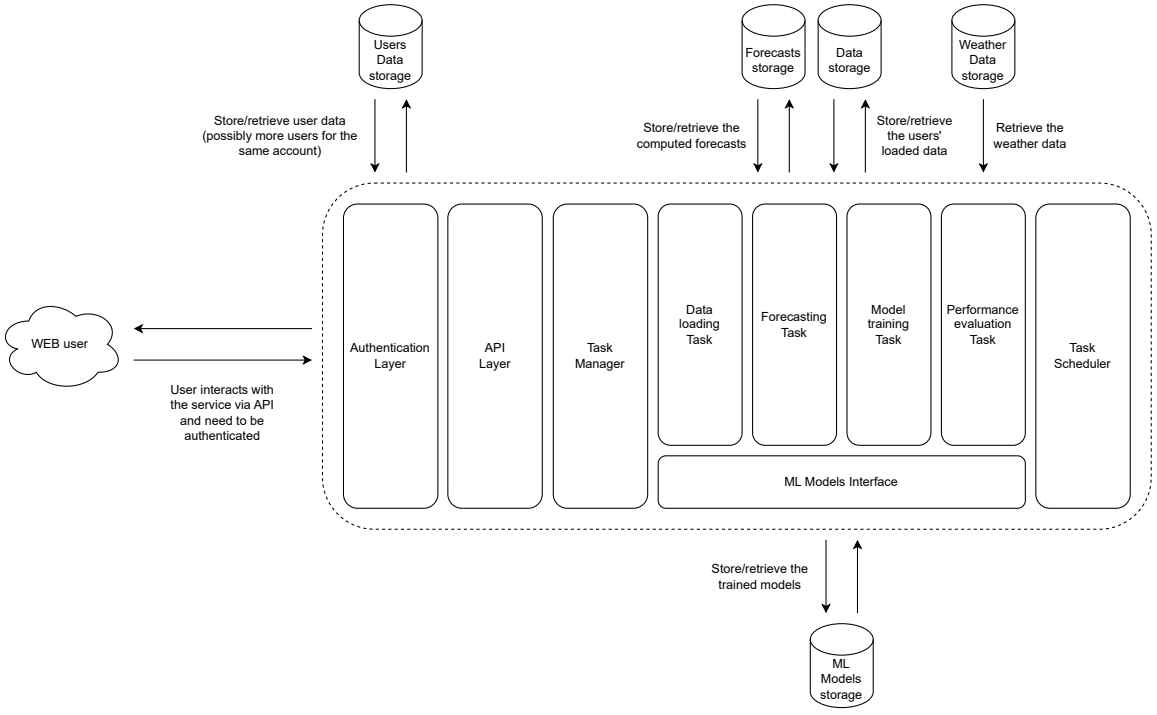


Figure 3.2: The architecture designed for the proposed system.

Figure 3.3 shows the interactions among the components of the designed architecture for achieving the presented use cases. In the following subsections, the logic for each use case is explained with the relative components' interactions.

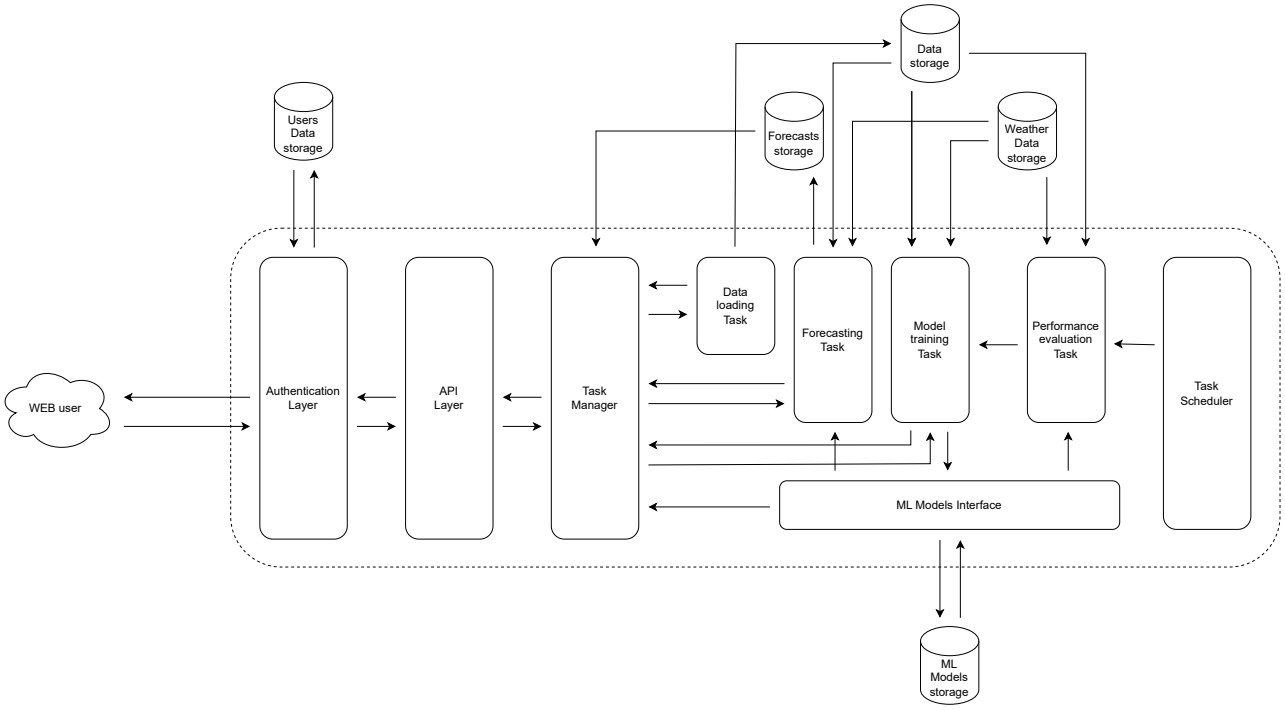


Figure 3.3: The interactions among the components in the designed architecture.

3.1.1 Data loading

The diagram representing the components' interactions for achieving the data loading use case is presented in figure 3.4.

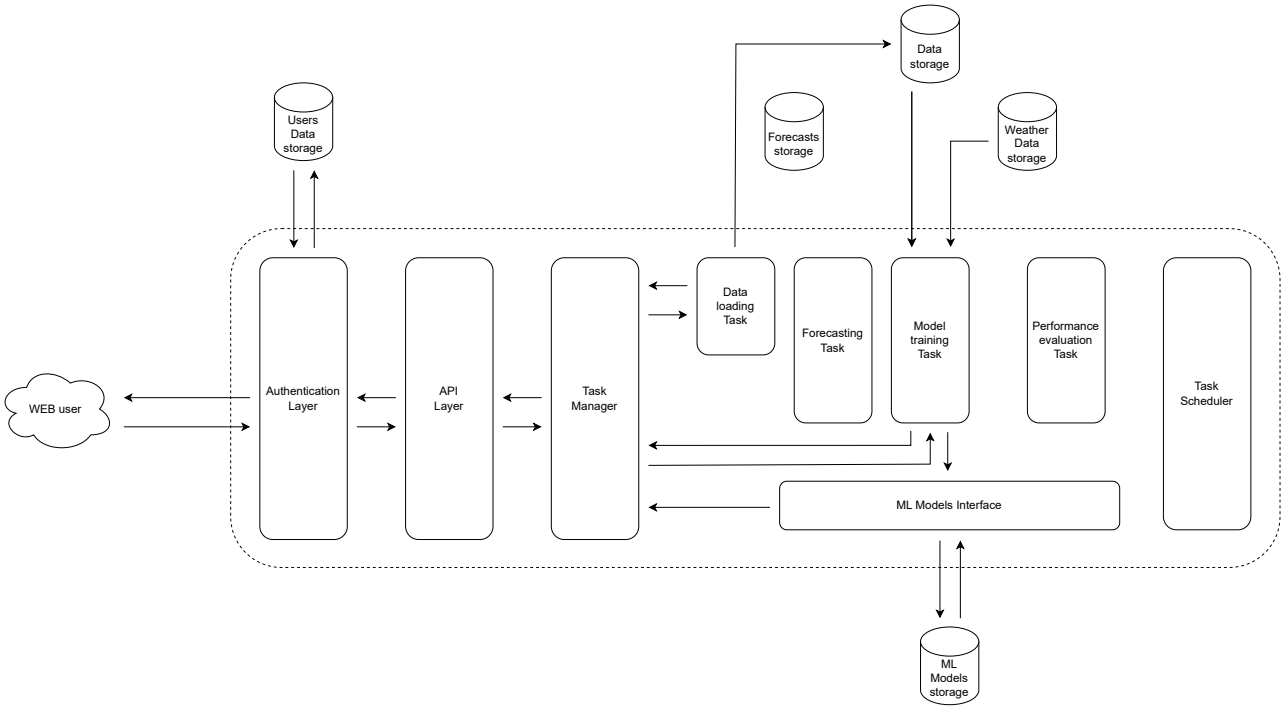


Figure 3.4: The interactions among the components for achieving the data loading use case.

The user sends the data to the system, for example uploading CSV files. First, the authentication layer verifies the authentication of the user and checks whether it has a subscription to load the data. The endpoint dedicated to data loading takes charge of the user request and provides the task manager with the details for creating a data loading task. The identifier of the task is then returned to the

user, who can request the status of the data loading request and check the result of the operation when completed.

The data loading task starts when the task manager schedules it. As shown by the diagram in figure 3.5, the logic inside this task can be formulated in 2 steps: parse the data and store the data. The data parser block transforms the user data in a system-compatible format and passes it to the data storage block which stores the data inside the data storage. If there is no model for the specific use case for which data is loaded, the task manager will also create a model training task to train all the available models on this data. This allows the system to be ready for the forecast requests on the loaded data. The model training task is described more in detail in subsection 3.1.2.

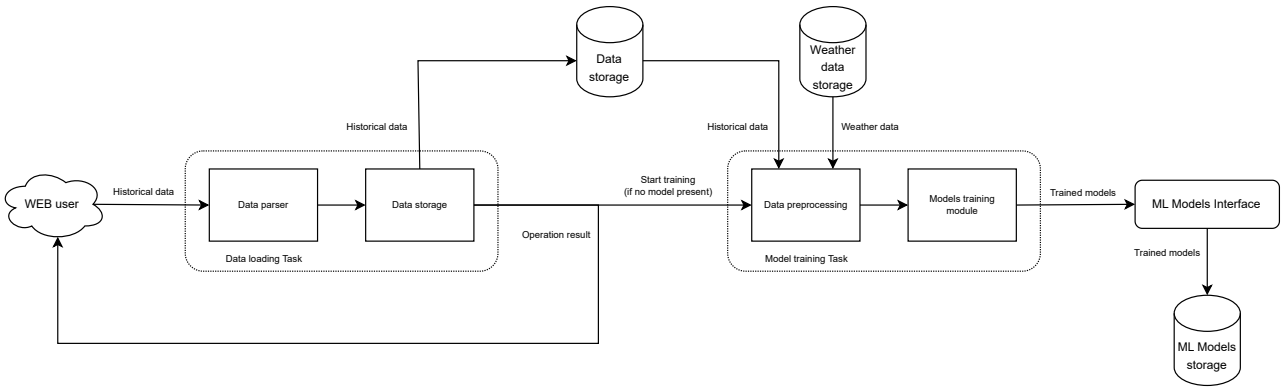


Figure 3.5: Diagram representing the data loading flow.

The sequence diagram representing the complete data loading procedure is presented in figure 3.6.

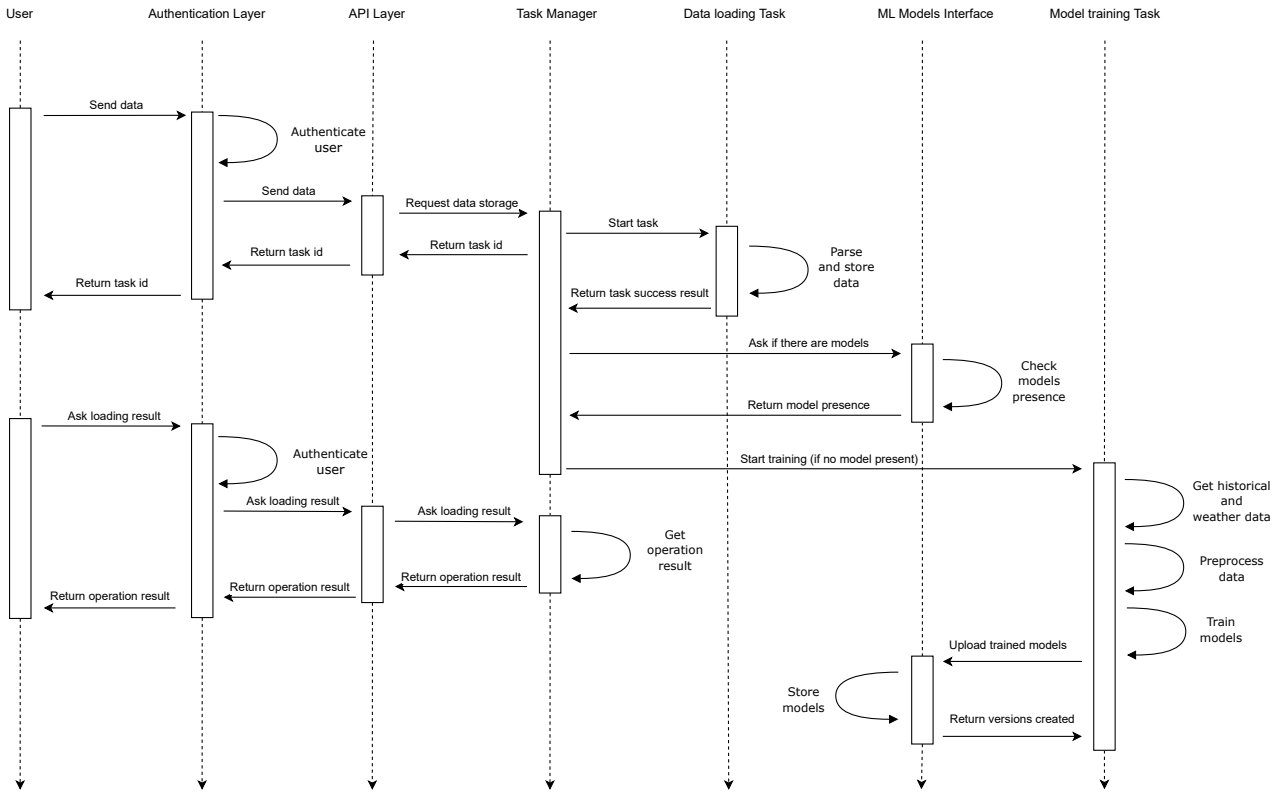


Figure 3.6: Sequence diagram of the data loading procedure.

3.1.2 Model training

The diagram representing the components' interactions for achieving the model training use case is presented in figure 3.7.

The user sends the specification of the model to train to the system. First, the authentication layer verifies the authentication of the user and checks whether it has a subscription to request the training of a model. The endpoint dedicated to model training takes charge of the user request and provides the task manager with the details for creating a model training task. The identifier of the task is then returned to the user, who can request the status of the model training request and get the trained model when completed.

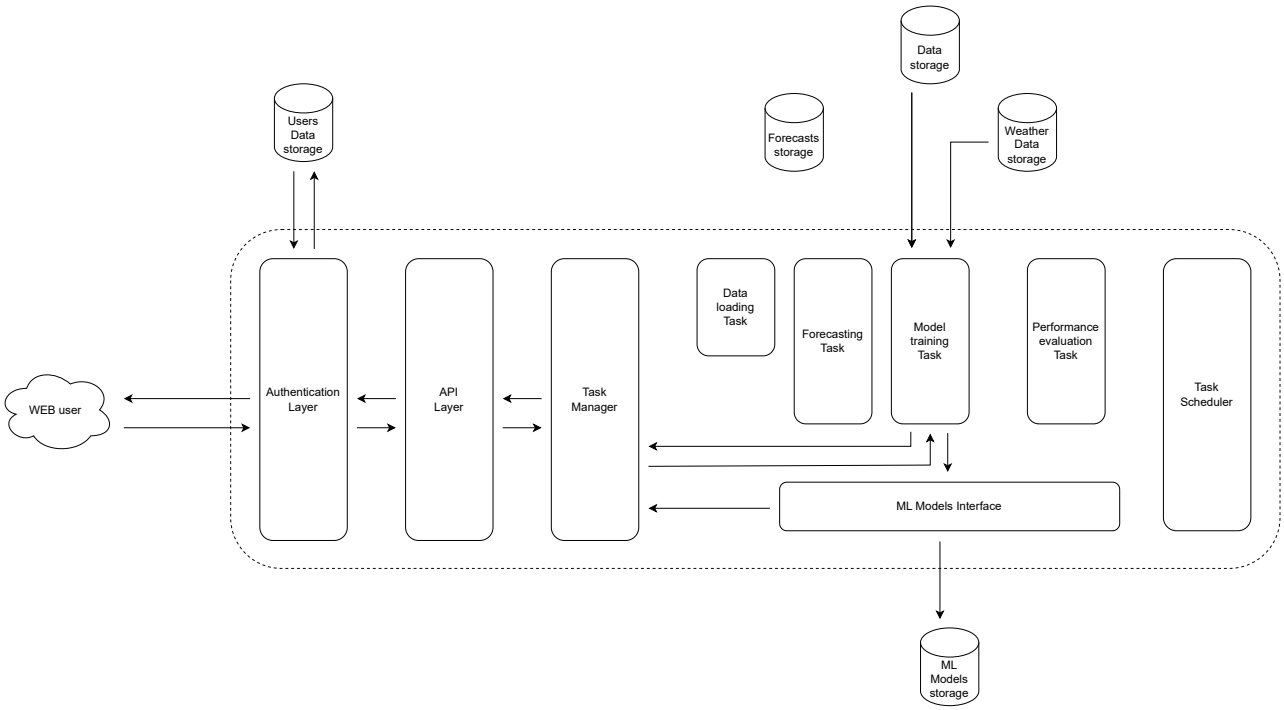


Figure 3.7: The interactions among the components for achieving the model training use case.

The model training task starts when the task manager schedules it. As shown by the diagram in figure 3.8, the logic inside this task can be formulated in 2 steps: preprocess the available data and train the model. The data preprocessing block collects the available data for the specific use case for which the model will be trained, aggregates it correctly, cleans it, and enriches it with weather data. The models training module takes the preprocessed data and trains the model with the provided specifications. It was thought of as an extendable framework on which it is possible to support different models and possibly integrate new ones over time. The resulting model is then passed to the ML models interface which stores it inside the ML models storage.

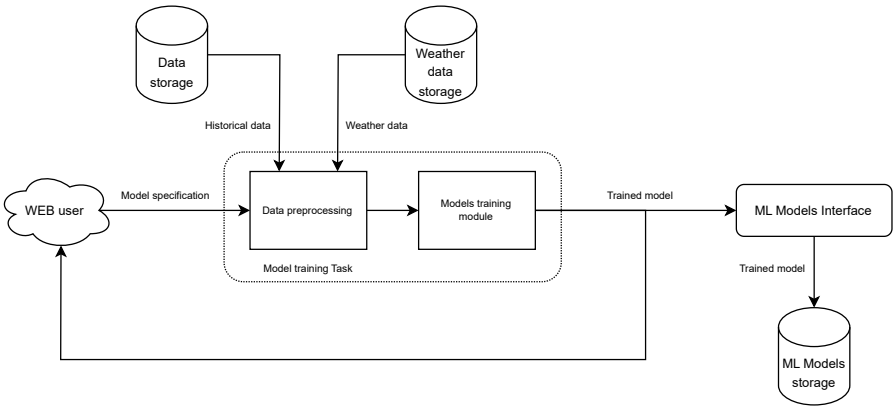


Figure 3.8: Diagram representing the model training flow.

The sequence diagram representing the complete model training procedure is presented in figure 3.9.

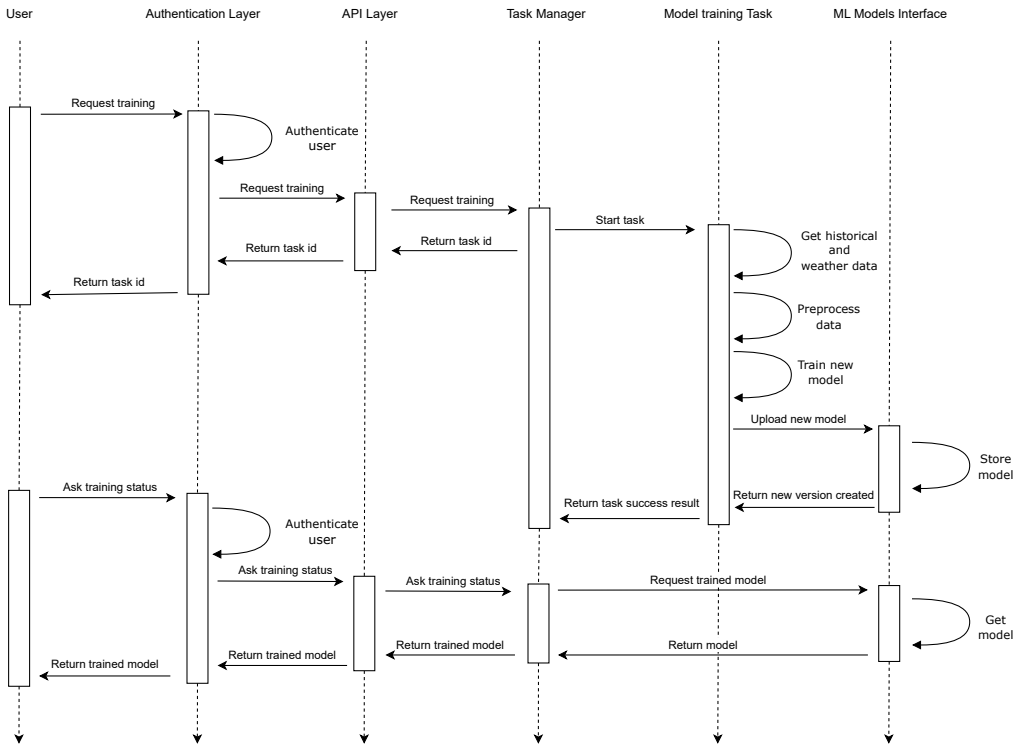


Figure 3.9: Sequence diagram representing the model training procedure.

3.1.3 Forecasting

The diagram representing the components' interactions for achieving the forecasting for a specific use case is presented in figure 3.10.

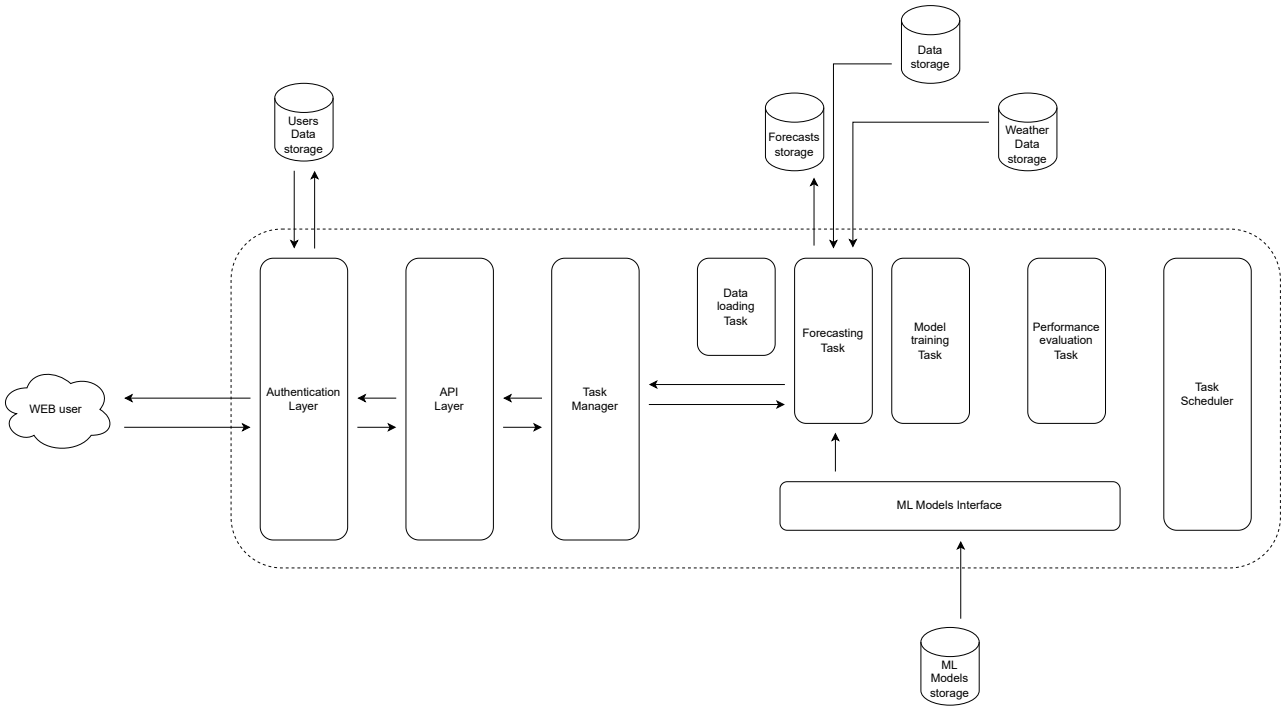


Figure 3.10: The interactions among the components for achieving the forecasting for a specific use case.

The user sends the specification for computing forecasts for a specific use case to the system. First, the authentication layer verifies the authentication of the user and checks whether it has a subscription

to request the forecasts for a specific use case. The endpoint dedicated to forecasting takes charge of the user request and provides the task manager with the details for creating a forecasting task. The identifier of the task is then returned to the user, who can request the status of the forecasting request and get the computed forecasts when completed.

The forecasting task starts when the task manager schedules it. As shown by the diagram in figure 3.11, the logic inside this task can be formulated in 2 steps: preprocess the needed data by the model and forecast future data. The data preprocessing block collects the needed data by the model for the specific use case for which the forecasts will be made, aggregates it correctly, cleans it, and enriches it with weather data. In addition to past weather data also weather forecasts are collected to help produce more accurate future data. The prediction module takes the preprocessed data, requests the specified model, and computes the forecasts. The computed forecasts are then stored in the forecasts storage.

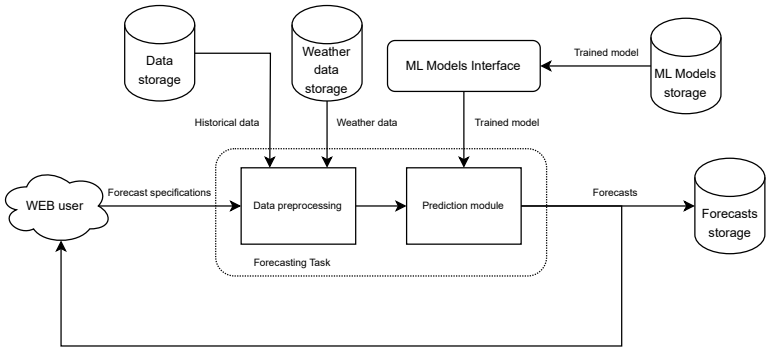


Figure 3.11: Diagram representing the forecasting flow.

The sequence diagram representing the complete forecasting procedure for a specific use case is presented in figure 3.12.

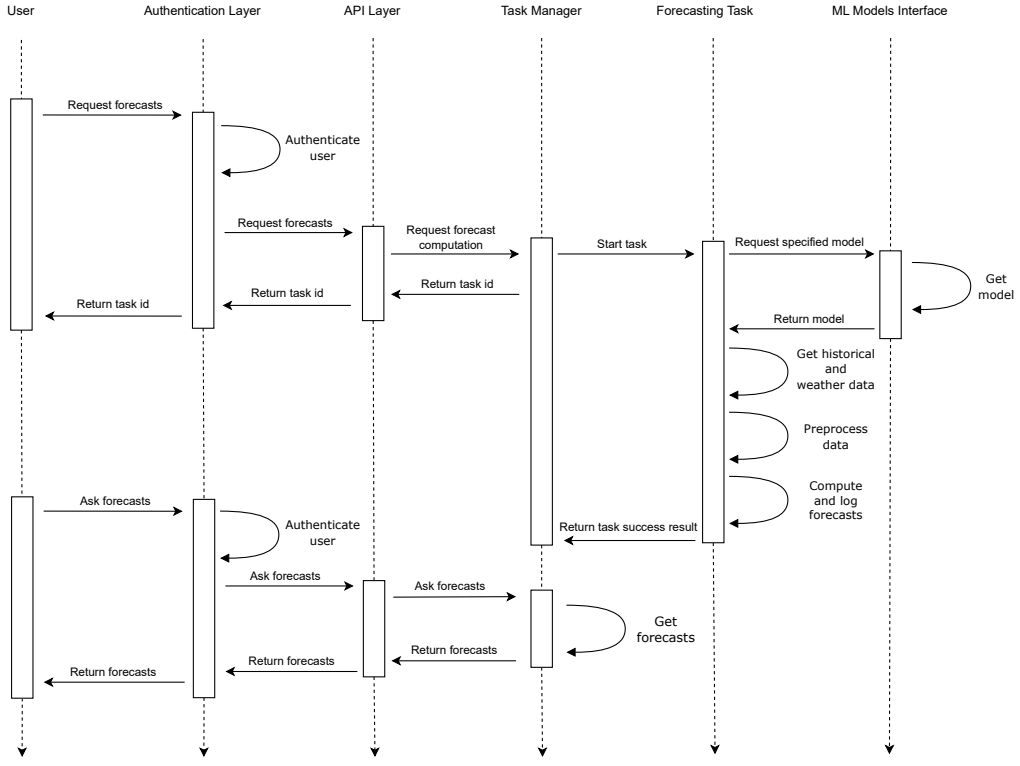


Figure 3.12: Sequence diagram representing the forecasting procedure.

3.1.4 Performance evaluation

The diagram representing the components' interactions for achieving the performance evaluation use case is presented in figure 3.13.

The task scheduler periodically creates and triggers a performance evaluation task. As shown by the diagram in figure 3.14, the logic inside the performance evaluation task can be formulated in 2 steps: preprocess the data and evaluate the performance. The data preprocessing block collects the data for the specific use case for which the model will be trained, aggregates it correctly, cleans it, and enriches it with weather data. The performance evaluation module takes the preprocessed data and evaluates the performance of the last available models. The performance evaluation procedure with the obtained results is described in chapter 5. When performance drops significantly, it triggers the models re-training by creating a model training task for having up-to-date models with respect to the user data so that they might perform better on future forecasts. The model training task is described more in detail in subsection 3.1.2.

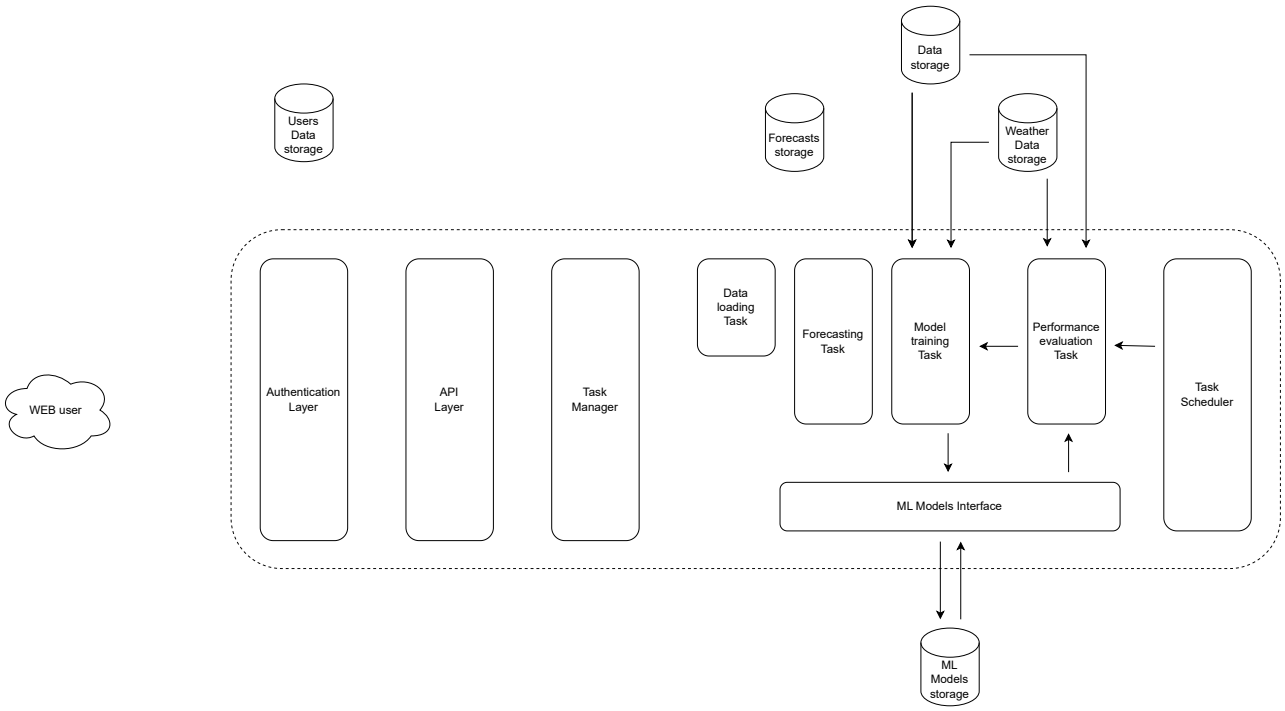


Figure 3.13: The interactions among the components for achieving the performance evaluation use case.

