LUISS

Degree Program in Data Science and Management

Course of Data Visualization

Comparative Analysis of XGBoost, LSTM, and TCN for Energy Demand

Prof. Blerina Sinaimeri

SUPERVISOR

Prof. Laura Luigi

CO-SUPERVISOR

Daniyar Meiram

CANDIDATE

Academic Year 2024/2025

Contents

Chapter 1

Introduction

1.1 Motivation and Background

In recent years, many countries have been fundamentally re-evaluating their energy systems, with a growing focus on the transition toward renewable energy sources and low-carbon technologies. This transformation extends beyond technological innovation and encompasses strategic political, economic, and social decisions that reshape national energy architectures and long-term development trajectories. At the same time, the increasing electrification of economies, the diffusion of digital technologies, and the integration of intermittent renewable sources have significantly increased the complexity of modern power systems.

A prominent example of such strategic decision-making is the national referendum held in Kazakhstan on October 6, 2024, regarding the adoption of nuclear energy. The debate surrounding this decision revealed a fundamental uncertainty in projections of future electricity demand. While some experts argue that existing and planned energy capacities will be insufficient to meet future demand, others contend that the expansion of wind and solar generation will be adequate to ensure energy security and system stability. This divergence of views highlights the central role of demand forecasting in shaping long-term energy strategies.

Inaccurate demand forecasts can lead to two opposite yet equally critical outcomes. On the one hand, overinvestment in generation capacity may result in inefficient allocation of financial resources and substantial economic losses, particularly in the context of capital-intensive nuclear projects. On the other hand, underinvestment in energy infrastructure may cause supply shortages, price volatility, and constraints on economic growth. Beyond economic implications, forecasting errors may also undermine grid reliability and slow down the pace of energy transition.

Moreover, the modernization of power grids and the increasing penetration of renewable energy sources have introduced nonlinear dynamics and heightened uncertainty into electricity demand patterns. Traditional statistical forecasting methods often struggle to capture such complexity, which motivates the adoption of advanced data-driven approaches. In this context, neural network–based models offer a promising framework for improving predictive accuracy, quantifying uncertainty, and enhancing interpretability, thereby supporting more informed policy-making and investment decisions in modern energy systems.

1.2 Problem Statement

While there have been advancements in developing energy systems' forecasting methodologies, numerous basic challenges still exist. First, forecast precision tends to decrease under extreme weather conditions as well as when dealing with complex environments; this is primarily due to traditional methods being unable to identify and depict the non-linear and dynamic nature of electricity demand and supply over time. Second, predictive uncertainty (an important consideration in managing risk and making strategic decisions) has yet to be sufficiently addressed within a majority of forecasting frameworks.

Third, there are many regions where access to reliable and adequate historical data on energy usage and meteorological information is severely limited. Emerging economies and geographically diverse locations experience gaps in available historical data, short time frames and inconsistent measurement standards which significantly limit the ability to develop and test reliable forecasting models. The availability of historical data limits model applicability and also increases the risk of providing inaccurate/biased/unstable forecasts.

Furthemore, the rapidly evolving structural composition of energy systems (e.g., renewable energy penetration, increased use of electric appliances, changes in consumer behavior) introduces non-stationarity and distributional shifts that further challenge the applicability of traditional forecasting frameworks. Finally, while machine learning and deep learning models may provide improved predictive performance relative to traditional models, their limited interpretability reduces their utility as decision-making tools for policymakers, system operators and investors.

1.3 Research Objectives and Questions

The aim of this thesis is to evaluate a neural network–based framework for short-term electricity demand forecasting. The primary objectives of the study are:

To compare multiple data-driven forecasting models for short-term electricity demand in Italy using high-frequency (15-minute) load data, incorporating population-weighted meteorological variables to account for regional climatic heterogeneity.

To assess the added value of probabilistic short-term load forecasting by applying Monte Carlo–based methods and prediction intervals, and to evaluate their relevance for operational risk assessment and short-term planning in power system applications.

To enhance the transparency and practical usability of machine learning–based forecasting systems through interpretability techniques (e.g., SHAP values and feature perturbation methods), in order to identify the most influential determinants of electricity demand and to provide actionable insights for grid operators, policymakers, and market participants.

Chapter 2

Literature Review and Theoretical Foundations

2.1 Traditional Energy Demand Forecasting Methods

At present, it is important not only to forecast future values but also to identify underlying patterns, relationships, and structural regularities within the available data [2]. While traditional forecasting methods are primarily designed to generate predictions, modern approaches aim to provide deeper insights into the mechanisms driving observed phenomena.

Traditional forecasting techniques are generally based on mathematical and statistical models, such as regression analysis, ARIMA models, exponential smoothing, and other econometric methods[1]. These approaches are effective when relationships between variables are stable and linear; however, they are limited in their ability to capture complex nonlinear dependencies and interactions among multiple factors.

In contrast, machine learning methods and neural networks enable not only improved forecasting accuracy but also the discovery of deeper and more complex patterns in large-scale and high-dimensional datasets[20]. This makes them particularly valuable in environments characterized by uncertainty, nonlinearity, and dynamic behavior.

2.1.1 Regression Methods

Regression models establish a functional relationship between electricity demand (the dependent variable) and a set of explanatory variables. They are widely used in energy forecasting because they allow the integration of external drivers such as weather conditions, calendar effects, and socio-economic factors[3].

2.1.1.1 Multiple Linear Regression (MLR)

Multiple Linear Regression (MLR) models electricity demand y_t as a linear combination of k predictors:

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_k x_{k,t} + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma^2), \tag{2.1}$$

where $x_{i,t}$ represent explanatory variables and β_i are regression coefficients. Typical predictors include temperature, humidity, binary indicators for holidays and weekends, and sine/cosine transformations of time variables to capture cyclical patterns.

MLR models offer high interpretability through estimated coefficients and enable straightforward statistical inference, including confidence intervals and hypothesis testing[2].

2.1.1.2 Generalized Additive Models (GAMs)

Generalized Additive Models (GAMs) extend linear regression by allowing nonlinear relationships between predictors and the dependent variable through smooth functions[3]:

$$y_t = \beta_0 + f_1(x_{1,t}) + f_2(x_{2,t}) + \cdots + f_k(x_{k,t}) + \varepsilon_t, \tag{2.2}$$

where $f_j(\cdot)$ are smooth functions, typically implemented using splines. GAMs are particularly effective in modeling nonlinear relationships between temperature and electricity demand, which often exhibit a U-shaped pattern due to combined heating and cooling effects.

GAMs capture nonlinearities while preserving interpretability through visualized smooth functions.

2.1.2 Time Series Analysis Models

Classical time-series models constitute the methodological foundation of forecasting and have been widely applied in economics, finance, and energy systems analysis[1]. These models aim to capture temporal dependencies in observed data through structured statistical representations, relying on assumptions of linearity, stationarity, and stochastic noise processes.

2.1.2.1 ARIMA Model

The Autoregressive Integrated Moving Average model, denoted as ARIMA(p, d, q), extends ARMA to non-stationary time series through differencing:

$$(1 - L)^d y_t = \alpha + \sum_{i=1}^{p} \phi_i (1 - L)^d y_{t-i} + \varepsilon_t + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j}, \tag{2.3}$$

where d is the order of differencing and L is the lag operator. ARIMA models are effective in capturing linear temporal dependencies [1].

2.1.2.2 SARIMA Model

The Seasonal ARIMA model, denoted as SARIMA$(p, d, q)\times$(P,D,Q)$_s$, incorporates seasonal dynamics:

$$(1 - L)^d (1 - L^s)^D y_t = \mu + \Phi(L)\Theta(L^s)\varepsilon_t, \tag{2.4}$$

where s denotes the seasonal period, and P, D, and Q represent seasonal autoregressive, differencing, and moving average orders. SARIMA models are particularly suitable for electricity demand data characterized by strong periodic patterns.

2.1.2.3 ARIMAX Model

The ARIMAX model generalizes ARIMA by incorporating exogenous explanatory variables:

$$(1 - L)^d y_t = \mu + \sum_{k=1}^{n} \beta_k x_{t,k} + \Theta(L)\varepsilon_t, \tag{2.5}$$

where $x_{t,k}$ denotes external regressors such as temperature, economic indicators, or calendar effects. ARIMAX models allow a more realistic representation of demand drivers.

2.1.2.4 SARIMAX Model

The SARIMAX model combines seasonal dynamics and exogenous variables:

$$(1 - L)^d (1 - L^s)^D y_t = \mu + \sum_{k=1}^{n} \beta_k x_{t,k} + \Phi(L)\Theta(L^s)\varepsilon_t. \tag{2.6}$$

SARIMAX models represent one of the most advanced extensions of classical time-series approaches and are widely applied in energy forecasting.

Despite their widespread adoption, ARIMA-based models exhibit several fundamental limitations motivating the adoption of machine learning models [20].They are inherently linear and therefore unable to capture complex nonlinear relationships and interactions among variables. Moreover, they rely on assumptions of stationarity and homoscedasticity, which are often violated in modern energy systems undergoing rapid technological and structural transformation. These limitations have motivated the increasing adoption of machine learning and neural network–based models, which offer greater flexibility and predictive power in complex environments.

2.1.3 Classical Machine Learning Techniques

2.1.3.1 Support Vector Regression (SVR)

Support Vector Regression (SVR) is a machine learning method derived from Support Vector Machines (SVM), designed to model nonlinear relationships between input variables and a continuous target variable. Unlike traditional regression methods, SVR [4] focuses on minimizing prediction error within a predefined tolerance margin while controlling model complexity.

SVR approximates the target function by constructing a regression hyperplane that maximizes the margin of tolerance, defined by the parameter ε. The optimization problem balances model accuracy and smoothness through a regularization parameter C, which controls the trade-off between model complexity and tolerance of prediction errors.

The SVR model can be expressed as:

$$f(x_t) = \mathbf{w}^\top \phi(x_t) + b, \tag{2.7}$$

where $\phi(\cdot)$ denotes a nonlinear mapping of input variables into a high-dimensional feature space, $\mathbf{w}$ is a weight vector, and b is a bias term. Kernel functions, such as the linear, polynomial, and radial basis function (RBF) kernels, are commonly used to capture nonlinear pattern

2.1.3.2 Random Forests (RF)

Random Forests (RF) is an ensemble machine [5] method based on decision trees, designed to improve predictive performance and reduce overfitting. The model constructs a large number of decision trees during training and aggregates their predictions, typically by averaging in regression tasks.

RF introduces randomness in two ways: bootstrap sampling of training data and random selection of features at each split. This combination increases model diversity and enhances generalization performance compared to single decision trees.

Formally, the Random Forest prediction can be expressed as:

$$\hat{y}_t = \frac{1}{B} \sum_{b=1}^{B} f_b(x_t), \tag{2.8}$$

where $f_b(\cdot)$ denotes the prediction of the b-th decision tree, and B is the total number of trees in the forest.

In electricity demand forecasting, Random Forests are effective in capturing nonlinear relationships and complex interactions between explanatory variables such as weather conditions, temporal factors, and socio-economic indicators.

Random Forests are robust to overfitting, capable of modeling nonlinear dependencies, and relatively insensitive to noise and outliers. They also provide measures of feature importance, which enhances model interpretability.

2.2 The Rise of Neural Networks in Time Series Forecasting

Neural networks have fundamentally transformed time-series forecasting by providing flexible, data-driven frameworks capable of modeling nonlinear dynamics, complex temporal dependencies, and high-dimensional interactions among variables[20]. Unlike traditional statistical models, which rely on explicit assumptions of linearity, stationarity, and parametric structure, neural networks learn representations directly from data, enabling them to adapt to evolving patterns and structural changes in modern energy systems. This makes them particularly suitable for electricity demand forecasting, where consumption

Table 2.1: Comparison of forecasting methods

Method	**Strengths**	**Limitations**
SARIMA	Statistical rigor, interpretable, captures seasonality	Linear assumptions, manual differencing, limited nonlinear modeling
MLR	Interpretable, integrates multiple explanatory variables, supports statistical inference	Linearity assumption, sensitivity to multicollinearity, error independence often violated
SVR	Captures nonlinear relationships via kernels, robust to outliers	Kernel selection and tuning, computational cost for large datasets
RF	Nonlinear pattern learning, handles interactions, feature importance estimation	Risk of overfitting, reduced transparency compared to linear models

patterns are influenced by heterogeneous factors such as weather conditions, economic activity, technological change, and behavioral dynamics.

2.2.1 Feedforward Neural Networks (FNNs)

Feedforward Neural Networks (FNNs), also known as multilayer perceptrons (MLPs), represent the most fundamental class of neural architectures. They consist of an input layer, one or more hidden layers, and an output layer, where information flows unidirectionally through nonlinear transformations.

In time-series forecasting, FNNs are typically applied by reformulating temporal data into a supervised learning framework, using lagged observations and exogenous variables as input features. Formally, an FNN approximates an unknown nonlinear function $f(\cdot)$ such that[20]:

$$y_t = f(y_{t-1}, y_{t-2}, \ldots, x_t) + \varepsilon_t, \tag{2.9}$$

where x_t denotes external predictors.

Although FNNs are capable of capturing nonlinear relationships between inputs and outputs, they do not explicitly model temporal dependencies due to the absence of internal memory mechanisms. Consequently, their effectiveness depends heavily on feature engineering and the selection of appropriate lag structures.

2.2.2 Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM)

Recurrent Neural Networks (RNNs) extend feedforward architectures by introducing recurrent connections that enable information to persist across time steps. This recursive structure allows RNNs to model sequential dependencies by maintaining a hidden state that evolves over time:

$$h_t = g(h_{t-1}, x_t), \tag{2.10}$$

where h_t denotes the hidden state and x_t the input at time t.

Despite their conceptual suitability for sequential data, standard RNNs suffer from vanishing and exploding gradient problems, which limit their ability to learn long-term dependencies [6, 19]. Long Short-Term Memory (LSTM) networks address these issues through gated mechanisms that regulate information flow within memory cells.

LSTM models have been widely adopted in electricity demand forecasting due to their ability to capture both short-term fluctuations and long-range temporal dependencies. Empirical evidence suggests that LSTM architectures outperform traditional statistical models in environments characterized by nonlinear dynamics and complex temporal structures.

2.2.3 Convolutional Neural Networks (CNNs) and Temporal Convolutional Networks (TCNs)

Convolutional Neural Networks (CNNs), originally developed for spatial data, have been successfully adapted to time-series forecasting by applying convolutional filters along the temporal dimension. CNNs extract local temporal patterns through convolutional kernels and hierarchical feature representations, enabling efficient modeling of short-term dependencies.

Temporal Convolutional Networks (TCNs) extend CNNs by employing causal and dilated convolutions, which allow the receptive field to grow exponentially with network depth while preserving temporal causality [7]. This architecture enables TCNs to capture long-range dependencies without the sequential processing constraints of RNNs.

Compared to recurrent architectures, CNNs and TCNs offer advantages in terms of parallel computation, training stability, and scalability, making them attractive for large-scale forecasting problems.

2.2.4 Attention Mechanisms and Transformer-Based Models

Attention mechanisms represent a paradigm shift in sequence modeling by allowing neural networks to dynamically focus on the most relevant parts of an input sequence. Instead of relying on recursive processing, attention-based models compute weighted interactions between all elements of a sequence, enabling direct modeling of long-range dependencies.

Transformer architectures, built entirely on self-attention mechanisms [8, 9], have recently emerged as powerful tools for time-series forecasting. By leveraging multi-head attention and positional encoding, Transformers can capture complex temporal relationships and interactions among multiple variables without recurrent structures.

In energy forecasting, Transformer-based models have shown superior performance in capturing multi-scale temporal patterns and cross-variable dependencies, particularly in high-dimensional and multivariate settings.

2.2.5 Hybrid and Ensemble Models

Hybrid and ensemble models combine multiple forecasting approaches to exploit their complementary strengths. Hybrid models typically integrate traditional statistical methods, such as ARIMA or SARIMA, with neural networks, allowing linear and nonlinear components of time series to be modeled separately. Ensemble models aggregate predictions from multiple machine learning or deep learning architectures, often through averaging, stacking, or weighted combinations.

