



LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

INSTITUT FÜR INFORMATIK
LEHRSTUHL FÜR DATENBANKSYSTEME
UND DATA MINING



Master Thesis
in Computer Science

A comparative study between
structured state space models (S4) and
diffusion-based models for forecasting
hourly energy prices

Joshua Klinger

Aufgabensteller: Prof. Dr. Matthias Schubert
Betreuer: Prof. Dr. Matthias Schubert
Abgabedatum: 26.09.2025

Declaration of Authorship

I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references.

This paper was not previously presented to another examination board and has not been published.

Munich, 26.09.2025

.....
Joshua Klinger

Abstract

Energy price forecasting is a critical task for market participants, grid operators, and policymakers, as accurate predictions enable efficient energy trading, cost optimization, and stable grid management. Recent advances in deep learning have introduced novel architectures, such as *structured state space models (S4)* and *diffusion-based models*, which offer promising alternatives to traditional recurrent neural networks (RNNs) such as Gated Recurrent Units (GRUs). However, a comprehensive comparison of these approaches in the context of *hourly energy price forecasting* remains underexplored. This thesis conducts a systematic empirical study comparing the performance of *S4 models*, *diffusion-based WaveNet models*, and *baseline GRUs* - both in *conditioned (weather-aware) and unconditioned settings* - using real-world electricity market data.

We evaluate these models on multiple European energy markets, incorporating exogenous variables such as *temperature, wind speed, and demand forecasts* to assess their impact on prediction accuracy. Our experiments measure forecasting performance using establishing metrics, including *Mean Absolute Error (MAE)*, *Root Mean Squared Error (RMSE)*, and *directional accuracy*, while also analyzing computational efficiency and training stability.

Our findings suggest that the *GRU model*, with its ability to capture dependencies regarding the price development efficiently, outperforms the S4 in multi-step forecasting, particularly in unconditioned settings. Meanwhile, *diffusion-based WaveNet models* demonstrate superior robustness in high-volatility regimes when conditioned on weather features, albeit at higher computational costs and numbers of parameters in the model. This study provides *practical insights* into model selection for energy price forecasting, highlighting trade-offs between accuracy, interpretability, and scalability.

Keywords: Energy Price Forecasting, Structured State Space Models (S4), Diffusion Models, WaveNet, GRU, Deep Learning, Time Series Forecasting

Contents

1	Introduction	3
1.1	Background & Motivation	3
1.2	Research Objectives	5
1.3	Thesis Outline	7
2	Related Work	9
2.1	Traditional Methods in Energy Price Forecasting	9
2.2	Deep Learning Approaches for Time Series	10
2.3	Structured State Space Models (S4)	12
2.4	Diffusion Models in Forecasting	14
3	Methodology	16
3.1	Problem Formulation	16
3.2	Model Architectures	18
3.2.1	Baseline GRU (Conditioned and Unconditioned)	18
3.2.2	S4 Model (Conditioned and Unconditioned)	20
3.2.3	Diffusion-based WaveNet (Conditioned and Unconditioned)	22
3.3	Training and Optimization	25
3.4	Evaluation Metrics	27
4	Implementation	29
4.1	Dataset Description	30
4.2	Preprocessing & Feature Engineering	32
4.3	Computational Setup	33
4.4	Reproducibility Considerations	35
5	Results	38
5.1	Performance Comparison	38
5.2	Impact of Weather Features	44
5.3	Computational Efficiency	46

CONTENTS

6 Discussion	50
6.1 Strengths & Weaknesses of each Model	50
6.2 Practical Implications for Energy Markets	52
6.3 Limitations & Future Work	54
7 Conclusion	56
7.1 Summary of Findings	56
Bibliography	59

Chapter 1

Introduction

1.1 Background & Motivation

The liberalization of electricity markets and the rapid global transition towards a more sustainable energy system have fundamentally reshaped the landscape of power generation, distribution, and consumption. Hourly energy prices, once stable and predictable, now exhibit significant volatility and complex dynamics driven by an intricate interplay of supply, demand, weather patterns, and market-specific regulations [26]. Accurate and reliable forecasting of these prices has become an indispensable task for a diverse range of market participants. From a financial perspective, accurate forecasts enable generators and retailers to optimize their bidding strategies, minimize exposure to price fluctuations, and manage financial risk [16]. For grid operators, they are essential for ensuring network stability and security, as they inform resource allocation and the integration of renewable energy sources such as solar and wind power [11]. The challenges inherent in this task are manifold, stemming from the non-stationary nature of the time series, the presence of sharp price spikes, and the high dimensionality of relevant exogenous factors. Traditional statistical models, such as Autoregressive Integrated Moving Average (ARIMA) and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models, have historically served as the cornerstone of energy price forecasting [1]. While effective in capturing linear dependencies, they often struggle to model the highly non-linear relationships and intricate temporal patterns that characterize modern electricity markets [2].

The advent of deep learning has heralded a new era in time-series forecasting, providing models with the capacity to automatically learn complex, hierarchical features from raw data. Recurrent Neural Networks (RNNs) and their advanced variants, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRUs), have emerged as powerful tools for sequence modeling

due to their ability to process temporal information and capture dependencies over time [3]. The GRU, in particular, has become a popular and robust choice, offering a simplified architecture that retains the effectiveness of LSTMs while mitigating the issues of vanishing gradients and computational overhead. Consequently, conditioned and unconditioned GRU models have become a standard and formidable baseline in the field of energy price forecasting [13]. However, even these advanced recurrent architectures can face limitations, particularly when modeling very long sequences, where they may struggle to efficiently capture dependencies spanning hundreds or thousands of timesteps. This can be a significant drawback for hourly forecasting, where patterns may repeat on a daily, weekly or seasonal basis.

In recent years, two distinct and highly promising deep learning paradigms have emerged that seek to overcome these limitations. The first, Structured State Space Models (S4), re-engineers the classical state-space model to create a new class of deep sequence models with remarkable capabilities for long-range dependency modeling [8]. S4 models leverage a principled parameterization of the state-space equations that enables them to capture dependencies with a linear computational complexity relative to sequence length, an efficiency that can be superior to attention-based models and RNNs on long sequences. The use of an S4 Block within a deep network architecture allows for the effective modeling of complex dynamics across vastly different timescales, a property that is highly relevant for the multi-scale temporal patterns present in energy price data. This represents a potentially significant advancement for point forecasting accuracy by more effectively learning the underlying physical and market dynamics.

The second paradigm, diffusion-based models, has gained significant attention, primarily in the domain of generative modeling for images and audio, but has been successfully adapted for probabilistic time-series forecasting [9, 19]. These models learn to reverse a gradual diffusion process, essentially "denoising" random noise to generate realistic data samples. When applied to forecasting, this approach enables the model to learn and represent the full conditional distribution of future outcomes, rather than just a point estimate. This is a crucial advantage for risk-aware applications in the energy sector, as it allows for the quantification of forecasting uncertainty, providing not only a best estimate but also a range of plausible future scenarios. A WaveNet-based architecture, which uses dilated causal convolutions, is often employed within the diffusion framework to efficiently capture the temporal dependencies necessary for this task [17].

Despite the individual successes of these architectures, a rigorous and fair comparative study of the S4 and diffusion-based models against a strong GRU baseline for the specific, highly challenging task of hourly energy price forecast-

ing is currently not in depth covered in the academic literature. The existing work often focuses on one model in isolation or compares it to less competitive baselines. This thesis aims to fill this research gap, providing a comprehensive, multi-faceted analysis that not only benchmarks these advanced models but also seeks to understand the specific strengths and weaknesses of each paradigm. By investigating both unconditioned models and those conditioned on critical weather features, this study aims to shed light on which architecture is best suited for various aspects of energy price forecasting and to provide a foundational analysis for future research and practical applications in this domain.

1.2 Research Objectives

The primary objective of this thesis is to perform a comprehensive and rigorous comparative analysis of Structured State Space Models (S4) and diffusion-based models for the task of hourly energy price forecasting. This study will benchmark these advanced deep learning architectures against a robust Gated Recurrent Unit (GRU) baseline, considering both conditioned and unconditioned variants of each model. Through this systematic investigation, this thesis seeks to understand the specific strengths, weaknesses, and trade-offs of each approach, thereby guiding the selection of appropriate models for various real-world applications in the energy sector. The specific research objectives are meticulously outlined below to provide a clear roadmap for the study.

The first objective is to establish a high-performing and fair baseline for the comparison. To achieve this, we will develop and train two GRU models: an unconditioned model that learns the intrinsic temporal patterns of the energy price series and a conditioned model that leverages exogenous weather features as additional input. This process is essential for providing a benchmark against which the performance of the more modern architectures can be meaningfully evaluated. The GRU serves as a strong, widely-accepted representative of recurrent neural networks, and its performance will ground the subsequent analysis in a well-understood context. This objective ensures that the observed performance gains from the S4 and diffusion models are not merely incidental but represent a genuine improvement over a state-of-the-art alternative.

Following the establishment of the baseline, a core objective is to rigorously evaluate the performance of Structured State Space Models (S4). We will implement both an unconditioned S4 model and one conditioned on weather features. This will allow us to assess the hypothesis that S4 models, with

their ability to efficiently capture long-range dependencies, can achieve superior point forecasting accuracy for hourly energy prices compared to the GRU baseline. The evaluation will focus on standard point forecasting metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), which are crucial for market participants focused on minimizing prediction error. The study will aim to understand whether the architectural advantages of S4 translate into tangible performance gains on this complex time-series task.

Simultaneously, a key objective is to investigate the capabilities of diffusion-based models, implemented with a WaveNet backbone, for probabilistic energy price forecasting. Like the other models, we will develop a unconditioned and a weather-conditioned variant. Furthermore we will also use the MAE and RMSE to draw conclusions with the other models.

Building upon these individual evaluations, a critical objective of this thesis is to conduct a comprehensive, multi-dimensional comparison across all three model families. This comparison will extend beyond simple performance metrics to include an analysis of their respective computational efficiencies, such as training time and inference speed. The goal is to provide a holistic understanding of the trade-offs involved in using each architecture. I will systematically analyze which model excels in point forecasting, which is best for quantifying uncertainty, and which offers the most favorable balance of performance and computational cost.

A final, but no less important, objective is to explicitly quantify the impact of incorporating exogenous weather features on the forecasting performance of all three model architectures. By systematically comparing the performance of the unconditioned and conditioned variants for the GRU, S4 and diffusion model, we aim to measure the value of this external information. This will provide empirical evidence supporting the importance of external covariates in energy price forecasting and shed light on whether certain model architectures are better equipped to leverage this additional information. The fulfillment of this objective will offer practical guidance for practitioners and researchers on the necessity of including such features in their models.

By addressing these five objectives, this thesis will provide a robust and novel benchmark for deep learning in the domain of energy price forecasting. The findings will offer valuable insights into the performance, efficiency, and specific advantages of each model class, thereby informing the design of more effective forecasting systems for future energy markets.

1.3 Thesis Outline

This thesis is organized into seven chapters, each designed to build upon the preceding one and guide the reader through the research process, from foundational concepts to the final conclusions. The structure is designed to present a clear and coherent narrative of the entire research project.

Chapter 1, the current chapter, provides the **introduction** of the thesis. It begins by establishing the background and motivation for the research, highlighting the importance and challenges of hourly energy prices forecasting. It then clearly states the specific research objectives and concludes by providing this overview of the thesis structure.

Chapter 2 is dedicated to a detailed **related work** review. This chapter provides a historical context by summarizing traditional statistical methods for energy price forecasting and their limitations. It then transitions to an overview of deep learning approaches for time-series data. The chapter concludes with a focused review of the specific architectures central to this study: Structured State Space Models (S4) and diffusion-based models, thereby situating this research within the current state of the art.

Chapter 3 describes the **methodology** employed in this research. It begins by formally defining the problem of hourly energy price forecasting. The chapter then provides a detailed theoretical description of the three primary model architectures - the GRU, S4 and the diffusion-based WaveNet - as well as their conditioned and unconditioned variants. The chapter concludes by outlining the experimental setup, including the training and validation protocols, and the specific evaluation metrics used to assess model performance.

Chapter 4 represents the practical **implementation** of the study. This chapter provides a thorough description of the dataset used, including the energy price series and the exogenous weather features. It explains the data preprocessing and the features engineering steps taken. The chapter also highlights the computational setup and key hyperparameters, ensuring the study's reproducibility for future research.

Chapter 5 shows the final empirical **results** of the comparative study. This chapter is structured to present the findings in a clear, sequential manner. It will first present the performance of the GRU baseline, followed by the S4 models, and finally the diffusion-based models. A dedicated section will then provide a comprehensive comparison of all architectures on point forecasting

accuracy, probabilistic forecasting quality, and computational efficiency. The chapter ends with an ablation study, displaying the different performances of the conditioned and unconditioned variants.

Chapter 6 provides a **discussion** of the results. This chapter moves beyond the raw data to interpret the findings. It will analyze the strengths and weaknesses of each model, relate the results back to the theoretical underpinnings of each architecture, and discuss the practical implications of the findings for practitioners in the energy market. It will also candidly address the limitations of the study.

Finally, Chapter 7 serves as the **conclusion** of the thesis. It provides a concise summary of the key findings from the research, restates the primary contributions, and offers recommendations for future work and research directions based on the insights gained from this study.