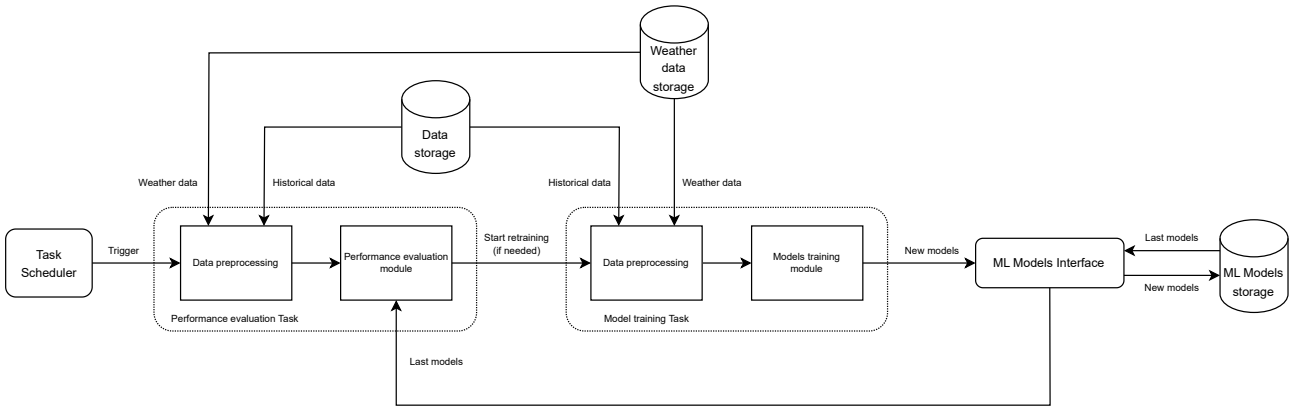


Figure 3.14: Diagram representing the performance evaluation flow.

The sequence diagram representing the complete performance evaluation procedure is presented in figure 3.15.

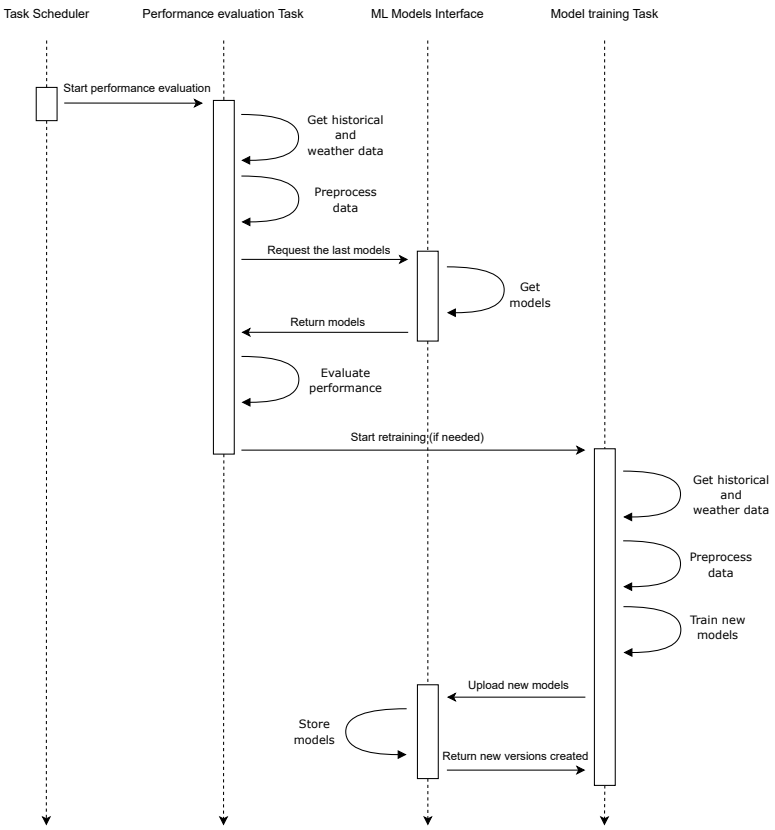


Figure 3.15: Sequence diagram representing the performance evaluation procedure.

3.2 System's common components

In this section, the system's common components across the various specific use cases are described in detail.

The authentication layer manages the user authentication process. In particular, for accessing the system the user needs to be authenticated. When users interact with the system via dedicated APIs, to be authorized they must provide one valid token and have an account with an active subscription. It is also possible to have more users associated with the same account.

The API layer manages the user interactions with the system. Many endpoints are available for managing the users' possible operations, such as data loading, model training, and forecasting requests. Data loading management endpoints consist of requesting the loading of new data and checking the status of the data loading operations. Model training management endpoints consist of requesting the training of new models and checking the status of the model training operations. Forecasting requests management endpoints consist of requesting the forecasts of future data and checking the status of the forecasting requests operations.

The task manager coordinates the execution of the different tasks requested by the users. Where the possible tasks, as described in the previous section, are the following:

- Data loading task, which parses and stores the users' loaded data on the basis of the type of data;
- Model training task, which trains new models based on available data. The specialization of this task for the specific use cases is described in the dedicated sections;
- Forecasting task, which forecasts future data using a specified model. The specialization of this task for the specific use cases is described in the dedicated sections.

For the data loading task, the main blocks are:

- Data parser, which transforms the input user data in a system-compatible format, the expected information in data depends on the specific use case for which data is loaded;

- Data storage, which stores the parsed data inside the data storage with a create or update logic specifying the type of data.

The base blocks across the various specific use cases that are then described more in detail in the following sections are the following:

- Data preprocessing, which retrieves the needed data from the data storage, aggregates it correctly, cleans it by filling the gaps by using a linear interpolation, and enriches it using the weather information retrieved from the weather data storage. For future data forecasts in addition to past weather data also weather forecasts are collected;

- Models training module, which takes the preprocessed data and trains the specified model with the provided specifications;

- Prediction module, which takes the preprocessed data, requests the specified model, and computes the forecasts.

The ML model interface handles the interactions with the ML models storage where the trained ML models are stored. In particular, this component provides the functionalities to store and retrieve models in the ML models storage, This component allows the system to support lots of models for each account and new ones can be added over time. A versioning mechanism is also handled in the model storage. This component allows the system to provide users with the latest available models and choose the best-performing ones.

The task scheduler manages the periodic scheduled tasks. In particular, as described in the previous section, it is present just the performance evaluation task, which evaluates the performance of the available models and if needed triggers the models re-training. The main block of this task is the performance evaluation module, which takes the preprocessed data and evaluates the performance of the last available models using specific use case error metrics such as mean absolute percentage error (MAPE) and mean absolute error (MAE).

3.3 Electricity demand forecasting module

In this section, the electricity demand forecasting module is described. The module handles the specialization of data models, training, and forecasts for the electricity demand forecasting use case.

The UML (Unified Modeling Language) data class for electricity demand forecasting is represented in figure 3.16. The considered attributes are the following: the date and time of the detection, the active incoming to the customers in the past hour, and the number of customers. The basic data is enhanced with the following weather information, the reported value is the average of the value in the last hour: air temperature, apparent temperature, and relative humidity. The mean active incoming per customer is calculated as the division of the total active incoming by the number of customers.

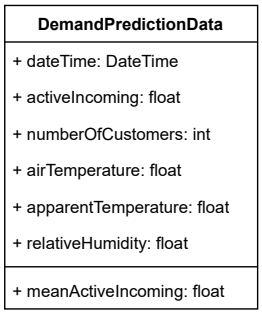


Figure 3.16: The UML data class for electricity demand forecasting.

The schematic representation of the model training procedure for electricity demand forecasting is presented in figure 3.17. It consists of 2 modules: the data preprocessing module and the models training module. The data preprocessing module takes historical aggregated consumption data and historical weather data as input and prepares the data for the model training. The models training module based on the prepared data trains the set of available models for electricity demand forecasting: SARIMA, support vector regressor, hist gradient boosting regressor, extreme gradient boosting regressor, prophet, LSTM, GRU, CNN, TFT, and the combination of different techniques which is also tested. Some baseline approaches which consider repeating past days and weeks, and an AutoML approach are also considered as comparisons.

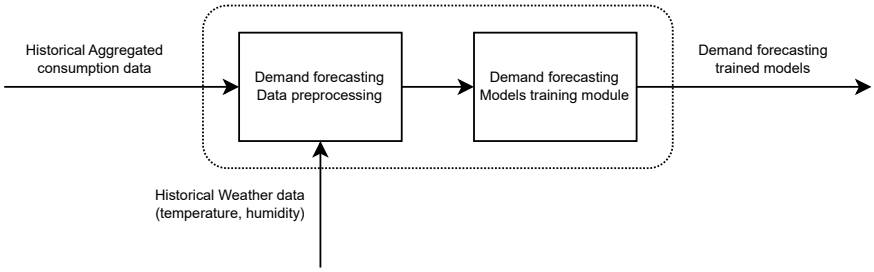


Figure 3.17: The schematic representation of the model training procedure for electricity demand forecasting.

The schematic representation of the forecasting procedure for electricity demand forecasting is presented in figure 3.18. It consists of 2 modules: the data preprocessing module and the prediction module. The data preprocessing module takes forecast specifications, historical aggregated consumption data, historical weather data, and weather forecasts as input and prepares the data for the forecasting. The prediction module based on the prepared data computes the aggregated demand forecasts for the trained available models.

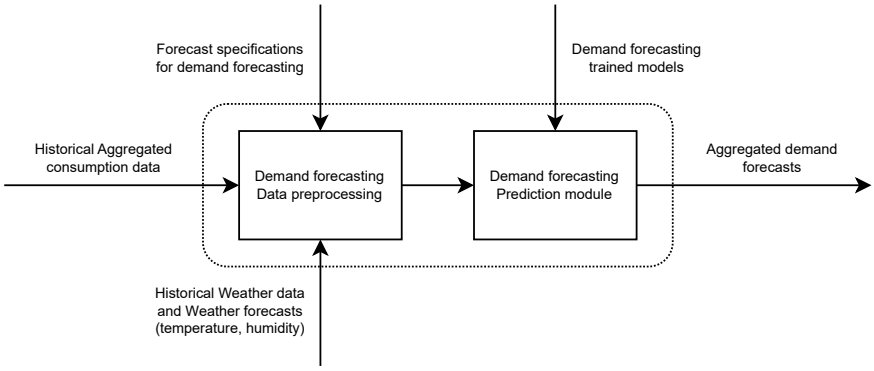


Figure 3.18: The schematic representation of the forecasting procedure for electricity demand forecasting.

3.4 Consumption baseline forecasting module

In this section, the consumption baseline forecasting module is described. The module handles the specialization of data models, training, and forecasts for the consumption baseline forecasting use case.

The UML data class for consumption baseline forecasting is represented in figure 3.19. The considered attributes are the following: the date and time of the detection, the CUPS code (Universal Supply Point Code) which identifies the customer, and the active incoming to the customer in the past hour. In the case of other customer information, this might be used to try to improve the performance of the models. The basic data is enhanced with the following weather information, the reported value is the average of the value in the last hour: air temperature, apparent temperature, and relative humidity.

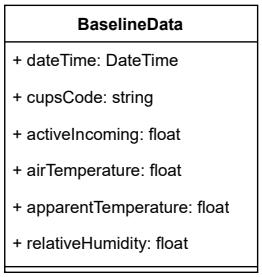


Figure 3.19: The UML data class for consumption baseline forecasting.

The schematic representation of the model training procedure for consumption baseline forecasting is presented in figure 3.20. It consists of 2 modules: the data preprocessing module and the models training module. The data preprocessing module takes historical single customer consumption data and historical weather data as input and prepares the data for the model training. The models training module based on the prepared data trains the set of available models for consumption baseline forecasting: SARIMA, support vector regressor, hist gradient boosting regressor, extreme gradient boosting regressor, prophet, LSTM, GRU, CNN, and TFT. Some baseline approaches which consider repeating past days and weeks, and an AutoML approach are also considered as comparisons.

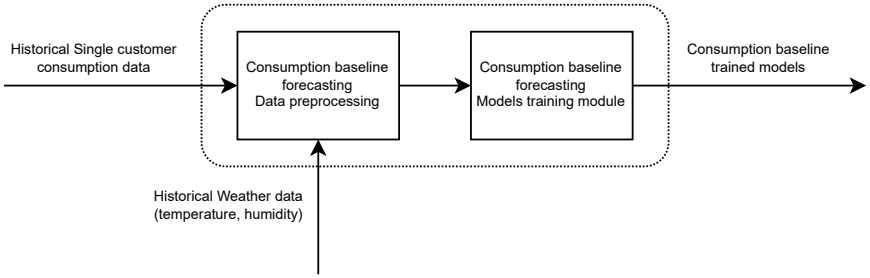


Figure 3.20: The schematic representation of the model training procedure for consumption baseline forecasting.

The schematic representation of the forecasting procedure for consumption baseline forecasting is presented in figure 3.21. It consists of 2 modules: the data preprocessing module and the prediction module. The data preprocessing module takes forecast specifications, historical single customer consumption data, historical weather data, and weather forecasts as input and prepares the data for the forecasting. The prediction module based on the prepared data computes the single customer consumption baseline forecasts for the trained available models.

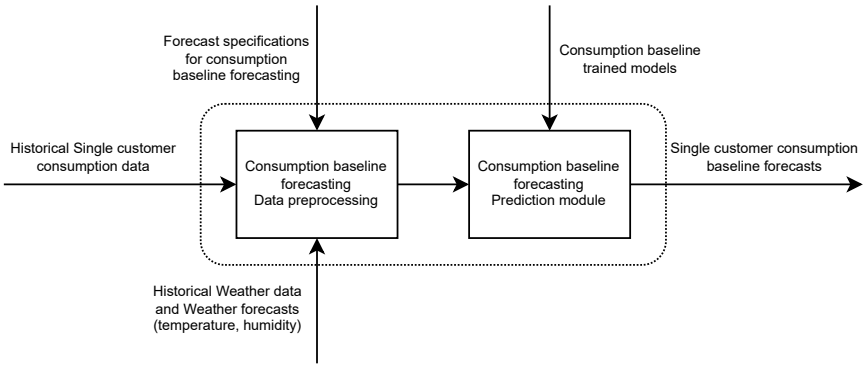


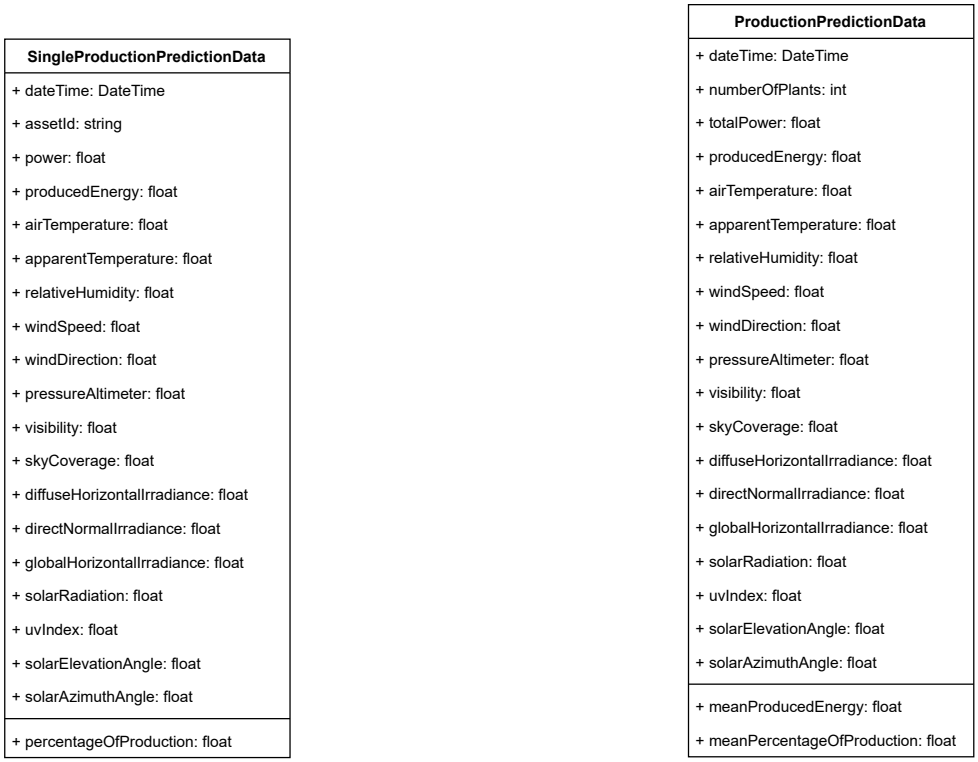
Figure 3.21: The schematic representation of the forecasting procedure for consumption baseline forecasting.

3.5 Electricity production forecasting module

In this section, the electricity production forecasting module is described. The module handles the specialization of data models, training, and forecasts for the electricity production forecasting use case.

The UML data class for a single PV plant data for electricity production forecasting is represented in figure 3.22a. The considered attributes are the following: the date and time of the detection, the id of the asset (intended as the PV plant), the power of the PV plant (intended as its maximum theoretical production power per hour), and the produced energy in the past hour by the PV plant. The basic data is enhanced with the following weather information, the reported value is the average of the value in the last hour: air temperature, apparent temperature, relative humidity, wind speed, wind direction, pressure altimeter, visibility, sky coverage, diffuse horizontal irradiance, direct normal irradiance, global horizontal irradiance, solar radiation, UV index, solar elevation angle, and solar azimuth angle. The percentage of production is calculated by the division of the produced energy by the power of the PV plant.

While the UML data class for the aggregated data over PV plants for electricity production forecasting is represented in figure 3.22b. The considered attributes are the following: the date and time of the detection, the number of PV plants, the total power of the PV plants (intended as the sum of the maximum theoretical production power per hour of all PV plants), and the produced energy in the past hour by the PV plants. The basic data is enhanced with the following weather information, the reported value is the average of the value in the last hour: air temperature, apparent temperature, relative humidity, wind speed, wind direction, pressure altimeter, visibility, sky coverage, diffuse horizontal irradiance, direct normal irradiance, global horizontal irradiance, solar radiation, UV index, solar elevation angle, and solar azimuth angle. The mean produced energy for a single PV plant is calculated as the division of the total produced energy by the number of PV plants. The mean percentage of production is calculated as the division of the total produced energy by the total power of the PV plants.



(a) (b)

Figure 3.22: (a) The UML data class for a single PV plant data and (b) the UML data class for the aggregated data over PV plants for electricity production forecasting.

The schematic representation of the model training procedure for electricity production forecasting

is presented in figure 3.23. It consists of 2 modules: the data preprocessing module and the models training module. The data preprocessing module takes historical PV plants production data and historical weather data as input and prepares the data for the model training. The models training module based on the prepared data trains the set of available models for electricity production forecasting: ARIMA, support vector regressor, hist gradient boosting regressor, extreme gradient boosting regressor, prophet, LSTM, GRU, CNN, TFT, and the combination of different techniques which is also tested. A baseline approach which considers repeating past days and an AutoML approach are also considered as comparisons.

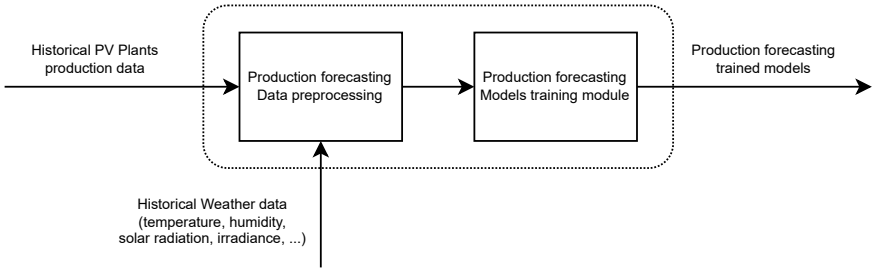


Figure 3.23: The schematic representation of the model training procedure for electricity production forecasting.

The schematic representation of the model training procedure for electricity production forecasting is presented in figure 3.24. It consists of 2 modules: the data preprocessing module and the prediction module. The data preprocessing module takes forecast specifications, historical PV plants production data, historical weather data, and weather forecasts as input and prepares the data for the forecasting. The prediction module based on the prepared data computes the aggregated PV plants production forecasts for the trained available models.

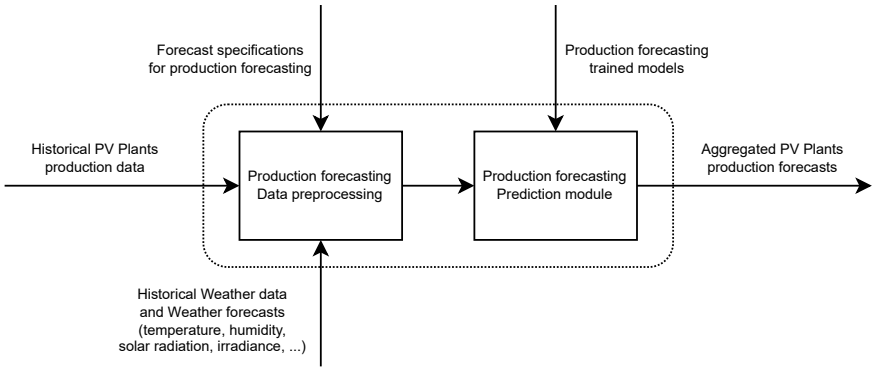


Figure 3.24: The schematic representation of the model training procedure for electricity production forecasting.

4 Prototype Implementation

This chapter presents which components of the system have been implemented and how. Only a subset of the components of the designed architecture in chapter 3 have been implemented. This is because a prototype was developed as a Proof of Concept (PoC) with a focus on key components for validating the core system functionalities for each specific use case. These are the training of models, the forecast of new data, and the evaluation of the performance of the developed models. The remaining components were not implemented since they were not crucial for having a working system, in fact, the overall architecture was designed in order to build a Software as a Service (SaaS) on the implemented core system functionalities.

The first section describes the implementation of the system's common components across the various specific use cases and the others explain the implementation details of the use case-specific modules. After this chapter, it will be clear how the system prototype was implemented and ready for validation and testing phases, which are discussed in chapter 5.

4.1 System's common components

The effectively implemented components of the architecture designed for the proposed system are reported in figure 4.1. The interactions among them are reported in figure 4.2.

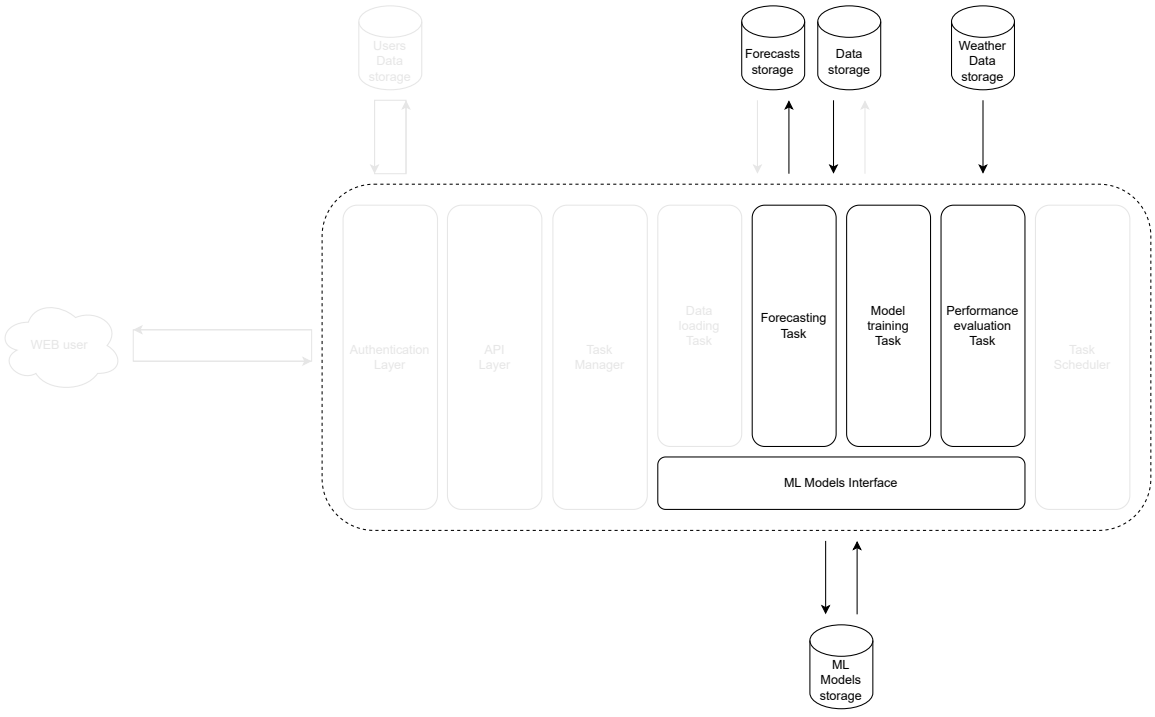


Figure 4.1: The effectively implemented components of the architecture designed for the proposed system.

The ML model interface was implemented using the pickle library[6] for interfacing with the ML models storage for storing the ML models and retrieving the stored ones making them available for prediction and evaluation. The use of the MLflow client[7] was thought of as a possible interface in the

[6] `https://docs.python.org/3/library/pickle.html`
[7] `https://mlflow.org/`

complete system integration for interfacing with the ML models storage.

The model training task trains new models based on available data for the specific use case. The details about the models and their training process for each specific use case are reported in the dedicated sections.

The forecasting task forecasts future data using the available models for the specific use case. The details about the models and their forecasting process for each specific use case are reported in the dedicated sections.

The performance evaluation task evaluates the performance of the available models for the specific use case. The details about the performance evaluation process for each specific use case are reported in chapter 5.

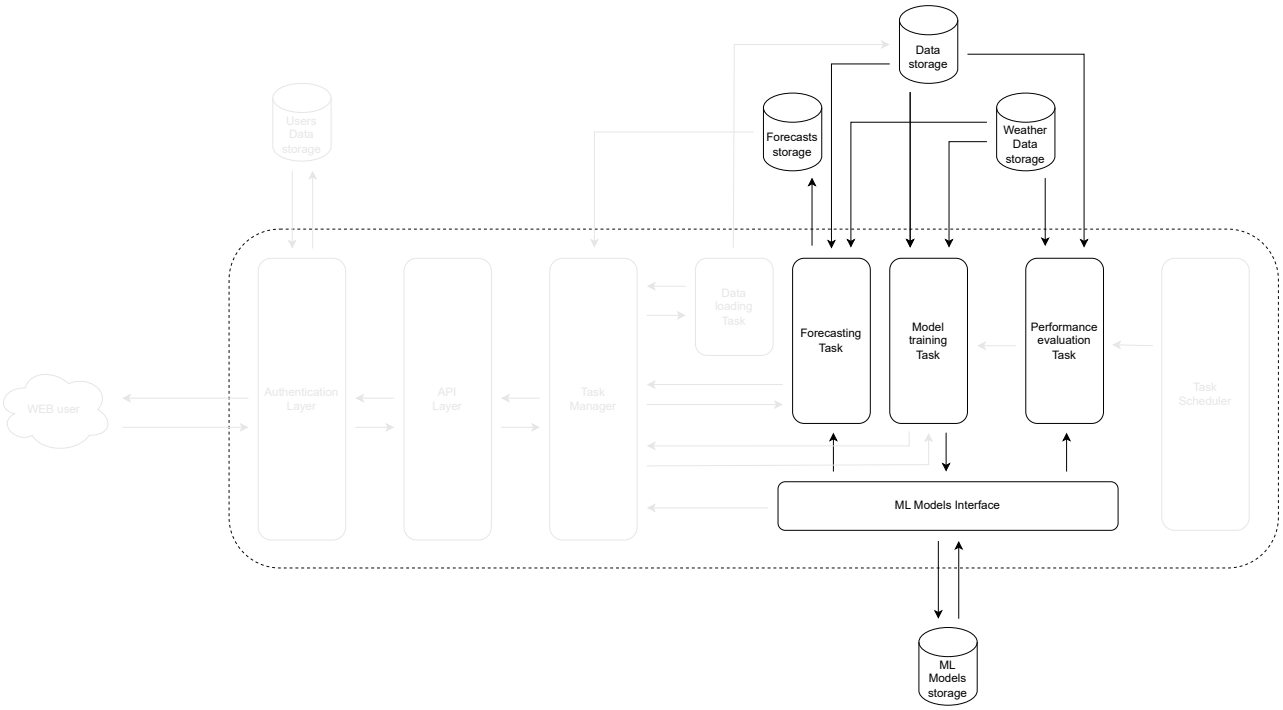


Figure 4.2: The interactions among the effectively implemented components of the architecture designed for the proposed system.

The system interacts with the following external components: data storage, weather data storage, ML models storage, and forecasts storage.

The data storage consists of CSV files containing the data for the different use cases, which are loaded using the pandas library[8]. InfluxDB[9] was thought of as a possible database in the complete system integration for managing the data.

The weather data storage consists of a CSV file and a JSON file containing the weather data, which are loaded using the pandas library. Weather data in the CSV file is manually obtained from Iowa Environmental Mesonet[10] for the weather station in the Murcia airport, and it was used for customers demand forecasting. Weather data in the JSON file is manually obtained from the Weatherbit[11] APIs for a weather station near Murcia airport, these were used for PV plants production forecasting since it provides solar energy data. InfluxDB was thought of as a possible database in the complete system integration for managing the weather data with an automatic download of weather data using Weatherbit APIs.

The ML models storage consists of pickle files containing the ML models. MLflow was thought of as a possible ML model storage in the complete system integration for managing the ML models.

[8] `https://pandas.pydata.org/`
[9] `https://www.influxdata.com/`
[10] `https://mesonet.agron.iastate.edu/request/download.phtml`
[11] `https://www.weatherbit.io/`

The forecasts storage consists of CSV files containing the forecasts, which are stored using the pandas library. InfluxDB was thought of as a possible database in the complete system integration for managing the forecasts.

4.2 Electricity demand forecasting models

The electricity demand forecasting models rely on dedicated data preprocessing which consists of parsing the aggregated consumption data over the customers from a CSV file using the pandas library. Then, this basic data is enhanced with the air temperature, the apparent temperature, and the relative humidity parsed from the weather CSV file, in which the reported value is the average of the value in the last hour. Finally, data is cleaned up by filling in the gaps using the linear interpolation provided by the pandas library, which proved to be effective since there was just some missing data in the weather information. Two possible granularities are considered: hourly and daily. The data is with an hourly granularity, so for the daily granularity, data is also aggregated over the day summing the active incomings of the single hours, and averaging the weather data over the day.

The developed models are some baseline approaches that consider repeating past days and weeks, a SARIMA model, a support vector regressor model, a hist gradient boosting regressor model, an extreme gradient boosting regressor model, a prophet model, a LSTM model, a GRU model, a CNN model, some models derived by the combination of the previous techniques, a TFT model, and an AutoML approach.