In electricity demand forecasting, hybrid and ensemble approaches have demonstrated improved accuracy and robustness by mitigating model-specific weaknesses and reducing sensitivity to noise and structural changes[18]. Moreover, ensemble strategies can enhance model stability under uncertainty and improve generalization across different temporal regimes.

Table 2.2: Strengths and limitations of neural network models in time-series forecasting

Model	**Strengths**	**Limitations**
Feedforward Neural Networks (FNNs)	Flexible nonlinear approximation, simple architecture, low computational cost	Limited ability to model temporal dependencies, strong reliance on feature engineering
Recurrent Neural Networks (RNNs) / LSTM	Effective modeling of sequential data, ability to capture long-term dependencies	High computational complexity, difficult training, limited interpretability
CNNs / Temporal Convolutional Networks (TCNs)	Efficient extraction of local temporal patterns, parallel computation, stable training	Dependence on architectural design choices, limited transparency of learned features
Attention and Transformer-Based Models	Strong capability to model long-range dependencies, high predictive accuracy, suitability for multivariate data	High computational cost, large data requirements, complex architecture
Hybrid and Ensemble Models	Improved robustness and accuracy through model combination, reduced sensitivity to noise	Increased model complexity, reduced interpretability, higher implementation cost

2.3 Uncertainty Quantification in Machine Learning

In modern forecasting systems, predictive accuracy alone is insufficient for robust decision-making. In energy systems, forecasting errors can lead to significant economic losses, reliability issues, and suboptimal infrastructure planning. Therefore, quantifying uncertainty associated with model predictions has become a critical research direction in machine learning and time-series forecasting.

Uncertainty quantification (UQ) aims to characterize the reliability of predictions by estimating not only point forecasts but also the associated confidence intervals or predictive distributions. In the context of electricity demand forecasting, UQ enables system operators and policymakers to assess risks, optimize resource allocation, and improve resilience under uncertain conditions[18].

2.3.1 Epistemic and Aleatoric Uncertainty

Uncertainty in machine learning models is commonly decomposed into two fundamental types: epistemic and aleatoric uncertainty [12].

Epistemic uncertainty arises from limited knowledge about the underlying data-generating process and model parameters. It reflects uncertainty due to insufficient data, model misspecification, or structural limitations of the model. Epistemic uncertainty can, in principle, be reduced by collecting more data or improving model design.

Aleatoric uncertainty, in contrast, originates from inherent randomness in the data. It reflects irreducible variability caused by stochastic processes, measurement noise, or unobserved factors. Unlike epistemic uncertainty, aleatoric uncertainty cannot be eliminated, even with infinite data.

Distinguishing between these two types of uncertainty is particularly important in energy forecasting, where both data scarcity and intrinsic variability play a significant role.

2.3.2 Bayesian Neural Networks

Bayesian Neural Networks (BNNs) provide a principled probabilistic framework for uncertainty quantification [10] by treating model parameters as random variables rather than fixed values. In a Bayesian setting, the posterior distribution of model parameters is inferred given the observed data:

$$p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta)\, p(\theta), \tag{2.11}$$

where θ denotes model parameters and $\mathcal{D}$ the training dataset.

Predictions are obtained by marginalizing over the posterior distribution:

$$p(y_t \mid x_t, \mathcal{D}) = \int p(y_t \mid x_t, \theta)\, p(\theta \mid \mathcal{D})\, d\theta. \tag{2.12}$$

Although BNNs offer a theoretically rigorous approach to uncertainty estimation, exact Bayesian inference is computationally intractable for large neural networks. As a result, approximate inference methods, such as variational inference and Monte Carlo sampling, are commonly employed.

2.3.3 Monte Carlo Dropout and Ensemble Methods

Monte Carlo Dropout (MC Dropout) represents an efficient approximation to Bayesian inference in deep neural networks, originally proposed by Gal and Ghahramani (2016) and further discussed in modern uncertainty quantification literature. Under this framework, conventional dropout regularization [11] is interpreted as a variational approximation to a deep Gaussian process.

Unlike standard training procedures, where dropout is disabled during inference, MC Dropout retains stochastic dropout masks at prediction time. As a result, repeated forward passes through the network correspond to sampling from an implicit posterior distribution over model parameters. This interpretation enables uncertainty estimation without explicitly specifying prior distributions or performing computationally expensive posterior sampling.

From a probabilistic perspective, each stochastic forward pass reflects a different realization of the model weights, induced by randomly deactivated neurons. The resulting distribution of predictions therefore captures uncertainty associated with limited data, model misspecification, and parameter estimation. In practice, this uncertainty is primarily epistemic in nature, although it may also incorporate residual noise effects.

A key advantage of MC Dropout lies in its simplicity and scalability. Since it relies on standard dropout layers already present in most deep learning architectures, it can be integrated into existing models with minimal modification. This property makes MC Dropout particularly attractive for large-scale time-series forecasting applications, where fully Bayesian neural networks are often computationally prohibitive.

In the context of short-term electricity demand forecasting, MC Dropout provides a practical mechanism for quantifying confidence in neural network predictions. By identifying periods associated with elevated predictive uncertainty, system operators can better assess forecast reliability and incorporate risk-aware decision-making strategies.

Nevertheless, MC Dropout remains an approximation and does not represent the true posterior distribution. Its performance depends on architectural choices, dropout placement, and sampling size. Consequently, careful empirical validation is required to ensure adequate calibration of the resulting prediction intervals.

2.3.4 Quantile Regression and Conformal Prediction

Quantile regression provides a distribution-free approach to uncertainty estimation by directly modeling conditional quantiles of the target variable. For a given quantile level $\tau \in (0, 1)$, the model estimates the corresponding conditional quantile $Q_\tau(y_t \mid x_t)$.

Conformal prediction extends this idea by constructing prediction intervals with guaranteed coverage probabilities under mild assumptions [13]. Unlike Bayesian methods, conformal prediction does not require explicit probabilistic modeling and can be applied to a wide range of machine learning algorithms.

These methods are particularly attractive in energy forecasting, where reliable prediction intervals are often more informative than point estimates.

2.4 Interpretability and Explainable Artificial Intelligence

The increasing adoption of complex machine learning and deep learning models in energy systems has raised concerns regarding their transparency and trustworthiness [16]. While neural networks often achieve superior predictive performance, their black-box nature makes it difficult to understand how predictions are generated.

Interpretability and Explainable Artificial Intelligence (XAI) aim to bridge this gap by providing tools and methodologies to explain model behavior, identify key drivers of predictions, and enhance trust in automated decision-making systems. In the context of electricity demand forecasting, interpretability is crucial for regulatory compliance, policy analysis, and operational decision-making.

2.4.1 Post-hoc and Intrinsic Interpretability

Interpretability methods can be broadly categorized into intrinsic and post-hoc approaches.

Intrinsic interpretability refers to models that are inherently transparent [16], such as linear regression, decision trees, and generalized additive models. These models offer direct insights into the relationship between inputs and outputs but often lack the expressive power required to model complex nonlinear systems.

Post-hoc interpretability, in contrast, focuses on explaining the predictions of complex models after training. Post-hoc methods can be applied to any black-box model, including deep neural networks, and are therefore widely used in modern machine learning applications.

2.4.2 SHAP and LIME

SHapley Additive exPlanations (SHAP) is a game-theoretic framework for attributing model predictions to individual input features. SHAP values quantify the marginal contribution of each feature to a prediction by approximating Shapley values from cooperative game theory.

LIME (Local Interpretable Model-agnostic Explanations) provides local explanations by approximating a complex model with a simpler interpretable model in the neighborhood of a specific prediction.

Both SHAP and LIME have been extensively applied in energy forecasting to identify the relative importance of weather variables, temporal features, and socio-economic factors in driving electricity demand [14, 15].

2.4.3 Saliency Maps and Attention-Based Interpretability

Saliency maps are gradient-based techniques that highlight input features most influential for model predictions. These methods are particularly relevant for neural networks, where gradients can be used to quantify the sensitivity of predictions to changes in input variables.

Attention mechanisms provide an additional layer of interpretability by explicitly modeling the relevance of different time steps or features. In Transformer-based models, attention weights can be visualized to reveal temporal dependencies and cross-variable interactions [16].

Although saliency and attention-based methods offer valuable insights into model behavior, their interpretations must be treated with caution, as they do not always provide causal explanations.

2.5 Research Gap

Although many papers exist about the subject of electricity demand forecasting, there are significant deficiencies in the current state-of-the-art with respect to the relationship of forecasting error, uncertainty (quantifying) and model interpretability.

One important deficiency is that most prior research was focused almost exclusively on improving the accuracy of point predictions by developing new statistical models, machine learning models, or deep neural network models. Although these prior research efforts have significantly improved the accuracy of point predictions, they usually do not address uncertainty quantification in a systematic way. However, uncertainty quantification is critical for developing reliable decision-making processes for risk management, reserve planning, and maintaining stability in operational energy systems; consequently, models which only produce point forecasts will be less useful for practical use in operational environments.

A second major deficiency is that, although many researchers are starting to develop probabilistic forecasting models for estimating uncertainty, few prior research efforts examine uncertainty quantification in conjunction with model interpretability[11, 13]. Therefore, Bayesian neural networks, Monte Carlo Dropout, and conformal prediction methods are typically evaluated in terms of coverage and/or interval width, rather than examining how uncertainty relates to the underlying factors which drive electricity demand. Consequently, system operators/analysts are generally unable to gain insight into what causes uncertainty to increase during particular temporal regimes or when exposed to certain environmental conditions.

A third important deficiency in the literature is that interpretability studies within the area of load forecasting are generally limited to a single modeling paradigm, which is typically tree-based models with SHAP explanations [14]. Deep learning models, such as LSTM and TCN architectures, are infrequently examined through the lens of comprehensive temporal and feature-level interpretability assessments. Thus, it is difficult to understand how different architectures utilize temporal, autoregressive, and exogenous information.

Lastly, most empirical studies depend upon fixed feature representations, and limited ablation studies. Therefore, the interaction between feature representation, model design, and uncertainty behavior is insufficiently understood. In particular, little attention has been given to how rolling statistics, lag structures, and meteorological interactions affect both predictive reliability and model explanations.

As a response to the above stated limitations, this paper presents a unified approach to examine forecasting accuracy, uncertainty estimation, and interpretability for multiple model types, including XGBoost, LSTM, and TCN. By integrating systematic benchmarking, uncertainty assessment, and multi-level interpretability analysis, this study intends to provide a deeper and more practical understanding of short-term electricity demand forecasting.

Chapter 3

Methodology and Experimental Design

3.1 Overall Research Framework

To ensure methodological rigour, reproducibility, and analytical transparency, this study adopts a structured and modular workflow for short-term electricity demand forecasting (Figure 3.1). The framework integrates data engineering, comparative model implementa-

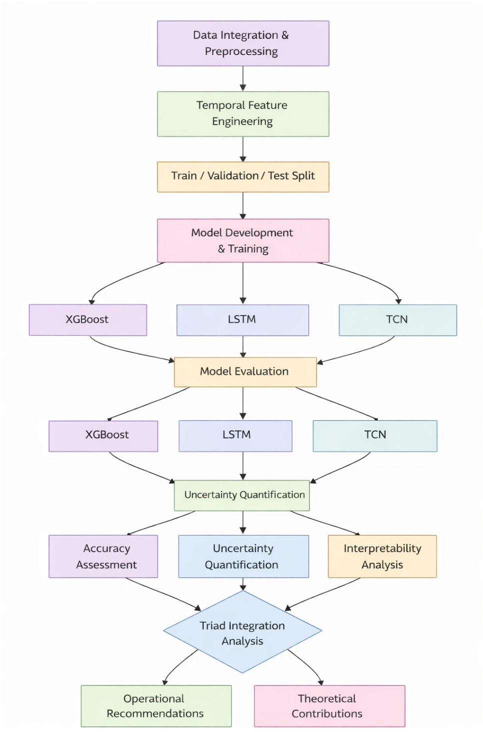

Figure 3.1: Research Framework

tion, uncertainty quantification, and interpretability analysis within a unified evaluation pipeline.

The analysis begins with the collection of heterogeneous data sources, including national electricity load data and high-resolution meteorological observations. These datasets are merged and preprocessed through systematic data cleaning, temporal alignment, outlier treatment, and normalization procedures to ensure internal consistency and reliability.

Subsequently, temporal feature engineering is performed to construct calendar-based, cyclic, lagged, and rolling statistical variables. These features enable the models to capture seasonality, autocorrelation structures, and short-term volatility inherent in electricity demand dynamics.

The processed dataset is partitioned into training, validation, and test subsets using a strict chronological split to replicate real-world forecasting conditions and prevent information leakage.

Following dataset partitioning, three established forecasting paradigms are implemented and comparatively evaluated:

- XGBoost, representing tree-based ensemble learning,
- Long Short-Term Memory (LSTM) networks, representing recurrent deep learning,
- Temporal Convolutional Networks (TCN), representing convolutional sequence modeling.

Each model is trained on the training subset and tuned using the validation subset through systematic hyperparameter optimization. The objective is not architectural innovation, but a controlled empirical comparison under a unified experimental framework.

Model performance is evaluated across three complementary analytical dimensions:

Predictive accuracy, assessed using standardized error metrics;

Predictive uncertainty, quantified through Monte Carlo dropout and conformal prediction methods to construct probabilistic prediction intervals and evaluate forecast reliability;

Model interpretability, analysed using SHAP values, permutation importance, and feature occlusion techniques to identify the primary determinants of electricity demand.

The results obtained from these three dimensions are integrated into a unified analytical framework designed to investigate trade-offs between predictive performance, forecast uncertainty, and model transparency.

Finally, cross-model comparison is conducted to identify structural strengths, limitations, and domain suitability of each forecasting paradigm. The findings are translated into practical recommendations for power system applications and into broader methodological insights concerning trustworthy artificial intelligence in energy systems.

Overall, the proposed methodology establishes a reproducible and comprehensive empirical pipeline linking data processing, machine learning implementation, uncertainty modelling, and explainable artificial intelligence within a coherent analytical structure.

3.2 Data Description and Sources

This study employs a multivariate time-series dataset with 15-minute temporal resolution. The dataset was constructed by integrating electricity load data provided by Terna [21] with hourly meteorological observations retrieved from the Open-Meteo historical forecast API [22]. The merged dataset simultaneously captures the internal temporal dynamics of electricity demand and the external climatic drivers influencing it at an operationally relevant scale for modern power system management.

Electricity load values were preserved at their native 15-minute resolution to retain intra-hour load dynamics, which are essential for grid balancing, frequency regulation, and short-term operational planning. Let $\text{Load}_t^{(15\text{min})}$ denote the electricity demand measured at quarter-hour interval t, where $t = 1, 2, \ldots, T$ indexes consecutive 15-minute periods. The high temporal resolution enables the identification of rapid demand fluctuations associated with industrial activity, commercial operations, and short-term weather variations, which may be obscured under hourly aggregation.

All timestamps were converted to the `Europe/Rome` time zone, and daylight saving time transitions were explicitly handled to ensure temporal consistency throughout the observation period.

Meteorological data were collected for all twenty Italian administrative regions using official geospatial coordinates [23] . To ensure completeness and reliability over the multi-annual study horizon, data were retrieved in weekly batches using an automated procedure with retry logic and error handling.

For each region r, meteorological variables were extracted and synchronised to a 15-minute resolution. These include temperature, relative humidity, apparent temperature, precipitation, solar irradiance, wind speed, and a day/night indicator, denoted as $X_{r,t}^{(15\text{min})}$.