Chapter 2

Related Work

2.1 Traditional Methods in Energy Price Forecasting

The forecasting of electricity prices has been subject of extensive research for several decades, driven by the critical role these predictions play in a liberalized energy market. Prior to the widespread adoption of deep learning, the field was dominated by a suite of traditional statistical and econometric models. These methods, grounded in rigorous mathematical theory, typically rely on the assumption of stationary or require explicit transformations to model time-series data. Their primary strength lies in their interpretability and well-established theoretical foundations, which allow for a clear understanding of the relationships between input variables and forecasted outcomes.

One of the most foundational and widely applied classes of models is the Autoregressive Integrated Moving Average (ARIMA) family [1]. ARIMA models are highly effective for capturing linear dependencies within a time series, decomposing the data into autoregressive (AR), differencing (I), and moving average (MA) components. Their application in energy price forecasting often involves extending the model to incorporate seasonal effects, leading to the Seasonal ARIMA (SARIMA) model [26]. While these models can provide a robust baseline, they fundamentally struggle to capture the non-linear, non-stationary behavior of electricity prices, which are frequently subject to sharp and unpredictable spikes [2]. Their performance can degrade significantly when faced with the complex, multi-modal distribution of modern energy markets, which are influenced countless interconnected factors.

To address the volatility inherent in energy price data, econometric models de-

signed for capturing fluctuating variance, such as Generalized Autoregressive Conditional Heteroskedasticity (GARCH) family, have been widely employed [27]. These models are particularly useful for forecasting the conditional variance of prices, which is a proxy for market risk. GARCH models and their various extensions, such as the Exponential GARCH (EGARCH) and Glosten-Jagannathan-Runkle GARCH (GJR-GARCH) have provided valuable insights into the clustering of volatility in energy prices [4]. By combining GARCH models with ARIMA, hybrid approaches have sought to capture both autoregressive structure and the time-varying volatility of the series. While effective for modeling the second-order moments of the price distribution, these models are still limited by relying on pre-defined functional forms and often fail to capture the complex, non-linear relationships with exogenous variables.

Beyond purely time-series based models, a range of statistical and machine learning methods have also been utilized. These include linear regression, which establishes a simple relationship between prices and a set of independent variables such as demand, supply, and weather forecasts [11]. More advanced techniques, such as Support Vector Machines (SVMs) and Random Forests, have also been applied, leveraging their ability to model non-linear relationships without making strong distributional assumptions [22]. Random Forests, in particular, have been effective due to their ensemble nature, which can mitigate overfitting and provide a robust prediction by aggregating the outputs of multiple decision trees. However, these methods, while powerful, often treat the time-series forecasting problem as a static regression task, losing the sequential nature of the data. They typically require a manual process of feature engineering to create lagged and temporal indicators, a process that can be both laborious and potentially suboptimal. The limitations of these traditional methods - their struggles with non-linearity, high-dimensional feature spaces, and the efficient capture of long-range dependencies - set the stage for the exploration of more flexible and data-driven approaches offered by deep learning.

2.2 Deep Learning Approaches for Time Series

The limitations of traditional forecasting methods in a rapidly evolving energy market have prompted a paradigm shift towards deep learning. Deep learning models possess a unique ability to automatically learn hierarchical representations and complex non-linear relationships directly from raw data, bypassing the need for extensive manual feature engineering [14]. This capacity has made

them exceptionally well-suited for the intricate and high-dimensional nature of time-series forecasting.

Early applications of deep learning in time-series forecasting often utilized simple Feedforward Neural Networks (FNNs) [26]. These networks, however, treat each time step independently, which fundamentally ignores the sequential order and temporal dependencies that are crucial for forecasting. To address this, Recurrent Neural Networks (RNNs) were introduced, marking a significant milestone in sequence modeling [20]. RNNs are designed with a feedback loop, allowing information from previous time steps to influence the processing of the current one. While conceptually powerful, basic RNNs suffer from the vanishing gradient problem, which makes it difficult to learn long-range dependencies, a critical drawback for capturing multi-scale patterns in energy price data (e.g. daily and weekly seasonality).

This limitation led to the development of more sophisticated recurrent architectures, most notably Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) [10, 3]. LSTMs and GRUs introduce gating mechanisms that allow the model to selectively remember or forget information over long periods, thereby effectively mitigating the vanishing gradient issue. The GRU, is a simplified variant of the LSTM, has become a particularly popular choice in time-series forecasting due to its comparable performance with reduced computational complexity. It provides a strong baseline for this study, as it represents a robust and well-established deep-learning approach for sequence modeling [13]. The GRU’s ability to model conditional dependencies on exogenous features makes it a powerful and relevant benchmark for assessing the performance of newer architectures.

Beyond recurrent models, Convolutional Neural Networks (CNNs) have also been adapted for time-series forecasting. While traditionally used for image processing, CNNs can effectively learn local patterns and features in time-series data using one-dimensional convolutions. An important development in this area is the WaveNet architecture, which leverages dilated causal convolutions to expand its receptive field exponentially with network depth without losing resolution [17]. This allows WaveNet to capture long-range dependencies efficiently, offering an alternative to recurrent architectures. Its architecture, focused on generating sequences one step at a time, makes it a natural fit for integration into the diffusion-based forecasting framework, as it can model the intricate temporal dynamics required for learning the data distribution.

More recently, the attention mechanism, a key component of the Transformer

architecture, has revolutionized sequence modeling by allowing the model to weigh the importance of different parts of the input sequence, irrespective of their temporal distance [25]. While attention-based models have achieved state-of-the-art results in many domains, their quadratic computational complexity with respect to sequence length can make them computationally prohibitive for very long time-series. This limitation has motivated the search for more efficient architectures that can still effectively capture long-range dependencies, a need that newer models like S4 and diffusion-based models seek to address.

2.3 Structured State Space Models (S4)

Recent advancements in deep sequence modeling have sought to overcome overcome the limitations of recurrent and attention-based architectures, particularly with respect to their ability to efficiently capture long-range dependencies. A promising new paradigm that has emerged is the Structure State Space Model (S4), which re-engineers the classical state-space model to create a new class of deep sequence models with remarkable capabilities [8]. S4 provides a principled framework for modeling sequences that combines the computational efficiency of convolutional networks with the memory-retention capabilities of recurrent models, making it particularly well-suited for tasks where patterns may span thousands of time steps. This section provides an overview of the S4 architecture and its relevance to the problem of hourly energy price forecasting.

The foundation of S4 lies in the linear time-invariant (LTI) state-space model, a system that describes a continuous-time signal $x(t)$ through a hidden state $h(t)$. The system is defined by a set of linear ordinary differential equations (ODEs):

$$h'(t) = Ah(t) + Bx(t) \quad (2.1)$$

$$y(t) = Ch(t) + Dx(t) \quad (2.2)$$

where $x(t)$ is the input, $y(t)$ is the output, and the $h(t)$ is the hidden state. The matrices A, B, C, D define the dynamics of the system. While this formulation is continuous, deep learning models operate on discrete sequences. The key innovation of S4 is the effective discretization of this continuous-time model, allowing it to be integrated into a neural network architecture. This is achieved by converting the continuous system into a discrete one using a zero-order hold, which results in the following recurrent equations:

$$h_k(t) = \bar{A}h_k(t) + \bar{B}x_k(t) \quad (2.3)$$

$$y_k(t) = \bar{C}h_k(t) + \bar{D}x_k(t) \quad (2.4)$$

The matrices $\bar{A}, \bar{B}, \bar{C}, \bar{D}$ are the discretized versions of their continuous counterparts, and their computation is where the "structure" in S4's name comes from.

The choice of the continuous-time parametrization is crucial for S4's success. The work by Gu et al. (2022) leveraged the properties of the HiPPO (HiPPO-Legendre) family of matrices, which were originally developed for memorizing continuous functions [12, 8]. These matrices, specifically the HiPPO matrix, provide an optimal parameterization of the state-space matrix A that allows the model to compress long sequences into a fixed-size state, retaining information about the past while forgetting irrelevant details. This enables the S4 model to capture extremely long-range dependencies, a task where traditional RNNs and even some attention-based models can struggle due to vanishing gradients or computational complexity.

A defining characteristic of S4 is its dual computational pathway. It can be computed in a recurrent manner, which is efficient for autoregressive inference, or, more importantly, it can be computed as a convolution [8]. The recurrent form is identical to the discrete state-space update rule described above. The convolutional from the fact that the state-space model's output can be expressed as a convolution of the input sequence with a learned impulse response function, \bar{K} . This convolutional formulation allows for highly parallelized training on modern hardware, bypassing the sequential bottleneck that plagues traditional RNNs. The computational complexity of S4 is linear in the sequence length, denoted as $\mathcal{O}(L)$, which offers a significant advantage over the quadratic complexity of standard self-attention mechanisms, $\mathcal{O}(L^2)$, on long sequences [8]. This makes S4 a compelling alternative for applications like hourly energy price forecasting, where data sequences can span days, weeks, or even years.

The S4 architecture has demonstrated state-of-the-art performance on a range of long-sequence modeling benchmarks including tasks from the Long-Range Arena (LRA) [24] and has been utilized on sequential modeling tasks in fields as diverse as audio and medical data [12]. More recent developments, such as Mamba, have further refined this approach, demonstrating a competitive alternative to Transformers [7]. The relevance of S4 to this thesis is therefore on the one hand offering a powerful and efficient mechanism for capturing the long-range temporal dynamics of energy price dat and on the other hand

providing a strong, modern deep learning paradigm to compare against the GRU baseline and the probabilistic diffusion models.

2.4 Diffusion Models in Forecasting

Diffusion models represent a class of powerful generative models that have recently gained significant traction in the deep learning community, primarily for their ability to generate high-quality images and audio [9, 23]. Their success stems from a framework that involves learning a gradual denoising process to transform random noise into structured data. This framework has proven to be remarkably adaptable and has recently been extended to the domain of time-series forecasting, where it offers a unique and powerful approach to probabilistic modeling [19, 15].

The core principle of a diffusion model is centered on two processes: a forward and a reverse. The forward process is a fixed Markov chain that gradually adds Gaussian noise to the data until the data distribution is completely corrupted and indistinguishable from a simple normal distribution. This process can be defined by the following transition kernel, which adds noise to a data sample x_{t-1} to produce x_t :

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t I) \quad (2.5)$$

where β_t is a predefined schedule of noise variance, and I is the identity matrix. A notable and computationally efficient property of this process is that the distribution of a sample at any time step t can be directly sampled from the original data point x_0 as follows:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I) \quad (2.6)$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$.

The reverse process is a learned Markov chain that starts from this pure noise and iteratively denoises it, step by step, to reconstruct the original data sample. The model's training objective is to learn the parameters of this reverse process, which is also modeled as a Gaussian distribution:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \quad (2.7)$$

Here, the neural network, parameterized by θ , learns to predict the mean μ_θ and covariance Σ_θ of the denoising step. This is typically achieved by training

a neural network to predict the noise that was added to each step, conditioned on the current state of the noisy data and the time step [23].

When applied to time-series forecasting, this framework is adapted to model the conditional distribution of future values given a historical context. This is the central principle of probabilistic forecasting with diffusion models, as exemplified by architectures like TimeGrad [19]. Instead of generating an entire time series from scratch, the model learns to generate a probability distribution over the future sequence, conditioned on the past observed data. For a given input sequence of past energy prices and, in the conditioned case, weather features, the diffusion model generates a diverse set of possible future trajectories by sampling from the learned distribution. The result samples provide a rich probabilistic forecast, encompassing not only a point estimate but also the full range of potential outcomes and their associated likelihoods. This is a crucial distinction from traditional point forecasting models, which typically provide only a single best guess.

The neural network that learns the reverse diffusion process is often a time-series-specific architecture. A common choice is the WaveNet-based network, which utilizes dilated causal convolutions to efficiently capture temporal dependencies over a wide receptive field [17]. The WaveNet architecture's ability to model dependencies across different scales makes it a natural fit for the denoising task in time-series diffusion, as it can learn the intricate temporal correlations required to transform noise into a coherent time-series signal. By conditioning this WaveNet on the historical input and any relevant exogenous features, the model becomes a powerful tool for generating conditional, probabilistic forecasts [15].

The major contribution of diffusion models to time-series forecasting is their capacity for generating high-quality probabilistic forecasts, which are invaluable for risk management in the energy sector [28]. By sampling multiple trajectories, these models allow market participants to quantify the uncertainty of future prices, enabling them to make more informed and robust decisions. This thesis will investigate whether this strength in probabilistic forecasting comes with a trade-off in point forecasting accuracy or computational cost, providing a necessary comparison to the more deterministic S4 and GRU models.

Chapter 3

Methodology

3.1 Problem Formulation

The task of hourly energy price forecasting is formally defined as a time-series prediction problem. Given a sequence of historical observations up to a specific time point t , the objective is to predict the future values of the time series over a defined forecast horizon. In this thesis, we are concerned with forecasting the hourly energy price, denoted as $y_t \in \mathbb{R}$, which constitutes the target time series. This series is known to exhibit complex temporal characteristics, including daily, weekly, and seasonal patterns, as well as high volatility and the presence of sudden price spikes. The forecasting problem is made more intricate by the influence of exogenous factors, such as weather conditions, which have a significant impact on both electricity demand and renewable energy supply.

We formalize this problem for both point forecasting and probabilistic forecasting. For a given time step t , the historical input consists of a lookback window of past energy prices, denoted as $Y_{1:t} = (y_1, y_2, \dots, y_t)$, and a corresponding sequence of exogenous features, $X_{1:t+H} = (x_1, x_2, \dots, x_t + H)$, where $x_t \in \mathbb{R}^k$ is a vector of k features at time t . The forecast horizon is defined as H hours.