The baseline approaches are developed using the NumPy library[12]. In particular, the tile method is used in different ways depending on the considered baseline. For the one-day baseline, the last day of the training active incomings is replicated for the days of the test set. For the one-week baseline, the last week of the training active incomings is replicated for the weeks of the test set.

The SARIMA model is developed using the pmdarima library[13]. The auto_arima method is used to automatically discover the optimal order for the SARIMA model to better fit the training active incomings using the week as the period for seasonal differencing and in case of the hourly granularity this is done for each hour. For the prediction, it simply takes in input the length of the test set and computes the predictions.

Support vector regressor and hist gradient boosting regressor models are developed using the scikit-learn library[14]. The extreme gradient boosting regressor model is developed using the XGBoost library[15]. Support vector regressor, hist gradient boosting regressor, and extreme gradient boosting regressor models take advantage of the following features for the hourly granularity: the hour, the day, the weekday, the month, and the year information of the time instant for which to compute the prediction, the air temperature, the apparent temperature, and the relative humidity in the considered hour. In addition, the active incomings of the past 24 hours and the ones of the same hour considered in the previous 14 days are also used. For the daily granularity instead, the hour information of the date for which to compute the prediction is not considered as a feature, the weather information is obtained as the average over the day, and in place of the active incomings described for the hourly granularity, the daily active incomings of the past 14 days are used. The training is performed using the fit method on the training data for one-step ahead forecasting. The specific models' parameters are discussed in the dedicated section of chapter 5. For the prediction, an recursive strategy is used by forecasting one time instant at a time using weather forecasts, and then uaing the predicted demand data as previous information for the next time instants.

The prophet model is developed using the prophet library[16]. The fit method tasks as input a pandas data frame with just the information related to the time instants and the training active incomings and fits the optimal prophet model. For the prediction, it takes as input a pandas data frame with the time instants for which to compute the predictions and computes them, providing also uncertainty intervals.

[12] `https://numpy.org/`
[13] `http://alkaline-ml.com/pmdarima/`
[14] `https://scikit-learn.org/`
[15] `https://xgboost.readthedocs.io/`
[16] `https://facebook.github.io/prophet/`

LSTM, GRU, and CNN models are developed using the Keras library[17]. These models take advantage of the following features for the hourly granularity: the hour, the day, the weekday, the month, and the year information of the time instant for which to compute the prediction, the air temperature, the apparent temperature, and the relative humidity in the considered hour. In addition, the active incoming of the previous hour is also used. Since these models are able to deal with sequences, a look back at the past 3 hours and at the same hour over the past 14 days is considered and passed as input to the models with the same features as the time instant for which to compute the prediction but with the actual consumption data of the past time instants. For the daily granularity instead, the hour of the date for which to compute the prediction is not considered as a feature, the weather information is obtained as the average over the day, and the considered look back is at the past 14 days. The training is performed using the fit method on the training data for one-step ahead forecasting. The specific models' architecture and parameters are discussed in the dedicated section of chapter 5. For the prediction, an recursive strategy is used by forecasting one time instant at a time using weather forecasts, and then uaing the predicted demand data as previous information for the next time instants.

The models derived by the combination of the previous techniques consist of the combination of one-day baseline, one-week baseline, and prophet with possibly one of LSTM, GRU, or CNN. Further details of the optimal combinations are discussed in the dedicated section of chapter 5.

The TFT model is developed using the PyTorch Forecasting library[18]. A TimeSeriesDataSet is constructed using the following features for the hourly granularity: the time index, the hour, the day, the weekday, the month, and the year information of the time instant for which to compute the prediction, the air temperature, the apparent temperature, and the relative humidity in the considered hour. For the daily granularity instead, the hour of the date for which to compute the prediction is not considered as a feature, and the weather information is obtained as the average over the day. The training is performed using the fit method of the Trainer object provided by the PyTorch Lightning library[19] on the training data. The specific models' parameters are discussed in the dedicated section of chapter 5. For the prediction, it takes as input a TimeSeriesDataSet with the time instants for which to compute the predictions and computes them.

The AutoML approach is based on the TimeSeriesForecastingTask of the Auto-PyTorch library[20]. The AutoML approach takes advantage of the following features for the hourly granularity: the hour, the day, the weekday, the month, and the year information of the time instant for which to compute the prediction, the air temperature, the apparent temperature, and the relative humidity in the considered hour. For the daily granularity instead, the hour of the date for which to compute the prediction is not considered as a feature, and the weather information is obtained as the average over the day. The search method is used to search for the best pipeline configuration for the given training data optimizing the machine learning models and building an ensemble out of them. For the prediction, it takes as input a pandas data frame with the time instants for which to compute the predictions and computes them.

4.3 Consumption baseline forecasting models

The consumption baseline forecasting models rely on dedicated data preprocessing which consists of parsing single customer consumption data from a CSV file using the pandas library. Then, this basic data is enhanced with the air temperature, the apparent temperature, and the relative humidity parsed from the weather CSV file, in which the reported value is the average of the value in the last hour. Finally, data is cleaned up by filling in the gaps using the linear interpolation provided by the pandas library, which proved to be effective since there was just some missing data in the weather information. Three possible granularities are considered: hourly, daily, and by tariff. The data is with an hourly granularity, so for the daily granularity, data is also aggregated over the day summing the active incomings of the single hours, and averaging the weather data over the day. The other

[17] `https://keras.io/`
[18] `https://pytorch-forecasting.readthedocs.io/`
[19] `https://www.pytorchlightning.ai/index.html`
[20] `https://automl.github.io/Auto-PyTorch/master/`

possible granularity consists in aggregating the data by tariff, in fact, the hourly tariff proposed by MIWenergía is different based on the day and the hour of the day. For the considered residential users the possible tariffs are 3:

1. Working days from 10 AM to 2 PM, and from 6 PM to 10 PM;
2. Working days from 8 AM to 10 AM, from 2 PM to 6 PM, and from 10 PM to midnight;
3. Saturday, Sunday, National Holidays, and working days from midnight to 8 AM.

For the tariff granularity, data is aggregated over the day summing the active incomings of the single hours for each tariff, and averaging the weather data over the tariff ranges.

The developed models are some baseline approaches that consider repeating past days and weeks, a SARIMA model, a support vector regressor model, a hist gradient boosting regressor model, an extreme gradient boosting regressor model, a prophet model, a LSTM model, a GRU model, a CNN model, a TFT model, and an AutoML approach.

The baseline approaches are developed using the NumPy library. In particular, the tile method is used in different ways depending on the considered baseline. For the one-day baseline, the last day of the training active incomings is replicated for the days of the test set. For the one-week baseline, the last week of the training active incomings is replicated for the weeks of the test set. The same was also done considering the average of the previous 4 weeks and of the previous 12 weeks of the training active incomings.

The SARIMA model is developed using the pmdarima library. The auto_arima method is used to automatically discover the optimal order for the SARIMA model to better fit the training active incomings using the week as the period for seasonal differencing and in case of the hourly granularity this is done for each hour. For the prediction, it simply takes in input the length of the test set and computes the predictions.

Support vector regressor and hist gradient boosting regressor models are developed using the scikit-learn library. The extreme gradient boosting regressor model is developed using the XGBoost library. Support vector regressor, hist gradient boosting regressor, and extreme gradient boosting regressor models take advantage of the following features for the hourly granularity: the hour, the day, the weekday, the month, and the year information of the time instant for which to compute the prediction, the air temperature, the apparent temperature, and the relative humidity in the considered hour. In addition, the active incomings of the past 24 hours and the ones of the same hour considered in the previous 14 days are also used. For the daily and tariff granularities instead, the hour information of the date for which to compute the prediction is not considered as a feature, the weather information is obtained as the average over the day or tariff ranges, and in place of the active incomings described for the hourly granularity, the daily or tariff active incomings of the past 14 days are used. The training is performed using the fit method on the training data for one-step ahead forecasting. The specific models' parameters are discussed in the dedicated section of chapter 5. For the prediction, an recursive strategy is used by forecasting one time instant at a time using weather forecasts, and then uaing the predicted demand data as previous information for the next time instants.

The prophet model is developed using the prophet library. The fit method tasks as input a pandas data frame with just the information related to the time instants and the training active incomings and fits the optimal prophet model. For the prediction, it takes as input a pandas data frame with the time instants for which to compute the predictions and computes them, providing also uncertainty intervals.

LSTM, GRU, and CNN models are developed using the Keras library. These models take advantage of the following features for the hourly granularity: the hour, the day, the weekday, the month, and the year information of the time instant for which to compute the prediction, the air temperature, the apparent temperature, and the relative humidity in the considered hour. In addition, the active incoming of the previous hour is also used. Since these models are able to deal with sequences, a look back at the past hour and at the same hour over the past 14 days is considered and passed as input to the models with the same features as the time instant for which to compute the prediction but with the actual consumption data of the past time instants. For the daily and tariff granularities instead,

the hour of the date for which to compute the prediction is not considered as a feature, the weather information is obtained as the average over the day or tariff ranges, and the considered look back is at the past 14 days. The training is performed using the fit method on the training data for one-step ahead forecasting. The specific models' architecture and parameters are discussed in the dedicated section of chapter 5. For the prediction, an recursive strategy is used by forecasting one time instant at a time using weather forecasts, and then uaing the predicted demand data as previous information for the next time instants.

The TFT model is developed using the PyTorch Forecasting library. A TimeSeriesDataSet is constructed using the following features for the hourly granularity: the time index, the hour, the day, the weekday, the month, and the year information of the time instant for which to compute the prediction, the air temperature, the apparent temperature, and the relative humidity in the considered hour. For the daily and tariff granularities instead, the hour of the date for which to compute the prediction is not considered as a feature, and the weather information is obtained as the average over the day or tariff ranges. The training is performed using the fit method of the Trainer object provided by the PyTorch Lightning library on the training data. The specific models' parameters are discussed in the dedicated section of chapter 5. For the prediction, it takes as input a TimeSeriesDataSet with the time instants for which to compute the predictions and computes them.

The AutoML approach is based on the TimeSeriesForecastingTask of the Auto-PyTorch library. The AutoML approach takes advantage of the following features for the hourly granularity: the hour, the day, the weekday, the month, and the year information of the time instant for which to compute the prediction, the air temperature, the apparent temperature, and the relative humidity in the considered hour. For the daily and tariff granularities granularity instead, the hour of the date for which to compute the prediction is not considered as a feature, and the weather information is obtained as the average over the day or tariff ranges. The search method is used to search for the best pipeline configuration for the given training data optimizing the machine learning models and building an ensemble out of them. For the prediction, it takes as input a pandas data frame with the time instants for which to compute the predictions and computes them.

4.4 Electricity production forecasting models

The electricity production forecasting models rely on dedicated data preprocessing which consists of parsing single PV plant production data from a CSV file using the pandas library and aggregating the single PV plant data to obtain the aggregated production data over the PV plants. Then, this basic data is enhanced with the air temperature, the apparent temperature, the relative humidity, the wind speed, the wind direction, the pressure altimeter, the visibility, the sky coverage, the diffuse horizontal irradiance, the direct normal irradiance, the global horizontal irradiance, the solar radiation, the UV index, the solar elevation angle, and the solar azimuth angle parsed from the weather JSON file, in which the reported value is the average of the value in the last hour. Finally, data is cleaned up by filling in the gaps using the linear interpolation provided by the pandas library, which proved to be effective since there was just some missing data in the weather information. Only the hourly granularity is considered since PV plants are highly correlated with weather data and the aggregation over the day loses part of this correlation. The data is with an hourly granularity and the target of the predictions is the mean percentage of production, which is calculated as the division of the total produced energy by the total power of the PV plants. This allows to have a bounded value from 0 to 100 from which it is possible to obtain the total produced energy simply by multiplying it by the total power of the PV plants.

The developed models are a baseline approach that considers repeating past days, an ARIMA model, a support vector regressor model, a hist gradient boosting regressor model, an extreme gradient boosting regressor model, a prophet model, a LSTM model, a GRU model, a CNN model, some models derived by the combination of the previous techniques, a TFT model, and an AutoML approach.

The baseline approach is developed using the NumPy library. In particular, the tile method is used. For the one-day baseline, the last day of the training mean percentages of production is replicated for the days of the test set.

The ARIMA model is developed using the pmdarima library. The auto_arima method is used

to automatically discover the optimal order for the ARIMA model to better fit the training mean percentages of production and this is done for each hour. For the prediction, it simply takes in input the length of the test set and computes the predictions.

Support vector regressor and hist gradient boosting regressor models are developed using the scikit-learn library. The extreme gradient boosting regressor model is developed using the XGBoost library. Support vector regressor, hist gradient boosting regressor, and extreme gradient boosting regressor models take advantage of the following features: the hour, the day, the weekday, the month, and the year information of the time instant for which to compute the prediction, the air temperature, the apparent temperature, the relative humidity, the diffuse horizontal irradiance, the direct normal irradiance, the global horizontal irradiance, the solar radiation, the UV index, and the solar elevation angle in the considered hour. In addition, the mean percentages of production of the past 24 hours and the one of the same hour considered in the previous 14 days are also used. The training is performed using the fit method on the training data for one-step ahead forecasting. The specific models' parameters are discussed in the dedicated section of chapter 5. For the prediction, an recursive strategy is used by forecasting one time instant at a time using weather forecasts, and then uaing the predicted production data as previous information for the next time instants.

The prophet model is developed using the prophet library. The fit method tasks as input a pandas data frame with just the information related to the time instants and the training mean percentages of production and fits the optimal prophet model. For the prediction, it takes as input a pandas data frame with the time instants for which to compute the predictions and computes them, providing also uncertainty intervals.

LSTM, GRU, and CNN models are developed using the Keras library. These models take advantage of the following features: the hour, the day, the weekday, the month, and the year information of the time instant for which to compute the prediction, the air temperature, the apparent temperature, the relative humidity, the diffuse horizontal irradiance, the direct normal irradiance, the global horizontal irradiance, the solar radiation, the UV index, and the solar elevation angle in the considered hour. In addition, the mean percentage of production of the previous hour is also used. Since these models are able to deal with sequences, a look back at the past hour and at the same hour over the past 14 days is considered and passed as input to the models with the same features as the time instant for which to compute the prediction but with the actual production data of the past time instants. The training is performed using the fit method on the training data for one-step ahead forecasting. The specific models' architecture and parameters are discussed in the dedicated section of chapter 5. For the prediction, an recursive strategy is used by forecasting one time instant at a time using weather forecasts, and then uaing the predicted production data as previous information for the next time instants.

The models derived by the combination of the previous techniques consist of the combination of one-day baseline and prophet with possibly one of LSTM, GRU, or CNN. Further details of the optimal combinations are discussed in the dedicated section of chapter 5.

The TFT model is developed using the PyTorch Forecasting library. A TimeSeriesDataSet is constructed using the following features: the time index, the hour, the day, the weekday, the month, and the year information of the time instant for which to compute the prediction, the air temperature, the apparent temperature, the relative humidity, the diffuse horizontal irradiance, the direct normal irradiance, the global horizontal irradiance, the solar radiation, the UV index, and the solar elevation angle in the considered hour. The training is performed using the fit method of the Trainer object provided by the PyTorch Lightning library[21] on the training data. The specific models' parameters are discussed in the dedicated section of chapter 5. For the prediction, it takes as input a TimeSeriesDataSet with the time instants for which to compute the predictions and computes them.

The AutoML approach is based on the TimeSeriesForecastingTask of the Auto-PyTorch library. The AutoML approach takes advantage of the following features: the hour, the day, the weekday, the month, and the year information of the time instant for which to compute the prediction, the air temperature, the apparent temperature, the relative humidity, the diffuse horizontal irradiance, the direct normal irradiance, the global horizontal irradiance, the solar radiation, the UV index, and

[21] `https://www.pytorchlightning.ai/index.html`

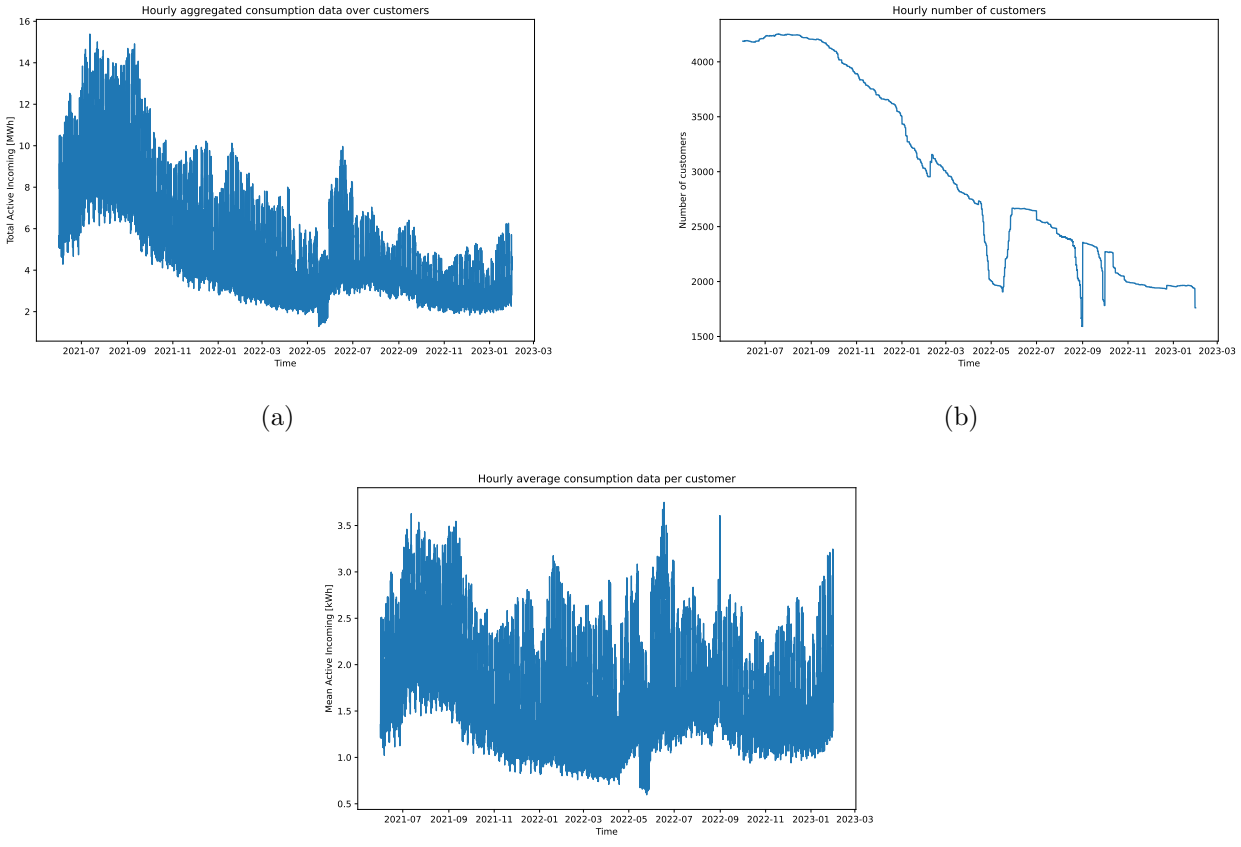
the solar elevation angle in the considered hour. The search method is used to search for the best pipeline configuration for the given training data optimizing the machine learning models and building an ensemble out of them. For the prediction, it takes as input a pandas data frame with the time instants for which to compute the predictions and computes them.

5 Performance Evaluation

In this chapter, the proposed system is validated and the performance of the models for the different use cases is evaluated. The first section presents the datasets provided by MIWenergía[22]. Subsequently, the adopted evaluation methodology is described. Finally, the evaluation of the performance of the models for the different use cases is presented. After this chapter, it will be clear how the system has been validated and what the performance achieved by the proposed system is.

5.1 MIWenergía datasets

In this section, the MIWenergía datasets are described. They provided 3 kinds of datasets: aggregated consumption data from all their customers, consumption data from single customers, and production data from PV plants.



(a) (b)

(c)

Figure 5.1: The graphical representation of the hourly (a) aggregated consumption over customers, (b) number of customers, and (c) average consumption per customer.

The aggregated consumption data from all their customers consists of hourly aggregated consumption data from June 2021 to January 2023 for a total of 14617 entries. The graphical representation of the hourly aggregated consumption data is reported in figure 5.1a. The number of customers is

[22] https://www.miwenergia.com/

variable, with a maximum of 4253, a minimum of 1591, a mean of 2988, and a standard deviation of 855. The graphical representation of the hourly number of customers is reported in figure 5.1b. It was thought to normalize the consumption on the basis of the number of customers in order to study the average consumption per user, as illustrated in figure 5.1c, and then multiply by the number of customers. However, this was not feasible since often this value changes significantly without reflecting on the consumption data, this information was deemed unreliable and not utilized. Despite this limitation, the average consumption per customer still provided valuable insights into consumption patterns. Specifically, it suggested the presence of two consumption peaks: one in the summer, likely attributed to air conditioning systems, and a second peak in the winter, likely caused by heating systems.

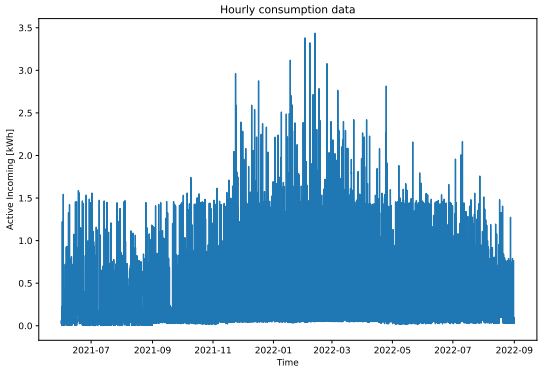
The consumption data from single customers consists of hourly aggregated consumption data of three customers:

1. from June 2021 to August 2022 for a total of 10952 total entries and it is represented in figure 5.2a;

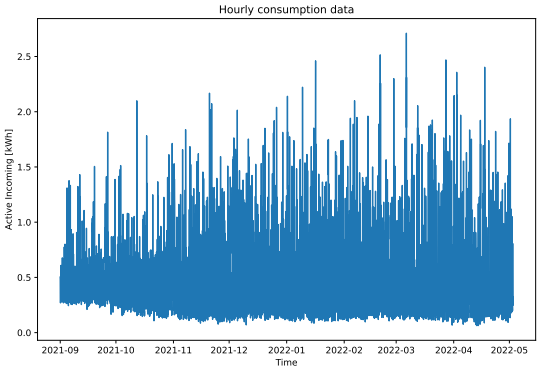
2. from September 2021 to May 2022 for a total of 5855 total entries and it is represented in figure 5.2b;

3. from September 2021 to August 2022 for a total of 8760 total entries and it is represented in figure 5.2c.

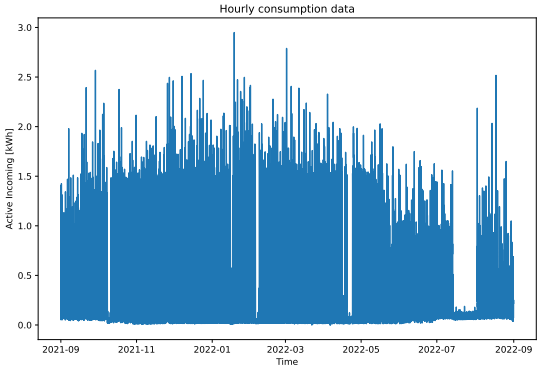
The total entries for the three customers are 25567 in the overall period from June 2021 to August 2022.



(a) First customer.



(b) Second customer.



(c) Third customer.

Figure 5.2: The graphical representation of the consumption data of the three customers.

Only consumption data for three customers were provided, as electricity consumption data of customers is considered personal data as stated in the Directive (EU) 2019/944 of the European

Parliament[23], which establishes common rules for the internal market for electricity. As such, the data falls under the scope of the General Data Protection Regulation (GDPR)[24], which requires explicit consent for processing personal data. Therefore, it is likely that the provided data came from MIWenergía's internal employees who consented to participate in the research. In fact, the provided data were needed for a technical feasibility study. However, to identify consumption patterns among a broader population, data from more customers would be required.

By analyzing the data plots, it can be observed that the consumption patterns of the three customers are quite distinct. For instance:

- The first customer's consumption is consistently below 1.5 kWh throughout the year except for the winter season;
- The second customer shows almost the same consumption pattern;
- The third customer has a dense series with some gaps with very low consumption in parts of the year, probably due to time periods away from home.

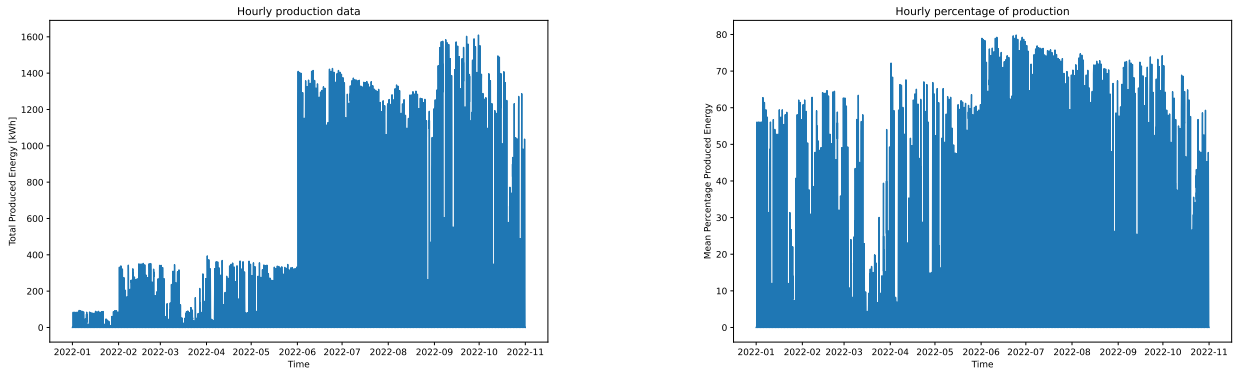
The production data from 8 PV plants consists of hourly aggregated production data:

1. from January 2022 to October 2022 for a total of 7296 total entries and it has a nominal power of 149.75 kW;
2. from February 2022 to October 2022 for a total of 6552 total entries and it has a nominal power of 237.6 kW;
3. from February 2022 to October 2022 for a total of 6552 total entries and it has a nominal power of 158.4 kW;
4. from June 2022 to October 2022 for a total of 3576 total entries and it has a nominal power of 1240 kW;
5. from September 2022 to October 2022 for a total of 1465 total entries and it has a nominal power of 126.2 kW;
6. from September 2022 to October 2022 for a total of 1465 total entries and it has a nominal power of 113 kW;
7. from September 2022 to October 2022 for a total of 1465 total entries and it has a nominal power of 45 kW;
8. from September 2022 to October 2022 for a total of 1465 total entries and it has a nominal power of 100 kW;

The total entries for the 8 PV plants are 29836 total entries in the overall period from January 2022 to October 2022, aggregating over plants the resulting entries are 7296. The graphical representation of the hourly aggregated total and mean percentage production data are reported respectively in figure 5.3a and figure 5.3b.

The same weather data from the same provider and the same weather station were used for all the tasks. While it is reasonable to assume that some weather conditions, such as clouds, may vary and affect consumption and production differently, this approach was taken since the customers and the PV plants were in the same area, with a maximum distance of around 100 km between them, and also because the value of the weather parameters is an average of the values recorded in the past hour. Therefore, using the same weather data is deemed appropriate for the analysis at hand, given that it reflects the weather conditions in the area over a broad time frame.

[23] `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32019L0944`
[24] `https://gdpr-info.eu/`



(a) (b)

Figure 5.3: The graphical representation of the hourly aggregated (a) total and (b) mean percentage production data.

5.2 Evaluation methodology

The evaluation methodology was based on two relevant error metrics: Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE). These are standard and widely used metrics in time series forecasting for different use cases, as reported in many articles and books such as [9, 26, 50]. The MAPE is defined as $\text{MAPE}(y, \hat{y}) = \frac{100\%}{N} \sum_{i=0}^{N-1} \frac{|y_i - \hat{y}_i|}{|y_i|}$. It is the most relevant error metric for all the tasks since it is a percentage-based error metric that takes into account the magnitude of the errors relative to the actual values. The MAE is defined as $\text{MAE}(y, \hat{y}) = \frac{\sum_{i=0}^{N-1} |y_i - \hat{y}_i|}{N}$. It is the most suitable error metric in consumption baseline forecasting where there is a high variability from very low to high values, and electricity production forecasting where there is a significant number of zeros when the sun is absent.

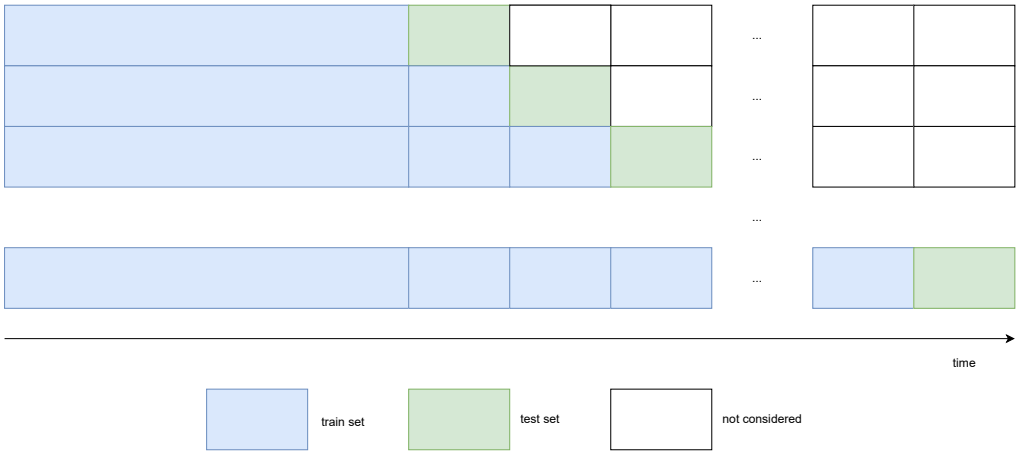


Figure 5.4: The schematic representation of the blocked k-fold cross-validation adopted.

For assessing the model performance when dealing with time series data, the traditional cross-validation techniques are not suitable as they assume that the data points are independent and identically distributed (i.i.d.), which is not the case in time series data, as also reported in the chapter 2 by the following articles [14, 19]. In fact, in time series data the order of the observations matters, and there are temporal dependencies between the observations. Therefore, a more appropriate technique for evaluating time series models is blocked k-fold cross-validation. The basic idea of blocked k-fold cross-validation is to split the data considering multiple training and test sets, where the training set only includes data from the past and the test set includes data from the future. This simulates the real-world scenario where we want to make predictions about the future based on past data. In figure 5.4, the schematic representation of how the blocked k-fold cross-validation was performed is

reported, 12 splits were used with a test size depending on the specific use case. The training set is increased by adding the previous test elements at every successive evaluation.

Another technique, reported in [19] is the repeated Holdout Out-of-sample tested in multiple testing periods with a Monte Carlo simulation using 70% of the total observations of the time series in each test. For each period, a random point is picked from the time series. The previous window comprising 60% of the time series is used for training and the following window of 10% of the time series is used for testing. In the paper, it was stated that the approach provided the most accurate estimates when the time series are non-stationary. However, for having the same evaluation mechanisms on all the models and simulating the fact that the model starts with limited data and then the training amount increases over time to obtain better performance, block validation was used using the TimeSeriesSplit provided by the scikit-learn library. Moreover, the considered data are without a strong trend as reported in the dedicated sections of the specific use cases where the data and the model forecasts are analyzed.

In addition to the blocked k-fold cross-validation results also the results on the last test split using the rest as training are reported. This was done since it provides insight on which could be the performance of the models in forecasting the near future data with the currently available training data. A table summarizing the metrics and the evaluation mechanisms adopted for the use cases is reported in table 5.1.

Use case	**Metrics**	**Evaluation mechanisms**
Electricity demand forecasting	MAPE	blocked k-fold cross-validation and test on the last split
Electricity production forecasting	MAE	blocked k-fold cross-validation and test on the last split
Consumption baseline forecasting	MAPE and MAE	blocked k-fold cross-validation and test on the last split

Table 5.1: Table summarizing the metrics and the evaluation mechanisms adopted for the use cases.