To account for spatial heterogeneity and regional differences in electricity consumption patterns, a population-weighted aggregation scheme was applied. Let pop_r denote the population of region r, and R the total number of regions. The weight assigned to region r is defined as:

$$w_r = \frac{\text{pop}_r}{\sum_{r=1}^{R} \text{pop}_r} \tag{3.1}$$

National-level meteorological indicators were computed as weighted averages:

$$X_{\text{Italy},t}^{(15\text{min})} = \sum_{r=1}^{R} w_r \cdot X_{r,t}^{(15\text{min})} \tag{3.2}$$

This aggregation procedure ensures that regions with larger populations (and therefore higher expected electricity demand) exert proportionally greater influence on national-level climatic indicators.

Continuous meteorological variables were aggregated using weighted averages, while binary indicators (e.g. day/night status) were thresholded after aggregation. Categorical weather condition codes were combined using a population-weighted modal approach.

Following regional aggregation, all meteorological series were resampled to a uniform 15-minute frequency and temporally aligned with the load data. Short temporal gaps were addressed using constrained interpolation to maintain local continuity while avoiding long-range extrapolation artefacts.

The final dataset comprises population-weighted meteorological indicators, electricity load observations, baseline forecasts, engineered temporal features, autoregressive lag variables, and rolling statistical measures. The resulting panel consists of $T = 114{,}290$ observations at 15-minute resolution and $N = 29$ explanatory variables and targets.

The resulting high-dimensional, high-frequency time series provides a comprehensive empirical framework for evaluating advanced forecasting techniques. By preserving native demand resolution and incorporating spatially representative meteorological indicators, the dataset enables rigorous assessment of predictive accuracy, uncertainty quantification, and model interpretability at a scale directly relevant to real-time grid operations and renewable energy integration.

3.3 Data Preprocessing and Feature Engineering

3.3.1 Missing Values and Normalization

A substantial proportion of zero values is observed across several meteorological and temporal variables. Importantly, these zero values do not correspond to missing observations or measurement errors. Instead, they arise from inherent physical, environmental, and behavioral mechanisms underlying the processes being modelled.

Precipitation exhibits a high frequency of zero observations, with approximately 52.49% of values equal to zero. This pattern reflects the intermittent nature of rainfall, particularly at a 15-minute temporal resolution, where precipitation events occur relatively infrequently. At high frequency sampling intervals, most time steps are characterised by the absence of rainfall, resulting in structured sparsity. Such sparsity is a well-documented characteristic of high-frequency meteorological time series and is consistent with findings reported in the short-term electricity demand forecasting literature.

A comparable proportion of zero values is observed for shortwave radiation (46.91%), corresponding to nighttime periods when solar irradiance is absent. This behaviour reflects the deterministic diurnal solar cycle. Consequently, the interaction variable `radiation_temp`, defined as the product of shortwave radiation and air temperature, exhibits the same proportion of zero values, as it collapses to zero whenever solar radiation equals zero.

The binary indicator `is_day` alternates deterministically between 0 and 1, with 0 representing nighttime. Approximately 49.52% of total observations correspond to nighttime periods, which is consistent with the expected alternation between day and night over the observation horizon.

Several calendar-based variables also display a high proportion of zero values due to their deterministic construction. The variable `weekend` equals 0 on weekdays, resulting in 71.40% zero values. Similarly, `dayofweek` and its cyclical transformation `dow_sin` equal zero at the start of the weekly cycle, accounting for 14.28% of observations. The variable `hour` and its cyclical representation `hour_sin` equal zero at midnight, which occurs once per day and therefore represents 4.17% of total observations.

Overall, the observed zero-value structure reflects normal environmental cycles, temporal periodicity, and behavioural regularities rather than data quality deficiencies. These structural patterns contain meaningful information about electricity demand dynamics and were therefore retained in the final dataset.

3.3.2 Feature Standardization

Prior to model training, all continuous input variables were standardised using z-score normalisation:

$$z = \frac{x - \mu}{\sigma} \tag{3.3}$$

where μ denotes the empirical mean of feature x, and σ its empirical standard deviation. Both statistics were computed exclusively on the training set. The same scaling parameters were subsequently applied to the validation and test sets to prevent information leakage.

Feature standardisation improves numerical stability during optimisation and enhances convergence behaviour, particularly for neural network architectures. Although tree-based models are generally invariant to monotonic feature scaling, the same standardised inputs were provided to all model classes to ensure experimental consistency across forecasting paradigms.

3.3.3 Temporal Features and Lag Selection

To capture temporal regularities and autocorrelation structures in electricity demand, a set of calendar-based, cyclic, lagged, and rolling statistical features was constructed from the original time index and historical load observations.

First, basic calendar variables were extracted from the timestamp index, including hour of day, day of week, month, and day of month. A binary indicator was also introduced to distinguish between weekdays and weekends. These variables allow the models to capture systematic variations in demand associated with daily, weekly, and monthly consumption patterns.

Since several temporal variables exhibit cyclic behavior, cyclic encoding was applied to represent periodic structures in a continuous and differentiable form. Specifically, sine and cosine transformations were used for both hour of day and day of week, defined as

$$\text{hour\_sin} = \sin\left(\frac{2\pi \cdot \text{hour}}{24}\right), \quad \text{hour\_cos} = \cos\left(\frac{2\pi \cdot \text{hour}}{24}\right), \tag{3.4}$$

$$\text{dow\_sin} = \sin\left(\frac{2\pi \cdot \text{dayofweek}}{7}\right), \quad \text{dow\_cos} = \cos\left(\frac{2\pi \cdot \text{dayofweek}}{7}\right). \tag{3.5}$$

This encoding preserves the circular nature of time variables and avoids artificial discontinuities between adjacent periods, such as between 23:00 and 00:00.

To model short-term temporal dependencies in electricity demand, multiple lagged versions of the target variable were generated. Given the 15-minute sampling frequency, lags of 1, 4, 96, and 672 observations were selected, corresponding respectively to 15 minutes, 1 hour, 24 hours, and 7 days. These lag features enable the models to exploit both immediate autocorrelation effects and recurring daily and weekly patterns.

In addition to discrete lags, rolling statistical features were computed to summarize recent demand dynamics over different temporal horizons. Moving averages and standard deviations were calculated using rolling windows of 4, 96, and 672 observations, representing approximately 1 hour, 24 hours, and 7 days, respectively. These features capture local trend behavior and short-term volatility in electricity consumption.

Finally, interaction variables between meteorological conditions were introduced to model nonlinear effects. In particular, multiplicative combinations of temperature with humidity, wind speed, and solar radiation were constructed. These interaction terms allow the models to represent compound weather influences on electricity demand, such as thermal discomfort and solar-related cooling effects.

The resulting feature set provides a comprehensive representation of temporal structure, persistence, and exogenous influences in the electricity demand time series.

3.3.4 Train-Validation-Test Split

To replicate real-world forecasting conditions and prevent information leakage, the full dataset was partitioned into three mutually exclusive subsets using a strict chronological split. The earliest 70% of observations were assigned to the training set, while the remaining 30% were divided equally between validation (15%) and test (15%) sets.

The training set was used to estimate model parameters for each candidate forecasting paradigm. During this phase, model weights were optimised exclusively on historical observations contained within the training period.

The validation set was subsequently employed to assess out-of-sample predictive performance and to guide hyperparameter tuning. This procedure ensures that model selection is based on performance evaluated on temporally subsequent data, thereby approximating operational forecasting settings in which models are trained on past observations and deployed on future time periods.

Regularisation techniques and early stopping criteria were applied during training to mitigate overfitting. Although early stopping monitors validation performance, model parameters were optimised solely with respect to the training data, ensuring that the validation set retained its role as an unbiased reference for model comparison and hyperparameter selection.

Once the best-performing configuration was identified based on validation metrics, the corresponding trained model was fixed without further retraining. Its generalisation performance was then evaluated on the independent test set, which remained entirely unseen during both training and validation phases.

The test set therefore provides an unbiased estimate of predictive performance under realistic forecasting conditions and serves as the final benchmark for cross-model comparison.

3.4 Model Selection

To ensure a comprehensive evaluation of different modeling paradigms for short-term electricity demand forecasting, three representative machine learning and deep learning architectures were selected: Extreme Gradient Boosting (XGBoost), Long Short-Term Memory networks (LSTM), and Temporal Convolutional Networks (TCN). These models were chosen to reflect complementary methodological approaches to time-series modeling, encompassing tree-based ensembles, recurrent neural networks, and convolutional sequence architectures.

3.4.1 XGBoost

The extreme gradient boost (XGBoost) method is a scalable and regularized form of gradient boosting decision trees that can be used to model complex nonlinear relationships in tabular data with structure [17]. Its wide adoption within energy forecasting is primarily due to its strong predictive capability, robustness to multicollinearity among predictor variables, and ability to leverage the richness of large numbers of engineered features.

The use of XGBoost in this study is based on three primary attributes. First, XGBoost provides a means to utilize the manual construction of features such as lagged demand values, rolling statistics, and cyclic calendar variables. Second, its built-in regularization methods (e.g., shrinkage, maximum tree depth limits, column sampling) assist in minimizing overfitting in high-dimensional datasets. Finally, XGBoost also provides a mature set of interpretive tools (i.e., feature importance measures and SHAP values [14]), enabling transparent analysis of model behavior.

Therefore, XGBoost represents a strong and interpretable baseline of current state-of-the-art machine learning techniques for modeling structured time-series data.

3.4.2 Long Short-Term Memory Networks (LSTM)

Long short-term memory (LSTM) networks represent a subset of recurrent neural networks (RNNs), specifically designed to capture long range temporal dependencies using gated memory architectures [6]. LSTMs avoid the vanishing gradient problem in traditional RNNs by selectively retaining or forgetting information over time. In this context, the LSTM has been shown to be an effective model for predicting electricity loads [24].

The reason for including LSTMs was that they could effectively utilize sequential data (the time series of electricity loads) with minimal need for extensive manual feature engineering. Through the use of sliding windows of historical load values, LSTMs can extract the underlying temporal relationships, non-linear dependencies and seasonal patterns from the time series data.

Moreover, LSTMs provide a natural foundation for uncertainty quantification via approximate Bayesian inference techniques such as Monte Carlo Dropout [11], allowing estimation of epistemic uncertainty. This capability is particularly valuable in operational forecasting contexts where risk awareness is critical.

3.4.3 Temporal Convolutional Networks (TCN)

Temporal Convolutional Networks (TCNs) are an example of convolutional architectures that are particularly developed for sequence modeling [7]. The architecture uses dilations and causal convolutions to enforce temporal order while allowing for very large receptive field sizes which can capture dependencies over long periods of time.

Unlike RNNs, TCNs have some additional benefits. TCNs enable parallel processing of all time steps simultaneously during training, thus improving both training speed and training scaleability. Additionally, using hierarchical dilation in TCN architectures enables the modeling of multi-scale temporal features, making TCNs suitable for the high frequency load forecasting applications [25].

For the purposes of this research, TCNs were chosen as a relatively new deep learning architecture compared to traditional recurrent architectures. TCNs provide a balance between expressiveness and computational efficiency.

3.5 Accuracy Metrics

Each model was assessed using a set of complementary performance metrics designed to capture different aspects of forecasting accuracy for the target variable *Total Load (GW).*

Mean Absolute Error (MAE) measures the average magnitude of prediction errors in the original unit of measurement and is defined as

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \tag{3.6}$$

where y_i and $\hat{y}_i$ denote the observed and predicted values, respectively.

Root Mean Squared Error (RMSE) penalizes larger deviations more strongly and is given by

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}. \tag{3.7}$$

Symmetric Mean Absolute Percentage Error (sMAPE) provides a scale-independent measure of relative forecasting accuracy and is defined as

$$\mathrm{sMAPE} = \frac{100}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2}. \tag{3.8}$$

The coefficient of determination (R^2) evaluates the proportion of variance in the observed data explained by the model and is expressed as

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}, \tag{3.9}$$

where $\bar{y}$ represents the mean of the observed values.

Together, these metrics provide a comprehensive evaluation framework. MAE and RMSE quantify absolute prediction errors in physical units, sMAPE enables relative comparison across time periods, and R^2 reflects the overall explanatory power of the forecasting models.

3.6 Uncertainty Quantification Methods

3.6.1 Epistemic vs. Aleatoric Uncertainty

Accurate prediction in machine learning involves not only point forecasting but also the quantification of predictive uncertainty. In energy forecasting and other time-series applications, understanding the sources and magnitudes of uncertainty is essential for reliable model selection and operational decision-making. Two fundamental types of uncertainty are commonly distinguished in the literature: epistemic and aleatoric uncertainty (Kendall & Gal, 2017; see also *Uncertainty in Machine Learning: A Modern Perspective*, Springer, 2017).

Aleatoric uncertainty refers to the inherent noise present in the data-generating process. It arises from stochastic environmental influences, measurement errors, and intrinsic variability in human behavior. Since this uncertainty reflects irreducible randomness, it cannot be eliminated by increasing the amount of data, but must instead be properly modeled within predictive distributions.

In contrast, epistemic uncertainty captures uncertainty associated with the model itself. It originates from limited training data, imperfect model assumptions, and parameter estimation errors. Unlike aleatoric uncertainty, epistemic uncertainty can be reduced through the acquisition of additional data, improved model architectures, or more expressive learning paradigms. In sparsely observed regions of the input space, epistemic uncertainty tends to be higher, reflecting limited model experience, while it decreases in well-sampled regimes.

Formally, let y denote the target variable and x the corresponding input features. A probabilistic forecasting model defines a predictive distribution conditioned on the training dataset D as

$$p(y \mid x, D) = \int p(y \mid x, \theta)\, p(\theta \mid D)\, d\theta,$$

where θ represents the model parameters and $p(\theta \mid D)$ denotes their posterior distribution.

Within this Bayesian framework, predictive uncertainty can be decomposed using the law of total variance:

$$\mathrm{Var}(y \mid x, D) = E_{p(\theta|D)}[\mathrm{Var}(y \mid x, \theta)] + \mathrm{Var}_{p(\theta|D)}(E[y \mid x, \theta])\,.$$

The first term corresponds to aleatoric uncertainty and represents the expected conditional variance under fixed model parameters. The second term captures epistemic uncertainty and reflects variability in predictions arising from uncertainty in the model parameters.

In practice, exact Bayesian inference is generally intractable for deep learning models. Therefore, approximate inference techniques are employed. Methods such as Bayesian neural networks, Monte Carlo Dropout, and deep ensembles approximate the posterior distribution by generating multiple stochastic forward passes and computing sample-based estimates of predictive uncertainty (Gal & Ghahramani, 2016). Conversely, heteroscedastic and quantile regression models explicitly parameterize aleatoric noise as a function of the input features.

In short-term load forecasting, both uncertainty components play distinct roles. Aleatoric uncertainty reflects irreducible demand variability driven by weather and behavioral factors, while epistemic uncertainty highlights regions characterized by limited historical precedent or structural model limitations. Proper estimation and decomposition of these uncertainty sources enables more robust and risk-aware forecasting, which is essential for modern power system operation.

3.6.2 Monte Carlo Dropout

Monte Carlo Dropout (MC Dropout) is an approximate Bayesian inference technique that enables uncertainty estimation in deep neural networks with minimal changes to the standard training procedure. Initially proposed by Gal and Ghahramani (2016) and further discussed in the context of modern uncertainty quantification (see *Uncertainty in Machine Learning: A Modern Perspective*, Springer, 2017), MC Dropout interprets the conventional dropout regularization used during training as a variational approximation to a deep Gaussian process.

In standard neural network training, dropout randomly zeroes out units with a fixed probability at each training iteration to prevent overfitting. In MC Dropout, the key idea is to retain dropout at inference time, thereby generating a distribution of predictions for the same input by performing multiple stochastic forward passes through the network. Specifically, given an input x, the network produces a set of predictions $\{\hat{y}_t(x)\}_{t=1}^{T}$ by evaluating the model T times with dropout enabled. These stochastic predictions can be used to estimate both the predictive mean and the uncertainty:

$$E[y \mid x, \mathcal{D}] \approx \frac{1}{T}\sum_{t=1}^{T} \hat{y}_t(x), \tag{3.10}$$

$$\mathrm{Var}(y \mid x, \mathcal{D}) \approx \frac{1}{T}\sum_{t=1}^{T} \hat{y}_t(x)^2 \;-\; \left(\frac{1}{T}\sum_{t=1}^{T} \hat{y}_t(x)\right)^2, \tag{3.11}$$

where $\mathcal{D}$ represents the training data and T is the number of stochastic forward passes. Under this interpretation, each forward pass corresponds to sampling from an implicit posterior distribution over the network's parameters induced by dropout.