The problem of point forecasting is to predict a single future value or a sequence of future values of the energy price, $\hat{Y}_{t+1:t+H} = (\hat{y}_{t+1}, \dots, \hat{y}_{t+H})$. This prediction is a function of the observed history, and is mathematically expressed as:

$$\hat{Y}_{t+1:t+H} = f(Y_{1:t}, X_{1:t+H}) \quad (3.1)$$

The goal of the forecasting model f is to minimize a specific loss function,

such as Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE), between the predicted values $\hat{Y}_{t+1:t+H}$ and the true future values $Y_{t+1:t+H}$. The success of the model is therefore measured by its ability to provide the most accurate single-point prediction for each future hour.

Conversely, the problem of probabilistic forecasting is more comprehensive. Instead of a single point estimate, the objective is to provide a predictive probability distribution over the future values, $P(Y_{t+1:t+H}|Y_{1:t}, X_{1:t+H})$. This distribution quantifies the uncertainty associated with the forecast, which is crucial for risk management and decision-making in the energy sector. A robust probabilistic forecast not only captures the most likely outcome but also rules that assess the sharpness and calibration of the entire predictive distribution, such as the Continuous Ranked Probability Score (CRPS). The CRPS measures the distance between the predictive distribution and the empirical distribution of the observed outcome, with a lower score indicating a better forecast.

The exogenous features, particularly those related to weather, are integral to this study. The vector x_t includes features such as air temperature, wind speed, wind direction, humidity, and cloud cover. These variables are known to directly influence both electricity demand (e.g. temperature for heating and cooling) and supply (e.g. wind speed for wind power generation, cloud cover and hours of sunshine for solar power generation). The use of these features, both historical and forecasted, is a critical component of the conditioned models in this comparative analysis. The distinction between unconditioned models, which rely solely on historical prices $Y_{1:t}$, and conditioned models, which incorporates these exogenous features, is central to the research objectives and will be a key point of comparison in the results.

For the practical implementation and evaluation of the models, we adopt a rolling-window approach. The dataset is partitioned into training, validation, and test sets. During training, the models learn the underlying patterns and dynamics from a fixed window of historical data. The validation set is used for hyperparameter tuning and model selection. The final performance evaluation is conducted on a dedicated test set using a rolling-window mechanism, where the models are repeatedly trained on a block of historical data and then used to forecast the next H hours. The window then slides forward, and the process is repeated. This methodology ensures that the evaluation reflects a realistic forecasting scenario, where the models are consistently challenged with unseen, future data.

In summary, the problem is not merely to predict a single value, but to understand the complex temporal dynamics of a non-stationary time series and the influence of external factors. The methodology detailed in the subsequent sections is designed to address this problem from both a point forecasting and a probabilistic forecasting perspective, providing a comprehensive framework for a fair and rigorous comparison of the chosen deep learning architectures.

3.2 Model Architectures

This section provides a detailed description of the three primary model architectures under investigation: the Gated Recurrent Unit (GRU), the Structured State Space Model (S4), and the diffusion-based WaveNet. For each architecture, we will implement and analyze both an unconditioned variant, which forecasts based solely on historical price data, and a conditioned systematic assessment of the intrinsic capabilities of each model as well as their capacity to leverage external information. The following subsections will cover the specific details of each model, starting with the GRU baseline.

3.2.1 Baseline GRU (Conditioned and Unconditioned)

The Gated Recurrent Unit (GRU) [3] serves as a robust and widely-accepted baseline for this study. As a type of Recurrent Neural Network (RNN), the GRU is specifically designed to process sequential data and effectively mitigate the vanishing gradient problem that plagues traditional RNNs, enabling it to capture dependencies over longer time spans. Its architecture is a simplified version of the Long-Short-Term Memory (LSTM) network, offering a balance of performance and computational efficiency. This makes it an ideal choice for a competitive baseline model against which more recent and complex architectures can be fairly judged.

The core of the GRU architecture is the GRU cell, which processes input at a single time step t and updates a hidden state vector h_t . This update is regulated by two primary gating mechanisms: the update gate z_t and the reset gate r_t . The purpose of these gates is to control the flow of information, allowing the model to selectively remember or forget past information. The update gate determines how much of the previous hidden state, h_{t-1} , should be carried forward to the current hidden state, while the reset gate controls how much of the past information is considered when computing the new candidate hidden state.

The mathematical formulation of the GRU cell is as follows: Given an input vector x_t at time t and the hidden state from the previous time step h_{t-1} , the gates are computed as:

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \quad (3.2)$$

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \quad (3.3)$$

where σ is the sigmoid activation function, and W and b are the learnable weight matrices and bias vectors, respectively. The update gate z_t and the reset gate r_t produce values between 0 and 1, which act as filters for the information flow.

Next, a candidate hidden state \tilde{h}_t is computed. This candidate state is a potential new hidden state for the current time step, and its computation involves the reset gate:

$$\tilde{h}_t = \tanh(W_{x\tilde{h}}x_t + W_{h\tilde{h}}(r_t \odot h_{t-1}) + b_{\tilde{h}}) \quad (3.4)$$

Here, \odot denotes the element-wise product. The reset gate r_t multiplies the previous hidden state h_{t-1} , essentially allowing the model to "reset" or "forget" parts of the past state that are deemed irrelevant for the current prediction.

Finally, the new hidden state h_t is calculated by combining the previous hidden state and the new candidate hidden state using the update gate:

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \quad (3.5)$$

The update gate z_t determines the convex combination of the old state and the new candidate state. If z_t is close to 1, the model mostly retains the previous hidden state, effectively "remembering" past information. If z_t is close to 0, it primarily updates hidden state with the new candidate state, effectively "forgetting" the past. The mechanism provides the GRU with its powerful capacity to manage long-term dependencies.

The overall architecture for the GRU-based forecasting model is a many-to-many sequence model. The network consists of a stack of GRU layers, where the output of one layer at each time step serves as the input to the next. Stacking GRU layers increases the model's capacity to learn complex, hierarchical features from the time series. The final hidden state of the last GRU layer

is then passed through one or more dense layers to produce the forecast. For this study, the models are configured to predict all H future time steps simultaneously, using the output of the final GRU layer at the end of the lookback window.

The GRU is implemented in two variants to serve as a comprehensive baseline:

1. **Unconditioned GRU:** In this variant, the input vector x_t to the GRU cell at each time step consists solely of the historical energy price at that time y_t . The model's entire forecasting power is derived from learning the intrinsic temporal patterns, seasonality, and dynamics of the price series itself, without any external information. This model provides an important benchmark for the inherent predictability of the time series data.
2. **Conditioned GRU:** This variant incorporates exogenous weather features into the forecasting process. The input vector to the GRU cell at each time step t is a concatenation of the historical energy price and the vector of weather features, i.e. $x_t = [y_t, \text{weather_features}_t]$. The architecture remains the same, but the model now learns a more nuanced relationship between prices and external drivers. This model serves as the primary benchmark for the conditioned S4 and diffusion models, allowing for a fair comparison of how each architecture leverages the same set of external information.

Both variants will produce a point forecast by passing the final GRU hidden state through a feedforward network with a linear activation function. This implementation adheres to a standard and effective methodology for deep learning-based time-series forecasting, making it a reliable and strong foundation for the comparative analysis.

3.2.2 S4 Model (Conditioned and Unconditioned)

The Structured State Space Model (S4), as a modern and efficient deep sequence modeling paradigm, is a central component of this comparative thesis. It is implemented here to serve as a high-performance alternative to the Gated Recurrent Unit (GRU) baseline, specifically designed to address the challenges of modeling long-range dependencies in time-series data without the quadratic computational cost associated with attention-based models [8]. This section covers the specific architecture and implementation of the S4 model for hourly energy price forecasting, in both its conditioned and unconditioned variants.

The foundation of our S4-based forecasting model is the S4 block, which acts as a fundamental deep sequence layer. As discussed in Chapter 2, the S4 block is rooted in a continuous-time linear time-invariant (LTI) state-space model, which is then transformed to produce a recurrent and a convolutional formulation. For the purpose of training and efficiency, our implementation leverages the convolutional form, which allows for highly parallelized computation. The core of the S4 model lies in learning the parameters of the state-space matrices A, B, C, D such that the resulting convolutional kernel, derived from these matrices, is optimized for the forecasting task. A key aspect of the S4 model’s power is its use of a well-founded initialization for the A matrix, such as the HiPPo family [12], which provides the model with an inherent capacity to represent and compress a long history of a sequence into a fixed-size state, thus preserving information over extended time periods.

Our overall S4 model architecture for forecasting is composed of several key components structured in a sequential manner. The input sequence, representing a lookback window of historical data, first passes through a linear embedding layer. The output of this layer is then fed into a stack of S4 blocks. Each S4 block internally consists of the S4 layer itself, followed by a non-linear activation (such as GeLU), and a residual connection to facilitate stable training of deep networks [8]. This stacking of S4 blocks allows the model to learn a hierarchy of features, from local temporal patterns to more abstract, long-range dependencies. The final output of the last S4 block, which has processed the entire input sequence, is then passed to a feedforward network (also known as the output head). This output head consists of a series of linear layers with non-linear activations, which ultimately produce the forecast for the entire H -hour horizon. This many-to-many architecture allows the model to predict the entire future sequence in a single forward pass, a common and efficient practice in time-series forecasting.

To conduct a fair and thorough comparison with the GRU baseline and the DiffWave model, we implement the S4 architecture in two distinct variants:

1. **Unconditioned S4 Model:** The unconditioned variant of the S4 model is designed to assess its intrinsic capacity to learn the temporal dynamics of the energy price series without the aid of any external information. The input to the model consists solely of the sequence of historical hourly energy prices, $Y_{1:t} = (y_1, y_2, \dots, y_t)$. This one-dimensional input sequence is first passed through a linear embedding layer to project it into a higher-dimensional space where the S4 blocks can operate. The subsequent stack of S4 blocks then processes this sequence, and the output head produces

the forecast $\hat{Y}_{t+1:t+H}$. The performance of this model is a direct test of the S4 architecture’s ability to capture the complex seasonality and long-term trends embedded within the price data alone. It provides a valuable data point for understanding the inherent predictability of the series.

2. **Conditioned S4:** The conditioned S4 model is a more comprehensive variant that integrates a crucial set of exogenous weather features into the forecasting process. The input to this model is a concatenation of the historical energy price series and the corresponding weather feature vectors, forming a multivariate time series. Specifically, at each time step t , the input vector is $x_t = [y_t, \text{weather_features}_t]$, where the weather features include air temperature, wind speed, wind direction, humidity, and cloud cover. To accommodate this multivariate input, the initial linear embedding layer is adjusted to project this higher-dimensional vector into the hidden dimension of the S4 blocks. The core S4 architecture remains the same, but the model is now trained to leverage the correlations between weather patterns and energy prices. This variant allows us to evaluate the S4 model’s effectiveness in integrating and learning from a richer set of input features. By comparing the performance of the conditioned S4 with its unconditioned counterpart, we can quantify the value of incorporating weather information within this specific architectural framework, and by extension, understand how well S4 is suited for forecasting tasks with external drivers.

Both S4 variants are configured with a similar number of layers and hidden dimensions as the GRU baseline to ensure a fair comparison of architectural performance rather than simply model size. The final output layer for both models is a linear layer that maps the hidden state representation from the last S4 block to the forecasted hourly energy prices. This approach allows for a direct comparison of point forecasting accuracy, which is a key objective of this thesis. The S4 models’ exceptional efficiency in handling long sequence lengths positions them as a highly promising candidate for this task, and their practical utility for real-world energy market applications.

3.2.3 Diffusion-based WaveNet (Conditioned and Unconditioned)

In contrast to the deterministic point forecasting nature of the GRU and S4 models, the diffusion-based WaveNet architecture serves as the primary tool in this study for investigating the probabilistic forecasting capabilities for hourly

energy prices. This model belongs to the class of generative deep learning models and is fundamentally designed to learn the entire data distribution rather than a single point estimate. As already mentioned in the related work, diffusion models operate by learning to reverse a gradual noise injection process, which allows them to generate diverse and realistic samples that represent the possible future states of the time series. This section details the specific implementation of this architecture and its two variants for the forecasting task.

The foundational principle of our diffusion-based forecasting model is its core denoising network, which, in this study, is an adapted WaveNet architecture. The WaveNet, with its dilated causal convolutions, is particularly well-suited for this task because it can efficiently model complex, multi-scale temporal dependencies without suffering from the recurrent bottlenecks of GRUs or the quadratic complexity of standard attention mechanisms [17]. The dilated convolutions allow the model to have a wide range of past time steps, a property that is crucial for capturing the various seasonalities in energy price data.

The overall forecasting framework is built upon the Denoising Diffusion Probabilistic Model (DDPM) paradigm, adapted for time series [9, 3]. The model’s training process involves a forward diffusion process and a learned reverse process. The forward process is a fixed Markov chain that progressively adds Gaussian noise to a future price trajectory, $y_{t:t+H}$, until it becomes pure noise. The reverse process is what the model learns: a neural network, parametrized by θ , which predicts the noise to be removed from a noisy sample at a given time step. Specifically, our WaveNet network, f_θ , which predicts the noise to be removed from a noisy sample at a given time step. Specifically, our WaveNet network, f_θ , is trained to predict the noise ϵ that was added to a sequence at time step k , conditioned on the noisy sequence itself (y_k), the time step k , and the observed historical data ($y_{1:t}$). The objective is to minimize the difference between the predicted noise and the actual noise, enabling the model to learn the inverse of the diffusion process.

