Shihab Mahmud Dhrobo

AI-Driven Load Forecasting and Optimization in Smart Grids using Cloud-Based Analytics

School of Technology and Innovations
Master of Science in Technology
Master Programme in Industrial Systems Analytics

Vaasa 2026

UNIVERSITY OF VAASA
School of Technology and Innovation
Author: Shihab Mahmud Dhrobo
Title of the thesis: AI-Driven Load Forecasting and Optimization in Smart Grids using Cloud-Based Analytics
Degree: Master of Sciences in Technology and Innovation
Degree Programme: Industrial Systems Analytics
Supervisor: Emmanuel Ndzibah
Year: 2026 Pages:107

ABSTRACT:

Smart grids combine digital communications, real-time sensing, and cloud-based computing to facilitate two-way electricity and information flows, generating large amounts of data produced by smart meters, IoT sensors, and distributed energy resources. With the increase in renewable energy penetration, and nonlinear demand patterns, accurate short-term load prediction is now required to maintain grid stability, day-ahead scheduling, and demand response activation. Conventional statistical models are not enough to capture such dynamics, but the state of the research in AI has been mainly based on forecast accuracy in the laboratory with no attention to deployment, monitoring, or the operational connection between forecasts and grid decisions.

This thesis explores the role of cloud-based AI models in improving the accuracy of short-term load forecasting and operational optimization in smart grids. The empirical basis is based on the Finnish national electricity consumption data of the Fingrid open data platform and weather covariates of Open-Meteo. Three AI models, Long Short-Term Memory networks, Prophet, and XGBoost are contrasted with ARIMA and a weekly persistence baseline under identical conditions, and evaluated based on RMSE, MAE, MAPE, and R square, which are all validated using Diebold-Mariano significance tests. On the Amazon Web Services, a six-stage MLOps pipeline is designed and implemented, which addresses data ingestion, preprocessing, training, validation, inference, and monitoring. XGBoost engineered lag, calendar and weather features have the best accuracy by all measures. The statistical baselines are surpassed by LSTM, whereas Prophet is viable under appropriate settings. The analysis of feature importance reveals that autoregressive structure prevails in short-horizon forecasting. At least twelve months of training data are determined to be operationally acceptable. The demand response scenarios are forecasted to allow automatic identification of the peak hours and low-load windows to be used as the decision support in the grid. The participant in the expert interview confirmed that 15-minute interval forecasting is the primary operational standard at the national transmission level, making short-term load forecasting the most operationally significant forecasting horizon in day-to-day grid management.

The study concludes that, as a system-design problem, cloud-based AI forecasting can provide good load prediction accuracy and operative demand response indicators when using publicly available data and open-source tooling.

KEYWORDS: short-term load forecasting, smart grid, XGBoost, LSTM, MLOps, demand response

DISCLAIMER AND DECLARATION OF INDEPENDENT AUTHORSHIP

I hereby declare that this Master of Science (MSc) thesis has been written independently and represents my own original work carried out in accordance with the academic rules, ethical standards, and research integrity principles of University of Vaasa.

I confirm that, to the best of my knowledge and intention, I have not engaged in any form of academic misconduct, including but not limited to plagiarism, falsification, ghostwriting, unauthorized collaboration, or the misuse of Artificial Intelligence (AI), Large Language Models (LLMs), or other automated content-generation tools in a manner that violates university regulations or academic integrity standards.

Furthermore, I declare that this thesis has not been subcontracted, outsourced, purchased, or produced in whole or in part by any paid service provider, third party, commercial writing agency, or external individual. All analysis, interpretation, writing, and presentation contained in this thesis are the result of my own independent academic effort unless explicitly and properly referenced.

Where AI-assisted tools or digital technologies may have been used for limited support purposes permitted under university guidelines (such as language refinement, grammar checking, formatting assistance, or idea organization), such usage has been conducted responsibly, transparently, and without compromising the originality, intellectual ownership, or academic integrity of the work.