The variance estimated from multiple passes combines both epistemic uncertainty — due to uncertainty in network weights — and residual noise. When dropout is applied to all layers of the network, MC Dropout can approximate model uncertainty without requiring explicit Bayesian inference or costly sampling of full posterior distributions. This method is computationally efficient and easily integrated into existing neural network workflows, making it particularly useful for uncertainty estimation in time-series forecasting applications such as short-term load prediction.

3.7 Interpretability Methods

In addition to predictive accuracy and uncertainty quantification, model interpretability plays a crucial role in understanding and validating forecasting models, particularly in high-stakes applications such as electricity demand forecasting. Interpretability methods aim to explain how input features contribute to model predictions, thereby improving transparency, trust, and diagnostic insight.

3.7.1 Global and Local Interpretability

Interpretability techniques are commonly categorized into global and local methods. Global interpretability focuses on understanding the overall behavior of a model across the entire dataset, identifying which features are generally most influential. Local interpretability, in contrast, explains individual predictions by quantifying the contribution of each feature to a specific forecast.

Tree-based ensemble models, such as Gradient Boosting and XGBoost, are particularly well-suited for post-hoc interpretability analysis due to their structured decision-making process and compatibility with additive explanation frameworks.

3.7.2 SHAP Values

To interpret the predictions of tree-based models, SHapley Additive exPlanations (SHAP) were employed. SHAP is a unified framework grounded in cooperative game theory, where each feature is treated as a "player" contributing to the prediction. The contribution of a feature is quantified using Shapley values, which represent the average marginal contribution of that feature across all possible subsets of features.

Formally, for a model f and an input instance x, the prediction can be decomposed as

$$f(x) = \phi_0 + \sum_{i=1}^{M} \phi_i, \tag{3.12}$$

where ϕ_0 is the expected model output over the training data, ϕ_i is the SHAP value associated with feature i, and M denotes the total number of input features. Each SHAP value represents the contribution of a feature to the deviation of the prediction from the baseline expectation.

SHAP values satisfy several desirable theoretical properties, including local accuracy, consistency, and missingness, which ensure faithful and stable explanations. For tree-based models, efficient algorithms such as TreeSHAP allow exact computation of Shapley values with polynomial complexity.

3.7.3 Interpretation of SHAP Results

Global SHAP importance can be evaluated through summary plots, which show how each feature is ranked based on the average SHAP (SHapley Additive exPlanations) values across all observations. The summary plots will illustrate the strength of each variable's influence on model output. Typically in the context of electricity demand forecasting, lagged load variables and baseline forecast variables will be at the top of the importance list due to strong autocorrelation over time. Temperature and solar radiation related features will have some nonlinear and seasonally dependent influences.

Additionally, dependence plots were employed to illustrate how changes to a particular feature affects model predictions and to capture the interaction of each feature with other features. The dependence plots will demonstrate that there are both asymmetric and nonlinear relationships present among the feature inputs, and that these relationships cannot be fully represented by linear importance scores.

3.7.4 Interpretability in Deep Learning Models

For deep learning models such as LSTM and Temporal Convolutional Networks (TCN), intrinsic interpretability is more limited due to their distributed representations and nonlinear dynamics. As a result, model-agnostic techniques, including perturbation-based feature importance and uncertainty-aware analyses, were considered to complement SHAP-based explanations obtained from tree-based models.

Overall, the combination of SHAP-based interpretability for gradient boosting models and uncertainty-aware analysis for neural networks provides a comprehensive understanding of model behavior, highlighting both dominant predictors and regions of increased uncertainty. This multi-perspective interpretability framework enhances the credibility and practical applicability of the proposed forecasting models.

3.8 Software and Hardware Environment

All experiments were implemented in Python (version 3.x) within the Google Colab environment. The computational pipeline integrates data preprocessing, model training, uncertainty estimation, and interpretability analysis into a unified and reproducible workflow.

Data manipulation and feature engineering were conducted using `NumPy` and `pandas`. Model development relied on the following libraries:

- `XGBoost` for gradient boosting models,
- `TensorFlow/Keras` for LSTM and TCN neural network architectures,
- `scikit-learn` for dataset splitting and evaluation metrics.

Uncertainty quantification via Monte Carlo Dropout was implemented within the `TensorFlow/Keras` framework by enabling dropout layers during inference. Model interpretability for tree-based models was conducted using the `SHAP` library, while permutation-based and occlusion-based feature importance methods were applied to neural network models.

Training was performed using GPU acceleration (NVIDIA Tesla T4) provided by Google Colab. To enhance reproducibility, fixed random seeds were set for `NumPy`, `TensorFlow`, and Python's built-in random module. Chronological data splitting was strictly enforced to prevent information leakage from future observations.

Although deterministic settings were applied where possible, minor variations in neural network results may occur due to stochastic optimization procedures and hardware-level non-determinism associated with GPU computation.

Chapter 4

Empirical Analysis I – Forecasting Accuracy

4.1 Benchmarking Experiments

This section presents a comparative evaluation of the considered forecasting models, namely the Terna baseline, LSTM, XGBoost, and TCN, using a common test dataset. The results are summarized in Table 4.1, which reports the main accuracy metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), symmetric Mean Absolute Percentage Error (sMAPE), and the coefficient of determination (R^2).

Model	MAE (GW)	RMSE	sMAPE	R^2
Terna (baseline)	0.592	0.768	1.72%	0.991
LSTM	0.26	0.34	0.75%	0.998
XGB	0.148	0.19	0.62%	0.999
TCN	0.209	0.27	0.99%	0.9988

Table 4.1: Model Performance Comparision

The Terna operational forecast serves as an external benchmark, representing the performance of a real-world industrial system. Although it achieves a high R^2 value, its error-based metrics remain substantially higher than those of the proposed machine learning models, with an MAE of 0.592 GW and an RMSE of 0.768 GW. This highlights the potential of data-driven approaches to improve short-term load forecasting accuracy.

Among the learning-based models, the LSTM network significantly improves upon the baseline, reducing the MAE to 0.260 GW and the sMAPE to 0.75%. These results confirm the ability of recurrent architectures to capture long-term temporal dependencies in electricity demand.

The XGBoost model achieves the best overall performance, attaining the lowest MAE (0.148 GW) and RMSE (approximately 0.19 GW), together with the smallest sMAPE value (0.62%). Its strong performance can be attributed to the effective exploitation of engineered temporal and autoregressive features, as well as to the robustness of gradient boosting methods in handling nonlinear relationships.

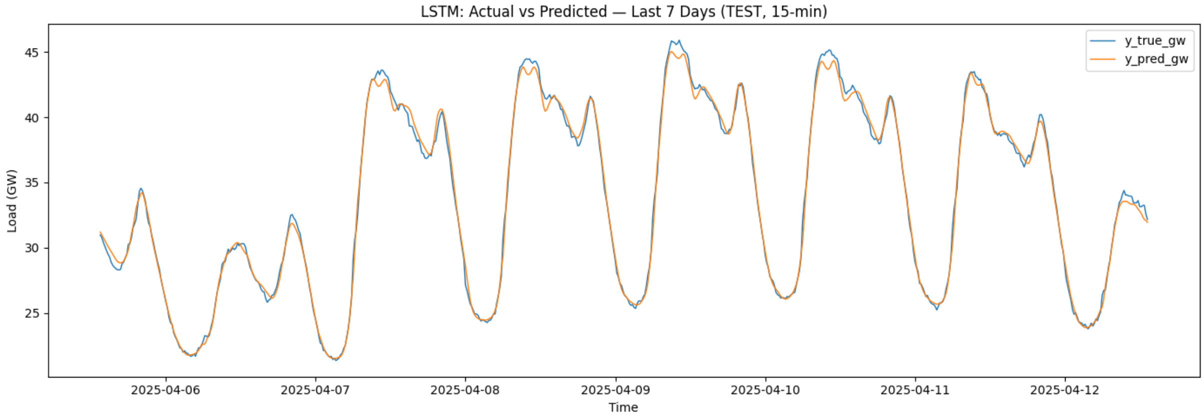

Figure 4.1: LSTM Actual vs Predicted Load — Last 7 Days (Test Set, 15-minute resolution)

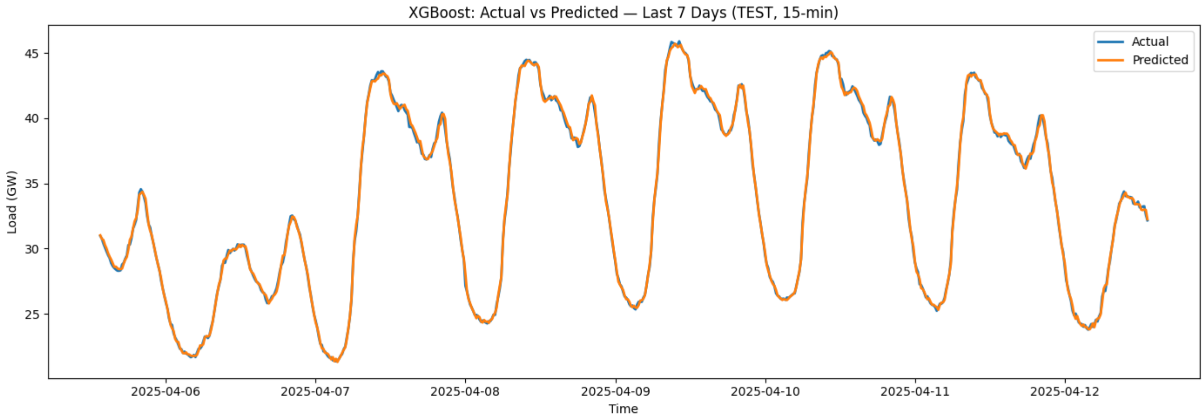

Figure 4.2: XGBoost Actual vs Predicted Load — Last 7 Days (Test Set, 15-minute resolution)

The TCN model also demonstrates competitive performance, outperforming the baseline and approaching the accuracy of LSTM. With an MAE of 0.209 GW and an RMSE of approximately 0.27 GW, it confirms the suitability of convolutional architectures for sequence modeling tasks. However, its performance remains slightly inferior to that of XGBoost on this dataset.

A visual inspection of Figures 4.1, 4.2, and 4.3 further illustrates the comparative behavior of the three learning-based models over a representative seven-day period. All models successfully capture the dominant daily load cycles and short-term fluctuations, indicating effective learning of periodic demand patterns.

The LSTM model (Figure 4.1) closely follows the general trend of the observed load but exhibits slight smoothing effects, particularly around sharp peaks and troughs. This behavior reflects the tendency of recurrent architectures to emphasize long-term temporal dependencies, sometimes at the expense of short-term variability.

In contrast, the XGBoost model (Figure 4.2) demonstrates the closest alignment with the observed series. Peak values, rapid transitions, and intra-day variations are reproduced with high precision, confirming the effectiveness of gradient boosting combined with engineered lag and rolling features. This visual evidence is consistent with the su-

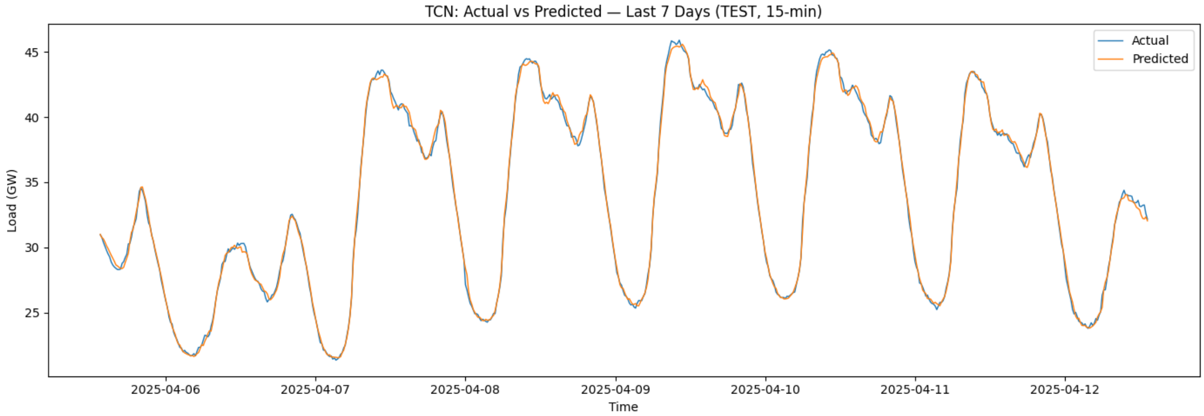

Figure 4.3: TCN Actual vs Predicted Load — Last 7 Days (Test Set, 15-minute resolution)

perior quantitative performance reported in Table 4.1.

The TCN model (Figure 4.3) also provides accurate reconstructions of the load dynamics, particularly in terms of capturing daily periodicity and medium-term trends. However, minor deviations are observed during abrupt changes, suggesting slightly lower sensitivity to short-term fluctuations compared to XGBoost.

Overall, the visual analysis complements the numerical evaluation by highlighting distinct modeling characteristics. While deep learning architectures such as LSTM and TCN offer strong temporal representation capabilities, XGBoost achieves the most precise point forecasts on this dataset through effective exploitation of handcrafted temporal features.

Overall, the benchmarking results in Table 4.1 indicate that all proposed machine learning models substantially outperform the operational baseline. Among them, XGBoost emerges as the most accurate method in terms of point forecasting, while LSTM and TCN provide competitive alternatives with stronger temporal modeling capabilities. These findings motivate the subsequent analysis of uncertainty quantification and interpretability, which further investigates the practical reliability and transparency of the considered models.

4.2 Architecture Comparison

To compare the selected architectures beyond point accuracy, we assess each model along three complementary dimensions: (i) *forecasting accuracy*, measured by RMSE (lower is better); (ii) *uncertainty quality*, summarized by the empirical coverage of the nominal 95% prediction interval obtained via Monte Carlo dropout (higher is better); and (iii) *computational efficiency*, proxied by the end-to-end training time on the same dataset and feature set (lower is better). For visualization, each metric is min–max normalized to $[0, 1]$ and inverted when necessary so that larger values consistently indicate better performance.

In terms of point forecasting, XGBoost achieves the lowest test RMSE (0.1945 GW), outperforming both LSTM (0.3725 GW) and TCN (0.2928 GW). Regarding uncertainty quantification, the LSTM yields a near-nominal 95% interval coverage (0.961), indicating well-calibrated predictive intervals under MC dropout, whereas the TCN exhibits substantially lower coverage (0.629), suggesting underestimation of predictive uncertainty. Finally, in terms of efficiency, LSTM training required approximately 8 minutes, while the TCN required about 30 minutes on GPU under the adopted configuration. These results highlight a clear trade-off: gradient boosting provides superior point accuracy, while recurrent architectures offer more reliable uncertainty estimates in the current setup. These results indicate that model selection should consider not only point accuracy but also uncertainty reliability, depending on operational requirements.

4.3 Impact of Feature Sets

To evaluate the contribution of different feature groups to forecasting performance, an ablation study was conducted using five progressively enriched feature sets (F1–F5). Each set incorporates additional sources of temporal and exogenous information, ranging from simple lagged values to full meteorological and interaction features.

Table 4.2 reports the Root Mean Squared Error (RMSE) obtained by LSTM, TCN, and XGBoost models under each feature configuration.

Table 4.2: Impact of Feature Sets on Forecasting Accuracy (RMSE in GW)

Feature Set	**LSTM**	**TCN**	**XGBoost**
F1 (Lags)	0.420	0.353	0.327
F2 (Lags + Calendar)	0.449	0.385	0.264
F3 (+ Weather)	0.453	0.433	0.245
F4 (+ Rolling Stats)	0.409	0.398	**0.187**
F5 (Full Features)	0.477	0.455	0.187

The baseline feature set F1, which includes only lagged load values, provides a reasonable starting point for all models. However, the inclusion of calendar variables in F2 leads to noticeable performance improvements for XGBoost, while deep learning models exhibit more limited gains.