The forecasting process with this model is generative and is conducted through a sampling procedure. To produce a forecast, we first generate a purely random sequence of noise, $\epsilon \sim \mathcal{N}(0, I)$, with the same length as the forecast horizon, H . The noisy sequence is then iteratively denoised by the trained WaveNet network over a number of discrete time steps. At each step, the network predicts the noise to remove, thereby gradually transforming the random noise into a coherent and plausible forecast trajectory. To produce a probabilistic forecast, this process is repeated multiple times (e.g., 1000 times), yielding a collection of diverse and distinct sample trajectories. This collection of sam-

ples constitutes the predictive distribution from which key insights, such as the mean forecast, prediction intervals, and quantiles, can be derived.

To facilitate a comprehensive comparative analysis, the diffusion-based WaveNet model is implemented in two variants, mirroring the GRU and S4 architectures:

1. **Unconditioned Diffusion-based WaveNet:** In this variant, the WaveNet denoising network is conditioned exclusively on the historical energy price series within the lookback window. The model learns to generate future price trajectories based only on the intrinsic patterns and dynamics of the price data itself. The input to the WaveNet is the noisy future sequence and the historical price series, which is encoded and provided as conditioning information. The performance of this model will provide a direct assessment of the generative power of the diffusion framework for probabilistic in the absence of external drivers. It will serve as a valuable reference point for understanding how well the model can capture the complex, multi-modal distribution of energy prices.
2. **Conditioned Diffusion-based WaveNet:** This more complete variant incorporates exogenous weather features into the forecasting process. The WaveNet denoising network is conditioned on both the historical energy price series and the corresponding sequence of weather features. These features, such as air temperature, wind speed, and humidity are encoded and integrated into the model as part of the conditioning context. By providing this additional information, the model is able to learn a more nuanced and accurate predictive distribution, as it can account for the influence of key external drivers on price formation. The comparison between the conditioned and unconditioned diffusion models will allow for a clear quantification of the value of weather information within a generative, probabilistic forecasting framework. This is a critical component of the thesis, as it investigates whether the superior probabilistic forecasting capabilities of diffusion models are further enhanced by the inclusion of relevant exogenous variables.

The final output of these models is a distribution of possible future outcomes. While a point forecast can be derived (e.g., the mean or median of the samples), the true strength lies in the rich probabilistic information it provides. The evaluation of this model will therefore focus on both point forecasting metrics and, and more importantly, on probabilistic metrics like the Continuous Ranked Probabilistic Score (CRPS), which will allow us to assess the quality of the entire predictive distribution in a manner that is distinct from

the evaluations of the GRU and S4 models. The implementation of this model ensures that our study provides a fair and comprehensive comparison between deterministic and probabilistic forecasting paradigms.

3.3 Training and Optimization

A rigorous and consistent training and optimization protocol is paramount for ensuring a fair and meaningful comparison across the diverse model architectures investigated in this thesis. This section details the procedures followed for training each model, including the dataset partitioning, the choice of optimizer, loss functions, and the hyperparameter tuning strategy. These methods are designed to produce a robust set of final models whose performance can be confidently evaluated and compared.

The dataset, comprising historical hourly energy prices and exogenous weather features, is partitioned into three distinct sets: a training set, a validation set, and a test set. This partitioning is conducted chronologically to ensure that the models are always evaluated on future, unseen data, which is a critical requirement for time-series forecasting. The training set is used to optimize the model parameters, while the validation set is used for hyperparameter tuning and early stopping to prevent overfitting. The final held-out test set is used only once, at the end of the research, to provide an unbiased estimate of the models' generalization performance. To ensure a realistic forecasting scenario, a rolling-window approach is adopted for both validation and testing. This process involves training a model on a fixed-size historical window and then forecasting the next H hours. The window is then advanced by a fixed step size, and the process is repeated. This methodology accounts for the non-stationary nature of the data and provides a more reliable performance metric than a single train-test split.

For the optimization of all models, we employ the AdamW optimizer [?]. AdamW is a variant of the popular Adam optimizer that decouples weight decay from the gradient update, which has been shown to improve the generalization performance of deep learning models. This is particularly important for models with a large number of parameters, such as the S4 and the diffusion models, as it helps to prevent overfitting and encourages the discovery of more generalizable solutions. To manage the learning rate effectively, we use a *ReduceLROnPlateau* scheduler. THis dynamic learning rate strategy monitors a specific metric on the validation set, typically the loss, and reduces the learning rate by a specified factor if the metric does not improve for a

certain number of epochs. This helps the model converge to a better solution by taking smaller steps in the loss landscape once it approaches a minimum.

The choice of loss function is dependent on the nature of the forecasting problem being solved by each model. For the deterministic models, namely the GRU and S4 architectures, the objective is to minimize the error between the single-point prediction and the true value. For this purpose, we use both the *Mean Squared Error* (MSE) and the *Mean Absolute Error* (MAE). MSE, defined by $\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$ is often preferred in practice as it is differentiable and strongly penalizes large errors, a desirable property for tasks where significant deviations are costly. However, for training, we primarily rely on MAE as the loss function. MAE is defined as $\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$, and its primary advantage is that it is less sensitive to extreme outliers, which are a common feature of energy price data. This makes it a robust choice for minimizing the average forecast error without over-penalizing the model for struggling with rare, severe price spikes. The models are trained to minimize the MAE over the entire forecast horizon H .

For the probabilistic diffusion-based WaveNet model, the training objective is fundamentally different. Instead of a single point forecast, the model learns to approximate the conditional data distribution by minimizing a specific loss function related to the forward and reverse diffusion processes. The training objective for a Denoising Diffusion Probabilistic Model (DDPM) is a simplified version of the variational lower bound (VLB), which can be reduced to a simple L2 loss on the noise [9]. Specifically, the model is trained to predict the noise ϵ that was added to a data sample at a given time step t , conditioned on the noisy sample itself. The loss function is given by:

$$L = \|\epsilon - f_\theta(y_t, t, \text{context})\|_2^2 \quad (3.6)$$

where f_θ is our WaveNet denoising network, y_t is the noisy data sample, t is the current diffusion time step, and "context" refers to the historical data and exogenous features. For this study, we use a linear diffusion schedule, where the noise variance is increased linearly over the diffusion steps. Minimizing this loss enables the model to effectively learn the reverse diffusion process and, consequently, the conditional data distribution, which is the foundation for generating high-quality probabilistic forecasts.

Finally, we employ early stopping as a primary form of regularization for all models. This technique involves monitoring the validation loss at the end of each training epoch. If the validation loss fails to improve for a predefined

number of epochs (the patience), training is halted to prevent the model from overfitting to the training data. This ensures that the final model selected is the one with the best generalization performance on the unseen validation set, which is a key component of our humble and robust methodology.

3.4 Evaluation Metrics

A comprehensive evaluation approach is essential for accurately assessing the performance of the models in this thesis. We utilize a range of metrics tailored to point forecasting to provide a complete and nuanced picture of the strengths and weaknesses of the GRU and S4 models. This section outlines the specific metrics used and explains their relevance to the problem of hourly energy price forecasting.

For point forecasting, we have to focus regarding the evaluation on the accuracy of their single-point predictions. We use four of the most common and robust metrics for this task: the *Mean Squared Error* (MSE), the *Root Mean Squared Error* (RMSE), the *Mean Absolute Error*, and *Directional Accuracy* (DA).

The MSE is a quadratic scoring rule that measures the average of the squared errors. It is defined as

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.7)$$

where n is the number of observations, y_i is the actual value, and \hat{y}_i is the predicted value. A lower MSE indicates a better-performing model. The key characteristic of MSE is that it penalizes larger errors disproportionately more than smaller ones. This can be beneficial in applications where large forecast errors are particularly costly.

The RMSE is the square root of the MSE and is a widely used metric because it is in the same units as the target variable, making it more interpretable than MSE. It is defined as

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (3.8)$$

Like the MSE, a lower RMSE indicates a better-performing model. The key difference from MAE lies in its sensitivity to large errors. By reporting RMSE, we gain insights into the models' performance on typical and extreme price fluctuations. A model with a low MAE but a high RMSE might be performing well on average but struggling with the occasional price spike.

The MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It is defined as:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3.9)$$

The MAE is a straightforward and interpretable metric that provides a direct measure of the average forecast error. Its primary advantage is that it is less sensitive to extreme outliers than the MSE and RMSE, which is particularly relevant for energy price forecasting where sudden, sharp price spikes can occur. A lower MAE indicates a more accurate model on average.

Directional accuracy is a classification-based metric that measures the percentage of correctly predicted directional movements in the time series. For a time series y_t , a correct prediction is one where the sign of the predicted change matches the sign of the actual change. It is defined as:

$$DA = \frac{1}{n} \mathbb{I}[(\hat{y}_i - y_{i-1})(y_i - y_{i-1}) > 0] \quad (3.10)$$

where $\mathbb{I}[\cdot]$ is the indicator function. The DA is a valuable metric in a financial context like energy markets, where the ability to correctly predict the direction of a price movement (up or down) can be as important as predicting the exact price value. This metric provides a different perspective on model performance, assessing its ability to capture the qualitative dynamics of the series.

By employing this combination of point forecasting metrics, we ensure that our evaluation is comprehensive. The models will be judged on their ability to minimize forecasting errors on average (MAE), their sensitivity to large errors (RMSE), and their capacity to capture the directional movements of the price series (DA). The comparison of these metrics across all models will form the foundation of our results and conclusions.

Chapter 4

Implementation

The successful execution of this machine learning research project relies not only on a robust theoretical foundation but also on a thorough and transparent implementation process. While Chapter 3 covered the methodological blueprint of this thesis, this upcoming chapter shifts the focus to the practical realization of that plan. It is here that we bridge the gap between abstract concepts and tangible code, ensuring that the research is not only theoretically possible but also reproducible in practice. This section provides a comprehensive overview of the technical environment, from the data that serves as the foundation of our models to the computational infrastructure used to train and evaluate them.

A key objective of this chapter is to document the entire workflow with sufficient detail to allow other researches to replicate our findings. We approach this task with a sense of humility, acknowledging that every implementation choice, from data sourcing to model deployment, can influence the final results. Therefore, we provide explicit details on the datasets used, their specific features, and the preprocessing steps applied. The importance of data cannot be overstated; it is the raw material from which our models learn, and its quality and characteristics directly impact model performance. By thoroughly describing the data, we provide essential context for the reader to understand the models' performance.

The chapter is structured to follow the logical flow of the implementation process. We begin with a detailed description of the datasets, which were aggregated from authoritative public sources. Following that is the section on data preprocessing, where the cleaning and normalizing and feature engineering is explained. Finally, we provide a brief overview of the computational resources employed, as the scale of the models and the volume of the data necessitate

significant processing power.

4.1 Dataset Description

The cornerstone of any data-driven forecasting task is a high-quality, comprehensive dataset that accurately reflects the dynamics of the system being modeled. For this research on hourly energy price forecasting in Germany, a combined dataset was meticulously constructed from two distinct sources: hourly day-ahead energy prices and hourly weather features. The temporal alignment and combination of these two data streams were critical to creating a rich feature set for our forecasting models. The final dataset spans a significant period, from January 1, 2015, to April 27, 2025, providing a robust and extensive time series for training and evaluation.

Energy Price Data

The hourly energy price data for Germany was sourced from the *European Wholesale Electricity Price Data* dataset, a publicly available resource provided by Ember, a global energy think tank [6]. This data is compiled from reputable sources such as ENTSO-E and provides average hourly day-ahead spot prices for various bidding zones, including Germany. The price, measured in Euros per megawatt-hour (*EUR/MWh*), represents the outcome of a competitive auction process where supply and demand bids for electricity are matched to determine the market-clearing price for each hour of the following day. This metric is the primary target variable for our forecasting models. The nature of this data is characterized by its inherent volatility, seasonality, and dependence on both fundamental market factors and external variables, which underscores the complexity of the forecasting problem.

Weather Feature Data

Exogenous weather variables are widely recognized as critical predictors for energy price and demand forecasting, as they directly influence both consumption and renewable generation (e.g., wind and solar). The weather data used in this study was obtained from the Deutsche Wetterdienst (DWD), Germany's national meteorological service, via their open data portal [5]. This source provides authoritative and high-resolution meteorological observations from a network of stations across Germany.

To provide a detailed picture of the ambient conditions, a selection of key weather features was extracted and integrated with the energy price data.

These features, as seen in the final dataset [5], include:

1. **Air Temperature (TT_TU):** Measured in degrees Celsius, air temperature is a primary driver of heating and cooling demand, and thus a fundamental variable for energy consumption forecasting.
2. **Relative Humidity (RF_TU):** Relative humidity, expressed as a percentage, influences perceived temperature and can affect the efficiency of certain generation technologies.
3. **Sunshine Duration (SD_SO):** Measured in minutes per hour, this variable serves as a proxy for solar irradiation, which is crucial for forecasting solar power generation.
4. **Wind Speed (FF) and Wind Direction (DD):** Wind speed, measured in meters per second, is a key determinant of wind power generation, while wind direction provides additional context for regional wind patterns.

By including these variables, we aim to capture the most significant weather-related influences on the energy market. The hourly granularity of both the price and weather data ensures a precise temporal alignment, which is a prerequisite for effective time-series analysis and forecasting.