This evaluation methodology was used at training time to study the performance over time. At forecasting time, the performance on the last time slot was treated with more attention as the most relevant for the energy retailers since the most related to the performance on imminent future data. The combinations of the models were done just at forecasting time on the last time slot using the average of some approaches for studying whether there could be a beneficial effect on the forecasts and compensation of error between the different approaches. The same was also done for the AutoML approach, in fact, it was tested just at forecasting time on the last time slot for comparison to the tested models. It was included as a baseline to show how a general-purpose framework can perform in specific use cases compared to specific architectures designed for the task.

As explained in chapter 4, for all the use cases, there is a difference between training and forecasting time for the ML (support vector regressor, hist gradient boosting regressor, and extreme gradient boosting regressor) and DL (LSTM, GRU, and CNN) models. In particular, for these models, a recursive strategy to generate multi-step forecasts was adopted. This approach is a standard classical technique covered in various textbooks, including [70]. It offers advantages such as simplicity in understanding, clear temporal dependency as previous forecasts are used as input for subsequent ones, flexibility to be applied to different models, and adaptability to various forecasting horizons, which is crucial when integrating it into a SaaS solution. However, there are also disadvantages associated with this approach. Error propagation can occur, and this can be observed in the forecasting results, where small inaccuracies in initial predictions can accumulate and lead to larger errors over time. Sub-optimality is another drawback, as the optimality achieved for one-step forecasting may not generalize well to multi-step forecasting. Additionally, the computational complexity is high as the model needs to be run for each individual step. In contrast, the other approaches: baselines, ARIMA/SARIMA, Prophet, and TFT directly handle the concept of multi-step prediction and present no difference between training and forecasting time.

5.3 Electricity demand forecasting

The aggregated consumption data over the customers is analyzed to get some descriptive analytics before finding adequate models to forecast the demand. The time series decomposition using an additive model of the hourly aggregated consumption over the customers considering as period of the time series a week is reported in figure 5.5. It showed a considerable amount of noise, comparable to seasonality in magnitude. The trend showed 2 peaks during the summer season, the first being more emphasized since it also corresponds to a peak in the number of users.

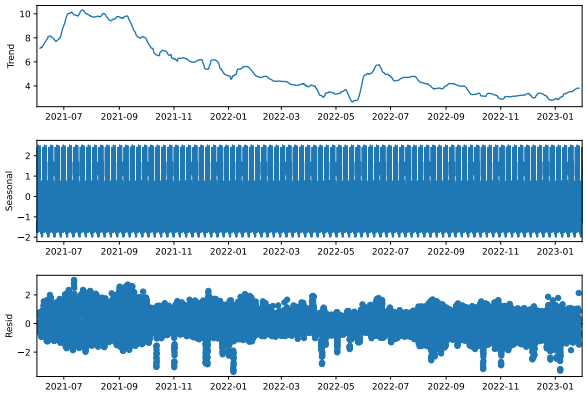


Figure 5.5: The time series decomposition of the hourly aggregated consumption over the customers considering as period of the time series a week.

The auto-correlation of the hourly aggregated consumption over the customers is reported in figure 5.6. It shows a high auto-correlation value in the close time lags and also at every 24 hours, along with an even higher value at a one-week distance. This indicates that the consumption data from the closest time lags, particularly up to three closest ones, as well as those corresponding to the same hour in the preceding days and even better in the preceding weeks, may be valuable features for predicting a time instant's demand. A reasonable balance can be achieved by incorporating the consumption data from the past 14 days.

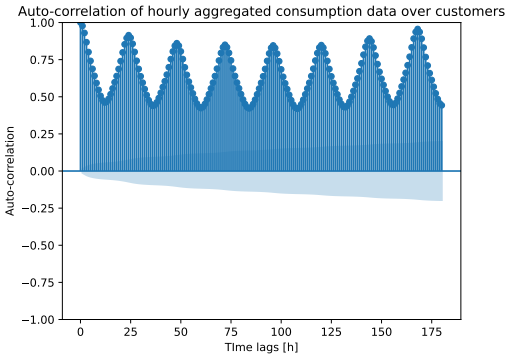


Figure 5.6: The auto-correlation of the hourly aggregated consumption over the customers.

The coefficients given by the Fourier transform for the hourly aggregated consumption over the customers are reported in figure 5.7. The graphical representation shows 2 main frequencies, one representing the weekly periodicity and one the daily periodicity. Other minor peaks are present mostly at multiples of the 1/week frequency.

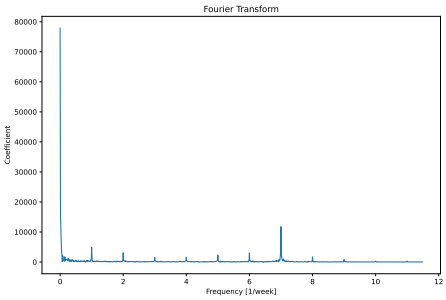


Figure 5.7: The coefficients given by the Fourier transform for the hourly aggregated consumption over the customers.

The daily aggregated consumption over the customers is reported in figure 5.8. Aggregating the data on a daily basis, it is possible to more clearly observe the weekly pattern of consumption, which exhibits an increase on weekdays and a decrease on weekends.

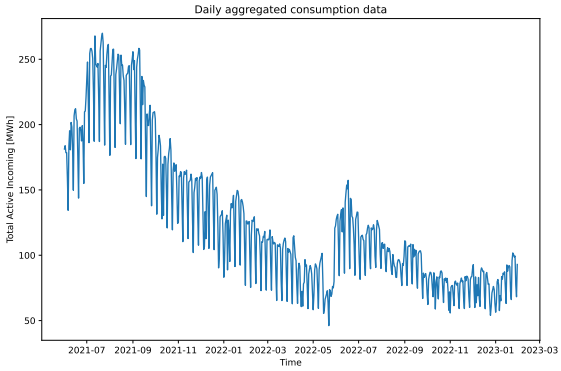


Figure 5.8: The daily aggregated consumption over the customers.

Figure 5.9 shows the time series decomposition of the daily aggregated consumption over the customers, using an additive model with a period of one week. The decomposition is consistent with the hourly aggregated consumption over the customers.

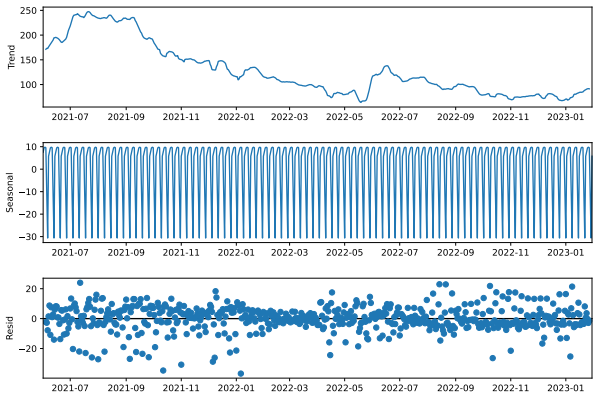


Figure 5.9: The time series decomposition of the daily aggregated consumption over the customers considering as period of the time series a week.

The auto-correlation of the daily aggregated consumption over the customers is reported in figure 5.10. It shows a high auto-correlation value in the closest time lag, with an even greater value at every week lag. This suggests that useful features, since highly correlated, for predicting a time

instant's demand may be the closest ones, but also those from the same day of the week in the previous weeks. A good trade-off can be achieved by considering the consumption data in the previous 14 days.

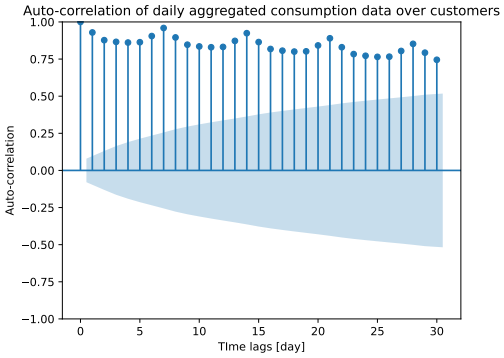


Figure 5.10: The auto-correlation of the daily aggregated consumption over the customers.

The coefficients given by the Fourier transform for the daily aggregated consumption over the customers are reported in figure 5.11. The graphical representation shows 1 main frequency representing the weekly periodicity and other minor peaks at multiples of the 1/week frequency.

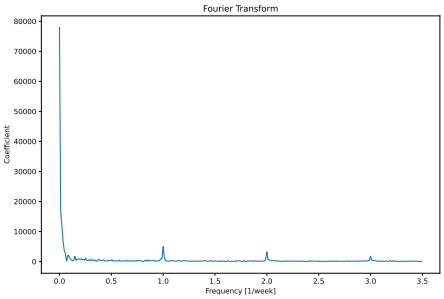


Figure 5.11: The coefficients given by the Fourier transform for the daily aggregated consumption over the customers.

Basic data is enhanced with the air temperature, the apparent temperature, and the relative humidity since they are considered the only weather features capable of influencing customers' energy consumption. To assess the relationship between these weather variables and the aggregated consumption over the customers, two correlation coefficients were used: Pearson's correlation coefficient and Spearman's rank correlation coefficient. Pearson's correlation coefficient measures the strength of the linear relationship between two variables, while Spearman's rank correlation coefficient measures the strength of the monotonic relationship. The pearsonr and spearmanr methods of the SciPy library[25] were used to compute the correlation with weather data. The results showed that the hourly aggregated consumption over the customers had:

- a Pearson correlation coefficient of 0.4480 and a Spearman's rank correlation coefficient of 0.4378 with respect to the air temperature;
- a Pearson correlation coefficient of 0.4475 and a Spearman's rank correlation coefficient of 0.4368 with respect to the apparent temperature;
- a Pearson correlation coefficient of -0.2689 and a Spearman's rank correlation coefficient of -0.3173 with respect to the relative humidity.

[25] `https://scipy.org/`

It can be noticed that both the coefficients indicate a moderate correlation between the weather variables and consumption, this suggests that incorporating weather data into the prediction models could be useful.

The results showed that the daily aggregated consumption over the customers had:

- a Pearson correlation coefficient of 0.3898 and a Spearman's rank correlation coefficient of 0.3119 with respect to the air temperature;
- a Pearson correlation coefficient of 0.3823 and a Spearman's rank correlation coefficient of 0.3114 with respect to the apparent temperature;
- a Pearson correlation coefficient of -0.0908 and a Spearman's rank correlation coefficient of -0.1547 with respect to the relative humidity.

It can be noticed that both the coefficients decreased considering the daily data meaning that the mean weather data over the day is less correlated to the daily aggregated consumption over the customers, compared to the hourly granularity.

After this data analysis, the specific parameters used in the models can be explained more in detail. The parameters for the different models were tested and verified on training data to result in the best-performing models, this was done by trying different values and configurations. The baseline approaches are built considering the repetition of past days and weeks since data presents a high correlation with that time instants and could lead to a reasonable baseline performance to achieve. The SARIMA model considers the week as the period for seasonal differencing since data presents a high correlation with that time instants and the model can try to take advantage of the weekly seasonality. The support vector regressor model uses a radial basis function kernel with a configuration of the C parameter (regularization parameter of squared l2 penalty) to 1.0 to penalize the complexity of the model, and the epsilon parameter defining the epsilon-tube for no penalty to 0.1 as a trade-off between accuracy and generalization. The hist gradient boosting regressor model uses as loss the absolute error since this was the supported metric closer to our evaluation metric. The learning rate was set to 0.1, 100 estimators were used and l2 regularization was not set. The parameters for tree definition are 31 as the maximum number of leaves, depth is not constrained, 20 as the minimum number of samples per leaf, and 255 as the maximum number of bins. For the extreme gradient boosting regressor model, tree-based models were used and the construction algorithm is automatically chosen by heuristic to choose the fastest method. It uses a squared loss since this was the supported metric closer to our evaluation metric. The learning rate was set to 0.3, 100 estimators were used and l2 regularization was set with a weight of 1.0. The parameters for tree definition are 6 as the maximum depth, and the maximum number of leaves is not constrained. The Prophet model was built keeping the default parameters for automatically detecting seasonalities and best fitting the training data.

The LSTM model was designed as a 2-layer model with 24 Bidirectional LSTM units in the first layer which use rectified linear unit (ReLU) activation function and sigmoid as recurrent activation function with both dropout and recurrent dropout of 0.02. Batch normalization is then applied before entering the second layer composed of 16 Bidirectional LSTM units which use ReLU activation function and sigmoid as recurrent activation function without dropout and recurrent dropout. The output of the layer goes inside a dense unit to output the final prediction. The model is trained using mean absolute percentage error as loss and Nadam (a version of Adam integrating the concept of Nesterov momentum) as optimizer with a learning rate of 0.005.

The GRU model was designed as a 3-layer model with 24 Bidirectional GRU units in the first layer which use ReLU activation function and sigmoid as recurrent activation function with both dropout and recurrent dropout of 0.02. Subsequently, the second layer is composed of 16 Bidirectional GRU units which use ReLU activation function and sigmoid as recurrent activation function without dropout and recurrent dropout. The output of the second layer goes inside a third dense layer composed of 4 units with ReLU activation function before entering the final dense unit to output the final prediction. The model is trained using mean absolute percentage error as loss and Nadam as optimizer with a learning rate of 0.005.

The CNN model was designed as a 3-layer model with a first layer composed of 24 1D Convolutional units with a kernel size of 5 which uses ReLU activation function. A 1D max polling operation is applied before entering the second layer composed of 16 1D Convolutional units with a kernel size of 3 which use ReLU activation function. A 1D max polling operation is applied before the flattening operation and entering the third dense layer composed of 4 units with ReLU activation function. The output of the layer goes inside a dense unit to output the final prediction. The model is trained using mean absolute percentage error as loss and Nadam as optimizer with a learning rate of 0.005.

The TFT model was configured with 2 LSTM layers with 16 as hidden size, 4 as attention size, and 8 as hidden continuous size. The dropout is set to 0.02. The model is trained using mean absolute percentage error as loss and Nadam as optimizer with a learning rate of 0.005.

12 splits were used for block validation with a test size of a month each time, namely 30 days of data. For the hourly granularity, the models start with 5977 entries as training and predict every time 720 entries until reaching the last prediction instant where the model has a training size of 13897. The results for hourly granularity are reported in table 5.2. Entries are sorted by best MAPE using the last split as the test set. The blocked k-fold cross-validation suggests that the one-week baseline has the best MAPE, but looking at the last split this is not confirmed. This is probably due to the fact that overall the weekly repetition seems to be quite effective over the year, but there are better approaches that with the increase of training data perform better, this can be the case in GRU and CNN since the results on the last split are quite good compared to the mean MAPE obtained over the splits. This can be explained by the fact that, due to their complexity, these models require more data and in the first splits there was not even a single year of data, so they didn't even have the possibility to infer a possible annual seasonality. Results seem also to confirm that not always sophisticated models perform better and some of them are no better than a baseline. Models treated in the literature that are performing well in similar tasks can also not be suited for this specific use case. Data makes a big difference and with a low amount of data, it is difficult for some complex networks to perform well and generalize. Instead, the TFT approach seems to have pretty much the same MAPE over the considered splits confirming the power of this architecture despite the limited data In contrast, the boosters that have a good blocked k-fold MAPE on the last split do not perform well.

Model	**Blocked k-fold cross-validation MAPE**	**Test on the last split MAPE**
TFT	15.00 ± 4.31	14.76
GRU	30.77 ± 14.48	16.82
CNN	34.64 ± 15.76	17.49
One Week Baseline	12.94 ± 4.27	17.55
SARIMA	15.44 ± 4.57	18.47
One Day Baseline	22.86 ± 7.07	20.54
LSTM	34.08 ± 11.92	23.18
HistGradientBoostingRegressor	14.54 ± 5.51	29.33
SVR	61.80 ± 22.63	29.47
Prophet	28.68 ± 7.07	31.62
XGBRegressor	16.88 ± 8.08	37.81

Table 5.2: Table summarizing the results for hourly granularity.

The results for hourly granularity at forecasting time on the last time slot including also the combinations of different techniques and the AutoML approach are reported in table 5.3. The combination of the previous techniques was performed by combining one-day baseline, one-week baseline, and Prophet with possibly one of LSTM, GRU, or CNN with the intent of trying to improve the robustness of DL models with some stable baselines and Prophet model which internally automatically deals with seasonalities, even though for this task it is not performing very well. The AutoML approach uses the mean MAPE forecasting as loss and it is launched for 10 hours with a maximum function evaluation time of 2 hours. Then the best ensemble of the found models is returned and can be used to forecast. In the table, it is also reported the performance on the one-week horizon, namely

7 days of data, from which it is possible to gain more insights on the models' forecasts. Including also combinations and AutoML, there are a few observations that can be done. The combinations of DL models and the one-week baseline are the most beneficial, reducing by many points the MAPE of the techniques on the one-month time horizon. This suggests that it can make sense to investigate the concept of ensemble learning further and optimize this combination at training time to possibly have stronger results at forecasting time. AutoML has great potential and obtains very good results compared to many other models. Potentially providing more total training time it can be further optimized to reach better performance. TFT still dominates on the one-month MAPE but not on the one-week one on which the one-week baseline and many combinations but also SARIMA and hist gradient boosting regressor perform better. This confirmed that the strength of TFT is to have good performance and preserve it also on larger prediction horizons.

Model	**Forecast on the last split one-week MAPE**	**Forecast on the last split one-month MAPE**
TFT	13.71	14.76
GRU + One Week Baseline	11.79	14.79
CNN + One Week Baseline	11.84	15.39
GRU	17.25	16.82
AutoML	14.68	16.96
CNN + One Week Baseline + Prophet	15.43	17.13
CNN + One Day Baseline + One Week Baseline + Prophet	13.60	17.25
GRU + One Day Baseline + One Week Baseline + Prophet	13.25	17.33
GRU + One Week Baseline + Prophet	15.63	17.37
CNN	14.59	17.49
One Week Baseline	10.87	17.55
LSTM + One Week Baseline + Prophet	13.16	17.89
LSTM + One Day Baseline + One Week Baseline + Prophet	12.58	18.10
LSTM + One Week Baseline	14.42	18.11
SARIMA	12.28	18.47
One Day Baseline	14.78	20.54
One Day Baseline + One Week Baseline + Prophet	15.15	20.97
One Week Baseline + Prophet	19.02	22.80
LSTM	20.94	23.18
HistGradientBoostingRegressor	13.00	29.33
SVR	29.06	29.47
Prophet	33.94	31.62
XGBRegressor	17.84	37.81

Table 5.3: Table summarizing the results for hourly granularity at forecasting time.

The forecasts produced by the TFT model for the last split in comparison to the actual values are reported in figure 5.12. It can be noticed that for the first week, it follows the signal very accurately, the second and third weeks are pretty good but it is missing the peaks over the day and the last week is decreasing while the actual values have high peaks. The absolute percentage error of forecasts produced by the TFT model for the last split in comparison to the actual values is reported in figure 5.13. It can be shown that except for some peaks, the percentage error stays below 30%, with

most values around 10%. Finally, the scatter plot representing the error of the forecasts produced by the TFT model for the last split in comparison to the actual values is reported in figure 5.14. This scatter plot demonstrates that there is no bias and can be also noticed that when the predicted values are low also the error is limited, while when the prediction is high there are both high positive and negative errors.

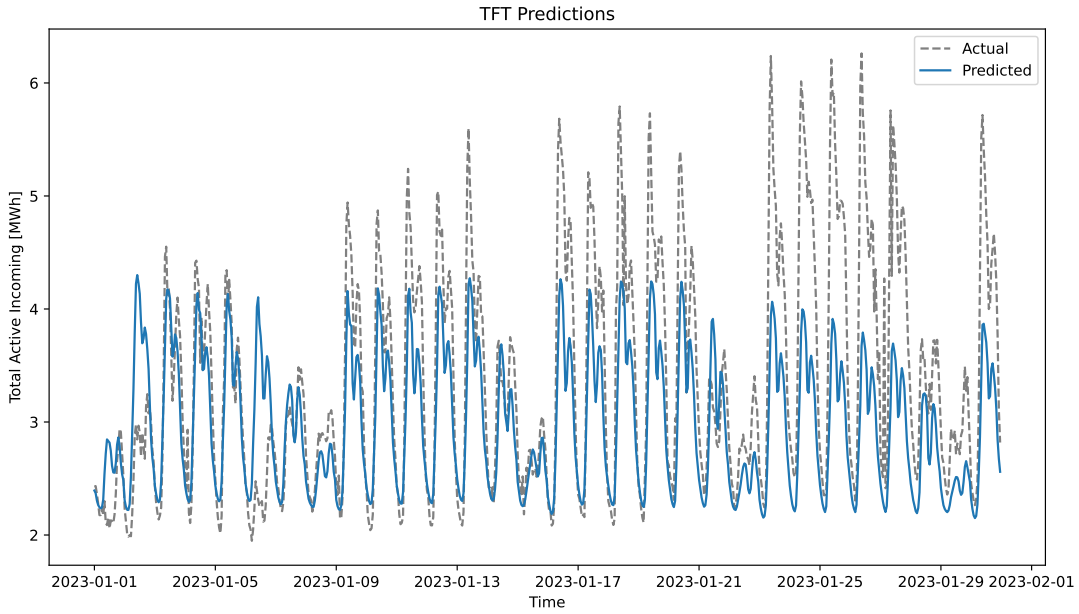


Figure 5.12: The forecasts produced by the temporal fusion transoformer model for the last split in comparison to the actual values.

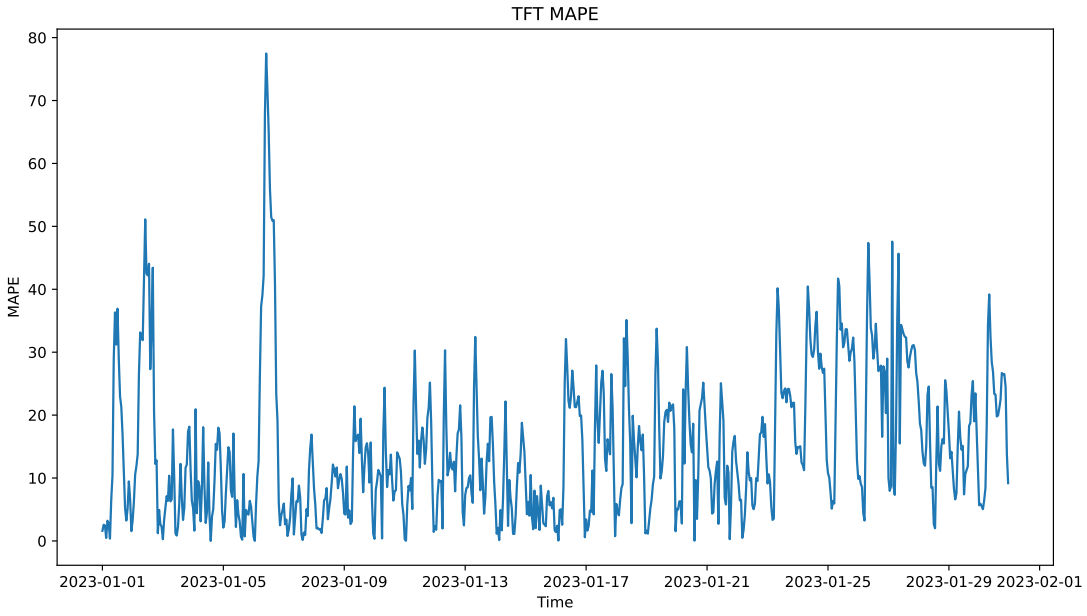


Figure 5.13: The absolute percentage error of forecasts produced by the temporal fusion transoformer model for the last split in comparison to the actual values.

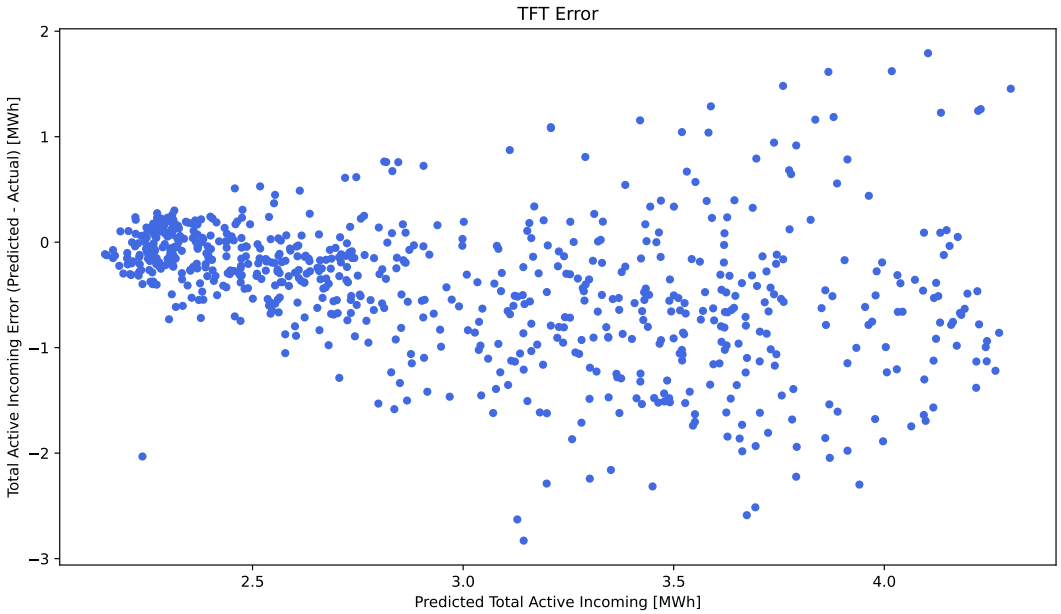


Figure 5.14: The scatter plot representing the error of the forecasts produced by the temporal fusion transoformer model for the last split in comparison to the actual values.

For the daily granularity, the models start with 249 entries as training and predict every time 30 entries until reaching the last prediction instant where the model has a training size of 579. The results for daily granularity are reported in table 5.4. In this case, TFT is no longer the best-performing model, probably due to the fact of 24 times less quantity of data due to the daily granularity. Though, it is still a well-performing model on both blocked k-fold cross-validation and last split results. LSTM and CNN perform better in terms of the last split MAPE but present a high blocked k-fold MAPE, as in the case of the hourly granularity probably that to the increase of data, the results on the last time slots improved in a significant way. Also, the boosters present very good results on the last time split and also on the blocked k-fold validation showing consistency over time and good results also with few training data.

Model	**Blocked k-fold cross-validation MAPE**	**Test on the last split MAPE**
CNN	50.55 ± 36.81	7.49
XGBRegressor	12.96 ± 4.36	7.65
HistGradientBoostingRegressor	13.02 ± 5.54	9.81
LSTM	29.60 ± 13.81	14.51
TFT	13.73 ± 5.06	14.88
One Week Baseline	11.50 ± 4.78	18.40
SARIMA	11.92 ± 7.04	19.63
GRU	28.76 ± 29.26	20.47
One Day Baseline	22.80 ± 8.21	21.51
Prophet	19.06 ± 10.81	26.33
SVR	53.63 ± 22.39	39.82

Table 5.4: Table summarizing the results for daily granularity.

The results for daily granularity at forecasting time on the last time slot including also the combinations of different techniques and the AutoML approach are reported in table 5.5. In this case, the impact of the combinations is not so evident and does not improve the results in all the cases. AutoML presents not very good results being worse than the one-week baseline, this is probably due

to the scarcity of training data and the fact of AutoML of trying very complex architectures when also simple ones may work well in certain cases. It can also be noticed that surprisingly for some of the best-performing models the MAPE on the first week of data is higher than the one on the whole month of data. This probably can be explained by the fact that the first week of the last split has a less regular pattern due to the Christmas holidays compared to the following ones.

Model	**Forecast on the last split one-week MAPE**	**Forecast on the last split one-month MAPE**
CNN	8.36	7.49
XGBRegressor	11.78	7.65
HistGradientBoostingRegressor	12.42	9.81
CNN + One Week Baseline	8.33	12.28
LSTM + One Week Baseline	11.19	12.91
LSTM	18.21	14.51
TFT	12.65	14.88
LSTM + One Week Baseline + Prophet	8.81	15.91
CNN + One Week Baseline + Prophet	11.34	16.96
LSTM + One Day Baseline + One Week Baseline + Prophet	9.82	17.29
CNN + One Day Baseline + One Week Baseline + Prophet	10.88	17.88
One Week Baseline	9.63	18.40
GRU + One Week Baseline	9.15	19.18
SARIMA	11.23	19.63
GRU	10.87	20.47
AutoML	10.61	20.98
GRU + One Day Baseline + One Week Baseline + Prophet	11.29	21.34
One Day Baseline	13.14	21.51
GRU + One Week Baseline + Prophet	11.89	21.56
One Day Baseline + One Week Baseline + Prophet	11.72	21.69
One Week Baseline + Prophet	12.83	22.21
Prophet	17.35	26.33
SVR	58.09	39.82

Table 5.5: Table summarizing the results for hourly granularity at forecasting time.

The forecasts produced by the CNN model for the last split in comparison to the actual values are reported in figure 5.15. It can be noticed that for the first three weeks, it follows the signal very accurately and the last week is increasing correctly but not enough as the actual values. The absolute percentage error of forecasts produced by the CNN model for the last split in comparison to the actual values is reported in figure 5.16. It can be shown that except for some peaks, the percentage error stays below 10%. Finally, the scatter plot representing the error of the forecasts produced by the CNN model for the last split in comparison to the actual values is reported in figure 5.17. This scatter plot demonstrates that there is no bias even though it seems that when predicting high values it is missing some extra quantity in the forecasts, this is probably true in the specific considered last split and in particular the last week.

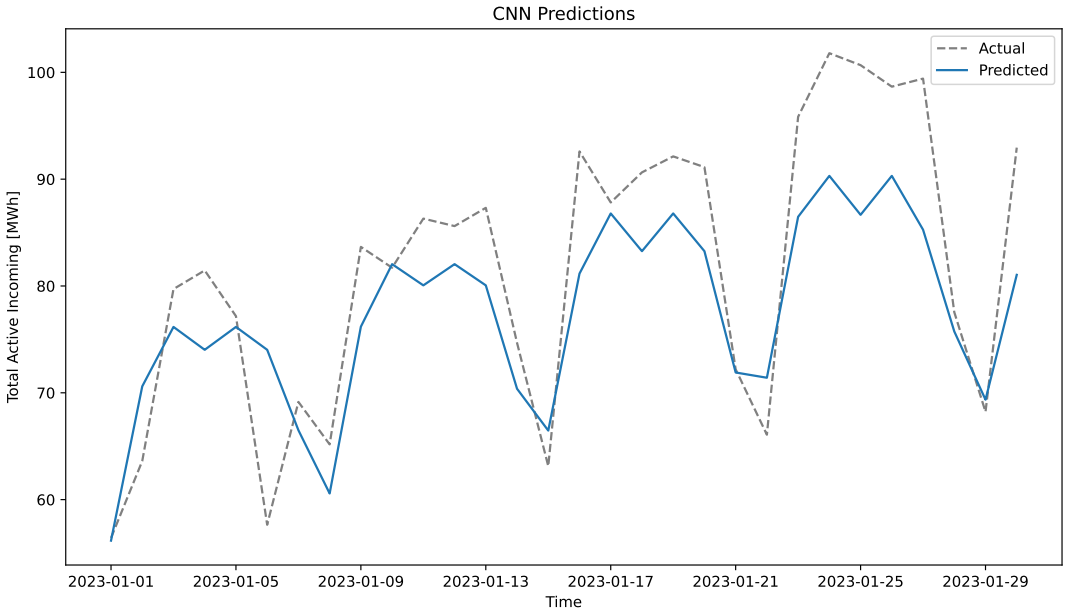


Figure 5.15: The forecasts produced by the convolutional neural network model for the last split in comparison to the actual values.