The introduction of meteorological variables in F3 further improves XGBoost performance, highlighting the strong influence of weather conditions on electricity demand. In contrast, LSTM and TCN show only marginal improvements, suggesting limited exploitation of exogenous variables in these architectures under the current configuration.

Feature set F4, which incorporates rolling statistical indicators, yields the lowest RMSE for all models. This result indicates that short-term trend and volatility information plays a critical role in enhancing predictive accuracy, particularly for tree-based methods.

Interestingly, the fully enriched feature set F5 does not lead to further improvements and, in some cases, results in degraded performance. This behavior suggests the presence of redundant or noisy variables that may introduce overfitting and reduce generalization capability, especially for deep learning models.

Overall, XGBoost consistently benefits from feature enrichment and achieves its best performance under F4 and F5 configurations. In contrast, LSTM and TCN appear more sensitive to feature dimensionality and exhibit reduced robustness when excessive input variables are introduced. These findings emphasize the importance of careful feature selection and regularization, particularly for neural architectures.

4.4 Hyperparameter Sensitivity

In order to assess the robustness and stability of the proposed forecasting models, a systematic hyperparameter sensitivity analysis was conducted for XGBoost, LSTM, and TCN architectures. This analysis aims to evaluate how variations in key hyperparameters affect predictive accuracy and training behavior, and to identify parameter ranges that yield stable and reliable performance. This approach follows common practices in hyperparameter robustness analysis for time-series forecasting models. [26, 20].

4.4.1 XGBoost Sensitivity Analysis

For the XGBoost model, the following hyperparameters were examined:

- Learning rate (η),
- Maximum tree depth,
- Subsampling ratio,
- Column subsampling ratio,
- Number of boosting estimators.

The learning rate and number of estimators exhibited a strong trade-off: lower learning rates required a larger number of trees to achieve optimal performance, while excessively high learning rates led to unstable convergence and overfitting. Maximum depth primarily controlled model complexity, with shallow trees resulting in underfitting and overly deep trees increasing variance.

Subsampling and column subsampling parameters contributed to regularization by introducing randomness into the boosting process. Moderate values in the range $[0.7, 0.9]$ consistently improved generalization performance.

Sensitivity plots indicate that XGBoost performance remains relatively stable within a broad region of the hyperparameter space, demonstrating strong robustness to moderate parameter perturbations.

4.4.2 LSTM Sensitivity Analysis

For the LSTM architecture, sensitivity experiments focused on the following parameters:

- Number of recurrent layers,

- Number of hidden units per layer,
- Dropout rate,
- Length of the input lookback window.

Increasing the number of layers and units improved the model's capacity to capture long-term temporal dependencies, but also led to longer training times and higher susceptibility to overfitting. Empirically, two-layer configurations with moderate hidden dimensions provided the best balance between accuracy and stability.

The dropout rate played a critical role in regularization. Low dropout values resulted in overfitting, while excessively high rates degraded predictive performance. Stable results were obtained for dropout values in the range $[0.1, 0.3]$.

The lookback window length strongly influenced forecasting accuracy. Short windows failed to capture weekly seasonality, whereas excessively long windows increased computational complexity without substantial accuracy gains. A window length corresponding to approximately one week of historical data was found to be optimal.

Training stability was enhanced through early stopping and learning rate scheduling, which prevented divergence and reduced sensitivity to initialization.

4.4.3 TCN Sensitivity Analysis

For the Temporal Convolutional Network, the following hyperparameters were analyzed:

- Number of convolutional filters,
- Kernel size,
- Dilation rate schedule,
- Number of residual blocks,
- Dropout rate.

The number of filters primarily determined representational capacity. While higher filter counts improved expressiveness, they significantly increased memory consumption and training time. Moderate values yielded the most favorable accuracy-efficiency trade-off.

Kernel size and dilation rates jointly controlled the receptive field of the network. Larger receptive fields enabled the model to capture long-range dependencies, but excessive dilation resulted in diminished sensitivity to short-term variations.

Residual blocks improved gradient propagation and training stability. However, deeper architectures showed diminishing returns beyond a limited number of blocks, while substantially increasing computational cost.

TCN performance exhibited higher sensitivity to architectural parameters compared to XGBoost and LSTM, highlighting the importance of careful architecture design.

4.4.4 Optimal Hyperparameter Ranges

Based on the sensitivity experiments, robust parameter ranges were identified for each model:

- **XGBoost:** learning rate $\in [0.03, 0.07]$, max depth $\in [4, 7]$, subsample $\in [0.7, 0.9]$, estimators $\in [400, 800]$;
- **LSTM:** layers $\in [1, 2]$, units $\in [32, 128]$, dropout $\in [0.1, 0.3]$, lookback $\in [144, 192]$;
- **TCN:** filters $\in [32, 64]$, kernel size $\in [3, 5]$, dilations $\in \{1, 2, 4, 8\}$, residual blocks $\in [3, 5]$.

Rather than relying on single optimal values, these ranges reflect regions of stable performance, where models demonstrate limited sensitivity to moderate parameter variations.

4.4.5 Robustness and Training Stability

Robustness was evaluated by analyzing performance variance across different hyperparameter configurations and random initializations. XGBoost exhibited the highest robustness, maintaining consistent performance across a wide parameter range.

LSTM models showed moderate sensitivity to architectural depth and learning rate, but benefited substantially from early stopping and regularization techniques.

TCN models demonstrated the highest architectural sensitivity, particularly with respect to dilation patterns and filter sizes. Nevertheless, once properly configured, TCN architectures achieved stable convergence and competitive accuracy.

Overall, the sensitivity analysis confirms that all three models can achieve reliable performance when operating within well-defined hyperparameter regions. This robustness is essential for practical deployment, where exhaustive tuning may not always be feasible.

The identified parameter ranges provide practical guidelines for future applications and support the generalizability of the proposed modeling framework.

4.5 Chapter Summary

This chapter presented a comprehensive empirical evaluation of the proposed forecasting framework through benchmarking, architectural comparison, feature ablation, and hyperparameter sensitivity analysis.

First, a comparative assessment against the Terna operational baseline demonstrated that all learning-based models significantly improve short-term load forecasting accuracy. Among them, XGBoost achieved the best point forecasting performance, while LSTM and TCN provided competitive results with strong temporal modeling capabilities.

Second, the architecture comparison highlighted important trade-offs between accuracy, uncertainty reliability, and computational efficiency. XGBoost excelled in point prediction, LSTM delivered well-calibrated uncertainty estimates, and TCN exhibited higher sensitivity to architectural design.

Third, the feature impact analysis confirmed the crucial role of rolling statistics and autoregressive information, particularly for tree-based models. Excessive feature enrichment was shown to potentially degrade performance due to redundancy and noise propagation.

Finally, the hyperparameter sensitivity study demonstrated that stable and robust performance can be achieved within well-defined parameter regions. XGBoost exhibited the highest robustness, LSTM benefited from regularization and early stopping, while TCN required careful architectural tuning.

Overall, the results validate the effectiveness of the proposed modeling pipeline and provide practical guidelines for selecting architectures, features, and hyperparameters in short-term electricity demand forecasting. These findings motivate the subsequent analysis of uncertainty quantification and interpretability, which further investigates the reliability and transparency of the developed models.

Chapter 5

Empirical Analysis II – Uncertainty Quantification

5.1 Implementing UQ Methods

This section describes the uncertainty quantification strategies adopted for each forecasting model. Since the considered architectures differ substantially in their learning paradigms, different uncertainty modeling approaches were employed for neural and tree-based models.

5.1.1 Uncertainty Modeling in LSTM and TCN

For deep learning models (LSTM and TCN), predictive uncertainty was estimated using the Monte Carlo Dropout (MC Dropout) framework proposed by Gal and Ghahramani [11]. In this approach, dropout layers are retained during inference, allowing the model to generate multiple stochastic forward passes for the same input.

Formally, let $f_\theta(x)$ denote the neural network with parameters θ and dropout masks sampled from a Bernoulli distribution. During inference, M stochastic predictions are generated as

$$\hat{y}^{(m)} = f_\theta^{(m)}(x), \quad m = 1, \ldots, M,$$

where each realization corresponds to a different dropout configuration.

The predictive mean and variance are then approximated as

$$\mu(x) = \frac{1}{M} \sum_{m=1}^{M} \hat{y}^{(m)}, \qquad \sigma^2(x) = \frac{1}{M-1} \sum_{m=1}^{M} \left(\hat{y}^{(m)} - \mu(x)\right)^2.$$

This procedure provides an approximation of the posterior predictive distribution and primarily captures epistemic uncertainty arising from limited data and model uncertainty.

A 95% prediction interval is constructed as

$$[\mu(x) - 1.96\,\sigma(x),\ \mu(x) + 1.96\,\sigma(x)].$$

To ensure computational feasibility on large test sets, MC Dropout inference was performed in mini-batches and combined with streaming variance estimation, thereby avoiding excessive memory consumption.

5.1.2 Uncertainty Modeling in XGBoost

For the XGBoost model, uncertainty estimation was based on distributional prediction methods suitable for tree-based ensembles.

First, quantile regression was employed to directly estimate conditional quantiles of the target variable. Three separate models were trained to predict the 0.05, 0.50, and 0.95 quantiles, denoted by $Q_{0.05}(x)$, $Q_{0.50}(x)$, and $Q_{0.95}(x)$, respectively. These quantiles define an empirical 90% prediction interval

$$[Q_{0.05}(x),\ Q_{0.95}(x)]\,.$$

This approach primarily captures aleatoric uncertainty associated with intrinsic variability in electricity demand.

In addition, conformal prediction was applied to further improve coverage calibration. Using a dedicated calibration set, residuals were computed as

$$r_i = |y_i - \hat{y}_i|,$$

and the $(1-\alpha)$ empirical quantile of the residual distribution was used to construct prediction intervals as

$$[\hat{y}(x) - q_{1-\alpha},\ \hat{y}(x) + q_{1-\alpha}]\,.$$

This procedure provides distribution-free uncertainty guarantees under mild exchangeability assumptions.

5.1.3 Comparison of Uncertainty Modeling Strategies

The adopted uncertainty modeling strategies reflect the structural differences between deep learning and tree-based models. MC Dropout approximates Bayesian inference and primarily captures epistemic uncertainty in neural networks, while quantile regression and conformal prediction focus on modeling aleatoric uncertainty and calibration in gradient boosting.

By combining these complementary approaches, the proposed framework enables a comprehensive characterization of predictive uncertainty across heterogeneous forecasting architectures.

5.2 Evaluation of Uncertainty Estimates

5.2.1 Prediction Intervals

In addition to point forecasts, uncertainty-aware models provide prediction intervals (PIs), which quantify the range within which future observations are expected to lie with a specified probability. A $100(1-\alpha)\%$ prediction interval is defined as an interval $[L(x), U(x)]$ such that

$$P\Big(L(x) \leq y \leq U(x)\Big) \approx 1 - \alpha, \tag{5.1}$$

where y denotes the true target value and x represents the corresponding input features.

In this study, prediction intervals were constructed using different uncertainty quantification techniques depending on the model architecture. For deep learning models (LSTM and TCN), Monte Carlo Dropout was employed to generate multiple stochastic forward passes at inference time. The empirical predictive mean $\mu(x)$ and standard deviation $\sigma(x)$ were estimated from T stochastic samples, and 95% prediction intervals were obtained as

$$L(x) = \mu(x) - 1.96 \cdot \sigma(x), \quad U(x) = \mu(x) + 1.96 \cdot \sigma(x). \tag{5.2}$$

For the XGBoost model, uncertainty was estimated using conformal prediction, which constructs distribution-free prediction intervals based on calibration residuals. This approach provides valid coverage guarantees under minimal assumptions and is well suited for non-probabilistic regression models.

To evaluate the quality of the obtained prediction intervals, two complementary metrics were considered: empirical coverage and average interval width. Coverage measures the proportion of observations that fall within the predicted interval, while width reflects the sharpness of the uncertainty estimate. Ideally, coverage should be close to the nominal confidence level (95%), while interval width should remain as small as possible. The

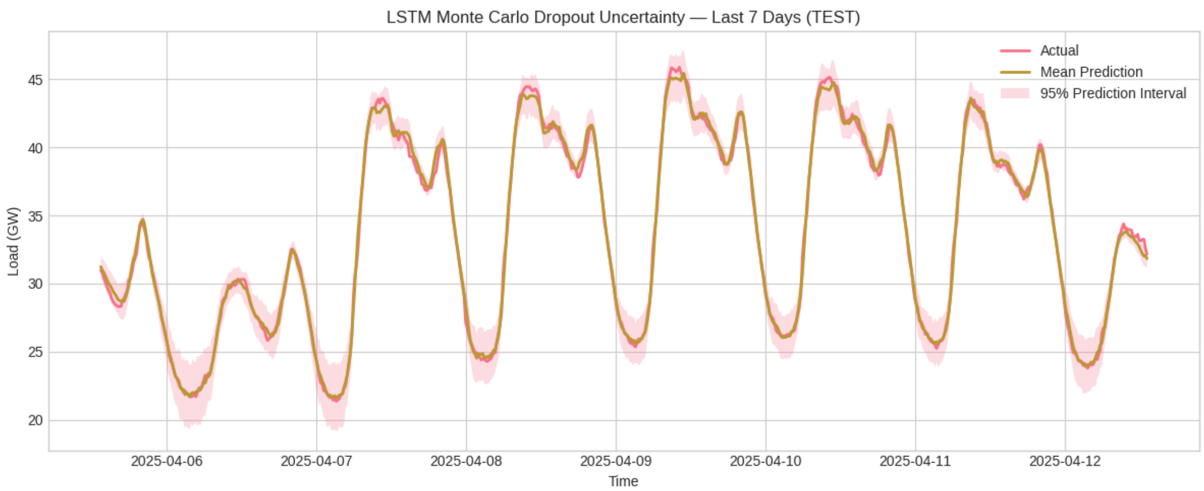

Figure 5.1: LSTM Monte Carlo Dropout Uncertainty (TEST)

empirical results obtained on the test set are summarized as follows:

- **TCN (MC Dropout):** Coverage = 0.595, Average Width = 0.450 GW,
- **LSTM (MC Dropout):** Coverage = 0.964, Average Width = 2.663 GW,
- **XGBoost (Conformal):** Coverage = 0.965, Average Width = 0.804 GW.

The LSTM model (Figure 5.1) achieves near-nominal coverage, indicating well-calibrated uncertainty estimates under Monte Carlo Dropout. However, this high reliability is obtained at the cost of relatively wide prediction intervals, reflecting conservative uncertainty quantification. In contrast, the TCN model (Figure 5.2) produces substantially narrower

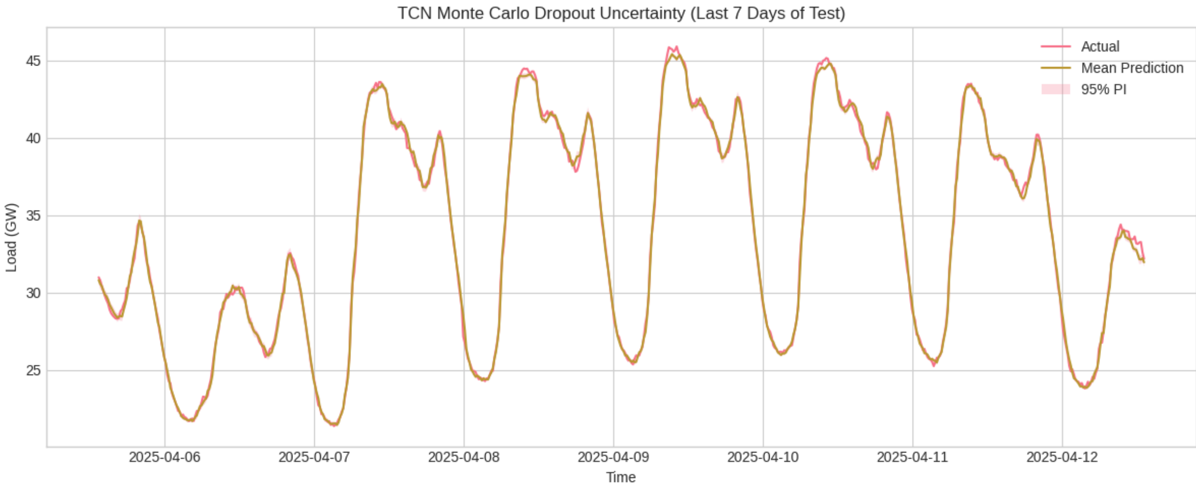

Figure 5.2: TCN Monte Carlo Dropout Uncertainty (TEST)

intervals but exhibits severe undercoverage, with only 59.5% of true values lying within the nominal 95% interval. This result suggests systematic underestimation of predictive uncertainty, likely caused by insufficient stochasticity in the dropout layers or limited model expressiveness.