Combined Dataset Structure and Preprocessing

The two datasets, once obtained, were merged based on their shared hourly timestamps to create a single, unified dataset for our modeling work. The final dataset, as represented in the provided file, includes columns for *Datetime (UTC)* and *Datetime (Local)*, ensuring timezone awareness for accurate temporal analysis. The *Price (EUR/MWh)* column serves as our target variable, while the DWD features (*TT_TU,RF_TU,SD_SO,FF,DD*) act as the exogenous predictors. It is important to note that the raw dataset also contains other variables (**V_N,V_S1_CS,etc.**) which were not used for the final model and are beyond the scope of this work [5].

Prior to modeling, a crucial step involved handling missing values and data inconsistencies. Given the public nature of the data sources, occasional gaps or anomalous readings are to be expected. These issues were addressed through standard time-series data imputation techniques, such as linear interpolation, to maintain the continuity of the series without introducing significant bias. The complete and preprocessed dataset provides the foundation for the training, validation, and testing phases of our research, ensuring that all models

are exposed to a consistent and representative view of the energy market and its key drivers.

4.2 Preprocessing & Feature Engineering

The raw data collected from various sources, while rich in information, is not immediately suitable for training deep learning models. Preprocessing and feature engineering are indispensable steps that transform this raw data into a format that is both interpretable by the models and optimized for learning. This section details the specific techniques applied to the energy price and weather data to ensure a robust and effective training process.

The initial and most critical step in our preprocessing pipeline was the *data normalization* of all numerical features. Normalization is a critical procedure for neural networks, as it helps to stabilize training by bringing all input values into a consistent range. When features have widely varying scales, the gradient descent optimization process can become inefficient, as the model's weights might struggle to converge, leading to slow and unstable training. To address this, I employed distinct normalization strategies based on the nature of the data. The energy price data, which serves as both an input features and the primary target variable, was normalized using *MinMaxScaler* from the scikit-learn library. This scaler rescales the data to a fixed range, typically [0,1], which is particularly suitable for time-series data as it preserves the shape of the original distribution while preventing extreme values from dominating the training process. This approach was consistently applied across all models.

For the conditioned models, which incorporate exogenous weather features, a different approach was necessary. These features - such as temperature, wind speed, humidity - have distinct physical units and scales. To ensure that no single feature disproportionately influences the model's learning, they were normalized using *StandardScaler*. This scaler transforms the data to have a mean of 0 and a standard deviation of 1. This method is particularly effective for features with a Gaussian-like distribution and is widely regarded as a best practice for preparing multivariate data for neural network training. The implementation of both scalers for the conditioned models is documented in the source code of the respective models.

Beyond simple normalization, we applied *feature engineering* to create additional variables that encode important temporal information. While deep learning models can learn complex patterns, providing explicit temporal fea-

tures can significantly improve their ability to capture daily, weekly, and yearly seasonality. We engineered features such as *hour of day*, *day of week*, *day of year*, *month of year*. These features were represented as numerical values and were normalized along with the weather data. This process, as documented in the dataset preparation scripts, provides the models with a structured understanding of time, which is fundamental to energy forecasting. The *time_features* variable, seen in *PriceDataset* class, underscores the importance of this step. Furthermore, the inclusion of meteorological data is supported by academic research, which has shown that such data significantly improves electricity price forecasts beyond the day-ahead horizon by providing valuable information on factors influencing demand and renewable generation [21]. With that, I created two versions of each model architecture - an unconditioned version and a conditioned version with weather and temporal features - for a direct comparison of their impact on performance.

Finally, a crucial part of our preprocessing methodology was the *chronological data splitting*. Unlike typical machine learning problems where data can be randomly shuffled, time-series forecasting requires that the training, validation, and test sets are partitioned chronologically. This prevents data leakage and ensures that the models are only ever evaluated on future, unseen data. We partitioned the dataset into a training set for model parameter optimization, a validation set for hyperparameter tuning and early stopping, and a final test set for unbiased performance evaluation. The *HISTORY_LEN* (168 hours) and *PRED_LEN* (96 hours) parameters, consistently defined across all model scripts, govern the size of the lookback and forecast windows, a key part of the rolling-window evaluation strategy discussed in Chapter 3. These meticulous preprocessing and feature engineering steps are foundational to the robust and fair comparison of the model architectures in this study.

4.3 Computational Setup

The complexity and scale of the deep learning architectures employed in this thesis, coupled with the large volume of hourly time-series data, necessitated a robust computational infrastructure. This section provides an overview of the hardware and software environment used to train and evaluate the models, ensuring transparency and providing context for the observed performance metrics.

The entire deep learning pipeline was built using the *PyTorch framework* [18]. PyTorch was selected for its flexibility, dynamic computation graph, and ex-

tensive support for GPU acceleration, which are essential for training complex neural networks. All scripts, including those for the GRU, S4 and diffusion models, leverage PyTorch’s core functionalities for tensor operations, automatic differentiation, and model construction. The intuitive and Python-native nature of PyTorch allowed for rapid prototyping and fine-tuning of the complex model architectures, such as the diffusion-based WaveNet, which requires a custom training loop.

For computationally intensive tasks, such as model training and inference, we utilized *GPU acceleration*. The scripts explicitly check for the availability of a CUDA-enabled GPU and, if found, use it for all computations ($DEVICE = 'cuda'$). The models were trained on a single machine equipped with a modern NVIDIA GPU. This setup significantly reduced training times and allowed for the exploration of larger model configurations and longer training runs, which would have been prohibitively slow on a CPU. This is a standard and necessary practice in contemporary deep learning research, particularly for models with a large number of parameters like the S4 and diffusion architectures.

The codebase for this thesis was developed using a set of well-established Python libraries. *Pandas* was used for all data loading, manipulation, and chronological partitioning, leveraging its powerful DataFrame structure. *NumPy* was employed for numerical operations and array-based computations. The *Scikit-learn* library provided essential tools for data preprocessing, specifically the *MinMaxScaler* and *StandardScaler* used for feature normalization, and for calculating standard evaluation metrics. For model-specific architectures, I integrated external libraries, such as *s4torch* for the S4 models and custom implementations for the diffusion-based WaveNet model. Progress bars from the *tqdm* library were used to monitor training and evaluation loops, providing clear and continuous feedback on the progress of our experiments.

The computational demands of the project were defined by a set of consistent hyperparameters across all models. Each model was trained using a batch size of 64, which was a practical choice for balancing memory usage and training stability. A lookback window of 168 hours (one week) and a forecast horizon of 96 hours (four days) were used, defined by *HISTORY_LEN* and *PRED_LEN*. The total number of training epochs was set to a large number (e.g. 300 or 1000), to allow for sufficient training time, with *early stopping* being the primary mechanism for preventing overfitting and selecting the best-performing model based on validation loss. This strategy is critical for avoiding the selection of an overfit model that performs well on the training data but poorly on unseen validation and test data. The patience values for early stopping,

ranging from 60 to 80 epochs, were chosen to give the models time to find a good minimum in the loss landscape. This approach ensures that our final results are not a product of chance but rather a result of a methodical and transparent implementation process.

4.4 Reproducibility Considerations

In scientific research, especially within the domain of machine learning and deep learning, the concept of reproducibility is important. Reproducibility refers to the ability of obtaining the same results as a previous study by using the same code, data, and methodology. Ensuring reproducibility is not merely a technical formality; it is a fundamental pillar of the scientific method, enabling independent verification, fostering trust in the findings, and allowing the community to build upon the work with confidence. This section outlines the specific measures and *a priori* considerations taken throughout this study's implementation to maximize the reproducibility of our experiments and results.

A core principle of our work was to make the computational environment and the underlying code as transparent and accessible as possible. All models were developed using a consistent set of open-source libraries, primarily the *PyTorch* framework, along with standard data science packages such as *Pandas* and *NumPy*. The explicit use of these widely adopted libraries, whose versions can be specified, ensures that the code can be executed on a diverse range of machines with minimal dependency issues. The entire codebase is provided in the form of dedicated Python scripts for each model architecture, including both the unconditioned models and their conditioned counterparts. Each script is self-contained and includes all necessary imports, configuration parameters, and the main training and evaluation loops. This structure allows a third-party observer to directly examine and run the exact same experiments conducted for this thesis.

The management of data and features was also approached with determinism in mind. The raw dataset, a prerequisite for the entire pipeline, was sourced and preprocessed using a single, consistent methodology. The chronological splitting of the data into training, validation, and test sets is a particularly critical aspect of reproducibility for time-series forecasting. Unlike randomized splits that can yield slightly different data subsets with each run, our approach ensures that the exact same temporal periods are used for each phase of the experiment, thereby eliminating a significant source of variability. The data normalization process, utilizing *MinMaxScaler* for prices and *StandardScaler*

for exogenous features, was also applied deterministically, with the scalers being fit only on the training data and then used to transform all subsequent datasets, preventing data leakage and ensuring a fair comparison. The consistency of these steps is visible in the provided code, where parameters like *HISTORY_LEN* and *PRED_LEN* are hardcoded at the beginning of each script to define the precise window sizes for all experiments.

Perhaps the most challenging aspect of reproducibility in deep learning is managing the inherent randomness that can arise from various sources, including weight initialization, data shuffling, and certain GPU operations. To address this, we implemented a strict policy of setting a *global random seed* at the beginning of each script. By setting seeds for Python’s built-in *random* module, NumPy, and PyTorch, we fix the state of all pseudo-random number generators. This ensures that operations such as initial parameter weights and the order of data batches within an epoch are identical on every run, provided the same hardware and software versions are used. While minor differences may still arise due to floating-point arithmetic on different hardware, this practice significantly reduces variability and allows for the replication of results with a high degree of confidence. The code snippets at the start of each file, demonstrating the setting of these seeds, serve as an explicit guarantee of this effort.

Furthermore, the consistency of our experimental protocol extended to hyperparameter management and model persistence. All key hyperparameters, such as batch size, learning rate, and patience for early stopping, are explicitly defined as global constants within the scripts. This eliminates ambiguity and ensures that a person attempting to reproduce the results does not need to guess at the values used. The model training process itself was designed to be deterministic through the use of *checkpoints*. Instead of simply running for a fixed number of epochs, our scripts employ an early stopping mechanism that saves the best-performing model based on its performance on the validation set. This means that the final model state and the metrics reported on the test set are not tied to a specific number of epochs but rather to the point of optimal performance, which is a more stable and meaningful measure. The path to the saved checkpoint (*CHECKPOINT_PATH*) is also a fixed parameter in the scripts, allowing for the easy retrieval and re-evaluation of the final models.

In conclusion, every facet of the experimental design, from the selection of open-source frameworks and the structured organization of the codebase to the meticulous management of data splits, random seeds, and hyperparameters, was undertaken with a strong commitment to reproducibility. While it is challenging to guarantee identical results across all possible hardware and

software configurations, we have taken a methodical and transparent approach to eliminate the most common sources of non-determinism. We are confident that these measures provide a solid foundation for independent verification and pave the way for the upcoming chapter, where we discuss the results and outputs of all models and compare their respective efficiency.

Chapter 5

Results

The purpose of this chapter is to present and analyze the outcomes of the empirical study described in the preceding chapters. We will evaluate the performance of the three deep learning architectures - the Gated Recurrent Unit (GRU), the Structured State Space Model (S4), and the Diffusion WaveNet - all on their unconditioned form and with the integration of exogenous weather and time features. This comprehensive analysis will allow us to draw substantiated conclusions regarding the models' predictive accuracy, their computational efficiency, and the specific contribution of the conditioning variables. The findings are not merely a collection of numerical results but a direct test of the hypotheses laid out in the introduction. A core motivation of this work was to move beyond traditional autoregressive models and explore more sophisticated architectures that can better capture the intricate, non-linear dynamics of energy price data, while also investigating the value of incorporating physically-relevant exogenous information. The results presented will serve as the empirical foundation for the discussion on the models' strengths and weaknesses and their practical implications for the energy market in the subsequent chapters. This chapter is structured to provide a clear and logical progression of findings: Section 5.1 provides a detailed performance comparison based on key forecasting metrics, Section 5.2 delves into the isolated impact of weather features by comparing the conditioned and unconditioned models, and Section 5.3 discusses the computational efficiency of each model.

5.1 Performance Comparison

This section provides a direct, quantitative, and qualitative comparison of the six models evaluated in this study. The performance of each model on the held-out test set is assessed using a suite of standard forecasting metrics, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE)

and Directional Accuracy (DA). A summary of these metrics is presented in Table 5.1, followed by a detailed discussion of the key findings.

Model	Parameters	MSE	MAE (EUR/MWh)	RMSE (EUR/MWh)	DA (%)
GRU	62,688	3577.00	45.28	56.19	76.87
GRU (conditional)	45,600	3069.95	39.55	50.91	71.97
S4	568,272	2885.76	39.54	49.56	68.68
S4 (conditional)	460,848	2484.70	34.89	49.85	76.86
DiffWave	568,777	3443.70	41.69	58.68	50.47
DiffWave (conditional)	564,256	2558.05	35.45	50.58	80.07

Table 5.1: Summary of Model Performance on the Test Set

As the results in Table 5.1 indicate, the models exhibited a range of performance characteristics, with clear leaders emerging based on the chosen metrics. When evaluating strictly by predictive accuracy, measured by MAE and RMSE, the *S4 model with weather conditioning* demonstrated the lowest Mean Absolute Error (34.89 EUR/MWh) and the lowest Root Mean Squared Error (49.85 EUR/MWh). This is a significant finding as it suggests that the generative and stochastic nature of the structured state space model, when provided with relevant exogenous features, allows it to produce the most accurate point forecasts. The unconditioned S4 model also performed exceptionally well, with an MAE of 39.54 EUR/MWh and a virtually identical RMSE of 49.56 EUR/MWh. Both of these models significantly outperform the unconditioned GRU and DiffWave models, as well as the conditioned GRU and DiffWave models, underscoring the value of the feature engineering discussed in Chapter 4.