I fully acknowledge and accept responsibility for the contents of this thesis. I understand that if evidence of plagiarism, unauthorized AI misuse, academic dishonesty, or third-party authorship is discovered at any stage before or after submission, the University reserves the right to take appropriate disciplinary and academic actions in accordance with its

regulations and policies. Such actions may include rejection of the thesis, annulment of the degree, disciplinary sanctions, or any other measures deemed necessary by the University. By signing this declaration, I affirm my commitment to honesty, transparency, and ethical academic conduct.

Student Name: Shihab Mahmud Dhrobo

Student Number: 2403098

Programme: Industrial Systems Analytics, Master of Science in Technology

Title of Thesis: AI-Driven Load Forecasting and Optimization in Smart Grids using Cloud-Based Analytics

Contents

1. Introduction 11

1.1 Background of study 11

1.2 Research gap, question and objectives 13

1.3 Definitions and scope of the study 16

1.4 Structure of the study 20

2 Literature Review 22

2.1 Smart grids and their challenges 23

2.2 Load forecasting 27

2.3 AI and machine learning techniques for energy data 30

2.4 Cloud-based analytics in energy management 33

2.5 Review of adaptive load management strategies 37

2.6 Summary of theoretical framework 41

3 Methodology 45

3.1 Data sources 48

3.2 Data preprocessing and feature engineering 50

3.3 AI models selected: rationale and implementation 53

3.3.1 Model selection criteria 54

3.3.2 Description of all models: LSTM, Prophet, XGBoost, ARIMA, and Naive baseline 55

3.3.3 Training and validation process 57

3.3.4 Hyper parameter tuning 59

3.3.5 Handling model limitations and assumptions 60

3.4 System architecture: cloud analytics platform 62

3.4.1 Cloud platform selection and setup 62

3.4.2 Data ingestion and storage 64

3.4.3 Real-time data processing and analytics pipeline 64

3.4.4 Deployment approach and automation (DevOps/MLOps) 65

3.4.5 Security, privacy and reliability considerations 65

3.5 Evaluation of metrics and approach 66

3.6 Ethical and Environmental 70

4 Results and Discussion 72

4.1 Experimental Setup and Scenarios 72

4.2 Model Performance Comparison 73

4.3 Visualizations and Analysis 75

4.4 Interpretation of Results 78

4.4.1 Engineered Feature Tables Outperform Deep Learning 78

4.4.2 Minimum Data Requirements Are Critical for Both Models 79

4.4.3 Correct Configuration Transforms Prophet from Failure to Viable Baseline 79

4.4.4 ARIMA's Strong Performance Reflects a Structural Evaluation Advantage 80

4.5 Adaptive Load Management Implications 80

4.6 MLOps Pipeline: Implementation and Operational Evaluation 82

4.6.1 Pipeline Architecture and Stage Description 83

4.6.2 MLOps Maturity Assessment 85

4.6.3 Pipeline Robustness: Observed Failure Modes and Mitigations 87

4.6.4 Comparison with Industrial MLOps Practice 88

4.7 Comparison with Prior Work 89

4.8 Limitations and Practical Implications 90
4.8.1 Data and Modeling Limitations 90
4.8.2 MLOps and Deployment Limitations 91
4.9 Recommendations for Smart Grid Operators 93
4.10 Industry Interview: Thesis Value and Future of Smart Grids 94
4.10.1 Interview Objective and Participant Description 94
4.10.2 Summary of Responses 95
5 Conclusion and Future Work 97
5.1 Summary of Findings 97
5.2 Contributions to the Field 100
5.3 Suggestions for Further Research 101
References 104

Figures

Figure 1. Conceptual framework linking smart grid data, AI analytics, and decision-making. 12

Figure 2. Conceptual Structure of the Thesis. 22

Figure 3. AI-based STLF in cloud-MLOps enabling adaptive load management and grid relief in smart grids. 44

Figure 4. Saunders' Research Onion (adapted from Musundire, 2025). 47

Figure 5. Data preprocessing and feature-engineering pipeline. 53

Figure 6. AWS-based architecture for AI-driven load forecasting. 63

Figure 7. Cross-validation performance metrics for XGBoost and LSTM. 75

Figure 8. SHAP summary plot for XGBoost showing the top 20 features ranked by mean absolute SHAP value. Red/pink dots indicate high feature values; blue dots indicate low values. 76