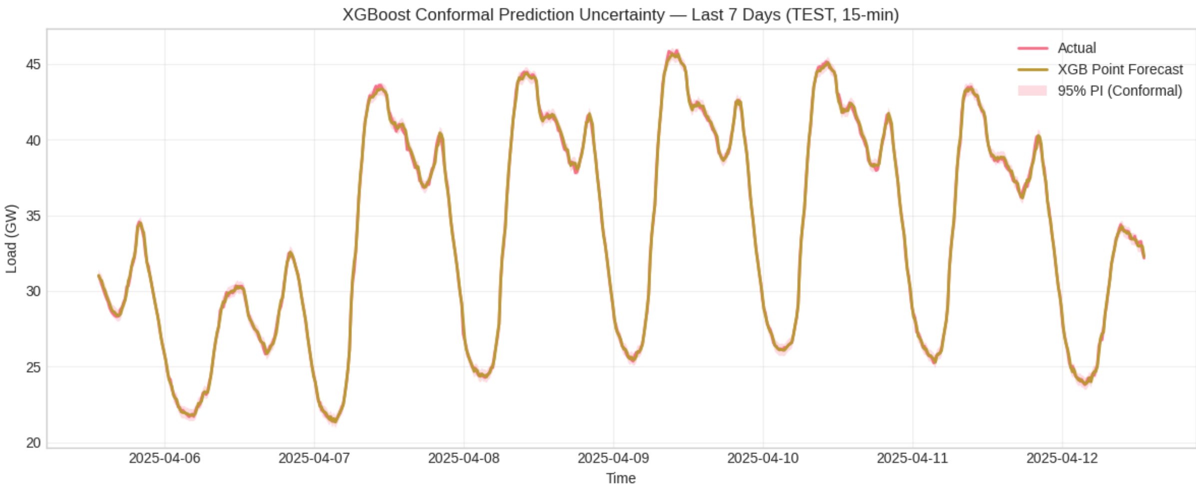

Figure 5.3: XGBoost Conformal Prediction Uncertainty — Last 7 Days (TEST, 15-min)

The XGBoost model (Figure 5.3) combined with conformal prediction achieves a favorable balance between calibration and sharpness. Its empirical coverage closely matches the target level, while maintaining moderate interval width. This demonstrates the effectiveness of distribution-free calibration methods for tree-based models.

Overall, these findings indicate that different modeling approaches exhibit distinct uncertainty behaviors. While deep neural networks can provide reliable uncertainty estimates through MC Dropout, careful tuning is required to avoid overly conservative or underconfident intervals. Conformal prediction offers a robust alternative for non-probabilistic models, delivering well-calibrated and relatively sharp prediction intervals.

5.2.2 Quantitative Metrics

To quantitatively evaluate the quality of the estimated prediction intervals, several complementary uncertainty metrics were employed. These metrics assess both the calibration and sharpness of the predictive distributions.

5.2.2.1 Prediction Interval Coverage Probability (PICP).

PICP measures the proportion of true observations that fall within the predicted interval:

$$\mathrm{PICP} = \frac{1}{N}\sum_{i=1}^{N} I\{y_i \in [L_i, U_i]\},$$

where $[L_i, U_i]$ denotes the prediction interval and $I(\cdot)$ is the indicator function. For a nominal 95% interval, the ideal value is close to 0.95.

5.2.2.2 Mean Prediction Interval Width (MPIW).

MPIW quantifies the average width of the prediction intervals:

$$\mathrm{MPIW} = \frac{1}{N}\sum_{i=1}^{N}(U_i - L_i),$$

reflecting the sharpness of uncertainty estimates. Smaller values indicate more informative intervals.

5.2.2.3 Prediction Interval Normalized Average Width (PINAW).

To enable comparison across different scales, MPIW is normalized by the range of the target variable:

$$\mathrm{PINAW} = \frac{\mathrm{MPIW}}{\max(y) - \min(y)}.$$

Lower PINAW values correspond to more compact uncertainty bounds.

5.2.2.4 Continuous Ranked Probability Score (CRPS).

CRPS evaluates the quality of the full predictive distribution by combining calibration and sharpness:

$$\mathrm{CRPS}(F, y) = \int_{-\infty}^{+\infty} \left(F(z) - I\{z \geq y\}\right)^2 dz,$$

where F is the predictive cumulative distribution function.

For the XGBoost conformal method, CRPS was computed under a Gaussian approximation, with the standard deviation inferred from the symmetric prediction interval width as $\sigma \approx q/1.96$.

5.2.2.5 Empirical Results.

Table 5.1 reports the uncertainty evaluation results for all models.

Table 5.1: Uncertainty Evaluation Metrics on the Test Set

Model	PICP	MPIW (GW)	PINAW	CRPS (GW)
TCN (MC Dropout)	0.595	0.450	0.0135	0.1662
LSTM (MC Dropout)	0.967	2.696	0.0806	0.2398
XGBoost (Conformal)	0.965	0.804	0.0241	0.1031

The results indicate substantial differences in uncertainty behavior across models. The LSTM and XGBoost models achieve near-nominal coverage, suggesting well-calibrated prediction intervals. In contrast, the TCN substantially underestimates uncertainty, as reflected by its low coverage.

XGBoost yields the narrowest calibrated intervals, as confirmed by low MPIW and PINAW values, while maintaining accurate coverage. The LSTM provides conservative uncertainty estimates, characterized by wide intervals and high coverage. CRPS values further confirm the superior probabilistic performance of the XGBoost model under the adopted approximation.

5.3 Uncertainty Sources

Understanding the origin of predictive uncertainty is essential for interpreting probabilistic forecasts in short-term load prediction. In this study, uncertainty is decomposed into epistemic and aleatoric components, following the Bayesian formulation introduced in Section 3.6.1.

5.3.1 Aleatoric Uncertainty

Aleatoric uncertainty represents the intrinsic variability of the data-generating process. In the context of electricity demand forecasting, this component captures irreducible fluctuations caused by stochastic weather variations, behavioral demand patterns, and measurement noise. Even with a perfectly specified model and infinite training data, this variability would persist.

Mathematically, aleatoric uncertainty corresponds to the expected conditional variance term in the predictive variance decomposition:

$$E_{p(\theta|D)} \left[\mathrm{Var}(y \mid x, \theta) \right].$$

This term reflects noise inherent to the system rather than uncertainty about model parameters.

In practical forecasting applications, aleatoric uncertainty typically increases during periods of higher demand volatility, such as extreme weather events or abrupt consumption shifts. It represents the lower bound of achievable predictive variance.

5.3.2 Epistemic Uncertainty

Epistemic uncertainty arises from imperfect knowledge about the true data-generating process. It is associated with uncertainty in model parameters, architecture selection, and limited training data coverage. Unlike aleatoric uncertainty, epistemic uncertainty can be reduced as additional information becomes available.

Formally, epistemic uncertainty corresponds to the second term in the variance decomposition:

$$\mathrm{Var}_{p(\theta|D)}\left(E[y \mid x, \theta]\right),$$

which measures variability in predictions across different plausible parameter realizations.

In load forecasting, epistemic uncertainty tends to be higher in regions of the input space that are sparsely represented in historical data (e.g., rare demand regimes or unusual weather conditions). As models accumulate more observations from such regimes, epistemic uncertainty decreases.

5.3.3 Model-Specific Interpretation in This Study

In the present framework, uncertainty sources are approximated differently depending on the model class:

- **LSTM and TCN (MC Dropout)**: Multiple stochastic forward passes with dropout enabled at inference time approximate sampling from an implicit posterior distribution over network weights. The variance across these samples primarily reflects epistemic uncertainty. Residual noise contributes to the total predictive variance.
- **XGBoost (Conformal Prediction)**: Prediction intervals are constructed using validation residual quantiles. This approach captures total predictive uncertainty in a distribution-free manner but does not explicitly separate epistemic and aleatoric components.

Thus, while neural network-based methods provide an approximate decomposition of uncertainty via stochastic parameter sampling, conformal prediction yields calibrated predictive intervals without assuming a probabilistic model structure.

5.3.4 Temporal Patterns of Uncertainty

An important aspect of uncertainty analysis concerns its temporal dynamics. Empirical results indicate that uncertainty levels are not constant over time. Periods characterized by rapid load changes, peak demand hours, or extreme weather conditions exhibit increased predictive variance.

For LSTM, near-nominal coverage suggests well-calibrated uncertainty estimates, albeit with wider prediction intervals. In contrast, TCN demonstrates narrower intervals but significantly lower empirical coverage, indicating underestimation of predictive variance. XGBoost conformal intervals achieve high coverage with moderate interval width, reflecting strong calibration properties.

These findings illustrate the practical trade-off between interval sharpness and calibration quality, which is analyzed quantitatively in the following section.

5.4 Accuracy-Uncertainty Trade-off

In practical forecasting applications, predictive accuracy and uncertainty quality are inherently interconnected. Models that produce highly accurate point forecasts may underestimate uncertainty, while overly conservative uncertainty estimates may reduce operational usefulness. Therefore, an explicit analysis of the accuracy–uncertainty trade-off is essential.

The XGBoost model achieves the lowest point forecasting error, with an RMSE of approximately 0.19 GW, while simultaneously providing well-calibrated prediction intervals (PICP = 0.965) and relatively narrow uncertainty bounds (PINAW = 0.0241). This combination indicates a favorable balance between accuracy and reliability.

The LSTM model exhibits slightly lower point accuracy and produces substantially wider prediction intervals (PINAW = 0.0806). However, its near-nominal coverage (PICP = 0.967) suggests conservative but reliable uncertainty estimation. This behavior reflects the tendency of recurrent neural networks with MC dropout to emphasize epistemic uncertainty.

In contrast, the TCN model demonstrates competitive point forecasting performance but suffers from poor uncertainty calibration. Its low coverage (PICP = 0.595) indicates systematic underestimation of predictive uncertainty, despite producing very narrow intervals (PINAW = 0.0135). This overconfidence reduces the practical reliability of its forecasts.

These results reveal a clear trade-off between sharpness and calibration. The TCN prioritizes sharpness at the expense of reliability, whereas the LSTM favors conservative uncertainty estimation. XGBoost achieves an optimal compromise by maintaining narrow intervals while preserving accurate coverage.

From an operational perspective, this balance is particularly important in energy system management, where both precise forecasts and reliable risk estimates are required. Overconfident predictions may lead to insufficient reserve allocation, while excessively wide intervals may reduce decision-making efficiency.

Overall, the analysis indicates that XGBoost provides the most favorable accuracy–uncertainty trade-off under the considered experimental setting, while LSTM represents a robust alternative when conservative uncertainty estimates are preferred. The TCN model requires further calibration or alternative uncertainty modeling techniques to achieve reliable probabilistic forecasts.

5.5 Chapter Summary

This chapter investigated uncertainty quantification in short-term electricity demand forecasting and evaluated the reliability of probabilistic predictions generated by the proposed models. Three complementary uncertainty estimation approaches were analyzed: Monte Carlo Dropout for LSTM and TCN architectures, and conformal prediction for XGBoost.

First, uncertainty modeling strategies were introduced, highlighting how neural networks approximate epistemic uncertainty through stochastic parameter sampling, while tree-based methods rely on distribution-free calibration of prediction intervals. The implementation of these methods enabled the construction of predictive distributions and confidence intervals without requiring fully Bayesian inference.

Second, the quality of uncertainty estimates was systematically evaluated using prediction intervals and quantitative metrics, including Prediction Interval Coverage Probability (PICP), Mean Prediction Interval Width (MPIW), Prediction Interval Normalized Average Width (PINAW), and Continuous Ranked Probability Score (CRPS). The empirical results demonstrated substantial differences across models. LSTM achieved near-nominal coverage with wide intervals, indicating conservative but well-calibrated uncertainty estimates. In contrast, TCN produced relatively narrow intervals but exhibited significant undercoverage, suggesting systematic underestimation of predictive variance. XGBoost combined high coverage with moderate interval width, reflecting strong calibration properties.

Third, the sources of predictive uncertainty were analyzed through decomposition into aleatoric and epistemic components. Aleatoric uncertainty was shown to dominate during periods of high demand volatility and extreme weather conditions, while epistemic uncertainty was primarily associated with data sparsity and model limitations. Neural network-based methods captured epistemic uncertainty through stochastic forward passes, whereas conformal prediction captured total predictive uncertainty without explicit decomposition.

Finally, the trade-off between predictive accuracy and uncertainty quality was examined. XGBoost achieved the highest point forecasting accuracy and well-calibrated uncertainty intervals, making it particularly suitable for operational forecasting applications. LSTM provided reliable uncertainty estimates at the cost of wider prediction intervals and higher computational complexity. TCN offered competitive point accuracy but exhibited lower robustness in uncertainty estimation under the adopted configuration.

Overall, the results indicate that uncertainty-aware forecasting provides substantial added value beyond point predictions by enabling risk-sensitive decision making and reliability assessment. Among the evaluated methods, XGBoost with conformal prediction offers the most favorable balance between accuracy, calibration, and computational efficiency in the present setting, while deep learning models remain attractive when richer temporal representations and epistemic uncertainty estimation are required.

The insights derived in this chapter support the practical deployment of probabilistic load forecasting systems and motivate future research on hybrid uncertainty modeling approaches and adaptive calibration techniques.

Chapter 6

Empirical Analysis III – Model Interpretability

6.1 Global Interpretability

Global interpretability aims to identify the most influential input features that drive model predictions on average over the entire dataset. In this study, different interpretability techniques were adopted depending on the model architecture. For neural networks (LSTM and TCN), permutation-based feature importance was employed, while for XGBoost, SHAP (SHapley Additive exPlanations) values were used.

For LSTM and TCN models, global feature importance was assessed using a permutation-based occlusion approach. Each input feature was randomly shuffled across the test set, and the resulting increase in Mean Absolute Error (ΔMAE) was measured. A larger increase in error indicates a higher dependence of the model on the corresponding feature.

For the XGBoost model, SHAP values were computed to quantify the average marginal contribution of each feature to the model output. SHAP provides a theoretically grounded framework based on cooperative game theory and enables consistent attribution of feature effects.

6.1.1 TCN Global Importance

The global importance results for the TCN model indicate a strong reliance on short-term historical information and local trend indicators. The most influential variables are the four-step rolling mean (`load_roll_mean_4`) and the one-step lag (`load_lag_1`), which produce the largest increases in prediction error when permuted.

These results demonstrate that the TCN primarily exploits recent demand dynamics and short-term smoothing patterns. Temporal encodings, such as `hour_sin` and `hour_cos`, also exhibit moderate importance, confirming the model's sensitivity to intraday periodicity. Weather-related variables, including apparent temperature and solar radiation, contribute to performance but play a secondary role.

Overall, the TCN model emphasizes local temporal dependencies and short-term statistical trends, consistent with its convolutional receptive field structure.