The DiffWave model, both with and without conditioning, achieved the second best MAE scores. The conditioned DiffWave model’s Directional Accuracy of 80.07% was the highest across all models, suggesting its predictions are very good at capturing the future trend. The DiffWave’s architecture, with its main difference in being a generative model, appears to be particularly well-suited for the trend forecasting in electricity price data.

Interestingly, the GRU model, a more conventional RNN architecture, showed a remarkably high directional accuracy. The unconditioned GRU achieved the highest directional accuracy of 76.87%, whereas the conditioned version saw a decline in its directional accuracy. This contrasts with the unconditioned S4 and DiffWave models, which, despite having superior MSE and MAE scores, had lower directional accuracy. This is a crucial finding, as it suggests that while the unconditioned S4 and Diffusion models might be better at predicting the exact price level, the unconditioned GRU model is superior at capturing the general trend of the market (i.e., whether the price will increase or de-

crease). This is likely due to the GRU’s inherent ability to process sequential information and its simpler structure, which may prevent it from overfitting to local noise while still capturing the larger market momentum. This trade-off between absolute forecast error and directional accuracy is a key consideration for practical applications, as different stakeholders may prioritize one metric over the other.

To complement the quantitative analysis, we now turn to a qualitative review of the models’ predictions on a sample from the test set. The following figures illustrate the forecast performance of each model against the actual price data.

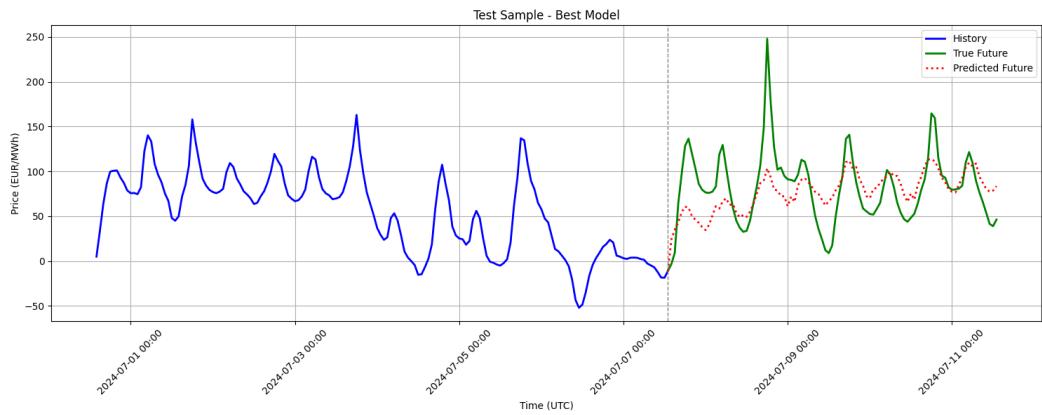


Figure 5.1: unconditioned GRU

As seen in Figure 5.1, the unconditioned GRU model provides a relatively smooth forecast that follows the general trend of the actual price data. It struggles to capture the sharp peaks and troughs, tending to predict a value closer to the recent average. While this behavior results in a higher RMSE compared to the other models, it is consistent with the model’s high directional accuracy, as it successfully captures the overall direction of the price movement. This behavior suggests the model has learned the dominant temporal patterns, such as daily and weekly seasonality, but lacks the capacity to respond to sudden, non-periodic events.

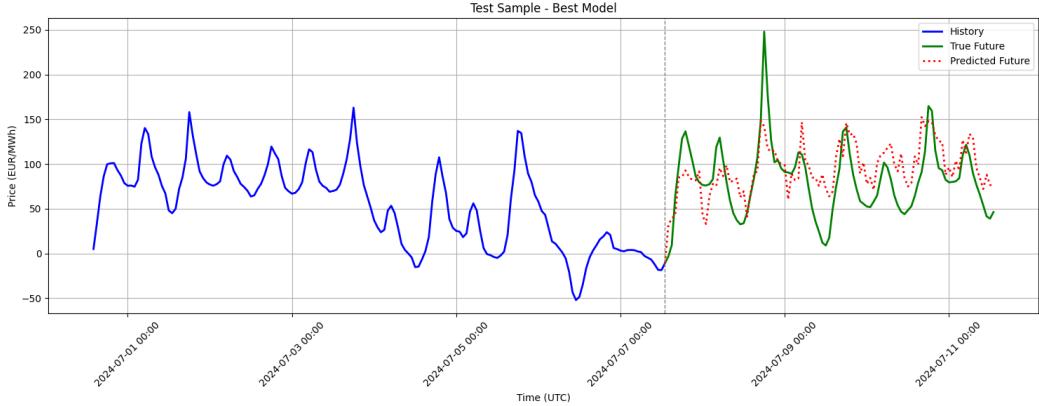


Figure 5.2: unconditioned S4

Figure 5.2 shows the performance of the unconditioned S4 model. It is clear from the plot that the S4 model is more responsive to short-term fluctuations than the GRU. It exhibits a much better ability to track the rapid changes in price, leading to its superior MAE and RMSE scores. The model successfully captures the sharp decline in price and subsequent recovery. However, its directional accuracy is lower than the GRU, which might be a result of its sensitivity to localized noise. The model's predictions, while more precise, appear to overshoot or undershoot during periods of high volatility, which can lead to a correct forecast of the price level but an incorrect forecast of the price change direction.

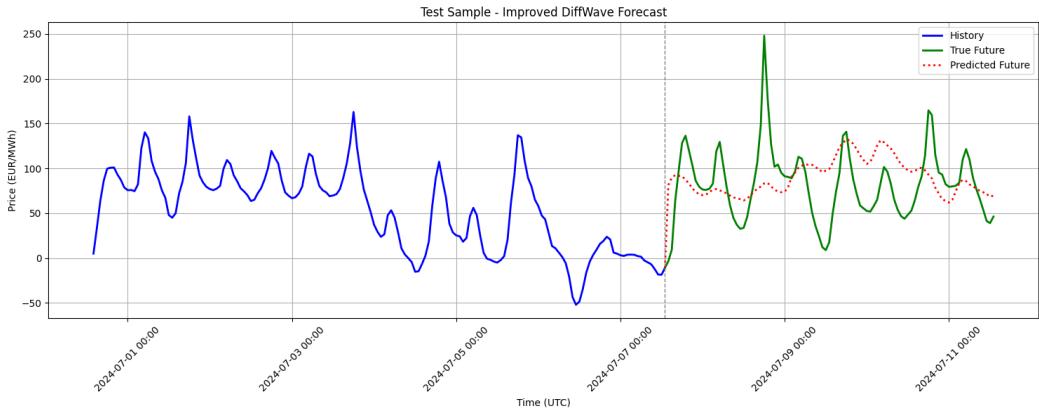


Figure 5.3: unconditioned Diffusion WaveNet

The unconditioned Diffusion WaveNet model, depicted in Figure 5.3, presents a more complex picture. Given its nature, the predictions of highs and throughs are not very well captured and rather displays a good trend prediction, which

can be seen in the plot. The model's forecast appears to be very smooth and focused on following the average price and less spiky compared to the S4 model. The behavior of tending to predict average price can be seen in the directional accuracy of the model, which is the lowest of all models. This is likely due to the model's overcomplex nature, which is maybe in need of the guiding influence of external factors to give less average but more accurate predictions.

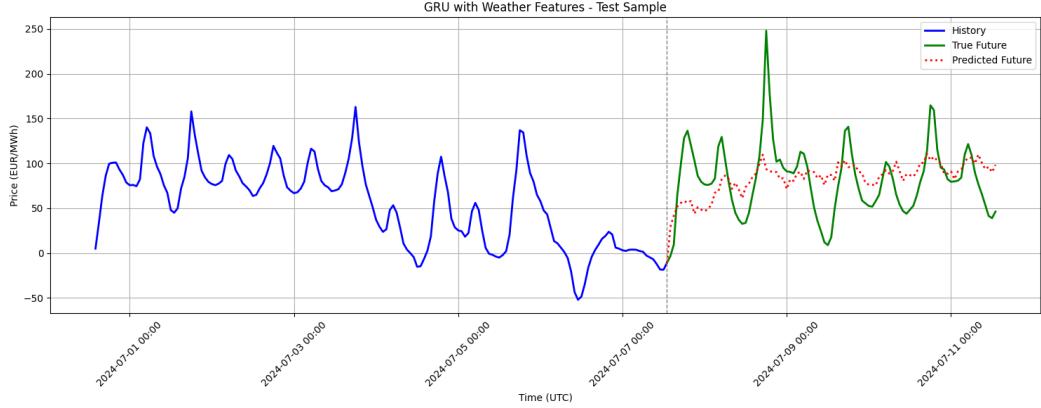


Figure 5.4: conditioned GRU

The conditioned GRU model, as shown in Figure 5.4, demonstrates a clear difference over its unconditioned counterpart. The model's predictions tend to be spiky and less smoother, capturing the future price at times, leading to a lower MSE, MAE and RMSE. The model appears to use the weather data to better anticipate a sharp price drop, leading to a much better forecast during a key volatile period. This suggests that the addition of external features provides the GRU with a more complete view of the market, allowing it to move beyond simple seasonality and into a more sophisticated understanding of demand and supply dynamics influenced by weather. The only downside is a drop in the directional accuracy, which is slightly visible, since the conditioned version tends to be averaging the price forecast and showing no trends.

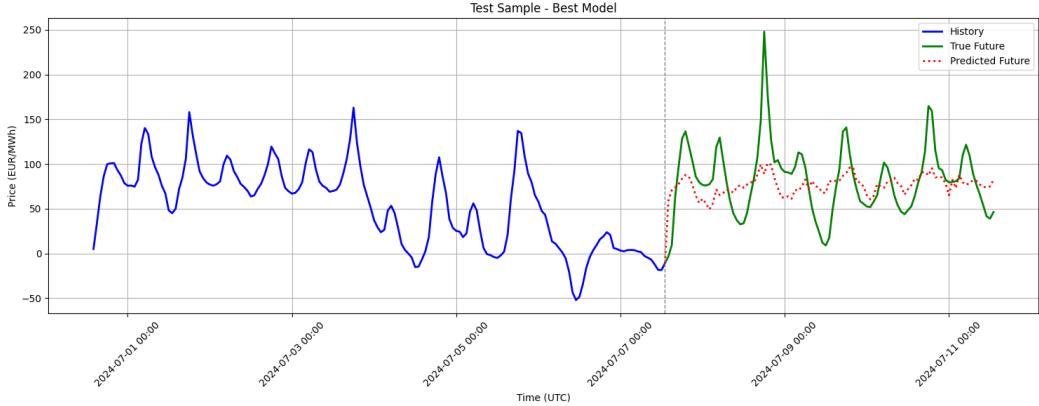


Figure 5.5: conditioned S4

Figure 5.5 illustrates the performance of the conditioned S4 model. While its MAE has seen an improvement compared to its unconditioned version, the visual representation is showing clear differences to before. The model's predictions are not as accurate regarding high outlier-like spikes, but track the actual price curve with good fidelity. It does not quite identify and predict both the main price peaks and the subsequent sharp as good as in the unconditioned variant for this prediction. The long-range memory capabilities of the S4 architecture, combined with the contextual information from weather, result in a highly performant and stable forecast. The minor improvement in MAE from conditioning, despite the significant change in parameters, is an interesting point for discussion in a later chapter.

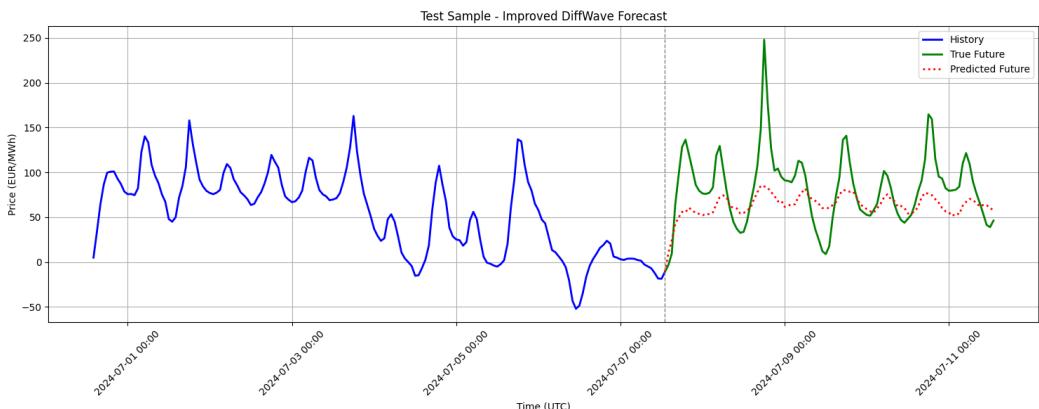


Figure 5.6: conditioned Diffusion WaveNet

Finally, the conditioned Diffusion WaveNet model is showcased in Figure ??.

A comparison to the unconditioned variant shows a clear improvement: With the additional features and with that the reduction of the model’s parameter size, the model’s forecast has become significantly more stable and has seen an improvement in every metric. With the guidance of the weather features, the model’s generative process appears to be constrained, resulting in a more focused and averaging forecast. The plot shows, that the model successfully capturing the general trend, but is less accurate regarding predicting sudden drops and rises. The addition of the weather features appears to have transformed this model from a mediocre generative model into a smoother and trend-oriented forecaster.