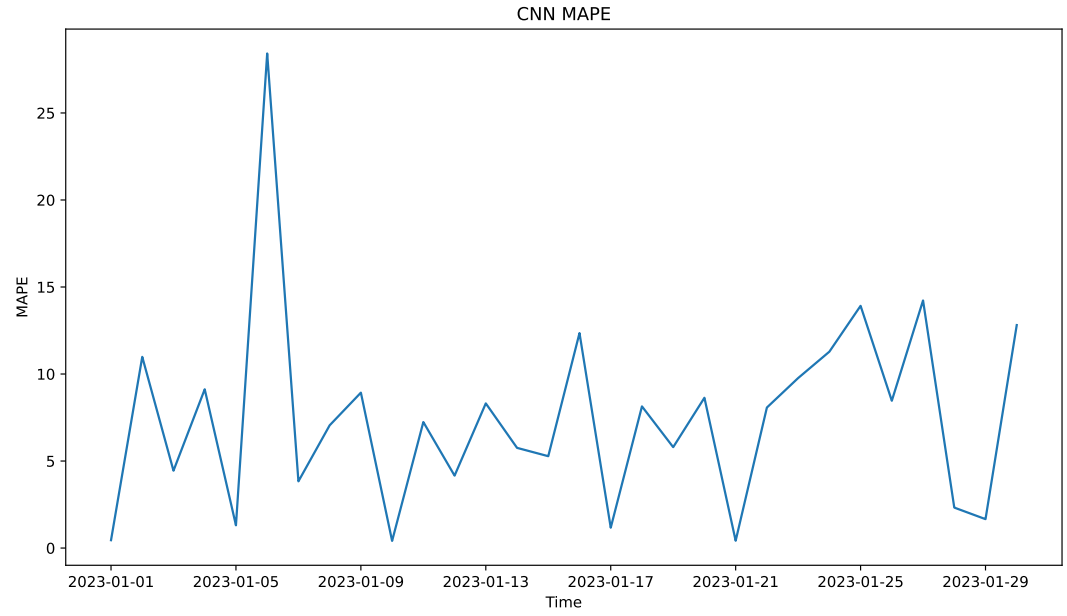


Figure 5.16: The absolute percentage error of forecasts produced by the convolutional neural network model for the last split in comparison to the actual values.

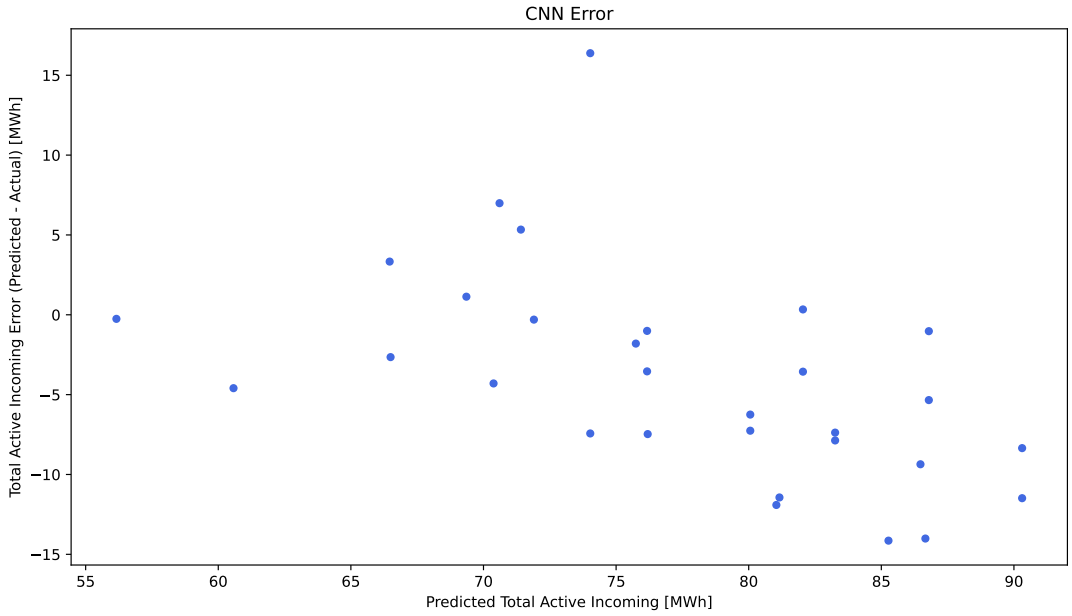


Figure 5.17: The scatter plot representing the error of the forecasts produced by the convolutional neural network model for the last split in comparison to the actual values.

5.4 Electricity production forecasting

As described in the data preprocessing in chapter 4, single PV plant production data are aggregated to obtain the aggregated production data over the PV plants. The target of the predictions is the mean percentage of production, which is calculated as the division of the total produced energy by the total power of the PV plants. This allows us to have a bounded value from 0 to 100 from which it is possible to obtain the total produced energy simply by multiplying it by the total power of the PV plants. This was also done since PV plants are added over time and this was unpredictable, this results in predicting the percentage of production of the PV plants. This data is analyzed to get some descriptive analytics before finding adequate models to forecast the production. The time series decomposition using an additive model of the hourly percentage of production considering as period of the time series a day is reported in figure 5.18. It showed a considerable amount of noise, comparable to seasonality and trend in magnitude. The trend component exhibits a clear peak during the summer season. As expected, the seasonality component appears to have the most significant impact on the time series.

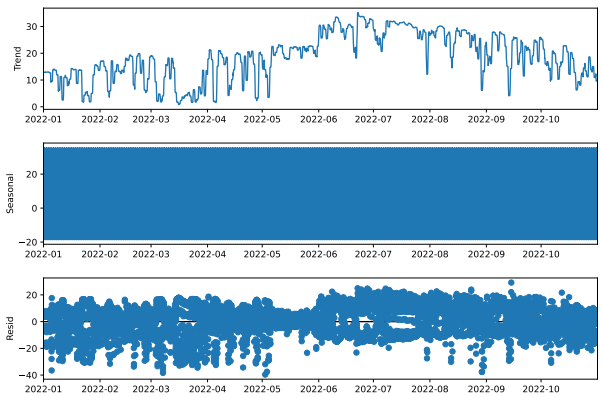


Figure 5.18: The time series decomposition of the hourly percentage of production considering as period of the time series a day.

The auto-correlation of the hourly percentage of production is reported in figure 5.19. It shows a high auto-correlation value in the closest time lag and also at every 24 hours, with the value slightly decreasing as the lag increases over days. This indicates that the production data from the closest time lag and those corresponding to the same hour in the preceding days may be valuable features for predicting a time instant's production. A reasonable balance can be achieved by incorporating the production data from the past 14 days.

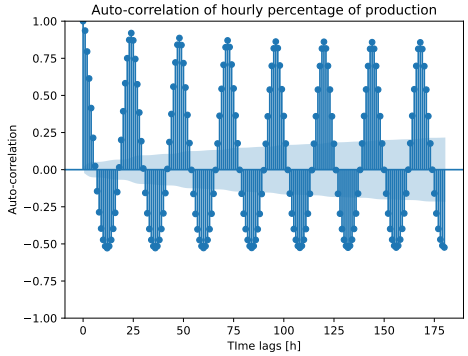


Figure 5.19: The auto-correlation of the hourly percentage of production.

The coefficients given by the Fourier transform for the hourly percentage of production are reported in figure 5.20. The graphical representation clearly shows a main frequency at the daily periodicity. Other minor peaks are present mostly at multiples of the 1/day frequency.

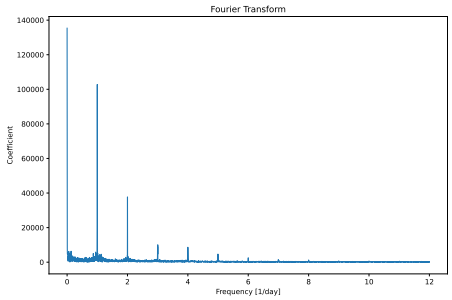


Figure 5.20: The coefficients given by the Fourier transform for the hourly percentage of production.

The daily percentage of production is reported in figure 5.21.

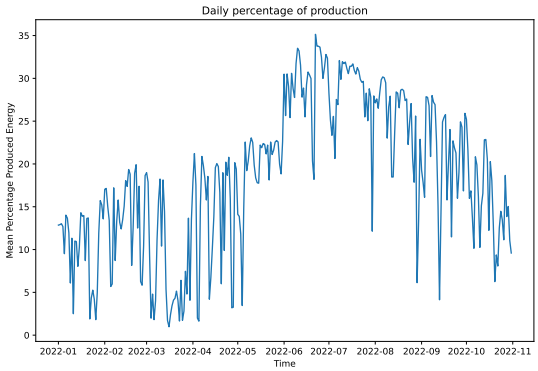


Figure 5.21: The daily percentage of production.

The time series decomposition using an additive model of the daily percentage of production considering as period of the time series a week is reported in figure 5.22. It showed a considerable

amount of noise, comparable to the trend in magnitude when not in the summer season. The trend component exhibits a clear peak during the summer season and appears to be the significant component of the time series. In this case, considering a weekly period, the seasonality component appears to not have a significant impact on the time series.

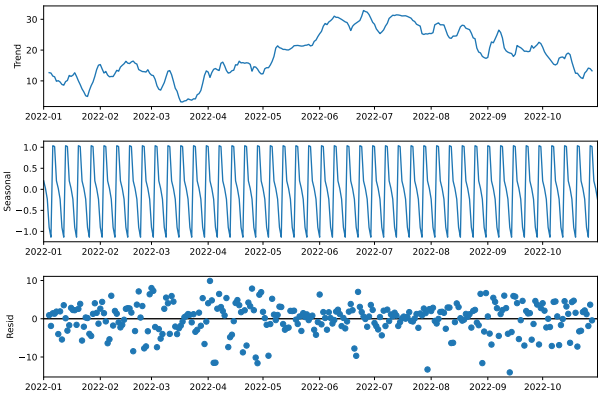


Figure 5.22: The time series decomposition of the daily percentage of production considering as period of the time series a week.

The auto-correlation of the daily percentage of production is reported in figure 5.23. It shows a high auto-correlation value in the closest time lag and then a slight decrease as the lag increases over days.

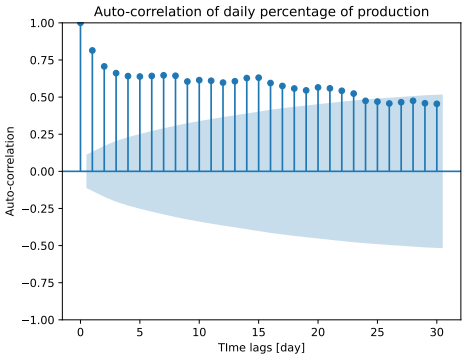


Figure 5.23: The auto-correlation of the daily percentage of production.

The coefficients given by the Fourier transform for the daily percentage of production are reported in figure 5.24. As expected from the time series decomposition, the graphical representation exhibits that there are no main frequencies in the daily percentage of production.

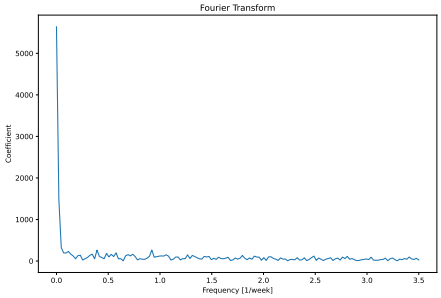


Figure 5.24: The coefficients given by the Fourier transform for the daily percentage of production.

Basic data is enhanced with the air temperature, the apparent temperature, the relative humidity,

the wind speed, the wind direction, the pressure altimeter, the visibility, the sky coverage, the diffuse horizontal irradiance, the direct normal irradiance, the global horizontal irradiance, the solar radiation, the UV index, the solar elevation angle, and the solar azimuth angle. To assess the relationship between these weather variables and the percentage of production, two correlation coefficients were used: Pearson's correlation coefficient and Spearman's rank correlation coefficient. Pearson's correlation coefficient measures the strength of the linear relationship between two variables, while Spearman's rank correlation coefficient measures the strength of the monotonic relationship. The pearsonr and spearmanr methods of the SciPy library were used to compute the correlation with weather data. The results showed that the hourly percentage of production had:

- a Pearson correlation coefficient of 0.5621 and a Spearman's rank correlation coefficient of 0.5185 with respect to the air temperature;
- a Pearson correlation coefficient of 0.5410 and a Spearman's rank correlation coefficient of 0.5032 with respect to the apparent temperature;
- a Pearson correlation coefficient of -0.6318 and a Spearman's rank correlation coefficient of -0.5969 with respect to the relative humidity;
- a Pearson correlation coefficient of 0.3105 and a Spearman's rank correlation coefficient of 0.3237 with respect to the wind speed;
- a Pearson correlation coefficient of -0.1919 and a Spearman's rank correlation coefficient of -0.2155 with respect to the wind direction;
- a Pearson correlation coefficient of -0.2092 and a Spearman's rank correlation coefficient of -0.2645 with respect to the pressure altimeter;
- a Pearson correlation coefficient of -0.0132 and a Spearman's rank correlation coefficient of 0.0355 with respect to the visibility;
- a Pearson correlation coefficient of -0.2822 and a Spearman's rank correlation coefficient of -0.2375 with respect to the sky coverage;
- a Pearson correlation coefficient of 0.8559 and a Spearman's rank correlation coefficient of 0.9340 with respect to the diffuse horizontal irradiance;
- a Pearson correlation coefficient of 0.8399 and a Spearman's rank correlation coefficient of 0.9315 with respect to the direct normal irradiance;
- a Pearson correlation coefficient of 0.8791 and a Spearman's rank correlation coefficient of 0.9357 with respect to the global horizontal irradiance;
- a Pearson correlation coefficient of 0.9073 and a Spearman's rank correlation coefficient of 0.9472 with respect to the solar radiation;
- a Pearson correlation coefficient of 0.8571 and a Spearman's rank correlation coefficient of 0.9392 with respect to the UV index;
- a Pearson correlation coefficient of 0.8107 and a Spearman's rank correlation coefficient of 0.9057 with respect to the solar elevation angle;
- a Pearson correlation coefficient of 0.0450 and a Spearman's rank correlation coefficient of 0.0465 with respect to the solar azimuth angle.

It can be noticed that both the coefficients indicate a moderate to strong correlation between the hourly percentage of production and several weather features, including the air temperature, the apparent temperature, the relative humidity, the diffuse horizontal irradiance, the direct normal irradiance, the global horizontal irradiance, the solar radiation, the UV index, and the solar elevation angle.

Therefore, these features are chosen to be incorporated into the prediction models, while the other weather features show weaker correlations and are not considered.

The results showed that the daily percentage of production had:

- a Pearson correlation coefficient of 0.7632 and a Spearman's rank correlation coefficient of 0.7721 with respect to the air temperature;
- a Pearson correlation coefficient of 0.7434 and a Spearman's rank correlation coefficient of 0.7471 with respect to the apparent temperature;
- a Pearson correlation coefficient of -0.7598 and a Spearman's rank correlation coefficient of -0.7474 with respect to the relative humidity;
- a Pearson correlation coefficient of 0.1773 and a Spearman's rank correlation coefficient of 0.3769 with respect to the wind speed;
- a Pearson correlation coefficient of -0.2405 and a Spearman's rank correlation coefficient of -0.2689 with respect to the wind direction;
- a Pearson correlation coefficient of -0.2506 and a Spearman's rank correlation coefficient of -0.2559 with respect to the pressure altimeter;
- a Pearson correlation coefficient of 0.2545 and a Spearman's rank correlation coefficient of 0.3259 with respect to the visibility;
- a Pearson correlation coefficient of -0.7004 and a Spearman's rank correlation coefficient of -0.6542 with respect to the sky coverage;
- a Pearson correlation coefficient of 0.6244 and a Spearman's rank correlation coefficient of 0.6939 with respect to the diffuse horizontal irradiance;
- a Pearson correlation coefficient of 0.6216 and a Spearman's rank correlation coefficient of 0.6923 with respect to the direct normal irradiance;
- a Pearson correlation coefficient of 0.6305 and a Spearman's rank correlation coefficient of 0.6988 with respect to the global horizontal irradiance;
- a Pearson correlation coefficient of 0.7878 and a Spearman's rank correlation coefficient of 0.8013 with respect to the solar radiation;
- a Pearson correlation coefficient of 0.8389 and a Spearman's rank correlation coefficient of 0.8680 with respect to the UV index;
- a Pearson correlation coefficient of 0.6502 and a Spearman's rank correlation coefficient of 0.7079 with respect to the solar elevation angle;
- a Pearson correlation coefficient of 0.1457 and a Spearman's rank correlation coefficient of 0.2813 with respect to the solar azimuth angle.

It can be noticed that both the coefficients decreased meaning that the mean weather data over the day is less correlated to the daily percentage of production, compared to the hourly granularity. Only the hourly granularity is considered since PV plants are highly correlated with weather data and the aggregation over the day loses part of this correlation.

If we were forecasting the production of each PV plant individually, the weather data at the specific locations of the plants would result in a higher correlation with the production data and likely allow the models to produce more accurate forecasts. However, since we are aggregating the production from multiple PV plants and looking for the overall percentage of production in the hour, using the weather data from a location located in the middle of the PV plants is also acceptable. In figure 5.25, the locations of the PV plants and of the weather station are reported.

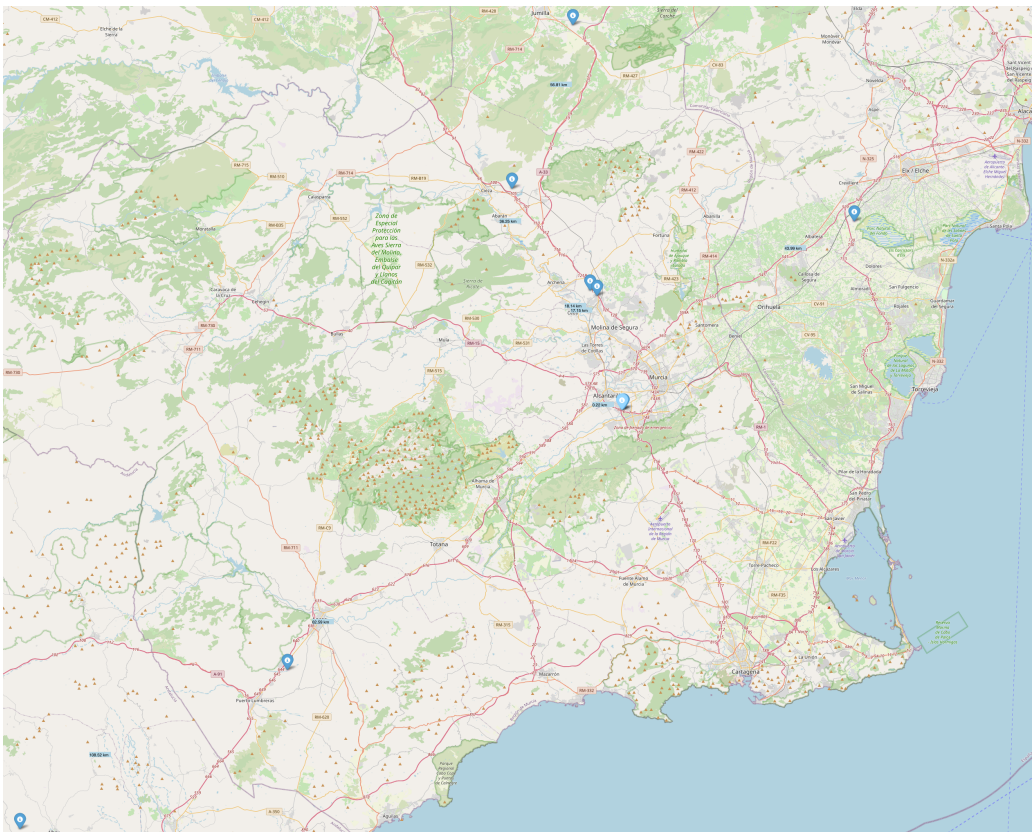


Figure 5.25: The map of the city of Murcia with indicated the location of the PV plants with a blue icon and of the revelation station with a lightblue icon. The distances between each PV plant and the weather station are reported in lightblue boxes near the PV plant locations.

After this data analysis, the specific parameters used in the models can be explained more in detail. The parameters for the different models were tested and verified on training data to result in the best-performing models, this was done by trying different values and configurations. The baseline approach is built considering the repetition of past days since data presents a high correlation with that time instant and could lead to a reasonable baseline performance to achieve. A simple ARIMA model considers the single hours without seasonally on the week period like for demand prediction since data does not present a relevant correlation with that time instants. The support vector regressor model uses a radial basis function kernel with a configuration of the C parameter (regularization parameter of squared l2 penalty) to 1.0 to penalize the complexity of the model, and the epsilon parameter defining the epsilon-tube for no penalty to 0.1 as a trade-off between accuracy and generalization. The hist gradient boosting regressor model uses as loss the absolute error. The learning rate was set to 0.1, 100 estimators were used and l2 regularization was not set. The parameters for tree definition are 31 as the maximum number of leaves, depth is not constrained, 20 as the minimum number of samples per leaf, and 255 as the maximum number of bins. For the extreme gradient boosting regressor model, tree-based models were used and the construction algorithm is automatically chosen by heuristic to choose the fastest method. It uses a squared loss since this was the supported metric closer to our evaluation metric. The learning rate was set to 0.3, 100 estimators were used and l2 regularization was set with a weight of 1.0. The parameters for tree definition are 6 as the maximum depth, and the maximum number of leaves is not constrained. The Prophet model was built keeping the default parameters for automatically detecting seasonalities and best fitting the training data.

The LSTM model was designed as a 2-layer model with 32 Bidirectional LSTM units in the first layer which use ReLU activation function and sigmoid as recurrent activation function with both dropout and recurrent dropout of 0.02. The second layer is composed of 16 Bidirectional LSTM units which use ReLU activation function and sigmoid as recurrent activation function without dropout and recurrent dropout. The output of the layer goes inside a dense unit to output the final prediction. The model is trained using mean absolute error as loss and Nadam as optimizer with a learning rate

of 0.005.

The GRU model was designed as a 2-layer model with 32 Bidirectional GRU units in the first layer which use ReLU activation function and sigmoid as recurrent activation function with both dropout and recurrent dropout of 0.02. Subsequently, the second layer is composed of 16 Bidirectional GRU units which use ReLU activation function and sigmoid as recurrent activation function without dropout and recurrent dropout. The output of the layer goes inside a dense unit to output the final prediction. The model is trained using mean absolute error as loss and Nadam as optimizer with a learning rate of 0.005.

The CNN model was designed as a 2-layer model with a first layer composed of 24 1D Convolutional units with a kernel size of 5 which uses ReLU activation function. A 1D max polling operation is applied before entering the second layer composed of 16 1D Convolutional units with a kernel size of 3 which use ReLU activation function. A 1D max polling operation is applied before the flattening operation and entering the final dense unit to output the final prediction. The model is trained using mean absolute error as loss and Nadam as optimizer with a learning rate of 0.005.

The TFT model was configured with 2 LSTM layers with 16 as hidden size, 4 as attention size, and 8 as hidden continuous size. The dropout is set to 0.02. The model is trained using mean absolute error as loss and Nadam as optimizer with a learning rate of 0.005.

12 splits were used for block validation with a test size of a week each time, namely 7 days of data. The models start with 5280 entries as training and predict every time 168 entries until reaching the last prediction instant where the model has a training size of 7128. The results for hourly granularity are reported in table 5.6. Entries are sorted by best MAE using the last split as the test set. The blocked k-fold cross-validation suggests that the hist gradient boosting regressor model has the best MAE, but looking at the last split this is not confirmed. In fact, the GRU model performs better on the last split, as the other DL models, this seems to confirm what is seen in demand prediction that with the increase of training data, these models perform better. It can also be noted that all the models are better than the one-day baseline. Prophet presents not very good results compared to the other models and is close to the baseline as results on the last split while considering the blocked k-fold cross-validation is the only model that performs worse than it, In this case, the TFT approach does not seem to perform in an optimal way, and its results are close to the ones of LSTM and ARIMA models.

Model	**Blocked k-fold cross-validation MAE percentage**	**Test on the last split MAE percentage**
GRU	5.70 ± 1.78	2.79
HistGradientBoostingRegressor	4.16 ± 0.99	2.98
SVR	5.20 ± 1.12	3.12
CNN	5.98 ± 1.52	3.27
XGBRegressor	4.57 ± 0.97	3.42
LSTM	5.23 ± 1.46	3.51
TFT	4.70 ± 1.39	3.60
ARIMA	5.09 ± 1.35	3.64
Prophet	7.90 ± 1.62	4.38
One Day Baseline	6.81 ± 3.14	4.58

Table 5.6: Table summarizing the results for hourly granularity.

The results for hourly granularity at forecasting time on the last time slot including also the combinations of different techniques and the AutoML approach are reported in table 5.7. The combination of the previous techniques was performed by combining one-day baseline and Prophet with possibly one of LSTM, GRU, or CNN with the intent of trying to improve the robustness of DL models with some stable baselines and Prophet model which internally automatically deals with seasonalities, even though for this task it is not performing very well. The AutoML approach uses the mean MAPE forecasting as loss and it is launched for 10 hours with a maximum function evaluation time of 2 hours. Also, the mean MAE forecasting was used but provided worse results. Then the best ensemble

of the found models is returned and can be used to forecast.

Model	**Forecast on the last split MAE percentage**
GRU	2.79
HistGradientBoostingRegressor	2.86
SVR	3.12
CNN	3.27
XGBRegressor	3.35
LSTM + One Day Baseline	3.37
GRU + One Day Baseline	3.45
CNN + One Day Baseline	3.48
LSTM	3.51
TFT	3.60
LSTM + One Day Baseline + Prophet	3.60
ARIMA	3.64
CNN + One Day Baseline + Prophet	3.67
GRU + One Day Baseline + Prophet	3.70
One Day Baseline + Prophet	4.38
Prophet	4.38
One Day Baseline	4.58
AutoML	5.89

Table 5.7: Table summarizing the results for hourly granularity at forecasting time.

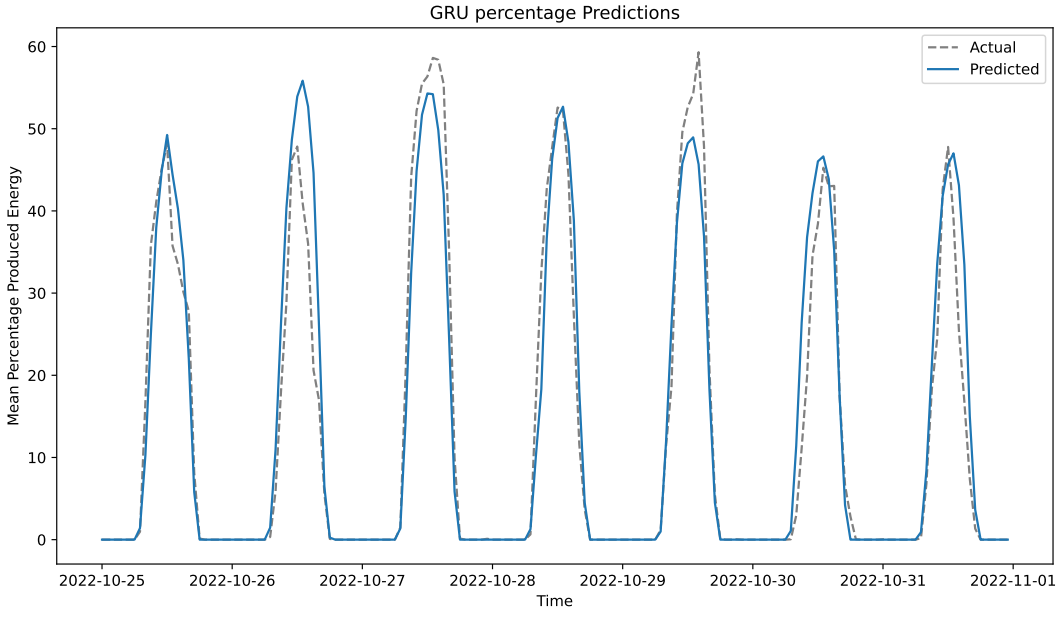


Figure 5.26: The forecasts produced by the gated recurrent unit model for the last split in comparison to the actual values.

Including also combinations and AutoML, there are a few observations that can be done. The combinations of DL models and the one-week baseline are the most beneficial, but just in the case of LSTM there is an improvement and it is very marginal. Probably in this case since the baseline and prophet are not well-performing models they are not bringing any significant improvement or robustness to the DL models. Nonetheless, it might still make sense to investigate the concept of

ensemble learning further and optimize this combination at training time to possibly have stronger results at forecasting time. In this case, AutoML is not able to achieve very good results compared to many other models. It still has great potential as demonstrated in demand prediction and potentially providing more total training time it can be able to reach better performance.

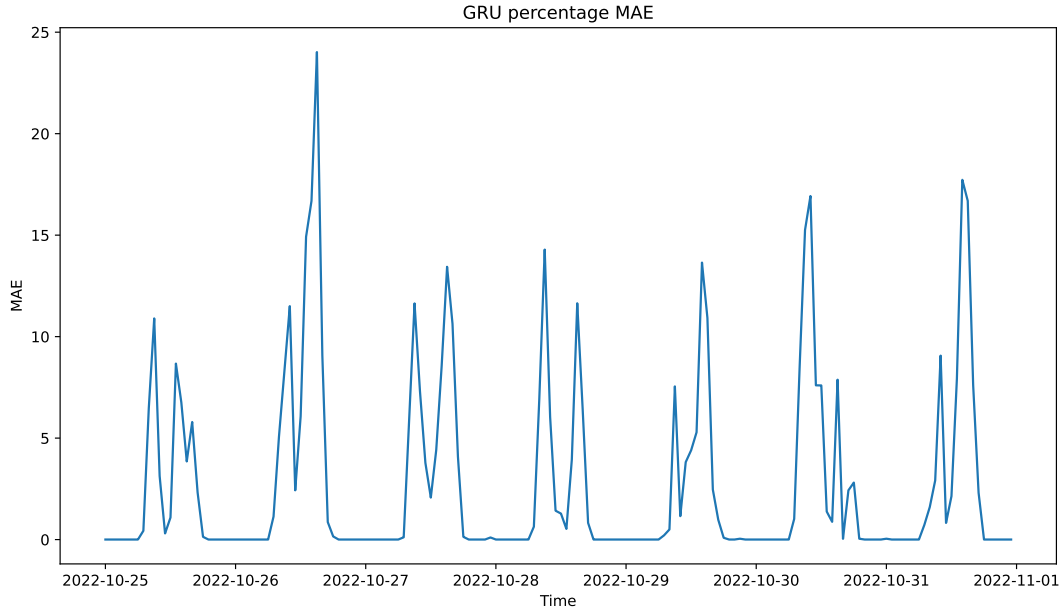


Figure 5.27: The absolute error of forecasts produced by the gated recurrent unit model for the last split in comparison to the actual values.

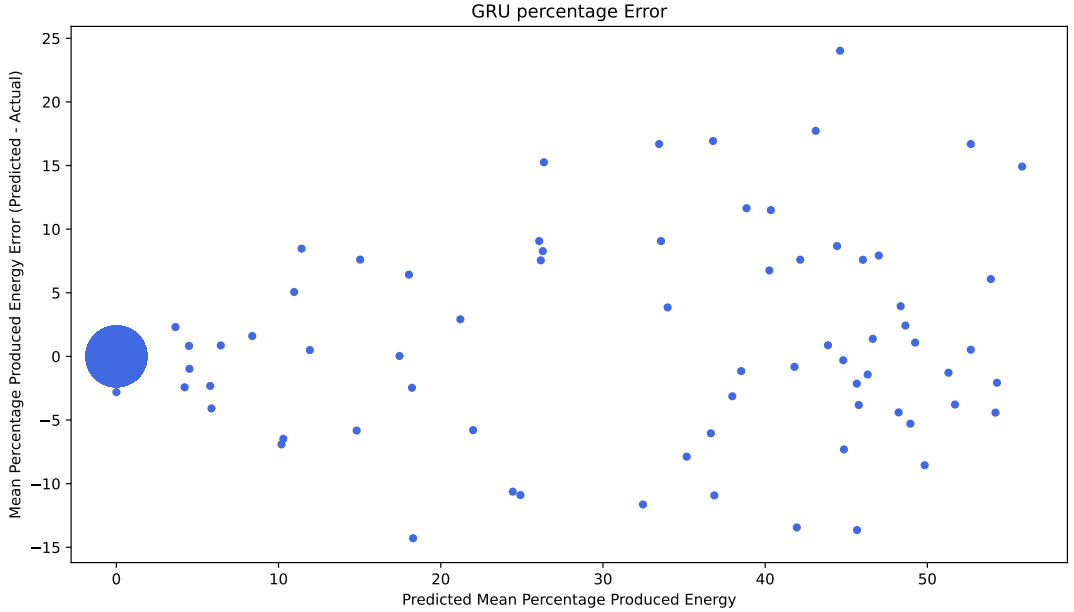


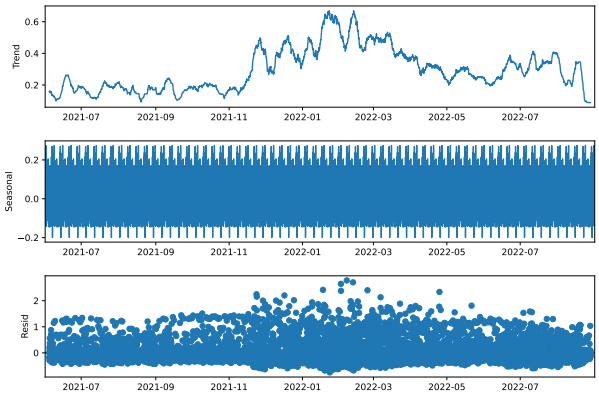
Figure 5.28: The scatter plot representing the error of the forecasts produced by the gated recurrent unit model for the last split in comparison to the actual values.