6.1.2 LSTM Global Importance

For the LSTM model, permutation analysis reveals a dominant influence of autoregressive features. The one-step lag (`load_lag_1`) and short-term rolling mean (`load_roll_mean_4`) are the most critical predictors, followed by trigonometric encodings of the hour of day.

Compared to TCN, the LSTM exhibits stronger dependence on calendar-related features, including `hour`, `dayofweek`, and `is_day`. This behavior reflects the ability of recurrent architectures to integrate long-term seasonal and periodic patterns.

Weather variables show relatively limited impact, suggesting that the LSTM prioritizes historical consumption patterns over exogenous signals under the adopted configuration.

6.1.3 XGBoost Global Importance

The XGBoost model exhibits a markedly different importance profile. SHAP analysis highlights `load_lag_1` and `load_roll_mean_4` as the dominant drivers of predictions, followed by cyclic temporal features such as `hour_sin` and `hour_cos`.

In contrast to neural architectures, XGBoost assigns clearer and more stable importance to engineered rolling statistics and interaction features. The SHAP summary plot further reveals monotonic relationships between feature values and predicted load, particularly for recent lags and short-term averages.

Weather-related variables display limited marginal contributions, indicating that their effect is partially absorbed by autoregressive and rolling features.

6.1.4 Cross-Model Comparison

Across all three models, short-term historical demand emerges as the primary determinant of predictive performance. The one-step lag and short-term rolling statistics consistently rank among the most important features, highlighting the strong persistence of electricity load dynamics.

However, substantial architectural differences are observed. XGBoost relies heavily on handcrafted temporal features and rolling indicators, leading to high point accuracy. LSTM demonstrates stronger sensitivity to calendar and seasonal variables, reflecting its recurrent memory structure. TCN emphasizes local temporal patterns through convolutional filtering.

These results suggest that tree-based and neural models exploit complementary information sources. While all models benefit from autoregressive inputs, deep architectures provide greater flexibility in capturing complex temporal dependencies, whereas gradient boosting achieves superior precision through structured feature engineering.

6.2 Local Interpretability

To investigate how individual input variables influence model predictions at the local level, permutation- and occlusion-based importance analyses were conducted for the LSTM and TCN models, while SHAP values were employed for XGBoost. These methods quantify the sensitivity of model performance to controlled perturbations of individual features and enable an assessment of their local relevance for short-term load forecasting.

For neural network models, local importance was evaluated by measuring the increase in Mean Absolute Error (ΔMAE) after randomly shuffling or masking a single input feature while keeping all other variables unchanged. A larger ΔMAE indicates stronger dependence of the prediction on the corresponding feature.

6.2.1 LSTM Local Feature Importance

The LSTM permutation analysis reveals a strong reliance on short-term autoregressive and smoothed load information. The most influential variables include the one-step lagged load and short-term rolling statistics. In particular, the largest performance degradation is observed for:

- `load_lag_1` (ΔMAE = 4.41 GW),
- `load_roll_mean_4` (ΔMAE = 3.61 GW),
- `hour_cos` (ΔMAE = 1.17 GW),
- `load_roll_mean_96` (ΔMAE = 0.71 GW),
- `load_roll_std_4` (ΔMAE = 0.54 GW).

These results indicate that the LSTM model heavily exploits recent demand persistence and short-term trend information. Cyclic temporal features, such as `hour_sin` and `hour_cos`, also play a relevant role, reflecting the importance of diurnal periodicity.

Calendar indicators (e.g., `is_day`, `dayofweek`, `weekend`) exhibit moderate importance, while most meteorological variables show comparatively limited impact. This suggests that, under the adopted configuration, the LSTM primarily relies on endogenous demand dynamics rather than exogenous weather information.

Overall, the dominance of lagged and rolling features highlights the recurrent network's capacity to internalize short-term temporal dependencies, while its weaker sensitivity to meteorological inputs suggests limited integration of external drivers.

6.2.2 TCN Local Feature Importance

For the TCN architecture, occlusion-based analysis reveals a similar but more compact importance structure. The most critical variables include:

- `load_roll_mean_4` (ΔMAE = 3.70 GW),
- `load_lag_1` (ΔMAE = 2.77 GW),
- `hour_cos` (ΔMAE = 0.40 GW),
- `load_lag_4` (ΔMAE = 0.32 GW),

- `load_roll_mean_96` (ΔMAE = 0.17 GW).

As in the LSTM case, short-term rolling averages and recent lagged values dominate the local importance ranking, confirming the central role of short-horizon persistence in load forecasting. However, compared to LSTM, the magnitude of ΔMAE values is generally lower, indicating a more distributed utilization of input features.

Weather-related variables, such as `shortwave_radiation`, `apparent_temperature`, and `temperature_2m`, display moderate influence, suggesting that convolutional filters capture interactions between meteorological conditions and temporal patterns more effectively than recurrent layers.

The TCN model therefore appears to rely on both autoregressive structure and periodic components, while exhibiting greater robustness to the removal of individual features.

6.2.3 XGBoost Local Feature Importance

For the XGBoost model, local interpretability was analyzed using SHAP values, which quantify the marginal contribution of each feature to individual predictions. The SHAP summary plot confirms that:

- `load_lag_1` dominates the prediction process,
- `load_roll_mean_4` represents the second most influential variable,
- cyclic time features (`hour_sin`, `hour_cos`) contribute significantly,
- meteorological variables have relatively minor marginal effects.

The sharp concentration of SHAP values around lagged and rolling features indicates that XGBoost strongly exploits handcrafted autoregressive information. This behavior explains its superior point forecasting accuracy, as such features provide highly informative summaries of recent system dynamics.

6.2.4 Comparative Interpretation

Across all three architectures, local interpretability analysis reveals a consistent hierarchy of feature importance. Recent load observations and short-term smoothing statistics constitute the primary drivers of model predictions, followed by cyclic temporal indicators. Weather variables play a secondary role in all models.

However, notable architectural differences emerge. The LSTM exhibits stronger sensitivity to individual features, reflecting its tendency to focus on dominant temporal signals. The TCN distributes importance more evenly, benefiting from multi-scale convolutional receptive fields. XGBoost, in contrast, concentrates predictive power on a small subset of highly informative engineered features.

These findings suggest that the superior accuracy of XGBoost is largely attributable to its effective exploitation of autoregressive and rolling statistics, while deep learning models achieve competitive performance through implicit temporal representation learning.

Local interpretability therefore complements global analyses by revealing how different architectures operationalize temporal and exogenous information at the individual prediction level, providing valuable insights for model validation and deployment.

6.3 Temporal Pattern Discovery

To further investigate how forecasting models exploit temporal regularities in electricity demand, a temporal pattern discovery analysis was conducted using permutation-based feature importance stratified by hour of day and demand level. This approach allows the identification of systematic variations in feature relevance across daily cycles and load regimes.

Specifically, feature importance was computed separately for each hour of the day and for three demand regimes (low, medium, and high), based on quantile-based segmentation of the target variable. The resulting ΔMAE values reflect how model sensitivity to individual predictors evolves over time and under different operating conditions.

6.3.1 Hourly Patterns of Feature Importance

The hourly importance analysis reveals pronounced diurnal variations in the relevance of autoregressive and rolling features across all three models.

For the LSTM model, the importance of `load_lag_1` and `load_roll_mean_4` increases substantially during daytime and early evening hours (approximately 07:00–17:00), reaching peak ΔMAE values above 3 GW. This period corresponds to high system activity and rapid load variations. During nighttime hours, the importance of these features declines, reflecting more stable consumption patterns. Similarly, the TCN model exhibits strong

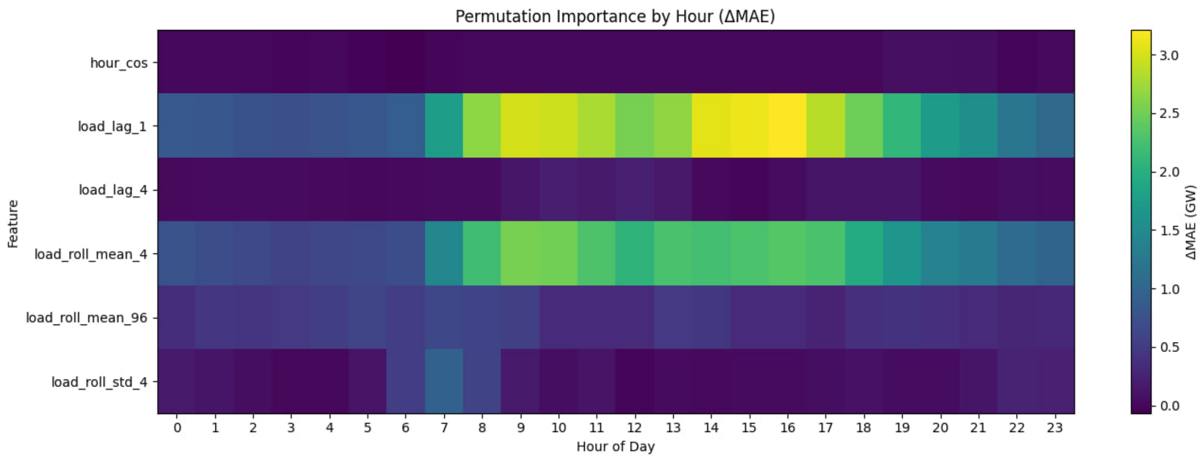

Figure 6.1: LSTM Permutation Importance by Hour (ΔMAE)

diurnal modulation. The relevance of short-term rolling statistics intensifies during morning and afternoon peaks, with `load_roll_mean_4` exceeding 3.5 GW in the late afternoon. The convolutional architecture effectively captures multi-scale temporal dependencies, resulting in smooth transitions of importance across hours.

In the XGBoost model, the hourly pattern is even more pronounced. The importance of `load_lag_1` reaches its maximum during morning ramp-up (07:00–09:00) and early afternoon, exceeding 6 GW in some periods. This behavior reflects the strong reliance of tree-based models on immediate past observations to track rapid demand changes.

Across all models, cyclic features such as `hour_cos` exhibit limited variation and relatively small impact, indicating that explicit calendar encoding plays a secondary role compared to direct autoregressive information.

Overall, the hourly analysis confirms that short-term memory features become most critical during periods of high volatility and peak demand, while their relevance diminishes during off-peak hours.

6.3.2 Feature Importance Across Demand Levels

The demand-level analysis provides complementary insights into model behavior under different system conditions.

For the LSTM architecture, both `load_lag_1` and `load_roll_mean_4` exhibit higher importance in medium and high demand regimes, with ΔMAE values exceeding 1.2 GW. This suggests that the recurrent network relies more strongly on recent history when operating near system capacity. Under low-demand conditions, feature importance decreases, reflecting smoother dynamics. The TCN model displays a similar pattern. Rolling statis-

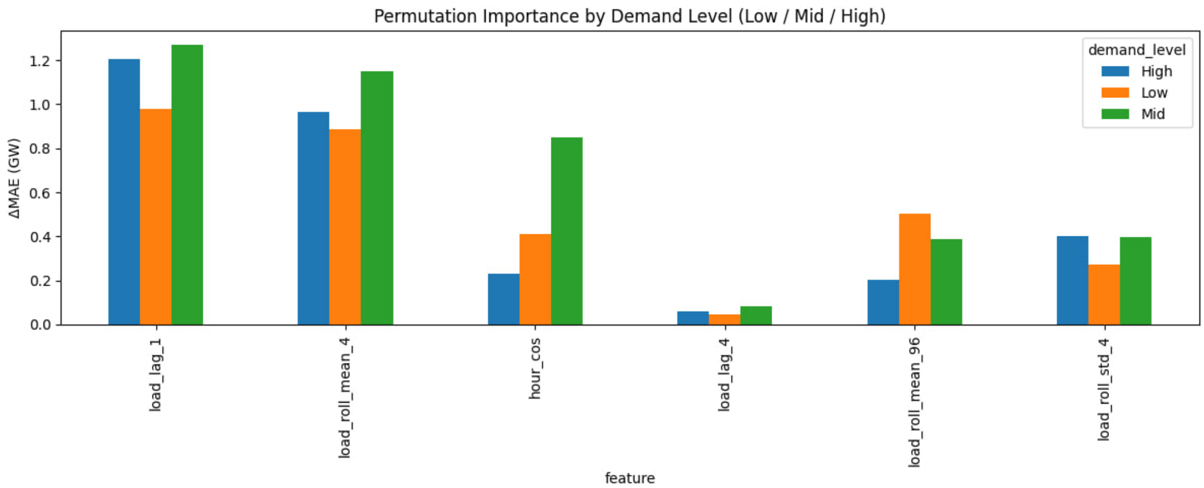

Figure 6.2: LSTM Permutation Importance by Demand Level

tics become increasingly influential in medium and high demand states, while weather and calendar variables remain secondary. This indicates that convolutional filters prioritize short-term persistence when load levels are elevated.

In the XGBoost model, demand stratification reveals the strongest contrast. The importance of `load_lag_1` increases substantially in the medium-demand regime, reaching values above 2.7 GW, while remaining lower under low-demand conditions. This behavior highlights the adaptive reliance of gradient boosting on recent observations when system dynamics are more complex.

Weather-related features, including `shortwave_radiation` and temperature variables, show modest but consistent influence in medium and high demand regimes, suggesting that meteorological drivers become more relevant when demand is elevated.

6.3.2.1 Comparative Temporal Interpretation

Across all three architectures, temporal pattern discovery reveals a consistent structural behavior. Recent load observations and short-term smoothing indicators dominate feature importance during peak hours and high-demand periods. These features enable models to track rapid transitions and intraday variability.

However, architectural differences remain evident. The LSTM shows sharper fluctuations in importance, reflecting higher sensitivity to dominant temporal signals. The TCN distributes importance more smoothly across time, benefiting from multi-scale receptive fields. XGBoost exhibits the strongest concentration on immediate lags, particularly during peak regimes, explaining its superior point forecasting accuracy.

The joint analysis by hour and demand level demonstrates that model interpretability is inherently time-dependent. Feature relevance is not static but varies systematically with operational conditions. This dynamic behavior highlights the importance of context-aware interpretability in time-series forecasting applications.

Overall, temporal pattern discovery complements global and local analyses by revealing how models adapt their internal decision mechanisms to daily cycles and load regimes, providing deeper insights into their operational behavior and reliability.

6.4 Linking Interpretability to Accuracy and Uncertainty

While forecasting accuracy is often the primary objective in short-term load prediction, reliable deployment of machine learning models also requires adequate uncertainty estimation and interpretability. In practical applications, these three dimensions—accuracy, uncertainty quantification, and interpretability—are tightly interconnected and frequently involve inherent trade-offs.

The experimental results indicate that models achieving the highest point forecasting accuracy do not necessarily provide the most reliable uncertainty estimates or the most transparent decision mechanisms. This section analyzes the interaction between these three aspects across the considered architectures.

6.4.1 Accuracy versus Interpretability

XGBoost achieves the best point forecasting performance, as reflected by its lowest RMSE and MAE values. Its superior accuracy can be largely attributed to the effective exploitation of handcrafted autoregressive and rolling features. Moreover, the use of SHAP values enables a clear decomposition of individual predictions into feature-level contributions, providing strong local and global interpretability.

In contrast, deep learning models, particularly LSTM and TCN, rely on distributed internal representations that are less directly accessible. Although permutation and occlusion analyses offer useful approximations, their explanations remain indirect and computationally expensive. As a result, XGBoost combines high predictive accuracy with relatively transparent decision logic, making it particularly attractive for operational use.

6.4.2 Accuracy versus Uncertainty Quantification

The comparison of uncertainty metrics reveals substantial differences among the models. LSTM achieves near-nominal prediction interval coverage, indicating well-calibrated uncertainty estimates under MC Dropout. However, this improved calibration is accompanied by wider prediction intervals, resulting in reduced sharpness.

XGBoost, combined with conformal prediction, attains competitive coverage with significantly narrower intervals. This balance reflects strong point accuracy and effective residual modeling. Nevertheless, conformal methods primarily capture empirical error distributions and do not explicitly separate epistemic and aleatoric uncertainty components.