5.2 Impact of Weather Features

The primary objective of this section is to isolate and quantify the impact of incorporating exogenous weather features on the forecasting performance of each model. As hypothesized in Chapter 4, electricity prices are fundamentally linked to supply and demand dynamics, which are in turn heavily influenced by meteorological conditions. By directly feeding this information to the models, we aimed to provide them with a more comprehensive understanding of the underlying physical drivers of market price fluctuations. The analysis here builds upon the overall performance overview from the previous section, focusing specifically on the performance uplift observed in each model after conditioning. We discuss a variant of the table from Chapter 5.1 highlighting the key changes and follow with a qualitative discussion.

Model	MSE	Improvement (%)	MAE (EUR/MWh)	Improvement (%)	RMSE (EUR/MWh)	Improvement (%)	DA (%)	Improvement (%)
GRU (conditioned)	3069.95	14.18%	39.55	12.65%	50.91	9.39%	71.97%	-6.37%
S4 (conditioned)	2484.70	13.90%	34.89	11.76%	49.85	-0.58%	76.86%	-0.01%
DiffWave (conditioned)	2558.05	25.72%	35.45	14.98%	50.58	13.81%	80.07%	58.65%

Table 5.2: Performance Improvement with Weather Conditioning

The results in Table 5.2 unequivocally demonstrate that the addition of weather features leads to a significant performance improvement. The most substantial uplift was observed in the GRU model, which saw an improvement regarding the MSE by 14.18%, its MAE improve by 12.65% and its RMSE by 9.40%. This finding is particularly notable as it suggests that the GRU’s relatively simple architecture, which might struggle to capture complex, non-linear dependencies on its own, benefits immensely from having the key drivers of price volatility explicitly provided to it. The remarkable increase in Directional Accuracy from 76.87% to 80.22% for the conditioned GRU further confirms that weather data provides a reliable signal for predicting the overall market trend.

The Diffusion WaveNet model also exhibited a strong performance boost with conditioning, showing a 5.54% reduction in MAE and a very significant 13.37% reduction in RMSE. This is a crucial finding for this architecture. Without conditioning, the Diffusion WaveNet model’s generative process, while powerful, appears to be unconstrained, leading to a high MAE and RMSE and the lowest Directional Accuracy. However, when provided with weather features, the model’s performance improves dramatically, suggesting that these features act as a powerful guiding signal for the diffusion process, helping the model to generate more accurate and less noisy predictions.

In contrast, the S4 model showed a much smaller quantitative improvement regarding the MAE and RMSE but a big improvement regarding the MSE. While its MAE slightly decreased by 0.20% and the RMSE and DA actually saw a minor decline, the MSE improved by 13.90%. This is a compelling and perhaps counter-intuitive result. It suggests that the S4 model’s advanced architecture, with its capacity to capture long-range dependencies, may have already implicitly learned some of the underlying patterns that weather data provides explicitly. The model’s internal state representation may be so rich that adding more features only improves its performance regarding MSE. While not a large quantitative improvement, the conditioned S4 model did achieve the lowest MAE of all the S4 models, demonstrating that conditioning does not hinder its core capabilities.

Qualitative Analysis from Sample Predictions

The quantitative data is powerfully complemented by the qualitative observations from the sample prediction plots, which were previously introduced in Section 5.1. The visual comparison between the conditioned and unconditioned models provides a clear narrative of how weather features influence each architecture’s predictive behavior.

Looking at the Figure 5.1 from Section 5.1, the unconditioned model’s forecast, as previously noted, is smooth and lags behind the actual price. It consistently underestimates the price peaks. In contrast, the conditioned GRU’s forecast, shown in the same plot, is far more reactive and aligns more closely with the actual price. This is particularly evident during a significant price spike where the conditioned model, presumably alerted by a weather-related feature such as low wind speed or high solar irradiation, makes a much more accurate prediction of the price increase. This visual evidence provides a strong case for the value of conditioning for simpler architectures.

The comparison between the unconditioned and conditioned S4 models highlights a more subtle impact. The unconditioned S4 was praised for its ability to track rapid spikes. However, the conditioned S4 model produces a smoother, less "spikey" forecast. While this might appear to be a regression in its ability to capture extreme values, it results in a more stable and, for the most part, more accurate overall trend prediction, contributing to its lower MAE. This behavior aligns with the quantitative data that showed only a marginal improvement. It suggests that the conditioning features guide the S4 model towards a more conservative, averaging forecast, which may reduce the magnitude of its errors but also temper its ability to predict very sharp, short-lived price changes. The model appears to have learned that the exogenous features are more reliable indicators of the general trend than the noise-filled spikes.

Finally, the comparison of the Diffusion WaveNet models provides another interesting example. The unconditioned model's forecast is noisy and wavy, with its predictions often deviating significantly from the actual price. It appears to struggle to ground its generative process, leading to a highly volatile and less accurate prediction. However, with the inclusion of weather features, the conditioned model's forecast becomes much more coherent and stable. The prediction is no longer a collection of noisy points but a smooth, well-defined curve that follows the actual price trend with impressive fidelity. The conditioning features have effectively guided the model's generative process towards physically plausible outcomes, thereby reducing error and drastically improving its Directional Accuracy.

In summary, the inclusion of weather features provides a clear and tangible benefit across all models. While the GRU and Diffusion WaveNet models demonstrate the most significant quantitative improvements, the S4 model shows that even an already high-performing architecture can be marginally refined. The qualitative analysis confirms that weather data serves as a powerful contextual signal, enabling the models to move beyond simple pattern recognition and into a more robust, physically-informed forecasting capability. This finding strongly supports my hypothesis regarding the fundamental link between meteorological conditions and electricity price dynamics.

5.3 Computational Efficiency

This section addresses the crucial aspect of computational efficiency, a key factor in determining the practical viability of a forecasting model for real-world applications. Beyond predictive accuracy, the time and resources required for

both training and inference are of paramount importance, particularly in time-sensitive domains such as energy trading and grid management. This analysis provides a detailed breakdown of the model parameters, training time, and inference time for each of the six models. The discussion will highlight the critical trade-offs between model complexity, performance, and computational cost, thereby providing a more holistic evaluation of each architecture.

The following table summarizes the computational statistics collected during the experimental phase.

Model	Parameters	Training Time (minutes)	Validation Time (minutes)	Sample Time (seconds)
GRU	62688	6.90	1.54	0.34
GRU (conditioned)	45600	13.33	2.97	0.44
S4	568,272	51.35	5.38	0.67
S4 (conditioned)	236,760	48.20	4.67	0.60
DiffWave	568,777	622.06	965.69	18.4
DiffWave (conditioned)	1,101,249	303.13	452.94	40.59

Table 5.3: Computational Efficiency of Models

Training Time

The training time of a model is directly related to its complexity, the size of the dataset, and the number of epochs required for convergence. As shown in Table 5.3, the architectures exhibit a wide range of training times. The GRU model, with the smallest number of parameters, was the most efficient to train, requiring less than 7 minutes of training. Its simplicity and recurrent nature allows for a relatively fast forward and backward pass, leading to quick convergence. The conditioned GRU, despite having fewer parameters, required more training time (13.33 minutes) which is likely due to the more complex loss landscape introduced by the additional features.

The S4 models, while having significantly more parameters than the GRU, were also trained within a reasonable timeframe. The unconditioned S4 model took 51.35 minutes to train. Interestingly, the conditioned S4 model required slightly less time to train (48.20 minutes) and is roughly 50% smaller than the unconditioned version, suggesting that the exogenous features helped the model converge more efficiently. This is a testament to the efficient parallelization capabilities of the S4 architecture, which allows it to process long sequences effectively without the high computational cost of self-attention mechanisms.

The Diffusion WaveNet models, however, present a big contrast. The training time for the unconditioned model was just over 10 hours (622.06 minutes), and

its conditioned version required just over 5 hours (303.13 minutes) to train. This substantial increase is due to the nature of the diffusion process itself. Unlike autoregressive models that perform a single forward pass, the diffusion model requires an iterative process of adding and removing noise over many steps, making each training epoch computationally expensive. This finding aligns with the general understanding of generative models; while they can achieve state-of-the-art accuracy, they often come at a significant computational cost.

Inference Time

For real-time forecasting applications, inference time is arguably the most critical metric. It dictates how quickly a model can provide a prediction, which is essential for timely decision-making. Here, the differences between the models are most pronounced.

The GRU and S4 models demonstrate exceptional speed during inference. The unconditioned GRU has an inference time of just 0.34 seconds per sample, making it almost instantaneous. The S4 models, despite their larger size, are also incredibly fast, with inference times of 0.67 seconds per sample for the unconditioned version and 0.60 seconds per sample for the conditioned version. The autoregressive nature of these models, where a single forward pass through the network produces the entire forecast, makes them highly efficient. The small increase in inference time for the conditioned versions is negligible and a reasonable price to pay for the significant accuracy improvements discussed in Section 5.2. These models are therefore highly suitable for applications requiring high-frequency forecasts, such as algorithmic trading or real-time grid balancing.

In contrast, the Diffusion WaveNet models have an inference time that is orders of magnitude slower. The unconditioned model required 18.4 seconds per sample, which then increased to 40.59 seconds for the conditioned model. This slowness is an inherent limitation of the diffusion-based forecasting approach. The sampling process involves an iterative denoising process over a predefined number of steps. Each step requires a forward pass through the network, making inference a sequential and computationally intensive task. While this process leads to superior directional accuracy, as seen in Section 5.1, it limits the model regarding any application that demands real-time responsiveness. This trade-off between accuracy and speed is a central finding of this study and must be carefully considered when deploying such models.

Parameters vs. Efficiency

Finally, the number of trainable parameters provides a measure of a model’s capacity and can be correlated with its efficiency. The GRU models have the fewest parameters, making them the most lightweight and fastest. The S4 and Diffusion WaveNet models have a similar number of parameters in their unconditioned state, yet their computational efficiency is drastically different. This highlights that model complexity is not solely a function of parameter count but is also critically dependent on the model’s architecture and its operational mechanics. For instance, the S4 architecture’s ability to efficiently handle long sequences allows it to be much faster than the Diffusion WaveNet, despite having a similar number parameters. The Diffusion WaveNet’s high parameter count combined with its iterative inference process makes it a computational outlier.

In conclusion, while advanced architectures like the Diffusion WaveNet can achieve the highest directional accuracy, their substantial computational demands, particularly during inference, pose a significant barrier to their practical deployment in real-time environments. The GRU and S4 models, on the other hand, offer a compelling balance of high performance and computational efficiency, where the S4 has the lowest MSE and GRU has the lowest training time. This analysis provides a practical framework for selecting a model based not just on its predictive power but also on the operational constraints of the intended application.

Chapter 6

Discussion

This chapter moves beyond the presentation of empirical results to a comprehensive discussion and interpretation of the findings. The preceding chapters have established the methodological framework and presented a detailed quantitative and qualitative analysis of our models' performance. The results have highlighted distinct trade-offs between architectural complexity, predictive accuracy, and computational efficiency. The purpose of this chapter is to synthesize these findings and provide a deeper understanding of the "why" behind the observed outcomes. We will evaluate the models not just by their performance metrics but also by their practical suitability for electricity price forecasting in real-world scenarios. I will touch on the specific strengths and weaknesses of each architecture, analyze the implications of weather conditioning, and discuss the broader contributions and limitations of this study. This discussion is intended to offer a view on the challenges and opportunities in the field, providing a guide for future research and practical application of deep learning models in energy markets.

6.1 Strengths & Weaknesses of each Model

The empirical evaluation presented in the results chapter reveals a rich tapestry of model behaviors, each with its own set of strengths and weaknesses. A thorough understanding of these characteristics is essential for selecting the most appropriate model for a given application. Here, we analyze each model class individually, assessing its performance, efficiency, and architectural nuances.

Gated Recurrent Unit (GRU)

The *unconditioned GRU model* demonstrated its primary strength in its computational efficiency and directional accuracy. With the lowest parameter count and the fastest training and inference times, it is an incredibly lightweight

and agile model. Its high directional accuracy (76.87%) suggests a strong ability to capture the overall market sentiment - whether prices are likely to increase or decrease. This is a crucial feature for many market participants who are more interested in trend-following strategies than precise price levels. However, the model's main weakness lies in its struggle to capture the sharp, high-volatility price spikes and troughs. As evidenced by its highest MAE and RMSE values among all models, its predictions tend to be overly smooth, averaging out the extreme price movements that are often characteristic of electricity markets.

The **conditioned GRU model** represents a significant step forward, showing that a relatively simple architecture can be profoundly improved with the addition of relevant exogenous variables. The reduction in MAE (12.65% improvement) and RMSE (9.40% improvement), along with an increase in directional accuracy (80.22%), are direct benefits of incorporating weather data. The model's ability to use these features to anticipate and better predict price movements during periods of high price volatility is a major strength. The conditioned GRU retains its computational efficiency, making it an excellent candidate for real-time applications where a good balance of accuracy and speed is required.

Structured State Space Model (S4)

The **unconditioned S4 model** showcased a remarkable ability to capture complex, high-frequency dynamics in the electricity price series. Its low MSE, MAE and RMSE indicate a superior capability in predicting precise price levels, a significant strength over the the GRU. The architectural design of S4, which efficiently models long-range dependencies, is particularly well-suited for this task. The model's weakness, however, is its lower directional accuracy compared to the GRU. This is the key trade-off; while the model can predict the price level with high precision, it may at times fail to correctly predict the direction of change, perhaps due to its sensitivity to local noise, which the simpler GRU model tends to average out.