The forecasts produced by the GRU model for the last split in comparison to the actual values are reported in figure 5.26. It can be noticed that it follows very accurately the signal for most of

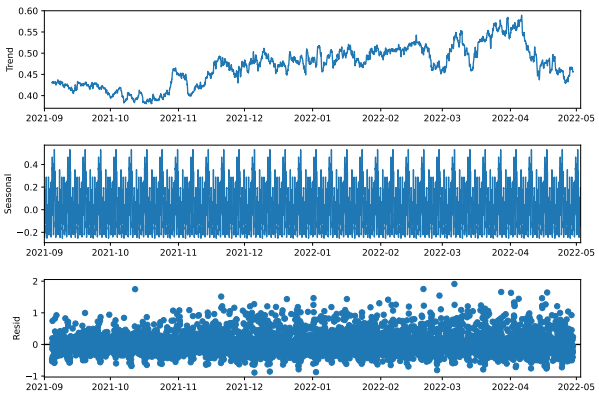
the test days, for 2 days it underestimates and for one overestimates the actual data. The absolute error of forecasts produced by the GRU model for the last split in comparison to the actual values is reported in figure 5.27. It can be shown that, except for the peak when it overestimates the actual data and some other minor peaks, the error stays below 10%. Finally, the scatter plot representing the error of the forecasts produced by the GRU model for the last split in comparison to the actual values is reported in figure 5.28. This scatter plot demonstrates that there is no bias and can be also noticed that when the predicted values are low also the error is limited, while when the prediction is high there are both high positive and negative errors.

5.5 Consumption baseline forecasting

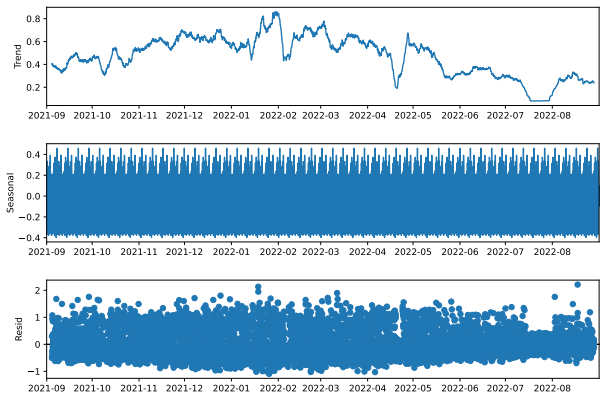
The consumption data of the three customers are analyzed to get some descriptive analytics before finding adequate models to forecast the consumption baseline of each customer. The time series decompositions using an additive model of the hourly consumption of the three customers considering as period of the time series a week are reported in figure 5.29a, figure 5.29b, and figure 5.29c. All three decompositions show a considerable amount of noise, which is dominant in magnitude compared to the trend and seasonality. For all the customers the seasonal component appears to not have a significant impact on the time series, with the exception of the second customer for which it is twice high in magnitude with respect to the others. The trend component has a slight impact, for the first customer it is possible to see a peak in the winter, for the second customer it is present a peak in April, and for the third customer there is not a clear trend.



(a) First customer.

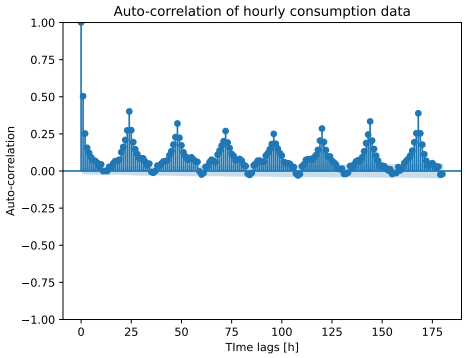


(b) Second customer.

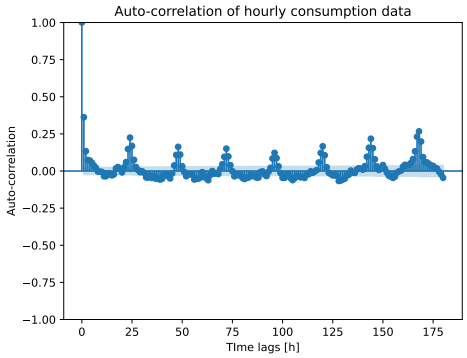


(c) Third customer.

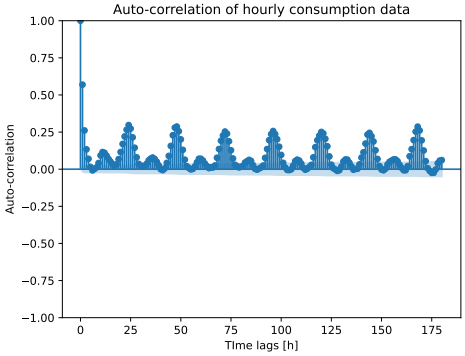
Figure 5.29: The time series decompositions of the hourly consumption of the three customers considering as period of the time series a week.



(a) First customer.

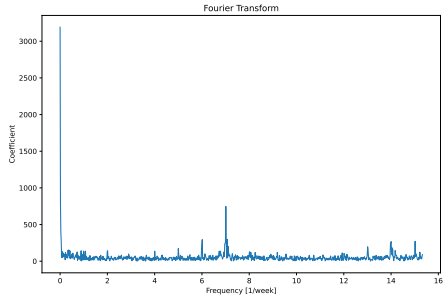


(b) Second customer.

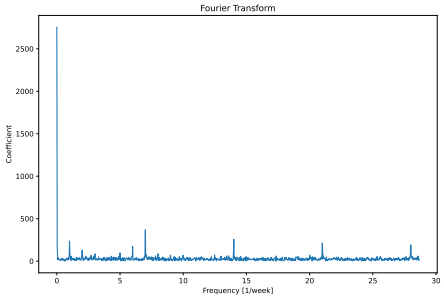


(c) Third customer.

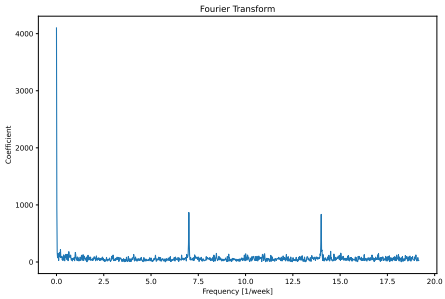
Figure 5.30: The auto-correlations of the hourly consumption of the three customers.



(a) First customer.



(b) Second customer.



(c) Third customer.

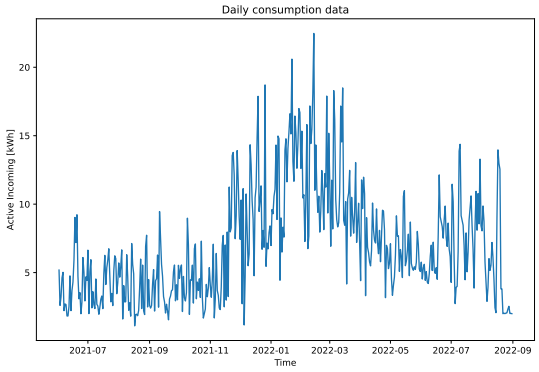
Figure 5.31: The coefficients given by the Fourier transform of the hourly consumption of the three customers.

The auto-correlations of the hourly consumption of the three customers are reported in figure 5.30a,

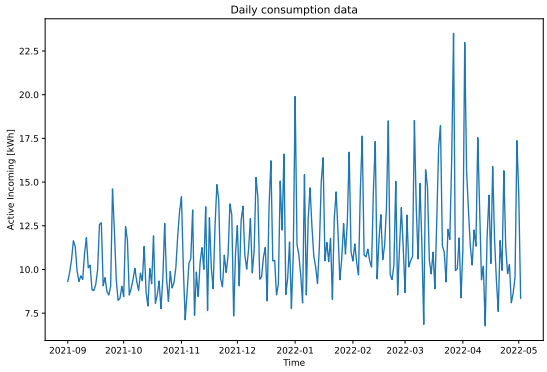
figure 5.30b, and figure 5.30c. All the customers' consumptions are not very auto-correlated, it can be shown that the maximum auto-correlation value is around 0.5 in the closest time lag and that there is a peak every 24 hours, with a slightly greater value at one week distance. This indicates that the consumption data from the closest time lag, as well as those corresponding to the same hour in the preceding days and even better in the preceding weeks, may be valuable features for predicting a time instant's demand. A reasonable balance can be achieved by incorporating the consumption data from the past 14 days.

The coefficients given by the Fourier transform for the hourly consumption of the three customers are reported in figure 5.31a, figure 5.31b, and figure 5.31c. The graphical representations show for all the customers a main frequency at the daily periodicity. Other minor peaks are present mostly at multiples of the 1/day frequency, in particular for the third customer where the 2/day frequency is almost equal to the 1/day.

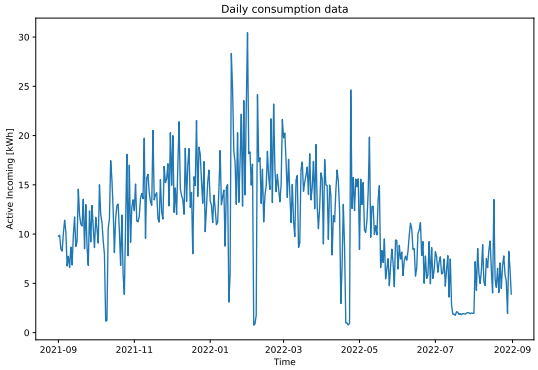
The daily consumption of the three customers are reported in figure 5.32a, figure 5.32b, and figure 5.32c.



(a) First customer.



(b) Second customer.



(c) Third customer.

Figure 5.32: The daily consumption of the three customers.

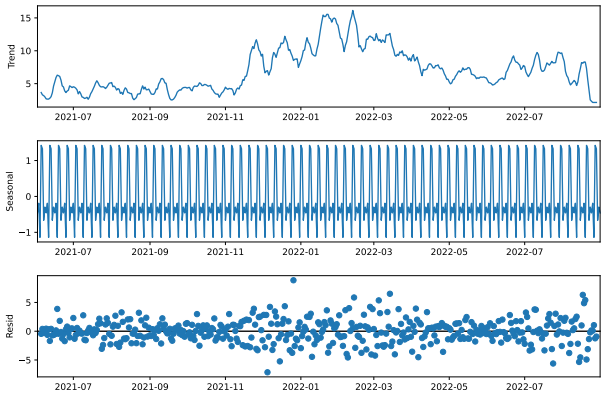
The time series decompositions using an additive model of the daily consumption of the three customers considering as period of the time series a week are reported in figure 5.33a, figure 5.33b, and figure 5.33c. The decomposition is consistent with the hourly aggregated consumption over the customers.

The auto-correlations of the daily consumption of the three customers are reported in figure 5.34a, figure 5.34b, and figure 5.34c. With the daily granularity, all the customers' consumptions show a slightly high value with respect to the hourly granularity. It can be shown that for the first and third customers, the auto-correlation value is slightly high in the closest time lag and then decreases with the days, with a slightly greater value at one week distance. Instead, for the second customer the only relevant values are only at a multiple of 7 days.

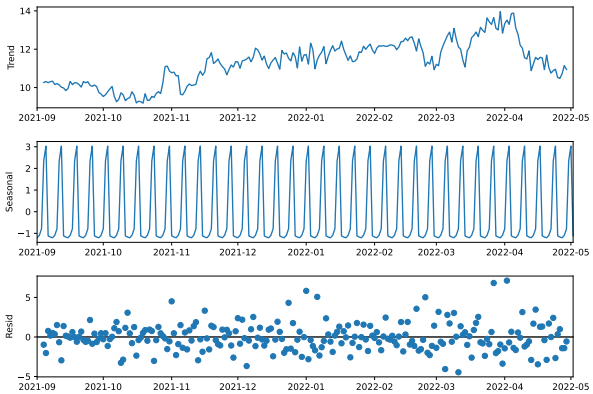
The coefficients given by the Fourier transform for the daily consumption of the three customers are reported in figure 5.35a, figure 5.35b, and figure 5.35c. The graphical representation exhibits that there are no main frequencies in the daily consumption of the first and third customers. Instead, for the second customer there is a small peak at the daily periodicity.

As can be noticed from the data, there is high variability in consumption and low auto-correlation, this suggests how it is difficult to produce highly accurate results on a single customer level. With just the time series of a few users, it is very difficult to learn a well-performing model, having more users it could be possible to learn certain generic trends or standard behaviors.

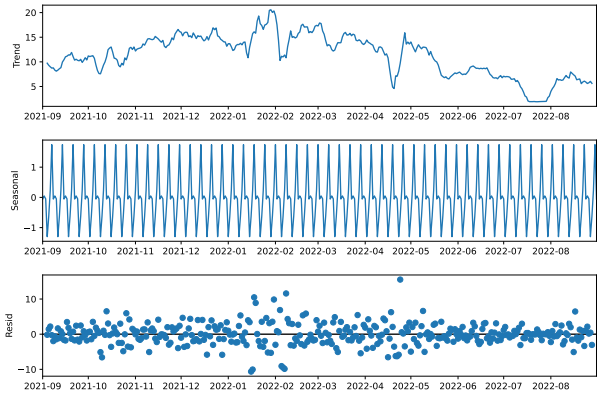
Basic data is enhanced with the air temperature, the apparent temperature, and the relative humidity since they are considered the only weather features capable of influencing customers' energy consumption. To assess the relationship between these weather variables and the consumption of the three customers, two correlation coefficients were used: Pearson's correlation coefficient and Spearman's rank correlation coefficient. Pearson's correlation coefficient measures the strength of the linear relationship between two variables, while Spearman's rank correlation coefficient measures the strength of the monotonic relationship. The pearsonr and spearmanr methods of the SciPy library were used to compute the correlation with weather data.



(a) First customer.

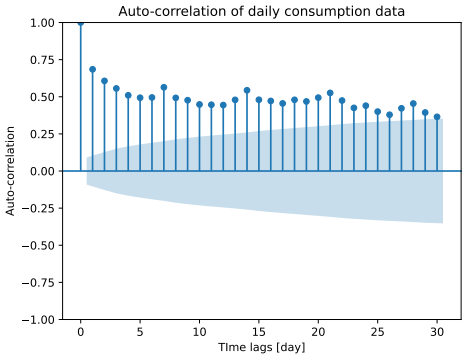


(b) Second customer.

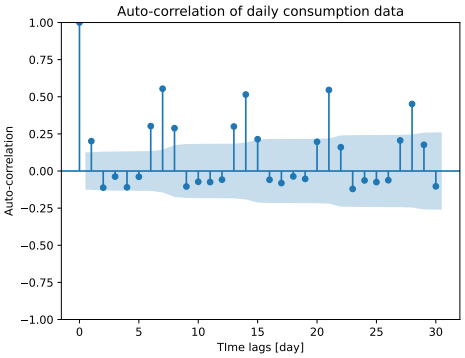


(c) Third customer.

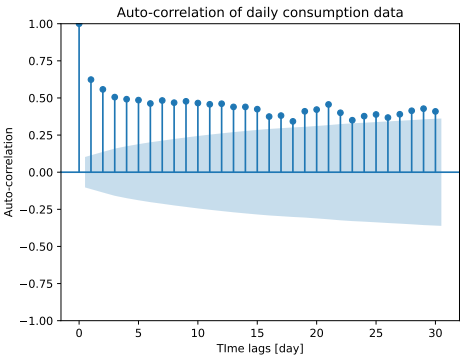
Figure 5.33: The time series decompositions of the daily consumption of the three customers considering as period of the time series a week.



(a) First customer.

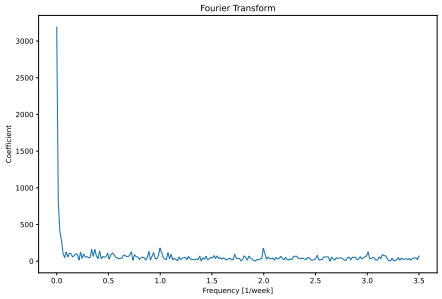


(b) Second customer.

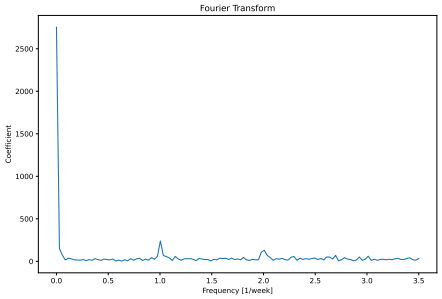


(c) Third customer.

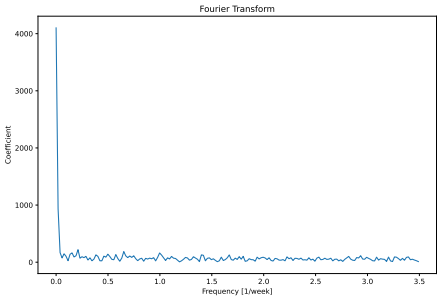
Figure 5.34: The auto-correlations of the daily consumption of the three customers.



(a) First customer.



(b) Second customer.



(c) Third customer.

Figure 5.35: The coefficients given by the Fourier transform of the daily consumption of the three customers.

The results showed that the hourly consumption of the three customers had:

- a Pearson correlation coefficient of -0.2491, 0.0240, and -0.1338 respectively and a Spearman's rank correlation coefficient of -0.2088, 0.1304, and -0.0196 respectively with respect to the air temperature;
- a Pearson correlation coefficient of -0.2434, 0.0266, and -0.1356 respectively and a Spearman's rank correlation coefficient of -0.2082, 0.1305, and -0.0207 respectively with respect to the apparent temperature;
- a Pearson correlation coefficient of 0.1193, -0.0845, and -0.0985 respectively and a Spearman's rank correlation coefficient of 0.1491, -0.1048, and -0.1123 respectively with respect to the relative humidity.

It can be noticed that both the coefficients indicate a weak correlation between the weather variables and the consumption of the three customers, nevertheless, it may still be useful to incorporate weather data into the prediction models since also the auto-correlation values are not particularly high.

The results showed that the daily consumption of the three customers had:

- a Pearson correlation coefficient of -0.5542, -0.2341, and -0.6727 respectively and a Spearman's rank correlation coefficient of -0.5164, -0.2592, and -0.6978 respectively with respect to the air temperature;
- a Pearson correlation coefficient of -0.5361, -0.2329, and -0.6787 respectively and a Spearman's rank correlation coefficient of -0.5166, -0.2584, and -0.7004 respectively with respect to the apparent temperature;
- a Pearson correlation coefficient of 0.1264, 0.0199, and 0.1407 respectively and a Spearman's rank correlation coefficient of 0.1412, 0.0506, and 0.2104 respectively with respect to the relative humidity.

It can be noticed that both the coefficients increased considering the daily data meaning that the mean weather data over the day is more correlated to the daily consumption of the three considered customers, compared to the hourly granularity.

As described in chapter 4, tariff granularity has also been thought of as a possible granularity. Tariff granularity assumes that the tariff is a control signal capable of modifying consumption. However, after some initial experimental results with this granularity showing unsatisfactory performance, this granularity was not further investigated. Though, this granularity can be investigated in a possible future work to understand whether it could be applicable to produce good results.

After this data analysis, the specific parameters used in the models can be explained more in detail. The parameters for the different models were tested and verified on training data to result in the best-performing models, this was done by trying different values and configurations. The baseline approaches are built considering the repetition of past days and weeks since data presents a high correlation with that time instants and could lead to a reasonable baseline performance to achieve. Instead of just considering the one-day and one-week baselines, for this use case also the four-week and the twelve-week baselines are considered taking an average of the consumption in the specified number of previous weeks. This is because averaging the consumption over the previous week may lead to a more robust prediction considering a month and a trimester range, The SARIMA model considers the week as the period for seasonal differencing since data presents a high correlation with that time instants and the model can try to take advantage of the weekly seasonality. The support vector regressor model uses a radial basis function kernel with a configuration of the C parameter (regularization parameter of squared l2 penalty) to 1.0 to penalize the complexity of the model, and the epsilon parameter defining the epsilon-tube for no penalty to 0.1 as a trade-off between accuracy and generalization. The hist gradient boosting regressor model uses as loss the absolute error. The learning rate was set to 0.1, 100 estimators were used and l2 regularization was not set. The parameters for tree definition are 31 as the maximum number of leaves, depth is not constrained, 20 as the minimum number of samples per leaf, and 255 as the maximum number of bins. For the extreme gradient boosting regressor model, tree-based models were used and the construction algorithm is

automatically chosen by heuristic to choose the fastest method. It uses a squared loss since this was the supported metric closer to our evaluation metric. The learning rate was set to 0.3, 100 estimators were used and l2 regularization was set with a weight of 1.0. The parameters for tree definition are 6 as the maximum depth, and the maximum number of leaves is not constrained. The Prophet model was built keeping the default parameters for automatically detecting seasonalities and best fitting the training data.

The LSTM model was designed as a 2-layer model with 32 Bidirectional LSTM units in the first layer which use ReLU activation function and sigmoid as recurrent activation function with both dropout and recurrent dropout of 0.02. The second layer is composed of 16 Bidirectional LSTM units which use ReLU activation function and sigmoid as recurrent activation function without dropout and recurrent dropout. The output of the layer goes inside a dense unit to output the final prediction. The model is trained using a combination of mean absolute error and mean absolute percentage error as loss and Nadam as optimizer with a learning rate of 0.005.

The GRU model was designed as a 2-layer model with 32 Bidirectional GRU units in the first layer which use ReLU activation function and sigmoid as recurrent activation function with both dropout and recurrent dropout of 0.02. Subsequently, the second layer is composed of 16 Bidirectional GRU units which use ReLU activation function and sigmoid as recurrent activation function without dropout and recurrent dropout. The output of the layer goes inside a dense unit to output the final prediction. The model is trained using a combination of mean absolute error and mean absolute percentage error as loss and Nadam as optimizer with a learning rate of 0.005.

The CNN model was designed as a 2-layer model with a first layer composed of 32 1D Convolutional units with a kernel size of 5 which uses ReLU activation function. A 1D max polling operation is applied before entering the second layer composed of 16 1D Convolutional units with a kernel size of 3 which use ReLU activation function. A 1D max polling operation is applied before the flattening operation and entering the final dense unit to output the final prediction. The model is trained using a combination of mean absolute error and mean absolute percentage error as loss and Nadam as optimizer with a learning rate of 0.005.

The TFT model was configured with 2 LSTM layers with 16 as hidden size, 4 as attention size, and 8 as hidden continuous size. The dropout is set to 0.02. The model is trained using mean absolute percentage error as loss and Nadam as optimizer with a learning rate of 0.005.

Model	**Blocked k-fold cross-validation MAPE — MAE [kWh]**	**Test on the last split MAPE — MAE [kWh]**
TFT	47.33 ± 5.57 — 0.287 ± 0.028	44.19 — 0.231
HistGradientBoostingRegressor	53.95 ± 7.02 — 0.260 ± 0.023	50.41 — 0.231
SVR	60.30 ± 6.81 — 0.278 ± 0.032	60.41 — 0.243
SARIMA	74.17 ± 10.01 — 0.271 ± 0.021	78.02 — 0.244
XGBRegressor	73.71 ± 9.54 — 0.275 ± 0.024	68.07 — 0.245
Prophet	83.84 ± 10.23 — 0.280 ± 0.020	81.92 — 0.249
GRU	58.88 ± 19.55 — 0.309 ± 0.041	42.55 — 0.257
LSTM	111.87 ± 212.07 — 0.464 ± 0.516	44.23 — 0.266
4 Week Baseline	74.06 ± 12.51 — 0.275 ± 0.027	79.07 — 0.266
12 Week Baseline	74.30 ± 13.66 — 0.266 ± 0.024	89.23 — 0.278
One Week Baseline	82.69 ± 12.83 — 0.326 ± 0.030	70.48 — 0.281
One Day Baseline	75.80 ± 13.96 — 0.318 ± 0.035	71.25 — 0.292
CNN	148.16 ± 30.05 — 0.510 ± 0.105	70.19 — 0.318

Table 5.8: Table summarizing the results for hourly granularity.

12 splits were used for block validation with a test size of a week each time, namely 7 days of data. For the hourly granularity, the models start with the total number of entries for the customer minus 2016 entries as training and predict every time 168 entries until reaching the last prediction instant where the model has a training size of the total number of entries for the customer minus the last 168

entries. The results for hourly granularity for the second customer are reported in table 5.8. Entries are sorted by best MAE using the last split as the test set. Only this customer is reported since it presents the best results and the models were optimized for this customer. The blocked k-fold cross-validation suggests that the TFT model has the best MAPE and this is not confirmed on the last split just because there is GRU that performs slightly better in terms of MAPE. The not-encouraging thing is that even if it is the best model in terms of MAPE, this value is still close to 50% indicating that on average the predictions are distant from the actual value by half of this value. TFT is also the best approach in terms of MAE on the last split, but not on the basis of the blocked k-fold cross-validation which indicates that the best model is the hist gradient boosting regressor model. In general, this data is not well predictable and also the different baselines confirm that having a look at the close history there is no relevant daily or weekly repetition with MAPE and MAE both higher than TFT and hist gradient boosting regressor model.

The AutoML approach uses the mean MAPE forecasting as loss and it is launched for 10 hours with a maximum function evaluation time of 2 hours. Then the best ensemble of the found models is returned and can be used to forecast. This approach obtained a one-week MAPE of 61.64% and a MAE of 0.338 kWh. In this case, AutoML is not able to achieve very good results compared to many other models but this is probably related to the quality of the data. It still has great potential as demonstrated in demand prediction and potentially providing more total training time and more data, potentially also of different customers, it can be able to reach better performance.

The forecasts produced by the TFT model for the last split in comparison to the actual values are reported in figure 5.36. It can be noticed that it is not able to accurately follow the signal since this is very varying from hour to hour. Although, it is able to somehow predict that there is a peak during the day but it is not able to reach the correct quantity.

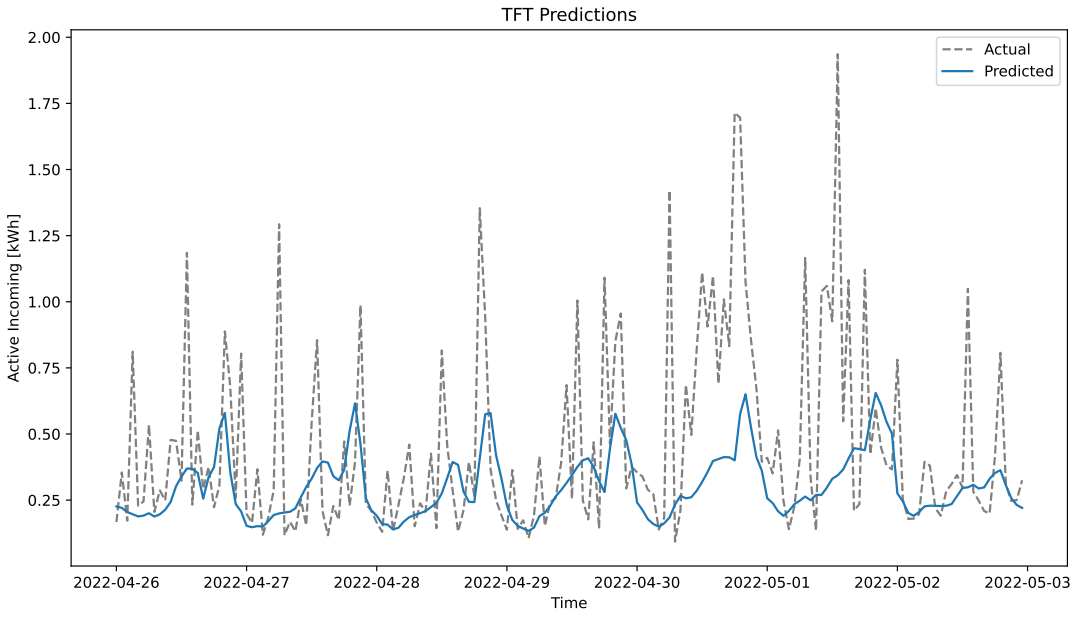


Figure 5.36: The forecasts produced by the temporal fusion transoformer model for the last split in comparison to the actual values.

The absolute error of forecasts produced by the TFT model for the last split in comparison to the actual values is reported in figure 5.37. While the absolute percentage error of forecasts produced by the TFT model for the last split in comparison to the actual values is reported in figure 5.38. It can be shown that the entity of the errors is quite high both in terms of absolute error and in terms of absolute percentage error. Moreover, the error signal seems to have some periodicity that the TFT model is not able to capture, probably due to the high variability of the data. Finally, the scatter plot representing the error of the forecasts produced by the TFT model for the last split in comparison

to the actual values is reported in figure 5.39. This scatter plot demonstrates that there is no bias and can be also noticed that the model is conservative and the positive error is contained while the negative ones are up to six times bigger. The error can be pretty high also when the predicted values are low.

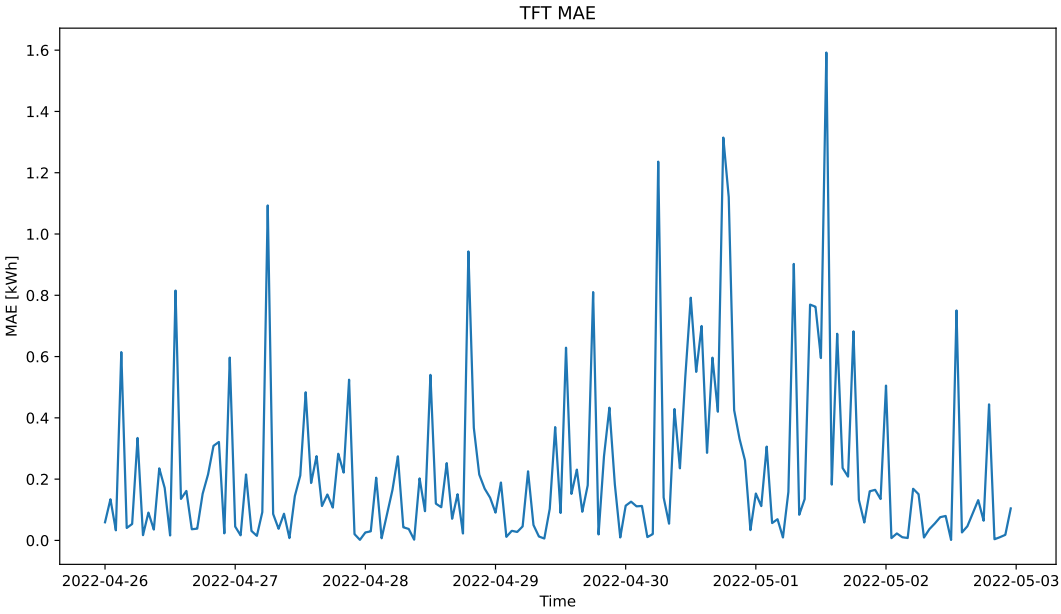


Figure 5.37: The absolute error of forecasts produced by the temporal fusion transoformer model for the last split in comparison to the actual values.