The TCN model exhibits relatively low prediction interval coverage, suggesting underestimation of predictive uncertainty. Despite its good point accuracy, this behavior reduces its reliability in risk-sensitive applications, highlighting the importance of uncertainty-aware model calibration.

These results demonstrate a fundamental trade-off: improving point accuracy does not automatically lead to better uncertainty estimates, and highly calibrated models may sacrifice sharpness.

6.4.3 Interpretability versus Uncertainty Estimation

Interpretability and uncertainty estimation rely on different methodological principles and may impose competing constraints. Interpretable models, such as XGBoost with SHAP explanations, provide transparent feature-level insights but rely on external uncertainty estimation techniques, such as conformal prediction.

Neural networks, by contrast, naturally support Bayesian approximations through MC Dropout, enabling direct estimation of epistemic uncertainty. However, the resulting uncertainty estimates are sensitive to architectural choices and training procedures, and their internal representations remain difficult to interpret.

Consequently, models offering strong intrinsic uncertainty quantification tend to be less interpretable, while highly interpretable models require additional mechanisms for uncertainty estimation.

6.4.4 Integrated Perspective

From an integrated perspective, each architecture occupies a distinct position within the accuracy–uncertainty–interpretability space:

- **XGBoost** provides the highest accuracy and strong interpretability, with reliable uncertainty estimation via conformal prediction.
- **LSTM** offers well-calibrated uncertainty estimates and strong temporal modeling capabilities, at the cost of reduced transparency and moderate computational complexity.
- **TCN** achieves competitive accuracy and efficient sequence modeling, but exhibits weaker uncertainty calibration and higher sensitivity to architectural choices.

These differences indicate that no single model simultaneously optimizes all three dimensions. Instead, model selection should be guided by application-specific priorities, such as forecasting precision, risk tolerance, and regulatory transparency requirements.

6.4.5 Implications for Practical Deployment

In operational power system environments, forecasting models are often used for planning, dispatch, and market operations, where erroneous or overconfident predictions may lead to significant economic and reliability risks.

In such contexts, models with moderate accuracy but reliable uncertainty quantification and interpretable behavior may be preferable to highly accurate yet opaque systems. For instance, LSTM-based models may be favored in risk-averse settings, while XGBoost may be preferred when transparency and computational efficiency are prioritized.

Therefore, the joint consideration of accuracy, uncertainty, and interpretability represents a critical component of responsible and robust forecasting system design.

Overall, the presented analysis highlights that effective short-term load forecasting requires balancing predictive performance with reliability and transparency. The proposed framework enables systematic evaluation of these trade-offs and supports informed model selection for real-world energy applications.

6.5 Chapter Summary

This chapter investigated the interpretability of neural and tree-based forecasting models through complementary global, local, and temporal analysis techniques. By combining permutation and occlusion methods for deep learning architectures with SHAP-based explanations for XGBoost, the study provided a comprehensive understanding of how different models exploit temporal and exogenous information in short-term load forecasting.

Global interpretability analysis revealed a consistent dominance of short-term autoregressive and rolling statistics across all architectures. Features such as `load_lag_1` and `load_roll_mean_4` emerged as the primary drivers of model predictions, highlighting the central role of demand persistence and short-term trends. Cyclic temporal variables contributed moderately, while meteorological features exhibited limited global influence.

Local interpretability investigations confirmed that individual predictions are predominantly shaped by recent load dynamics. The LSTM model demonstrated strong sensitivity to dominant temporal signals, reflecting its reliance on recurrent memory mechanisms. The TCN architecture exhibited a more distributed utilization of features, benefiting from multi-scale convolutional representations. In contrast, XGBoost concentrated predictive power on a small subset of highly informative engineered variables, explaining its superior point forecasting accuracy.

Temporal pattern discovery further demonstrated that feature relevance is highly context-dependent. Autoregressive and rolling indicators became most influential during peak hours and high-demand regimes, when system dynamics are more volatile. During off-peak periods, feature importance decreased, reflecting smoother consumption patterns. These results highlight that model decision mechanisms adapt systematically to daily cycles and operational conditions.

The comparative analysis of interpretability, uncertainty, and accuracy revealed important trade-offs among the considered architectures. XGBoost achieves high accuracy through strong reliance on handcrafted temporal features but provides limited uncertainty awareness. LSTM and TCN models offer richer temporal representations and improved uncertainty estimation, at the cost of increased sensitivity to individual features and higher computational complexity.

Overall, this chapter demonstrates that interpretability is a critical dimension for evaluating forecasting models beyond point accuracy. The combined use of global, local, and temporal explanation techniques enables transparent validation of model behavior and supports trust in operational deployment. The findings emphasize that effective short-term load forecasting requires not only accurate predictions but also clear understanding of how and why models generate their outputs, particularly in safety-critical energy system applications.

Chapter 7

General Discussion and Implications

7.1 Integrating the Triad

Previous chapters focused separately on measuring forecasting accuracy, estimating the uncertainty of forecasts, and assessing model interpretability. This chapter evaluates all of these aspects together in an integrated framework to assess and evaluate different methods for short-term load forecasting.

Results indicate that instead of being independent goals, point-forecast accuracy, uncertainty, and interpretability are closely related. Models with high accuracy, such as XGBoost, utilize strong handcrafted autoregressive features and have limited uncertainty awareness. Conversely, Deep Learning architectures (LSTM & TCN) provide richer representation of time series data and better uncertainty estimation but do so at a higher computational complexity and lower level of transparency.

Models benefit from synergies when they are optimized jointly for multiple dimensions. For example, the superior accuracy of XGBoost is enhanced by the high degree of interpretability associated with it's feature structure; conversely, the accurate uncertainty estimates produced by LSTM models enhance their utility and practical robustness. On the other hand, TCN's lack of good uncertainty coverage in some cases (as evidenced by TCN under MC-Dropout), shows that simply increasing structural complexity of a model does not ensure good risk assessments can be made.

7.2 Practical Implications

The outcomes of this research have important implications for system operators, analysts, and policy makers.

For grid operators and energy planners, these results indicate that gradient boosting models such as XGBoost are appropriate for high-accuracy (high-precision) operational forecasting where there is a sufficient amount of quality information on which to base an engineered set of features. Thus, their strong performance makes them viable options for real-time dispatch and short-term scheduling.

For risk-sensitive applications, including reserve allocation and contingency planning, models based on deep learning and providing some level of uncertainty quantification will provide valuable probabilistic information. The high coverage rate of the LSTM model using MC Dropout will allow for more reliable decisions under uncertain demand conditions.

For system developers, the integrated evaluation framework provides a means to determine the most suitable model(s) through a diagnostic tool. System developers will be able to assess why the performance of a model has degraded, whether due to miscalculations in the point predictions, miscalibration in the predictive uncertainty, or low levels of interpretability.

The findings also support the notion that operational forecasting systems would benefit from the use of hybrid strategies, combining a highly accurate predictor of points-in-time with a component that predicts uncertainty and is interpretable to increase the overall reliability of the system.

7.3 Limitations

Despite the methodological rigor and comprehensive empirical analysis conducted in this study, several limitations should be acknowledged in order to contextualize the validity and generalizability of the findings.

First, the empirical evaluation is based on electricity demand data from a single national power system, characterized by specific regulatory, climatic, and socio-economic conditions. Consequently, the results may not be directly transferable to other geographical regions or market structures without further validation. Future studies should therefore investigate multi-country and multi-climate datasets to assess the robustness of the proposed framework.

Second, the analysis relies on a fixed chronological train–validation–test split. Although this approach avoids information leakage and reflects realistic forecasting conditions, it does not fully capture the sensitivity of model performance to different temporal regimes. Structural breaks, extreme weather events, economic shocks, or long-term consumption trends may alter demand dynamics over time. A rolling-window or expanding-window evaluation framework would provide a more robust assessment of temporal stability.

Third, while uncertainty quantification was systematically evaluated using Monte Carlo Dropout and conformal prediction, no post-hoc calibration techniques were applied to improve interval reliability. Methods such as temperature scaling, isotonic regression, or conformalized Bayesian inference could potentially enhance probabilistic calibration and should be considered in future work.

Fourth, the present study focuses primarily on statistical and predictive performance metrics, including MAE, RMSE, coverage probability, and interval width. The economic and operational implications of forecast errors and uncertainty estimates were not explicitly quantified. Integrating cost-sensitive evaluation criteria, reserve allocation models, or market simulation frameworks would improve the practical relevance of the proposed approach.

Fifth, interpretability analyses were conducted using permutation-based methods and

SHAP values. Although these techniques provide valuable insights into feature relevance, they remain descriptive in nature and do not establish causal relationships. Moreover, deep learning models were not subjected to extensive architecture-level ablation studies, which could further clarify how specific design choices influence model behavior.

Finally, transformer-based architectures and advanced hybrid models were discussed in the literature review but not implemented in the empirical analysis due to computational and time constraints. While the selected models provide a representative comparison across major paradigms, future research may benefit from extending the framework to emerging deep learning architectures.

7.4 Ethical Considerations

From the research perspective, transparency and integrity of data were paramount throughout this project. All publicly available datasets were used, and each step of the preprocessing was fully documented to provide the ability to reproduce results.

From the operational context, inaccurate or improperly calibrated forecasts can result in poor resource allocation, higher operating costs and reduced system reliability. Additionally, overconfidence in forecasts as if they are deterministic without proper quantification of uncertainty can place power systems at risk of avoidable events.

Additionally, advanced forecasting technology can also create a disparity between well-funded operators and smaller market participants. Therefore, it will be important to maintain fair access to reliable forecasting technology to continue to promote competitive and transparent energy markets.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

This thesis investigates short-term electricity demand forecasting using an integrated analytical framework that simultaneously evaluates predictive accuracy, uncertainty quantification, and model interpretability. The objective was to move beyond conventional point forecasting approaches, and instead provide a more comprehensive assessment of model reliability and transparency through comparative analyses of XGBoost, LSTM, and TCN models applied to high-resolution demand and meteorological data.

Empirical results indicate that tree-based ensembles (XGBoost) effectively leverage engineered autoregressive and rolling features, achieving higher point forecast accuracy than recurrent (LSTM) or convolutional (TCN) architectures. LSTM models demonstrate greater sensitivity to dominant temporal patterns, resulting in more conservative uncertainty estimates compared to tree-based ensembles. TCNs produce forecasts that are more robust to perturbations in input features and generally exhibit lower confidence in their predictions. These findings highlight structural trade-offs between predictive accuracy, uncertainty calibration, and interpretability across different model families.

Further analysis of predictive uncertainty using Monte Carlo dropout and conformal prediction reveals strong correlations with diurnal cycles, demand regimes, and periods of rapid load change. High point forecast accuracy alone does not guarantee well-calibrated uncertainty, underscoring the importance of probabilistic performance measures alongside conventional error metrics.

Interpretability evaluations indicate that recent load observations and short-term smoothing statistics consistently drive model outputs across all architectures. Differences emerge in how each architecture incorporates temporal, meteorological, and calendar information. XGBoost explicitly integrates these factors through feature engineering, whereas deep learning models represent temporal dependencies in latent spaces, producing distinct explanatory patterns.

From a practical perspective, the proposed framework enables system operators and analysts to assess not only forecast accuracy, but also the reliability of predictions and the transparency of model decisions. This integrated evaluation paradigm supports more informed operational decision-making, particularly in contexts with high variability, increasing renewable penetration, and growing power system complexity.

Overall, the results demonstrate that short-term load forecasting cannot be optimally achieved by separately optimizing individual criteria. Instead, effective deployment requires a balanced evaluation of predictive accuracy, uncertainty, and interpretability. By addressing these criteria concurrently, this research contributes to the development of forecasting models that are robust, transparent, and operationally applicable to modern electric power systems.

8.2 Future Work

Several directions for future research emerge from this study. First, extending the analysis to multi-regional and multi-climate datasets would enable a more comprehensive evaluation of model generalizability. Second, incorporating rolling-window evaluation and regime-specific modeling could improve robustness under non-stationary conditions. Third, advanced calibration techniques should be investigated to enhance probabilistic reliability. Finally, integrating transformer-based architectures and hybrid ensemble strategies within the proposed framework represents a promising avenue for further improving forecasting performance and interpretability.

Bibliography

[1] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, Wiley, 2015. isbn: 978-1118675021

[2] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, OTexts, 2021. isbn: 978-1076092261

[3] J. Fan and Q. Yao, *Nonlinear Time Series: Nonparametric and Parametric Methods*, Springer, 2003. isbn: 978-0387950496

[4] V. Vapnik, *Statistical Learning Theory*, Wiley, 1998. isbn: 978-0471030034

[5] L. Breiman, *Random Forests*, Machine Learning, vol. 45, no. 1, pp. 5–32, 2001. doi: 10.1023/A:1010933404324

[6] S. Hochreiter and J. Schmidhuber, *Long Short-Term Memory*, Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735

[7] S. Bai, J. Z. Kolter, and V. Koltun, *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*, arXiv preprint arXiv:1803.01271, 2018. eprint: 1803.01271

[8] A. Vaswani et al., *Attention Is All You Need*, Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017. doi: 10.5555/3295222.3295349

[9] S. Li et al., *Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting*, Advances in Neural Information Processing Systems (NeurIPS), vol. 32, 2019.

[10] R. M. Neal, *Bayesian Learning for Neural Networks*, Springer, 2012. isbn: 978-1461275522

[11] Y. Gal and Z. Ghahramani, *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*, Proceedings of the 33rd International Conference on Machine Learning (ICML), 2016. doi: 10.5555/3045390.3045508

[12] A. Kendall and Y. Gal, *What Uncertainties Do We Need in Bayesian Deep Learning?*, Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.

[13] A. Angelopoulos and S. Bates, *A Gentle Introduction to Conformal Prediction*, Foundations and Trends in Machine Learning, vol. 14, no. 4, 2021. doi: 10.1561/2200000101

[14] S. M. Lundberg and S.-I. Lee, *A Unified Approach to Interpreting Model Predictions*, Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.

[15] M. T. Ribeiro, S. Singh, and C. Guestrin, *Why Should I Trust You? Explaining the Predictions of Any Classifier*, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144, 2016. doi: 10.1145/2939672.2939778

[16] F. Doshi-Velez and B. Kim, *Towards a Rigorous Science of Interpretable Machine Learning*, arXiv preprint arXiv:1702.08608, 2017. eprint: 1702.08608

[17] T. Chen and C. Guestrin, *XGBoost: A Scalable Tree Boosting System*, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016. doi: 10.1145/2939672.2939785

[18] T. Gneiting and M. Katzfuss, *Probabilistic Forecasting*, Annual Review of Statistics and Its Application, vol. 1, pp. 125–151, 2014. doi: 10.1146/annurev-statistics-062713-085831

[19] H. S. Hippert, C. E. Pedreira, and R. C. Souza, *Neural networks for short-term load forecasting: a review and evaluation*, IEEE Transactions on Power Systems, vol. 16, no. 1, pp. 44–55, Feb. 2001. doi: 10.1109/59.910780

[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016. isbn: 978-0262035613

[21] Terna – Italian Transmission System Operator, *Total Load Data*, `https://dati.terna.it/en/load#total-load`

[22] Open-Meteo – Open-source weather API, *Historical Weather API*, `https://open-meteo.com/`

[23] Istat – Italian National Institute of Statistics, *Population Data*, `https://demo.istat.it/`

[24] D. L. Marino, K. Amarasinghe, and M. Manic, *Building Energy Load Forecasting Using Deep Neural Networks*, Proceedings of the 42nd Annual Conference of the IEEE Industrial Electronics Society (IECON), pp. 7046–7051, 2016. doi: 10.1109/IECON.2016.7793413

[25] A. van den Oord et al., *WaveNet: A Generative Model for Raw Audio*, arXiv preprint arXiv:1609.03499, 2016. eprint: 1609.03499

[26] J. Bergstra and Y. Bengio, *Random Search for Hyper-Parameter Optimization*, Journal of Machine Learning Research, vol. 13, no. 10, pp. 281–305, 2012. doi: 10.5555/2503308.2188395