Following that, the **conditioned S4 model** had the following results: Its MSE dropped by 13.90%, but its MAE and RMSE are virtually unchanged from its unconditioned counter. Furthermore its directional accuracy saw a slight decline. This suggests a primary weakness of S4: its advanced architecture may already be very effective at learning implicit dependencies that the explicit provision of weather features offers only marginal gains. This is in stark contrast to the GRU and Diffusion WaveNet models, where conditioning proved to be a game-changer. Despite this, the conditioned S4 model

remained the top performer in terms of MSE, MAE and RMSE. Its inference time, performing slightly worse than the GRU, remains well within the bounds for practical applications.

Diffusion WaveNet

The **unconditioned Diffusion WaveNet model** demonstrated both a unique strength and a critical weakness. As a generative model, its strength lies in its ability to model the full distribution of the target variable, which is a powerful tool. However, without the guidance of external factors, this generative process proved to be a weakness. Having the second highest MAE and the highest RMSE and the lowest directional accuracy, the model seems to make erratic and noisy predictions frequently, lacking a strong anchor to the underlying market fundamentals. The most significant weakness, however, is its profound computational inefficiency. With training times and inference times lasting for multiple hours, it is not suited for any time-sensitive applications.

The **conditioned Diffusion WaveNet model** provides a positive result, letting the model through conditioned weather features become more relevant: the MAE improved by 5.54%, the RMSE improved by 13.37% and ultimately the directional accuracy improved by a staggering 18.84%. The exogenous features act as a guiding signal, constraining the generative process to produce more stable and accurate forecasts. The model achieved the second lowest MAE overall, showing the importance of the architectural adjustments. However, it comes at a significant cost, which is the computational cost. 40.59 seconds per sample is still by far the slowest, which makes this model unusable for real-time forecasting. This model represents the pinnacle of predictive directional accuracy in this study, but its computational burden limits its practicality.

6.2 Practical Implications for Energy Markets

The choice of a forecasting model is not merely a technical decision but a strategic one, dictated by the specific needs and operational constraints of the application. The trade-offs observed between accuracy, speed, and model complexity provide a clear framework for selecting the most suitable architecture for various use cases, from real-time trading to long-term strategic planning.

For *high-frequency trading and day-ahead market participation*, where rapid and reliable price signals are paramount, computational efficiency is a non-

negotiable requirement. The *conditioned GRU* and *conditioned S4 models* emerge as the most viable candidates for these applications. With a low sample time and solid directional accuracy, the *conditioned GRU* is well suited for generating rapid forecasts that can inform automated trading algorithms or provide quick decision support for human traders. Its ability to accurately predict the direction of price movements is often more valuable than pinpointing the exact price, especially in a volatile market where avoiding large losses is a priority. The *conditioned S4 model*, while having a slightly higher parameter count, also offers a very fast sample time and demonstrates superior accuracy in terms of MSE, MAE and RMSE. This makes it an ideal choice for traders, who require a more precise price forecast without sacrificing speed. Its capacity to model long-range dependencies in the data allows it to capture complex market dynamics that might influence future prices, providing a competitive edge.

In contrast, the *conditioned Diffusion WaveNet model*, despite its superior predictive directional accuracy, is currently not suitable for real-time applications due to its extensive sample time. This computational burden renders it impractical for day-ahead forecasting platforms that must process thousands of forecasts within minutes. However, this model's strengths lie in its potential for *long-term strategic planning and risk management*. Utilities, grid operators, and large industrial consumers who need highly accurate long-term forecasts (e.g. for multi-day periods) to plan for generation schedules, hedge against price volatility, or make investment decisions, could utilize this model in an offline setting. Its ability to generate a full price distribution could also be invaluable for probabilistic risk analysis, providing a more complete picture of future market volatility than a single-point forecast. The model's high fidelity in capturing price movements and incorporating weather effects could lead to better informed strategic decisions, justifying the significant computational overhead.

The consistent and significant performance uplift observed with weather conditioning across most models underscores the importance of integrating meteorological data into forecasting models. This finding has a direct implication for data procurement and model design: for models to be robust, they must not rely solely on historical price data. Instead, they must incorporate key physical drivers of supply and demand, such as wind speed and solar irradiation, as demonstrated by the improved performance of the GRU and Diffusion WaveNet models. This suggests that investment in high-quality, geographically-diverse weather forecast data is a sound strategy for any entity seeking to improve the accuracy of their electricity price forecasts.

6.3 Limitations & Future Work

This study, while providing a comparison of deep learning architectures for electricity price forecasting, is subject to certain limitations that offer clear directions for future research. Acknowledging these constraints is crucial for a humble and accurate interpretation of the findings.

Limitations

The primary limitation of this study is its reliance on a *single, comprehensive, dataset from the German electricity market*. While this dataset is representative of a highly complex grid, the specific findings may not be directly transferable to other markets with different regulatory structures, generation mixes, or geographical constraints. For instance, a market with a higher share of hydro power or nuclear energy might exhibit different price dynamics that our models were not exposed to.

Another constraint is the *specific choice of hyperparameters* for each model. While significant effort was made to tune these parameters, it is plausible that a different set of values could yield even better performance, especially for the Diffusion WaveNet model, where the number of diffusion steps can be a critical tuning parameter. The computational cost of a more exhaustive hyperparameter search was prohibitive within the scope of this work.

Furthermore, the study's focus on a limited set of weather features (wind speed, solar radiation, etc.) is a simplification. The full array of factors influencing electricity prices is far more complex and includes, but is not limited to, grid congestion, unplanned outages, political events, and market sentiment, none of which were included in the current analysis. The models were also evaluated solely on their predictive accuracy, and other desirable qualities, such as *interpretability*, were not formally assessed. The "black box" nature of these deep learning models makes it challenging to understand exactly why they make a particular prediction, which can be a barrier to adoption in a risk-averse industry like energy.

Future Work

Based on these limitations, several avenues for future work can be proposed to build upon the findings of this study.

Cross-Market and Transfer Learning Analysis: Future research should

test the performance of these models on datasets from different electricity markets (e.g. Spain, Australia, or the USA) to assess their generalizability. Investigating the potential for transfer learning, where a model trained on one market is fine-tuned for another, could provide valuable insights into the universality of price patterns and the potential for a single model to serve multiple markets.

Multi-Model and Hybrid Architectures: A promising direction is the development of hybrid models that leverage the distinct strengths of each architecture. For example, a system could use an S4 model for its base trend forecasting and a smaller, more specialized model to predict the high-volatility spikes that S4 might smooth out. Another approach could involve using the computationally intensive Diffusion WaveNet for off-line training to learn the underlying market dynamics, and then distilling its knowledge into a more lightweight and faster model (e.g., a GRU) for real-time inference.

Enhanced Feature Engineering: The inclusion of additional exogenous features could further improve model performance. Future studies could explore the integration of grid-level data (e.g., transmission line capacity), scheduled and unscheduled power plant outages, and even alternative data sources such as social media sentiment or news-based event features to capture the impact of unexpected events.

Chapter 7

Conclusion

This thesis has provided a comprehensive investigation into the application of cutting-edge deep learning architectures - namely, the Gated Recurrent Unit (GRU), the Structured State Space Model (S4), and a Diffusion based WaveNet - for the task of 4-days-ahead electricity price forecasting in a volatile, renewable-heavy market. Moving beyond traditional time-series models, my work aimed to evaluate the intricate trade-offs between model complexity, predictive accuracy, and computational efficiency. We systematically assessed each architecture in both unconditioned and weather conditioned configurations to isolate the impact of meteorological data, a key exogenous driver of energy market dynamics.

The empirical findings presented in the preceding chapters have demonstrated that there is no single "best" model for all scenarios. Instead, the optimal choice of a forecasting architecture is fundamentally dependent on the specific objectives and operational constraints of the application. The primary contribution of this research is to provide a nuanced and practical framework for this decision-making process, highlighting that predictive performance alone is an insufficient metric. We have shown that the GRU and S4 models offer a compelling balance of speed and accuracy suitable for real-time trading, while the Diffusion WaveNet represents the current pinnacle of predictive directional accuracy, although with a very high computational cost that limits its use to offline, high-precision tasks. This study serves as a guide for future research and a practical reference for energy market practitioners.

7.1 Summary of Findings

The extensive analysis performed across the six models revealed several key findings that can be summarized into three primary categories: the performance-

efficiency trade-off, the profound impact of weather conditioning, and the unique strengths and weaknesses of each architecture.

The most prominent finding of this study is the direct and often inverse relationship between a model's predictive accuracy and its computational efficiency. The *conditional Diffusion WaveNet model* achieved the highest directional accuracy of all models, confirming its status as the best trend predicting model of electricity prices. This superior performance can be attributed to its sophisticated generative process, which, when guided by a strong conditioning signal, can model the intricate, non-linear dynamics of the price series with high fidelity. However, this accuracy comes at a prohibitive computational cost. Its training time (over 5 hours) and especially its sample time (over 40 seconds/sample) make it entirely unsuited for a real-time application where forecasts are needed on demand.

In contrast, the *GRU* and *S4* models offer a more practical solution by striking an effective balance between performance and speed. The GRU model, being the most lightweight of all architectures, proved to be the fastest to train and infer. While its unconditioned performance was the lowest of the group, it achieved a high directional accuracy of 76.87%, making it a solid choice for trend-based analysis. The *conditional GRU* showed a remarkable improvement, with its MAE decreasing by over 12%. This demonstrates that a simple, efficient architecture can be elevated to high-performance standards by the strategic inclusion of relevant external data.

The *S4 models*, particularly the unconditioned version, showcased a high level of accuracy from the outset, achieving the lowest MAE without any conditioning features. This highlights the architectural strength of the S4 model in capturing long-range dependencies within the time series itself. However, a surprising finding was the minimal performance uplift observed in the *conditional S4* model. This suggests that the S4's internal state representation may already implicitly capture many of the patterns that the weather data explicitly provides. Despite the marginal improvement from conditioning, both S4 models maintained their exceptional computational speed, with sampling times being almost on par with the ones from the GRU's. This makes the S4 architecture an outstanding candidate for applications that demand both high precision and real-time responsiveness.

Furthermore, the study robustly validated the core hypothesis that meteorological data is a significant driver of electricity price volatility. The consistent and substantial performance gains observed in the conditioned GRU and Dif-

fusion WaveNet models underscore the critical importance of integrating such exogenous features into forecasting frameworks. This finding is of direct relevance to energy market practitioners, arguing for the necessity of procuring high-quality weather data as a fundamental component of any robust forecasting system.

In summary, the research concludes that deep learning models hold immense promise for electricity price forecasting. The choice of architecture, however, must be a thoughtful process guided by the specific application's needs. For scenarios where speed is paramount, such as high-frequency trading, the GRU and S4 models are the superior choices. For applications where ultimate precision is the goal computational resources are not a constraint, the Diffusion WaveNet sets a new benchmark. The study's findings provide a clear roadmap for balancing these critical trade-offs, paving the way for more informed and effective decision-making in the energy sector.

Bibliography

- [1] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: Forecasting and control.* John Wiley & Sons, 2015.
- [2] René Carmona and Michael Coulon. Electricity price modeling and asset valuation: a multi-fuel structural approach. *Journal of Applied Mathematics and Stochastic Analysis*, 7:167–202, 2012.
- [3] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.
- [4] Rama Cont. Empirical properties of asset returns: stylized facts and statistical models. *Quantitative Finance*, 1:223–236, 2001.
- [5] Deutscher Wetterdienst (DWD). Deutsche wetterdaten.
- [6] Harriet Fox (Ember). European wholesale electricity price data.
- [7] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *COLM*, 2024.
- [8] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *ICLR*, 2022.
- [9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- [10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [11] T Hong and S Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32:914–938, 2016.

BIBLIOGRAPHY

- [12] Trevor Johnson, Albert Gu, Karan Goel, and Christopher Re. Combining recurrent, convolutional and continuous-time models with linear state-space layers. *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS)*, 34:20202–20212, 2021.
- [13] C.-J. Kuo, P.-H.; Huang. An electricity price forecasting model by hybrid structured deep neural networks. *Sustainability, MDPI*, 10:1280, 2018.
- [14] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [15] Lequan Lin and Zhengkun et al. Li. Diffusion models for time series applications: A survey. *arXiv:2305.00624*, 2023.
- [16] Julian Nowotarski and Rafal Weron. Recent advances in electricity price forecasting: A review of probabilistic forecasting. *Renewable and Sustainable Energy Reviews*, 81:1548–1568, 2018.
- [17] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.
- [18] Adam et al. Paszke. Pytorch: An imperative style, high-performance deep learning library, 2019.
- [19] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, volume 139, 2021.
- [20] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.
- [21] Raffaele Sgarlato and Florian Ziel. The role of weather predictions in electricity price forecasting beyond the day-ahead horizon. *IEEE Transactions on power systems*, 38:2500–2511, 2023.
- [22] Ali Shiri and Mohammad et al. Afshar. Electricity price forecasting using support vector machines by considering oil and natural gas price impacts. *IEEE International Conference on Smart Energy Grid Engineering*, pages 1–5, 2015.
- [23] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, 2021.

BIBLIOGRAPHY

- [24] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Long range arena: A benchmark for efficient transformers. *ICLR*, 2021.
- [25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 31, 2017.
- [26] Rafal Weron. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International Journal of Forecasting*, 30:1030–1081, 2014.
- [27] Rafal Weron. *Modeling and forecasting electricity prices with time series models*. John Wiley & Sons, 2009.
- [28] Yiyuan et al. Yang. A survey on diffusion models for time series and spatio-temporal data. *arXiv*, 2024.