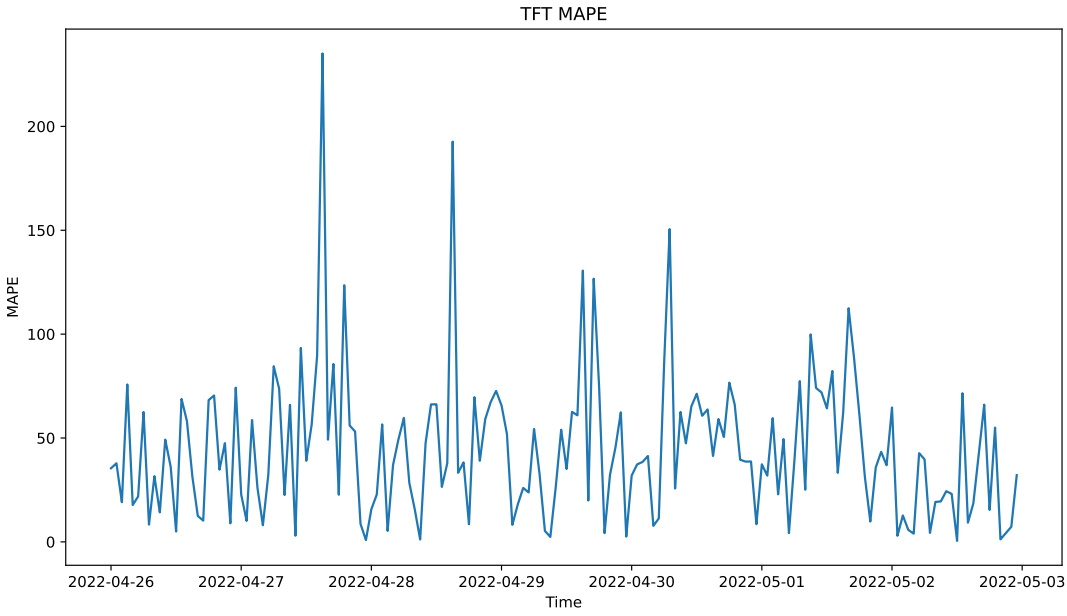


Figure 5.38: The absolute percentage error of forecasts produced by the temporal fusion transoformer model for the last split in comparison to the actual values.

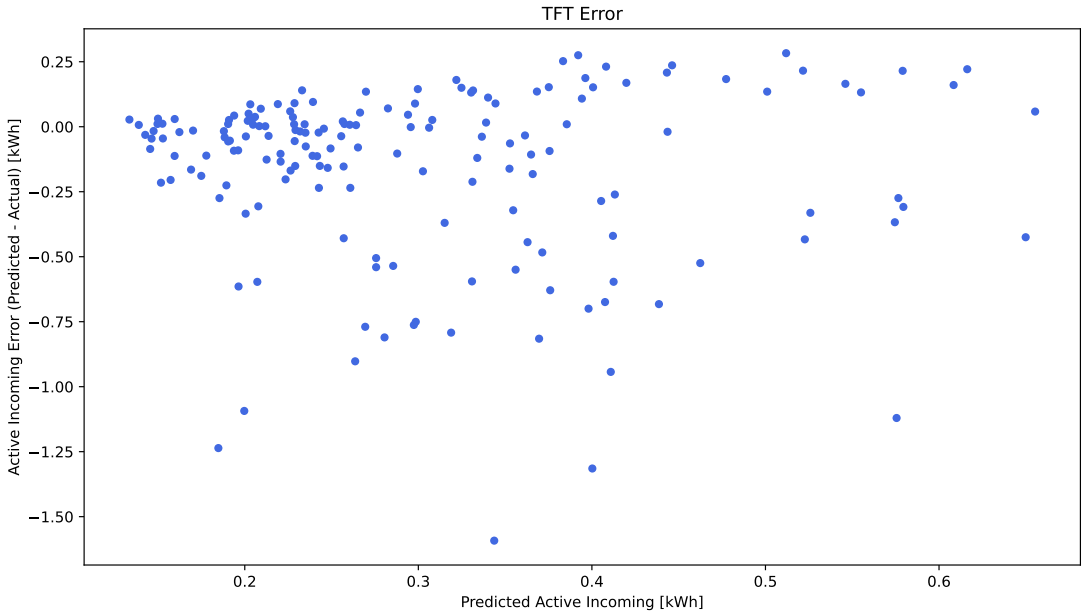


Figure 5.39: The scatter plot representing the error of the forecasts produced by the temporal fusion transoformer model for the last split in comparison to the actual values.

Model	**Blocked k-fold cross-validation MAPE — MAE [kWh]**	**Test on the last split MAPE — MAE [kWh]**
4 Week Baseline	16.16 ± 4.88 — 1.906 ± 0.572	14.40 — 1.318
One Week Baseline	19.56 ± 7.45 — 2.343 ± 0.899	14.45 — 1.554
CNN	28.45 ± 17.47 — 3.320 ± 1.939	13.46 — 1.751
SARIMA	16.38 ± 4.82 — 1.896 ± 0.454	17.07 — 1.773
TFT	16.67 ± 6.33 — 2.002 ± 0.643	14.55 — 1.786
HistGradientBoostingRegressor	15.36 ± 3.72 — 1.889 ± 0.355	16.80 — 1.877
12 Week Baseline	15.56 ± 4.43 — 1.861 ± 0.460	19.99 — 2.006
XGBRegressor	17.62 ± 4.88 — 2.135 ± 0.558	21.38 — 2.111
GRU	18.68 ± 3.36 — 2.463 ± 0.534	17.92 — 2.390
LSTM	18.96 ± 3.52 — 2.500 ± 0.616	18.41 — 2.432
One Day Baseline	20.36 ± 3.93 — 2.657 ± 0.551	18.95 — 2.432
Prophet	17.36 ± 5.18 — 1.947 ± 0.426	25.45 — 2.458
SVR	18.10 ± 3.24 — 2.449 ± 0.552	23.44 — 2.694

Table 5.9: Table summarizing the results for daily granularity.

For the daily granularity, the models start with the total number of entries for the customer minus 84 entries as training and predict every time 30 entries until reaching the last prediction instant where the model has a training size of the total number of entries for the customer minus the last 7 entries. The results for daily granularity for the second customer are reported in table 5.9. In this case, TFT is no longer the best-performing model, probably due to the fact of 24 times less quantity of data due to the daily granularity. Though, it is still a well-performing model on both blocked k-fold cross-validation and last split results. The four-week baseline and the one-week baseline models perform better in terms of the last split MAE. This gives an indication of how with a low amount of data a simple baseline may provide reasonable results and on average they perform better than sophisticated models since these models are not able to be trained well due to the low amount and irregular pattern of this data. Looking at the last split MAPE instead the CNN outperforms the other models even

though by less than a percentage point. Actually, in terms of blocked k-fold validation, it results to be the worst in both MAPE and MAE but on the last split it recovers, probably the increase of data lead to a significant improvement to this model.

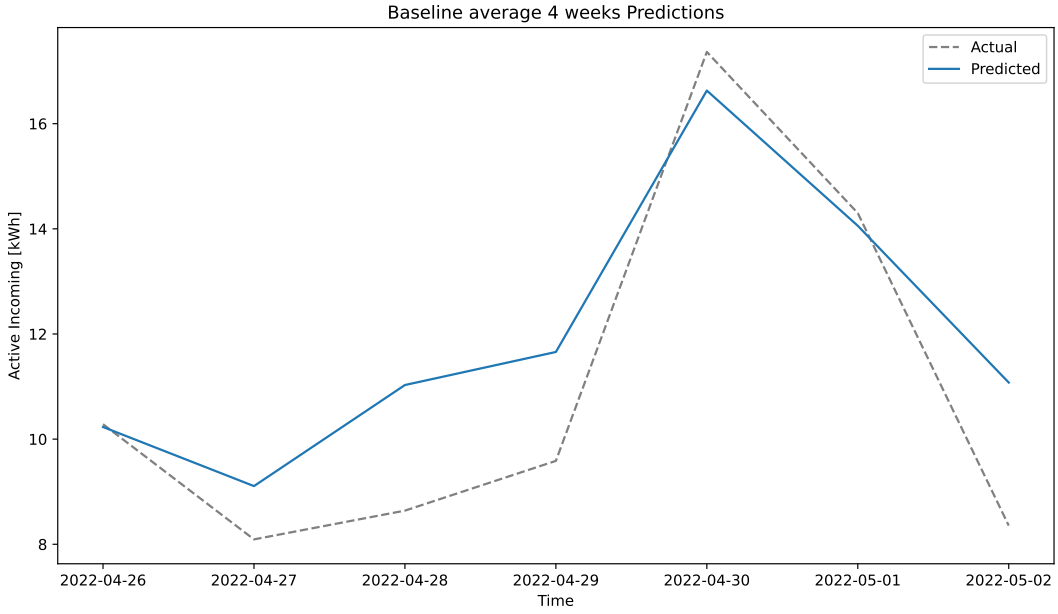


Figure 5.40: The forecasts produced by the four-week baseline for the last split in comparison to the actual values.

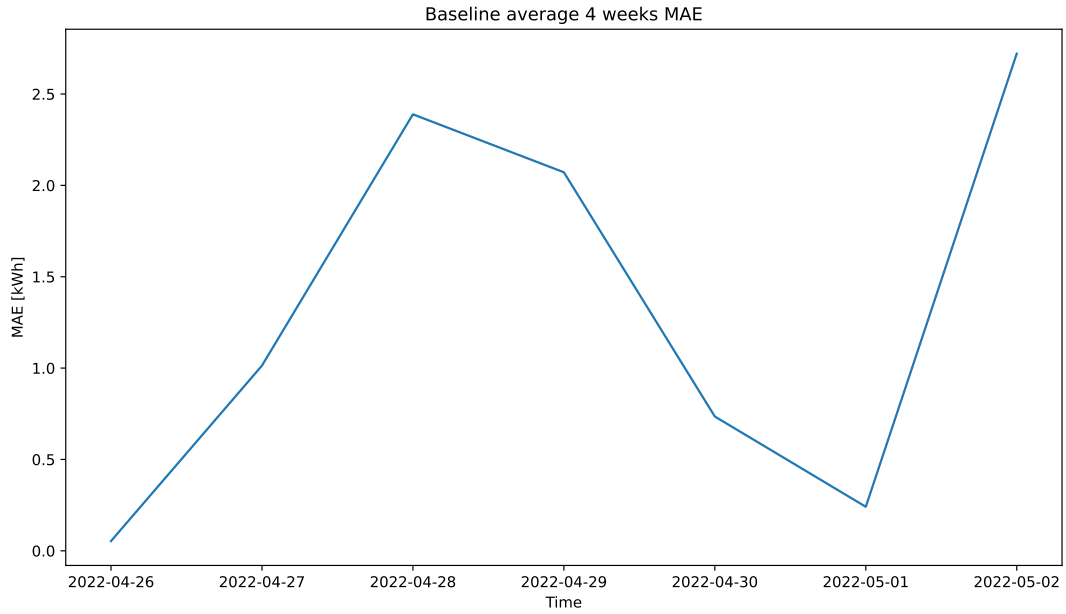


Figure 5.41: The absolute error of forecasts produced by the four-week baseline for the last split in comparison to the actual values.

Also, the SARIMA and hist gradient boosting regressor models present good results on the last time split and also on the blocked k-fold validation showing consistency over time and good results also with few training data. Instead, the AutoML approach obtained a one-week MAPE of 19.88%

and a MAE of 2.197 kWh. As for demand prediction, AutoML presents not very good results probably due to the scarcity of training data and the fact of AutoML of trying very complex architectures when also simple ones may work well in certain cases.

The forecasts produced by the four-week baseline for the last split in comparison to the actual values are reported in figure 5.40. It can be noticed that it accurately follows the signal meaning that this baseline can provide a good indication of what will be the daily consumption for this customer. The absolute error of forecasts produced by the four-week baseline for the last split in comparison to the actual values is reported in figure 5.41. While the absolute percentage error of forecasts produced by the four-week baseline for the last split in comparison to the actual values is reported in figure 5.42. It can be seen that the error increases until the third day, then decreases and explodes on the last day both in terms of absolute error and in terms of absolute percentage error. This confirms that there is no clear pattern and it really depends on the customer's overall daily consumption, which can be very different also on the same day of the week, although the four-week average does seem to capture some basic information about consumption.

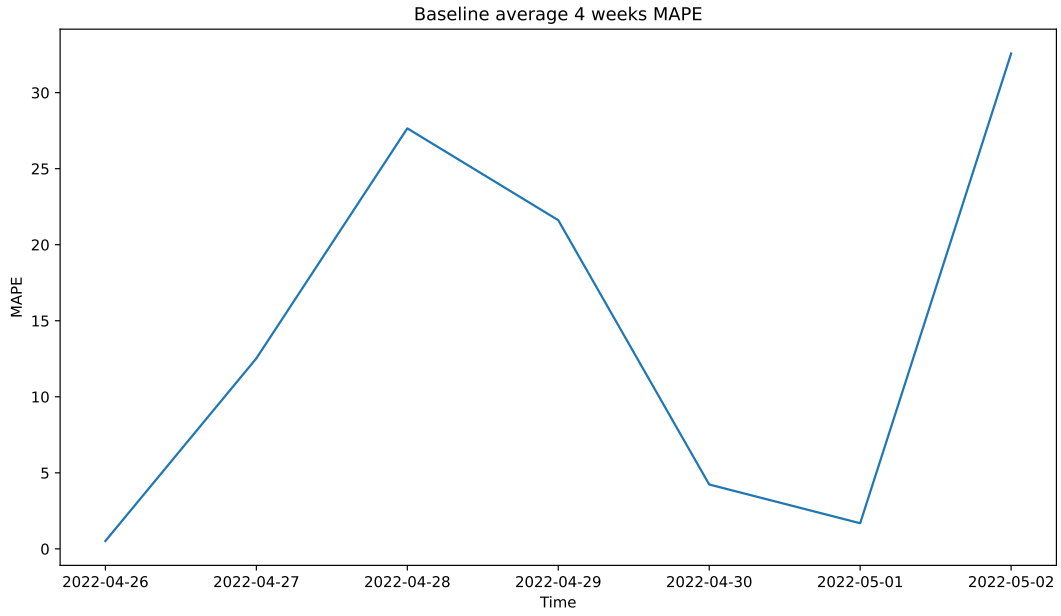


Figure 5.42: The absolute percentage error of forecasts produced by the four-week baseline for the last split in comparison to the actual values.

6 Conclusions

The chapter summarizes the work done and some conclusions are drawn from it. At the end of the chapter, some ideas for future work are suggested.

6.1 Summary

The accurate forecasting of customers' electricity demand is of vital importance for electricity suppliers. It allows them to optimize the purchase of the necessary electricity without having to rely solely on the instantaneous electricity market, thereby avoiding potential cost fluctuations. Furthermore, by accurately forecasting demand, suppliers can efficiently offer customers the necessary electricity and supply it at a competitive price, ensuring customer satisfaction and loyalty. Additionally, forecasting the production from their own photovoltaic plants becomes crucial in determining the amount of electricity that needs to be purchased. Lastly, understanding the consumption habits of individual customers by establishing a reference consumption baseline aids in developing personalized energy solutions and promoting energy efficiency initiatives. This allows the addition of potential new services connected to the field of Demand Side Management (DSM) which aims to increase customer retention.

The study addressed the following research questions:

1. Can the customers' electricity demand be forecasted based on past aggregated consumption data? The findings contribute to understanding the feasibility of utilizing historical data to develop a reliable forecasting model, enabling effective energy management and resource allocation.

2. Can the photovoltaic plants' production be forecasted based on past aggregated production data? The research examined the potential for accurately predicting PV plants' output using historical production data, facilitating the integration of renewable energy sources and optimizing energy procurement strategies.

3. Can a consumption baseline of individual customers be established based on past consumption data? The study explored the possibility of establishing consumption baselines for individual customers, offering insights into personalized energy solutions, energy efficiency, and tailored energy-saving recommendations.

By addressing these research questions, this study advances knowledge in forecasting electricity demand, photovoltaic production, and the establishment of consumption baselines. The findings provide practical implications for enhancing energy management strategies, optimizing resource allocation, promoting renewable energy integration, and improving customer-centric energy solutions.

A forecasting system was designed for addressing these research questions. The system architecture was designed in order to build a Software as a Service (SaaS) capable of satisfying different use cases and being used directly by energy retailers. A prototype was implemented with a focus on key components for validating the core system functionalities for each specific use case. In particular, the training of different models, the forecast of new data, and the evaluation of the performance of the developed models were implemented for all the use cases. As evaluation methodologies, the blocked k-fold cross-validation and the test on the last split were adopted using as metrics either the MAPE or MAE depending on the specific use case. In each of the following subsections, a comprehensive analysis of the results obtained for each specific use case is provided.

6.1.1 Electricity demand forecasting

For electricity demand forecasting the aggregated consumption data over the customers was used. This data was initially explored and analyzed and the correlation with weather data was also studied.

After this, many different models were developed for producing accurate forecasts on this use case. Experimental results suggested that the TFT model was the best-performing model for the hourly granularity with a MAPE of 14.76% on the test set composed of the last month of data. Instead, for the daily granularity, the best-performing model was the CNN model with a MAPE of 7.49%. These results can be considered a good indication of the soundness of the proposed approach. For answering whether the customers' electricity demand can be forecasted based on past aggregated consumption data the answer seems to be positive considering the promising results obtained. Taking into consideration the forecasts with a daily granularity, energy retailers can gain valuable insights into the expected energy demand for upcoming days and, instead of basing their purchases based on simple baselines or standard statistical methods, they can optimize their purchases saving many percentage points in excess/defect which results in monetary savings for the company. When a finer granularity is needed, as in the case of MIWenergía, forecasts results with an hourly granularity can be used and even though these are less accurate, they still offer useful information when considered across the entire period.

6.1.2 Electricity production forecasting

For electricity production forecasting the aggregated production data over the PV plants was used. The target of the predictions is the mean percentage of production, which is calculated as the division of the total produced energy by the total power of the PV plants. This data was initially explored and analyzed and the correlation with weather data, including solar energy data, was also studied. After this, many different models were developed for producing accurate forecasts on this use case. Experimental results suggested that the GRU model was the best-performing model for the hourly granularity with a MAE of 2.79% on the test set composed of the last week of the percentage of production data. These results can be considered a good indication of the soundness of the proposed approach. For answering whether the photovoltaic plants' production can be forecasted based on past aggregated production data the answer seems to be positive considering the promising results obtained. Taking into consideration the forecasts with an hourly granularity, energy retailers can access crucial insights into the expected energy production for upcoming days with higher accuracy compared to relying on simple baselines, standard statistical methods, or estimates just based on weather forecasts. Integrating these production forecasts with demand forecasts enables retailers to optimize their purchasing decisions effectively. By accurately forecasting production, retailers can avoid unnecessary excess or deficits in energy procurement, leading to substantial cost savings for the company. In fact, providing forecasts for both production and demand enables retailers to operate efficiently and maximize their financial returns.

6.1.3 Consumption baseline forecasting

For consumption baseline forecasting the consumption data of three customers was used. This data was initially explored and analyzed and the correlation with weather data was also studied. After this, many different models were developed for producing accurate forecasts on this use case with a focus on the second customer. Experimental results suggested that the TFT model was the best-performing model for the hourly granularity in terms of MAE with 0.231 kWh on the test set composed of the last week of data and it presents a MAPE of 44.19%. The GRU model performed slightly better in terms of MAPE with 42.55% but with a MAE of 0.257 kWh. The value of the MAPE is close to 50% indicating that on average the predictions are distant from the actual value by half of this value. In general, this data is not well predictable and also the different baselines confirmed that having a look at the close history there is no relevant daily or weekly repetition with MAPE and MAE both higher than the TFT model. Instead, for the daily granularity, no model is outperforming the four-week baseline in terms of MAE with 1.318 kWh and it presents a MAPE of 14.40%. The CNN model performed slightly better in terms of MAPE with 13.46% but with a MAE of 1.751 kWh. Even though the value of the MAPE with daily granularity is noticeably reduced with respect to the hourly granularity, the results provided an indication of how with a low amount of data a simple baseline may provide reasonable results and on average they perform better than sophisticated models since these models are not able to be trained well due to the low amount and irregular pattern of this data. For answering whether a

consumption baseline of individual customers can be established based on past consumption data the answer seems to be negative considering the quality of data of the provided customers and the limited amount of historical data. More studies should be conducted in this direction for being able to obtain accurate and robust consumption baseline forecasts.

6.2 Future works

In the following, further possibilities for future study that were not included in this thesis but could be addressed in subsequent research are reported. MIWenergía acknowledges the limitations of the provided data but expresses satisfaction with the initial results obtained through these solutions, indicating a desire to continue the collaboration. Therefore, ideas for future works are presented herein, aiming to expand upon the current research and explore additional avenues of investigation.

The first future work is to perform a long-term evaluation to explore the long-term performance and sustainability of the developed forecasting models. It would be beneficial to assess the accuracy and reliability of the predictions over extended time periods and analyze any potential deviations or biases that may arise over time. This evaluation will provide valuable insights into the stability and robustness of the models and their applicability in real-world scenarios.

Another critical point is related to data improvement and availability. Future research could focus on improving and cleaning the available data while exploring the use of additional features. Employing a more data-centric AI approach coupled with the increase of available data could allow for exploring more sophisticated and deeper architectures for the treated use cases. Furthermore, investigating alternative ways to train and use the models can contribute to enhancing their performance and efficiency.

An interesting future work can be of extending the concept of model combination adopted in demand and production forecasting use cases and considering the use of ensemble learning techniques could be valuable for finding optimal combinations of models for various use cases. Ensemble methods can harness the strengths of multiple models to improve overall forecasting accuracy and reliability. By combining the predictions of multiple models, for example using a weighted average, ensemble learning approaches can mitigate the weaknesses of individual models and enhance the overall forecasting performance. Based on this concept, with the continuous advancement of AutoML, this technology can be used to obtain stronger ensembles letting it find good models providing many days as available training time.

Future research should focus on improving the accuracy and reliability of the consumption baseline, which serves as a reference for individual customers' energy consumption. A deeper understanding of individual customers' energy consumption habits and the factors influencing their choices can be helpful in this objective. Exploring alternative models and techniques, as well as incorporating additional data sources, could enhance the precision of baseline forecasts. One possibility, having more customers available with more historical data, may also be to find a general model for dealing with all customers. Furthermore, investigating tariff granularity can be a good future direction to understand whether it could be applicable to produce good results for this use case.

Another future direction to offer scalability, flexibility, and accessibility for MIWenergía and potentially other energy retailers, is to implement the entire system as a SaaS solution with the architecture proposed in chapter 3. This would allow for easier deployment, maintenance, and continuous improvement of the developed forecasting models. Moreover, the presented architecture would be able to support additional use cases other than the ones discussed in this thesis.

Consumption disaggregation on a single customer level was also thought to be an interesting use case for MIWenergía. It consists of estimating individual appliances' energy consumption from individual customers' energy consumption data. This would be an interesting application to explore in the near future which could provide valuable insights to both MIWenergía and its customers and facilitate more targeted energy-saving strategies. However, this use case requires dedicated devices for collecting finer aggregated data with a minute-level granularity. Moreover, a first round of data collection and ground truth annotation is required for developing dedicated models for this use case.

These future research directions have the potential to build upon the initial findings of this thesis and contribute to the ongoing collaboration between MIWenergía and the research community in the

field of forecasting customers' electricity demand, optimizing energy procurement, and understanding consumption patterns. By exploring these possibilities, further insights can be gained, leading to more accurate forecasting models, improved energy management strategies, and enhanced customer-centric approaches in the electricity sector.

Bibliography

[1] Muhammad Waseem Ahmad, Monjur Mourshed, and Yacine Rezgui. "Tree-based ensemble methods for predicting PV power generation and their comparison with support vector regression". In: *Energy* 164 (2018), pp. 465–474. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2018.08.207. URL: https://www.sciencedirect.com/science/article/pii/S0360544218317432.

[2] Nesreen K. Ahmed, Amir F. Atiya, Neamat El Gayar, and Hisham El-Shishiny. "An Empirical Comparison of Machine Learning Models for Time Series Forecasting". In: *Econometric Reviews* 29.5-6 (2010), pp. 594–621. DOI: 10.1080/07474938.2010.481556. URL: https://doi.org/10.1080/07474938.2010.481556.

[3] R. Ahmed, V. Sreeram, Y. Mishra, and M.D. Arif. "A review and evaluation of the state-of-the-art in PV solar power forecasting: Techniques and optimization". In: *Renewable and Sustainable Energy Reviews* 124 (2020), p. 109792. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2020.109792. URL: https://www.sciencedirect.com/science/article/pii/S1364032120300885.

[4] Mohanad S. Al-Musaylh, Ravinesh C. Deo, Jan F. Adamowski, and Yan Li. "Short-term electricity demand forecasting with MARS, SVR and ARIMA models using aggregated demand data in Queensland, Australia". In: *Advanced Engineering Informatics* 35 (2018), pp. 1–16. ISSN: 1474-0346. DOI: https://doi.org/10.1016/j.aei.2017.11.002. URL: https://www.sciencedirect.com/science/article/pii/S1474034617301477.

[5] Mohammad H. Alobaidi, Fateh Chebana, and Mohamed A. Meguid. "Robust ensemble learning framework for day-ahead forecasting of household based energy consumption". In: *Applied Energy* 212 (2018), pp. 997–1012. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2017.12.054. URL: https://www.sciencedirect.com/science/article/pii/S0306261917317695.

[6] Ahmad Alsharef, Sonia, Karan Kumar, and Celestine Iwendi. "Time Series Data Modeling Using Advanced Machine Learning and AutoML". In: *Sustainability* 14.22 (2022). ISSN: 2071-1050. DOI: 10.3390/su142215292. URL: https://www.mdpi.com/2071-1050/14/22/15292.

[7] K.P. Amber, R. Ahmad, M.W. Aslam, A. Kousar, M. Usman, and M.S. Khan. "Intelligent techniques for forecasting electricity consumption of buildings". In: *Energy* 157 (2018), pp. 886–893. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2018.05.155. URL: https://www.sciencedirect.com/science/article/pii/S036054421830999X.

[8] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F.J. Martinez de Pison, and F. Antonanzas-Torres. "Review of photovoltaic power forecasting". In: *Solar Energy* 136 (2016), pp. 78–111. ISSN: 0038-092X. DOI: https://doi.org/10.1016/j.solener.2016.06.069. URL: https://www.sciencedirect.com/science/article/pii/S0038092X1630250X.

[9] J.S. Armstrong. *Principles of Forecasting: A Handbook for Researchers and Practitioners*. International Series in Operations Research & Management Science. Springer, 2001. ISBN: 9780792374015. URL: https://books.google.it/books?id=XdE4m_xMfL8C.

[10] Srihari Athiyarath, Mousumi Paul, and Srivatsa Krishnaswamy. "A Comparative Study and Analysis of Time Series Forecasting Techniques". In: *SN Computer Science* 1 (2020). ISSN: 2661-8907. DOI: 10.1007/s42979-020-00180-5. URL: https://doi.org/10.1007/s42979-020-00180-5.

[11] Florian Barbieri, Sumedha Rajakaruna, and Arindam Ghosh. "Very short-term photovoltaic power forecasting with cloud modeling: A review". In: *Renewable and Sustainable Energy Reviews* 75 (2017), pp. 242–263. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2016.10.068. URL: https://www.sciencedirect.com/science/article/pii/S136403211630733X.

[12] Jatin Bedi and Durga Toshniwal. "Deep learning framework to forecast electricity demand". In: *Applied Energy* 238 (2019), pp. 1312–1326. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2019.01.113. URL: https://www.sciencedirect.com/science/article/pii/S0306261919301217.

[13] Souhaib Ben Taieb, Gianluca Bontempi, Amir F. Atiya, and Antti Sorjamaa. "A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition". In: *Expert Systems with Applications* 39.8 (2012), pp. 7067–7083. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2012.01.039. URL: https://www.sciencedirect.com/science/article/pii/S0957417412000528.

[14] Christoph Bergmeir and José M. Benítez. "On the use of cross-validation for time series predictor evaluation". In: *Information Sciences* 191 (2012). Data Mining for Software Trustworthiness, pp. 192–213. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2011.12.028. URL: https://www.sciencedirect.com/science/article/pii/S0020025511006773.

[15] Anastasia Borovykh, Sander Bohte, and Cornelis W. Oosterlee. *Conditional Time Series Forecasting with Convolutional Neural Networks*. 2017. DOI: 10.48550/ARXIV.1703.04691. URL: https://arxiv.org/abs/1703.04691.

[16] Lim Bryan and Zohren Stefan. "Time-series forecasting with deep learning: a survey". In: *Philosophical Transactions of the Royal Society A* 379 (2021). ISSN: 1471-2962. DOI: https://doi.org/10.1098/rsta.2020.0209. URL: https://royalsocietypublishing.org/doi/full/10.1098/rsta.2020.0209.

[17] Mengmeng Cai, Manisa Pipattanasomporn, and Saifur Rahman. "Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques". In: *Applied Energy* 236 (2019), pp. 1078–1088. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2018.12.042. URL: https://www.sciencedirect.com/science/article/pii/S0306261918318609.

[18] Lijuan Cao. "Support vector machines experts for time series forecasting". In: *Neurocomputing* 51 (2003), pp. 321–339. ISSN: 0925-2312. DOI: https://doi.org/10.1016/S0925-2312(02)00577-5. URL: https://www.sciencedirect.com/science/article/pii/S0925231202005775.

[19] Vitor Cerqueira, Luis Torgo, and Igor Mozetič. "Evaluating time series forecasting models: an empirical study on performance estimation methods". In: *Machine Learning* 109 (2020), pp. 1997–2028. ISSN: 1573-0565. DOI: 10.1007/s10994-020-05910-7. URL: https://doi.org/10.1007/s10994-020-05910-7.

[20] Wen Chen, Kaile Zhou, Shanlin Yang, and Cheng Wu. "Data quality of electricity consumption data in a smart grid environment". In: *Renewable and Sustainable Energy Reviews* 75 (2017), pp. 98–105. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2016.10.054. URL: https://www.sciencedirect.com/science/article/pii/S1364032116307109.

[21] Yi-Wei Chen, Qingquan Song, and Xia Hu. "Techniques for Automated Machine Learning". In: *SIGKDD Explor. Newsl.* 22.2 (Jan. 2021), pp. 35–50. ISSN: 1931-0145. DOI: 10.1145/3447556.3447567. URL: https://doi.org/10.1145/3447556.3447567.

[22] Ying Chen, Wei Xian Xue, and Xing Long Xie. "Big-Data-Based Modeling of Electricity Consumption Behavior". In: *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. 2018, pp. 1380–1387. DOI: 10.1109/IAEAC.2018.8577770.

[23] Gopal Chitalia, Manisa Pipattanasomporn, Vishal Garg, and Saifur Rahman. "Robust short-term electrical load forecasting framework for commercial buildings using deep recurrent neural networks". In: *Applied Energy* 278 (2020), p. 115410. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2020.115410. URL: https://www.sciencedirect.com/science/article/pii/S0306261920309223.

[24] Utpal Kumar Das, Kok Soon Tey, Mehdi Seyedmahmoudian, Saad Mekhilef, Moh Yamani Idna Idris, Willem Van Deventer, Bend Horan, and Alex Stojcevski. "Forecasting of photovoltaic power generation and model optimization: A review". In: *Renewable and Sustainable Energy Reviews* 81 (2018), pp. 912–928. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2017.08.017. URL: https://www.sciencedirect.com/science/article/pii/S1364032117311620.

[25] Gabriel de Freitas Viscondi and Solange N. Alves-Souza. "A Systematic Literature Review on big data for solar photovoltaic electricity generation forecasting". In: *Sustainable Energy Technologies and Assessments* 31 (2019), pp. 54–63. ISSN: 2213-1388. DOI: https://doi.org/10.1016/j.seta.2018.11.008. URL: https://www.sciencedirect.com/science/article/pii/S2213138818301036.

[26] Jan G. De Gooijer and Rob J. Hyndman. "25 years of time series forecasting". In: *International Journal of Forecasting* 22.3 (2006). Twenty five years of forecasting, pp. 443–473. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2006.01.001. URL: https://www.sciencedirect.com/science/article/pii/S0169207006000021.

[27] Domingos S. de O. Santos Júnior, João F.L. de Oliveira, and Paulo S.G. de Mattos Neto. "An intelligent hybridization of ARIMA with machine learning models for time series forecasting". In: *Knowledge-Based Systems* 175 (2019), pp. 72–86. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2019.03.011. URL: https://www.sciencedirect.com/science/article/pii/S0950705119301327.

[28] Chirag Deb, Fan Zhang, Junjing Yang, Siew Eang Lee, and Kwok Wei Shah. "A review on time series forecasting techniques for building energy consumption". In: *Renewable and Sustainable Energy Reviews* 74 (2017), pp. 902–924. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2017.02.085. URL: https://www.sciencedirect.com/science/article/pii/S1364032117303155.

[29] Difan Deng, Florian Karl, Frank Hutter, Bernd Bischl, and Marius Lindauer. *Efficient Automated Deep Learning for Time Series Forecasting*. 2022. DOI: 10.48550/ARXIV.2205.05511. URL: https://arxiv.org/abs/2205.05511.

[30] Jianguang Deng and Panida Jirutitijaroen. "Short-term load forecasting using time series analysis: A case study for Singapore". In: *2010 IEEE Conference on Cybernetics and Intelligent Systems*. 2010, pp. 231–236. DOI: 10.1109/ICCIS.2010.5518553.

[31] Ashwini Doke and Madhava Gaikwad. "Survey on Automated Machine Learning (AutoML) and Meta learning". In: *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. 2021, pp. 1–5. DOI: 10.1109/ICCCNT51525.2021.9579526.

[32] Bing Dong, Zhaoxuan Li, S.M. Mahbobur Rahman, and Rolando Vega. "A hybrid model approach for forecasting future residential electricity consumption". In: *Energy and Buildings* 117 (2016), pp. 341–351. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2015.09.033. URL: https://www.sciencedirect.com/science/article/pii/S0378778815302735.

[33] Pei Du, Jianzhou Wang, Wendong Yang, and Tong Niu. "Multi-step ahead forecasting in electrical power system using a hybrid forecasting system". In: *Renewable Energy* 122 (2018), pp. 533–550. ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2018.01.113. URL: https://www.sciencedirect.com/science/article/pii/S096014811830123X.

[34] Shengdong Du, Tianrui Li, Yan Yang, and Shi-Jinn Horng. "Multivariate time series forecasting via attention-based encoder-decoder framework". In: *Neurocomputing* 388 (2020), pp. 269–279. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.12.118. URL: https://www.sciencedirect.com/science/article/pii/S0925231220300606.

[35] Salijona Dyrmishi, Radwa Elshawi, and Sherif Sakr. "A Decision Support Framework for AutoML Systems: A Meta-Learning Approach". In: *2019 International Conference on Data Mining Workshops (ICDMW)*. 2019, pp. 97–106. DOI: 10.1109/ICDMW.2019.00025.

[36] Radwa Elshawi, Mohamed Maher, and Sherif Sakr. *Automated Machine Learning: State-of-The-Art and Open Challenges*. 2019. DOI: 10.48550/ARXIV.1906.02287. URL: https://arxiv.org/abs/1906.02287.

[37] Cheng Fan, Jiayuan Wang, Wenjie Gang, and Shenghan Li. "Assessment of deep recurrent neural network-based strategies for short-term building energy predictions". In: *Applied Energy* 236 (2019), pp. 700–710. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2018.12.004. URL: https://www.sciencedirect.com/science/article/pii/S0306261918318221.

[38] Luís Ferreira, André Pilastri, Carlos Manuel Martins, Pedro Miguel Pires, and Paulo Cortez. "A Comparison of AutoML Tools for Machine Learning, Deep Learning and XGBoost". In: *2021 International Joint Conference on Neural Networks (IJCNN)*. 2021, pp. 1–8. DOI: 10.1109/IJCNN52387.2021.9534091.

[39] Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, and Frank Hutter. "Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning". In: (2020). DOI: 10.48550/ARXIV.2007.04074. URL: https://arxiv.org/abs/2007.04074.

[40] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. "Efficient and Robust Automated Machine Learning". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper/2015/file/11d0e6287202fced83f79975ec59a3a6-Paper.pdf.

[41] Mingming Gao, Jianjing Li, Feng Hong, and Dongteng Long. "Day-ahead power forecasting in a large-scale photovoltaic plant based on weather classification using LSTM". In: *Energy* 187 (2019), p. 115838. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2019.07.168. URL: https://www.sciencedirect.com/science/article/pii/S0360544219315105.

[42] Arpad Gellert, Adrian Florea, Ugo Fiore, Francesco Palmieri, and Paolo Zanetti. "A study on forecasting electricity production and consumption in smart cities and factories". In: *International Journal of Information Management* 49 (2019), pp. 546–556. ISSN: 0268-4012. DOI: https://doi.org/10.1016/j.ijinfomgt.2019.01.006. URL: https://www.sciencedirect.com/science/article/pii/S0268401218311368.

[43] Pieter Gijsbers, Erin LeDell, Janek Thomas, Sébastien Poirier, Bernd Bischl, and Joaquin Vanschoren. *An Open Source AutoML Benchmark*. 2019. DOI: 10.48550/ARXIV.1907.00909. URL: https://arxiv.org/abs/1907.00909.

[44] Jake Grigsby, Zhe Wang, and Yanjun Qi. *Long-Range Transformers for Dynamic Spatiotemporal Forecasting*. 2021. DOI: 10.48550/ARXIV.2109.12218. URL: https://arxiv.org/abs/2109.12218.

[45] Weihua He, Yongyun Wu, and Xiaohua Li. "Attention Mechanism for Neural Machine Translation: A survey". In: *2021 IEEE 5th Information Technology,Networking,Electronic and Automation Control Conference (ITNEC)*. Vol. 5. 2021, pp. 1485–1489. DOI: 10.1109/ITNEC52019.2021.9586824.

[46] Xin He, Kaiyong Zhao, and Xiaowen Chu. "AutoML: A survey of the state-of-the-art". In: *Knowledge-Based Systems* 212 (2021), p. 106622. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2020.106622. URL: https://www.sciencedirect.com/science/article/pii/S0950705120307516.

[47] Amirreza Heidari and Dolaana Khovalyg. "Short-term energy use prediction of solar-assisted water heating system: Application case of combined attention-based LSTM and time-series decomposition". In: *Solar Energy* 207 (2020), pp. 626–639. ISSN: 0038-092X. DOI: https://doi.org/10.1016/j.solener.2020.07.008. URL: https://www.sciencedirect.com/science/article/pii/S0038092X20307398.

[48] Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. "Recurrent Neural Networks for Time Series Forecasting: Current status and future directions". In: *International Journal of Forecasting* 37.1 (2021), pp. 388–427. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2020.06.008. URL: https://www.sciencedirect.com/science/article/pii/S0169207020300996.

[49] Rob J. Hyndman. "A brief history of forecasting competitions". In: *International Journal of Forecasting* 36.1 (2020). M4 Competition, pp. 7–14. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2019.03.015. URL: https://www.sciencedirect.com/science/article/pii/S016920701930086X.

[50] Rob J. Hyndman and Anne B. Koehler. "Another look at measures of forecast accuracy". In: *International Journal of Forecasting* 22.4 (2006), pp. 679–688. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2006.03.001. URL: https://www.sciencedirect.com/science/article/pii/S0169207006000239.

[51] Rich H. Inman, Hugo T.C. Pedro, and Carlos F.M. Coimbra. "Solar forecasting methods for renewable energy integration". In: *Progress in Energy and Combustion Science* 39.6 (2013), pp. 535–576. ISSN: 0360-1285. DOI: https://doi.org/10.1016/j.pecs.2013.06.002. URL: https://www.sciencedirect.com/science/article/pii/S0360128513000294.

[52] Indrajeet Y. Javeri, Mohammadhossein Toutiaee, Ismailcem B. Arpinar, John A. Miller, and Tom W. Miller. "Improving Neural Networks for Time-Series Forecasting using Data Augmentation and AutoML". In: *2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)*. 2021, pp. 1–8. DOI: 10.1109/BigDataService52369.2021.00006.

[53] Mei Jie, Gao Ciwei, Chen Xiao, and Yi Yongxian. "A customer baseline load prediction and optimization method based on non-demand-response factors". In: *2016 China International Conference on Electricity Distribution (CICED)*. 2016, pp. 1–5. DOI: 10.1109/CICED.2016.7576207.

[54] Haifeng Jin, Qingquan Song, and Xia Hu. "Auto-Keras: An Efficient Neural Architecture Search System". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. New York, NY, USA: Association for Computing Machinery, 2019, 1946–1956. ISBN: 9781450362016. DOI: 10.1145/3292500.3330648. URL: https://doi.org/10.1145/3292500.3330648.

[55] Shubhra Kanti Karmaker ("Santu"), Md. Mahadi Hassan, Micah J. Smith, Lei Xu, Chengxiang Zhai, and Kalyan Veeramachaneni. "AutoML to Date and Beyond: Challenges and Opportunities". In: *ACM Comput. Surv.* 54.8 (Oct. 2021). ISSN: 0360-0300. DOI: 10.1145/3470918. URL: https://doi.org/10.1145/3470918.

[56] Kübra Kaysal, Fatih Onur Hocaoğlu, and Nihat Öztürk. "Comparison the Performance of Different Optimization Methods in Artificial Intelligence Based Electricity Production Forecasting". In: *2022 10th International Conference on Smart Grid (icSmartGrid)*. 2022, pp. 236–239. DOI: 10.1109/icSmartGrid55722.2022.9848724.

[57] Junhong Kim, Jihoon Moon, Eenjun Hwang, and Pilsung Kang. "Recurrent inception convolution neural network for multi short-term load forecasting". In: *Energy and Buildings* 194 (2019), pp. 328–341. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2019.04.034. URL: https://www.sciencedirect.com/science/article/pii/S0378778819308072.

[58] Tae-Young Kim and Sung-Bae Cho. "Predicting residential energy consumption using CNN-LSTM neural networks". In: *Energy* 182 (2019), pp. 72–81. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2019.05.230. URL: https://www.sciencedirect.com/science/article/pii/S0360544219311223.

[59] Dan Li, Ya Tan, Yuanhang Zhang, Shuwei Miao, and Shuai He. "Probabilistic forecasting method for mid-term hourly load time series based on an improved temporal fusion transformer model". In: *International Journal of Electrical Power & Energy Systems* 146 (2023), p. 108743. ISSN: 0142-0615. DOI: https://doi.org/10.1016/j.ijepes.2022.108743. URL: https://www.sciencedirect.com/science/article/pii/S0142061522007396.

[60] Pengtao Li, Kaile Zhou, Xinhui Lu, and Shanlin Yang. "A hybrid deep learning model for short-term PV power forecasting". In: *Applied Energy* 259 (2020), p. 114216. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2019.114216. URL: https://www.sciencedirect.com/science/article/pii/S0306261919319038.

[61] Youru Li, Zhenfeng Zhu, Deqiang Kong, Hua Han, and Yao Zhao. "EA-LSTM: Evolutionary attention-based LSTM for time series prediction". In: *Knowledge-Based Systems* 181 (2019), p. 104785. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2019.05.028. URL: https://www.sciencedirect.com/science/article/pii/S0950705119302400.

[62] Huanyue Liao and Krishnanand Kaippilly Radhakrishnan. "Short-Term Load Forecasting with Temporal Fusion Transformers for Power Distribution Networks". In: *2022 IEEE Sustainable Power and Energy Conference (iSPEC)*. 2022, pp. 1–5. DOI: 10.1109/iSPEC54162.2022.10033079.

[63] Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister. "Temporal Fusion Transformers for interpretable multi-horizon time series forecasting". In: *International Journal of Forecasting* 37.4 (2021), pp. 1748–1764. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2021.03.012. URL: https://www.sciencedirect.com/science/article/pii/S0169207021000637.

[64] Tao Liu, Zehan Tan, Chengliang Xu, Huanxin Chen, and Zhengfei Li. "Study on deep reinforcement learning techniques for building energy consumption forecasting". In: *Energy and Buildings* 208 (2020), p. 109675. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2019.109675. URL: https://www.sciencedirect.com/science/article/pii/S0378778819324740.

[65] Yeqi Liu, Chuanyang Gong, Ling Yang, and Yingyi Chen. "DSTP-RNN: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction". In: *Expert Systems with Applications* 143 (2020), p. 113082. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2019.113082. URL: https://www.sciencedirect.com/science/article/pii/S0957417419307997.

[66] Peter Lusis, Kaveh Rajab Khalilpour, Lachlan Andrew, and Ariel Liebman. "Short-term residential load forecasting: Impact of calendar effects and forecast granularity". In: *Applied Energy* 205 (2017), pp. 654–669. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2017.07.114. URL: https://www.sciencedirect.com/science/article/pii/S0306261917309881.

[67] Gonçalo Luís, João Esteves, and Nuno Pinho da Silva. "Energy Forecasting Using an Ensamble of Machine Learning Methods Trained Only with Electricity Data". In: *2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*. 2020, pp. 449–453. DOI: 10.1109/ISGT-Europe47291.2020.9248865.

[68] Miguel López Santos, Xela García-Santiago, Fernando Echevarría Camarero, Gonzalo Blázquez Gil, and Pablo Carrasco Ortega. "Application of Temporal Fusion Transformer for Day-Ahead PV Power Forecasting". In: *Energies* 15.14 (2022). ISSN: 1996-1073. DOI: 10.3390/en15145232. URL: https://www.mdpi.com/1996-1073/15/14/5232.

[69] Zhitong Ma, Cantao Ye, and Weibin Ma. "Support vector regression for predicting building energy consumption in southern China". In: *Energy Procedia* 158 (2019). Innovative Solutions for Energy Transitions, pp. 3433–3438. ISSN: 1876-6102. DOI: https://doi.org/10.1016/j.egypro.2019.01.931. URL: https://www.sciencedirect.com/science/article/pii/S1876610219309762.

[70] Joseph Manu. *PMODERN TIME SERIES FORECASTING WITH PYTHON explore industry-ready time series forecasting using modern machine learning and deep learning*. PACKT PUBLISHING LIMITED, 2022. ISBN: 9781803246802.

[71] Ricardo P. Masini, Marcelo C. Medeiros, and Eduardo F. Mendes. "Machine learning advances for time series forecasting". In: *Journal of Economic Surveys* 37.1 (2023), pp. 76–111. DOI: https://doi.org/10.1111/joes.12429. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/joes.12429. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/joes.12429.

[72] A. Mellit, A. Massi Pavan, and V. Lughi. "Deep learning neural networks for short-term photovoltaic power forecasting". In: *Renewable Energy* 172 (2021), pp. 276–288. ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2021.02.166. URL: https://www.sciencedirect.com/science/article/pii/S0960148121003475.

[73] S. Mirasgedis, Y. Sarafidis, E. Georgopoulou, D.P. Lalas, M. Moschovits, F. Karagiannis, and D. Papakonstantinou. "Models for mid-term electricity demand forecasting incorporating weather influences". In: *Energy* 31.2 (2006), pp. 208–227. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2005.02.016. URL: https://www.sciencedirect.com/science/article/pii/S0360544205000393.

[74] Elena Mocanu, Phuong H. Nguyen, Madeleine Gibescu, and Wil L. Kling. "Deep learning for estimating building energy consumption". In: *Sustainable Energy, Grids and Networks* 6 (2016), pp. 91–99. ISSN: 2352-4677. DOI: https://doi.org/10.1016/j.segan.2016.02.005. URL: https://www.sciencedirect.com/science/article/pii/S2352467716000163.

[75] Shahzad Muzaffar and Afshin Afshari. "Short-Term Load Forecasts Using LSTM Networks". In: *Energy Procedia* 158 (2019). Innovative Solutions for Energy Transitions, pp. 2922–2927. ISSN: 1876-6102. DOI: https://doi.org/10.1016/j.egypro.2019.01.952. URL: https://www.sciencedirect.com/science/article/pii/S1876610219310008.

[76] Thiloshon Nagarajah and Guhanathan Poravi. "A Review on Automated Machine Learning (AutoML) Systems". In: *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*. 2019, pp. 1–6. DOI: 10.1109/I2CT45611.2019.9033810.

[77] Amril Nazir, Abdul Khalique Shaikh, Abdul Salam Shah, and Ashraf Khalil. "Forecasting energy consumption demand of customers in smart grid using Temporal Fusion Transformer (TFT)". In: *Results in Engineering* 17 (2023), p. 100888. ISSN: 2590-1230. DOI: https://doi.org/10.1016/j.rineng.2023.100888. URL: https://www.sciencedirect.com/science/article/pii/S2590123023000154.

[78] Cuong Nguyen and Jeremy J. Roberts. "Green Button Data-Access Model for Smart Cities: Lessons Learned on Security, Transfer, Authorization, and Standards-Compliance in Sharing Energy & Water Usage Data". In: *Proceedings of the 2nd ACM/EIGSCC Symposium on Smart Cities and Communities*. SCC '19. Association for Computing Machinery, 2019. ISBN: 9781450369787. DOI: 10.1145/3357492.3358624. URL: https://doi.org/10.1145/3357492.3358624.

[79] Duc Anh Nguyen, Anna V. Kononova, Stefan Menzel, Bernhard Sendhoff, and Thomas Back. "Efficient AutoML via Combinational Sampling". In: *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2021, pp. 01–10. DOI: 10.1109/SSCI50451.2021.9660073.

[80] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. "A review on the attention mechanism of deep learning". In: *Neurocomputing* 452 (2021), pp. 48–62. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2021.03.091. URL: https://www.sciencedirect.com/science/article/pii/S092523122100477X.

[81] Mariana Oliveira and Luis Torgo. "Ensembles for Time Series Forecasting". In: *Proceedings of the Sixth Asian Conference on Machine Learning*. Ed. by Dinh Phung and Hang Li. Vol. 39. Proceedings of Machine Learning Research. Nha Trang City, Vietnam: PMLR, Nov. 2015, pp. 360–370. URL: https://proceedings.mlr.press/v39/oliveira14.html.

[82] Rafael Pedro and Arlindo L. Oliveira. "Assessing the Impact of Attention and Self-Attention Mechanisms on the Classification of Skin Lesions". In: *2022 International Joint Conference on Neural Networks (IJCNN)*. 2022, pp. 1–8. DOI: 10.1109/IJCNN55064.2022.9892274.

[83] Ratchakit Phetrittikun, Kerdkiat Suvirat, Thanakron Na Pattalung, Chanon Kongkamol, Thammasin Ingviya, and Sitthichok Chaichulee. "Temporal Fusion Transformer for forecasting vital sign trajectories in intensive care patients". In: *2021 13th Biomedical Engineering International Conference (BMEiCON)*. 2021, pp. 1–5. DOI: 10.1109/BMEiCON53485.2021.9745215.

[84] Radu Platon, Vahid Raissi Dehkordi, and Jacques Martel. "Hourly prediction of a building's electricity consumption using case-based reasoning, artificial neural networks and principal component analysis". In: *Energy and Buildings* 92 (2015), pp. 10–18. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2015.01.047. URL: https://www.sciencedirect.com/science/article/pii/S0378778815000651.

[85] Xi Qi, Lihua Tian, Chen Li, Hui Song, and Jiahui Yan. "Singing Melody Extraction Based on Combined Frequency-Temporal Attention and Attentional Feature Fusion with Self-Attention". In: *2022 IEEE International Symposium on Multimedia (ISM)*. 2022, pp. 220–227. DOI: 10.1109/ISM55400.2022.00050.

[86] V. Sackdara, S. Premrudeepreechacharn, and K. Ngamsanroaj. "Electricity demand forecasting of Electricite Du Lao (EDL) using neural networks". In: *TENCON 2010 - 2010 IEEE Region 10 Conference*. 2010, pp. 640–644. DOI: 10.1109/TENCON.2010.5686767.

[87] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. "Financial time series forecasting with deep learning : A systematic literature review: 2005-2019". In: *Applied Soft Computing* 90 (2020), p. 106181. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2020.106181. URL: https://www.sciencedirect.com/science/article/pii/S1568494620301216.

[88] Minglei Shao, Xin Wang, Zhen Bu, Xiaobo Chen, and Yuqing Wang. "Prediction of energy consumption in hotel buildings via support vector machines". In: *Sustainable Cities and Society* 57 (2020), p. 102128. ISSN: 2210-6707. DOI: https://doi.org/10.1016/j.scs.2020.102128. URL: https://www.sciencedirect.com/science/article/pii/S2210670720301153.

[89] Zhipeng Shen, Yuanming Zhang, Jiawei Lu, Jun Xu, and Gang Xiao. "A novel time series forecasting model with deep learning". In: *Neurocomputing* 396 (2020), pp. 302–313. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2018.12.084. URL: https://www.sciencedirect.com/science/article/pii/S0925231219304461.

[90] Shun-Yao Shih, Fan-Keng Sun, and Hung-yi Lee. "Temporal pattern attention for multivariate time series forecasting". In: *Machine Learning* 108 (2019), pp. 1421–1441. ISSN: 1573-0565. DOI: 10.1007/s10994-019-05815-0. URL: https://doi.org/10.1007/s10994-019-05815-0.

[91] Arunesh Kumar Singh, S Khatoon Ibraheem, Md Muazzam, and DK Chaturvedi. "An overview of electricity demand forecasting techniques". In: *Network and complex systems* 3.3 (2013), pp. 38–48.

[92] Slawek Smyl. "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting". In: *International Journal of Forecasting* 36.1 (2020). M4 Competition, pp. 75–85. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2019.03.017. URL: https://www.sciencedirect.com/science/article/pii/S0169207019301153.

[93] Sobrina Sobri, Sam Koohi-Kamali, and Nasrudin Abd. Rahim. "Solar photovoltaic generation forecasting methods: A review". In: *Energy Conversion and Management* 156 (2018), pp. 459–497. ISSN: 0196-8904. DOI: https://doi.org/10.1016/j.enconman.2017.11.019. URL: https://www.sciencedirect.com/science/article/pii/S0196890417310622.

[94] Nivethitha Somu, Gauthama Raman M R, and Krithi Ramamritham. "A hybrid model for building energy consumption forecasting using long short term memory networks". In: *Applied Energy* 261 (2020), p. 114131. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2019.114131. URL: https://www.sciencedirect.com/science/article/pii/S0306261919318185.

[95] Nivethitha Somu, Gauthama Raman M R, and Krithi Ramamritham. "A deep learning framework for building energy consumption forecast". In: *Renewable and Sustainable Energy Reviews* 137 (2021), p. 110591. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2020.110591. URL: https://www.sciencedirect.com/science/article/pii/S1364032120308753.

[96] Evangelos Spiliotis, Andreas Kouloumos, Vassilios Assimakopoulos, and Spyros Makridakis. "Are forecasting competitions data representative of the reality?" In: *International Journal of Forecasting* 36.1 (2020). M4 Competition, pp. 37–53. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2018.12.007. URL: https://www.sciencedirect.com/science/article/pii/S0169207019300159.

[97] Stefano Frizzo Stefenon, Laio Oriel Seman, Viviana Cocco Mariani, and Leandro dos Santos Coelho. "Aggregating Prophet and Seasonal Trend Decomposition for Time Series Forecasting of Italian Electricity Spot Prices". In: *Energies* 16.3 (2023). ISSN: 1996-1073. DOI: 10.3390/en16031371. URL: https://www.mdpi.com/1996-1073/16/3/1371.

[98] James W. Taylor and Roberto Buizza. "Using weather ensemble predictions in electricity demand forecasting". In: *International Journal of Forecasting* 19.1 (2003), pp. 57–70. ISSN: 0169-2070. DOI: https://doi.org/10.1016/S0169-2070(01)00123-6. URL: https://www.sciencedirect.com/science/article/pii/S0169207001001236.

[99] Sean J Taylor and Benjamin Letham. "Forecasting at scale". In: *PeerJ Preprints* 5 (2017), e3190v2. ISSN: 2167-9843. DOI: 10.7287/peerj.preprints.3190v2. URL: https://doi.org/10.7287/peerj.preprints.3190v2.

[100] Ahmed Tealab. "Time series forecasting using artificial neural networks methodologies: A systematic review". In: *Future Computing and Informatics Journal* 3.2 (2018), pp. 334–340. ISSN: 2314-7288. DOI: https://doi.org/10.1016/j.fcij.2018.10.003. URL: https://www.sciencedirect.com/science/article/pii/S2314728817300715.

[101] Alper Tokgöz and Gözde Ünal. "A RNN based time series approach for forecasting turkish electricity load". In: *2018 26th Signal Processing and Communications Applications Conference (SIU)*. 2018, pp. 1–4. DOI: 10.1109/SIU.2018.8404313.

[102] Anh Truong, Austin Walters, Jeremy Goodsitt, Keegan Hines, C. Bayan Bruss, and Reza Farivar. "Towards Automated Machine Learning: Evaluation and Comparison of AutoML Approaches and Tools". In: *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. 2019, pp. 1471–1479. DOI: 10.1109/ICTAI.2019.00209.

[103] Lorenzo Vaccaro, Giuseppe Sansonetti, and Alessandro Micarelli. "An Empirical Review of Automated Machine Learning". In: *Computers* 10.1 (2021). ISSN: 2073-431X. DOI: 10.3390/computers10010011. URL: https://www.mdpi.com/2073-431X/10/1/11.

[104] William VanDeventer, Elmira Jamei, Gokul Sidarth Thirunavukkarasu, Mehdi Seyedmahmoudian, Tey Kok Soon, Ben Horan, Saad Mekhilef, and Alex Stojcevski. "Short-term PV power forecasting using hybrid GASVM technique". In: *Renewable Energy* 140 (2019), pp. 367–379. ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2019.02.087. URL: https://www.sciencedirect.com/science/article/pii/S0960148119302411.

[105] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. *Attention Is All You Need*. 2017. DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.

[106] Huaizhi Wang, Haiyan Yi, Jianchun Peng, Guibin Wang, Yitao Liu, Hui Jiang, and Wenxin Liu. "Deterministic and probabilistic forecasting of photovoltaic power based on deep convolutional neural network". In: *Energy Conversion and Management* 153 (2017), pp. 409–422. ISSN: 0196-8904. DOI: https://doi.org/10.1016/j.enconman.2017.10.008. URL: https://www.sciencedirect.com/science/article/pii/S019689041730910X.

[107] Jian Qi Wang, Yu Du, and Jing Wang. "LSTM based long-term energy consumption prediction with periodicity". In: *Energy* 197 (2020), p. 117197. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2020.117197. URL: https://www.sciencedirect.com/science/article/pii/S0360544220303042.

[108] Kejun Wang, Xiaoxia Qi, and Hongda Liu. "A comparison of day-ahead photovoltaic power forecasting models based on deep learning neural network". In: *Applied Energy* 251 (2019), p. 113315. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2019.113315. URL: https://www.sciencedirect.com/science/article/pii/S0306261919309894.

[109] Kejun Wang, Xiaoxia Qi, and Hongda Liu. "Photovoltaic power forecasting based LSTM-Convolutional Network". In: *Energy* 189 (2019), p. 116225. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2019.116225. URL: https://www.sciencedirect.com/science/article/pii/S0360544219319206.

[110] Ran Wang, Shilei Lu, and Wei Feng. "A novel improved model for building energy consumption prediction based on model integration". In: *Applied Energy* 262 (2020), p. 114561. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2020.114561. URL: https://www.sciencedirect.com/science/article/pii/S0306261920300738.

[111] Yi Wang, Dahua Gan, Mingyang Sun, Ning Zhang, Zongxiang Lu, and Chongqing Kang. "Probabilistic individual load forecasting using pinball loss guided LSTM". In: *Applied Energy* 235 (2019), pp. 10–20. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2018.10.078. URL: https://www.sciencedirect.com/science/article/pii/S0306261918316465.

[112] Lulu Wen, Kaile Zhou, and Shanlin Yang. "Load demand forecasting of residential buildings using a deep learning model". In: *Electric Power Systems Research* 179 (2020), p. 106073. ISSN: 0378-7796. DOI: https://doi.org/10.1016/j.epsr.2019.106073. URL: https://www.sciencedirect.com/science/article/pii/S037877961930392X.

[113] Binrong Wu, Lin Wang, and Yu-Rong Zeng. "Interpretable wind speed prediction with multivariate time series and temporal fusion transformers". In: *Energy* 252 (2022), p. 123990. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2022.123990. URL: https://www.sciencedirect.com/science/article/pii/S0360544222008933.

[114] Neo Wu, Bradley Green, Xue Ben, and Shawn O'Banion. *Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case*. 2020. DOI: 10.48550/ARXIV.2001.08317. URL: https://arxiv.org/abs/2001.08317.

[115] Weizhong Yan. "Toward Automatic Time-Series Forecasting Using Neural Networks". In: *IEEE Transactions on Neural Networks and Learning Systems* 23.7 (2012), pp. 1028–1039. DOI: 10.1109/TNNLS.2012.2198074.

[116] Yandong Yang, Weijun Hong, and Shufang Li. "Deep ensemble learning based probabilistic load forecasting in smart grids". In: *Energy* 189 (2019), p. 116324. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2019.116324. URL: https://www.sciencedirect.com/science/article/pii/S0360544219320195.

[117] M. Zamo, O. Mestre, P. Arbogast, and O. Pannekoucke. "A benchmark of statistical regression methods for short-term forecasting of photovoltaic electricity production, part I: Deterministic forecast of hourly production". In: *Solar Energy* 105 (2014), pp. 792–803. ISSN: 0038-092X. DOI: https://doi.org/10.1016/j.solener.2013.12.006. URL: https://www.sciencedirect.com/science/article/pii/S0038092X13005239.

[118] M. Zamo, O. Mestre, P. Arbogast, and O. Pannekoucke. "A benchmark of statistical regression methods for short-term forecasting of photovoltaic electricity production. Part II: Probabilistic forecast of daily production". In: *Solar Energy* 105 (2014), pp. 804–816. ISSN: 0038-092X. DOI: https://doi.org/10.1016/j.solener.2014.03.026. URL: https://www.sciencedirect.com/science/article/pii/S0038092X14001601.

[119] Haixiang Zang, Ruiqi Xu, Lilin Cheng, Tao Ding, Ling Liu, Zhinong Wei, and Guoqiang Sun. "Residential load forecasting based on LSTM fusing self-attention mechanism with pooling". In: *Energy* 229 (2021), p. 120682. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2021.120682. URL: https://www.sciencedirect.com/science/article/pii/S0360544221009312.

[120] G.Peter Zhang. "Time series forecasting using a hybrid ARIMA and neural network model". In: *Neurocomputing* 50 (2003), pp. 159–175. ISSN: 0925-2312. DOI: https://doi.org/10.1016/S0925-2312(01)00702-0. URL: https://www.sciencedirect.com/science/article/pii/S0925231201007020.

[121] Hao Zhang, Yajie Zou, Xiaoxue Yang, and Hang Yang. "A temporal fusion transformer for short-term freeway traffic speed multistep prediction". In: *Neurocomputing* 500 (2022), pp. 329–340. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2022.05.083. URL: https://www.sciencedirect.com/science/article/pii/S092523122200666X.

[122] Hai Zhong, Jiajun Wang, Hongjie Jia, Yunfei Mu, and Shilei Lv. "Vector field-based support vector regression for building energy consumption prediction". In: *Applied Energy* 242 (2019), pp. 403–414. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2019.03.078. URL: https://www.sciencedirect.com/science/article/pii/S0306261919304878.

[123] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. *Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting*. 2020. DOI: 10.48550/ARXIV.2012.07436. URL: https://arxiv.org/abs/2012.07436.

[124] Yi Zhou, Nanrun Zhou, Lihua Gong, and Minlin Jiang. "Prediction of photovoltaic power output based on similar day analysis, genetic algorithm and extreme learning machine". In: *Energy* 204 (2020), p. 117894. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2020.117894. URL: https://www.sciencedirect.com/science/article/pii/S036054422031001X.

[125] Lucas Zimmer, Marius Lindauer, and Frank Hutter. *Auto-PyTorch Tabular: Multi-Fidelity MetaLearning for Efficient and Robust AutoDL*. 2020. DOI: 10.48550/ARXIV.2006.13799. URL: https://arxiv.org/abs/2006.13799.

[126] Zeynep Çamurdan and Murat Can Ganiz. "Machine learning based electricity demand forecasting". In: *2017 International Conference on Computer Science and Engineering (UBMK)*. 2017, pp. 412–417. DOI: 10.1109/UBMK.2017.8093428.