Tables

Table 1. Gap to objective mapping. 14

Table 2. Challenges and forecasting implications. 26

Table 3. Forecasting horizons (adapted from Hong & Fan, 2016). 27

Table 4. Forecasting approaches. 28

Table 5. Comparison of LSTM, Prophet and XGBoost models. 30

Table 6. Cloud functions and forecasting value (adapted from Al-Jumaili et al., 2023; Masood et al., 2024; Ullah et al., 2021). 34

Table 7. Cloud versus edge comparison (adapted from Ullah et al., 2021; Masood et al., 2024). 36

Table 8. Core strategies and mechanisms. 37

Table 9. Strategy, forecast, and outcome. 39

Table 10. The five pillars. 41

Table 11. Main data sources used in the study. 49

Table 12. Model roles in comparative study. 57

Table 13. Aligned test-set performance metrics for all models (2,583-hour evaluation window, September–December 2024). 73

Table 14. Diebold-Mariano statistical significance test results with Harvey et al. (1997) small-sample correction. 77

Table 15. Adaptive load management outputs derived from XGBoost forecasts over the 2,583-hour test period. 81

Table 16. MLOps pipeline stage descriptions and AWS service mapping. 83

Table 17. MLOps maturity assessment against Kreuzberger et al. (2023) dimensions. 85

Table 18. Comparison of this thesis with selected recent STLF studies. 89

Table 19. Test-set performance summary aligned with 2,583-hour evaluation window. 97

Abbreviations

AI	Artificial Intelligence
ARIMA	Autoregressive Integrated Moving Average
AWS	Amazon Web Services
CI/CD	Continuous Integration / Continuous Delivery
DER	Distributed Energy Resource
DM	Diebold-Mariano (test)
DR	Demand Response
DSO	Distribution System Operator
EMS	Energy Management System

EV	Electric Vehicle
GDPR	General Data Protection Regulation
IoT	Internet of Things
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MLOps	Machine Learning Operations
NCCS	EU Network Code on Cybersecurity
RNN	Recurrent Neural Network
RMSE	Root Mean Square Error
R^2	Coefficient of Determination
S3	Amazon Simple Storage Service
SCADA	Supervisory Control and Data Acquisition
SHAP	SHapley Additive exPlanations
STLF	Short-Term Load Forecasting
TSO	Transmission System Operator
XGBoost	Extreme Gradient Boosting

1. Introduction

1.1 Background of study

The energy sector is currently experiencing a major transformation driven by digitalization, decentralization, and the increasing integration of renewable energy resources into electricity systems. Traditionally, electric power systems were constructed based on large, centralized power plants with a one-way transmission and distribution network between the generators and consumers. To address these shortcomings, smart grids were developed with sensing, communication and control technologies that allow two-way flows of electricity and information between generation, transmission, distribution, and consumption (Fang et al., 2012, pp. 944–945; Ullah et al., 2021, pp. 956–958).

This development creates large, heterogeneous data streams of IoT sensors, smart meters, and edge devices and fits the criteria of big data of volume, variety, and velocity (Ali Al-Jumaili et al., 2023, pp. 1–4). This scale, however, cannot be processed conventionally, and cloud and edge computing have become the key to scalable analytics and quicker decision support. (Ali Al-Jumaili et al., 2023, pp. 1–4; Ullah et al., 2021, pp. 956–958).

The concept of short-term load forecasting (STLF) is very important to the operations of a smart grid because it forecasts the demand within 1 hour to several days in advance, which is used to support day-ahead scheduling, real-time balancing, and demand-response activation (Hong & Fan, 2016, p. 914). Traditional statistical tools (ARIMA, regression) are only sufficient in stable conditions but not in conditions of renewable variability and non-linear demand trends (Hong & Fan, 2016, pp. 914–916; Masood et al., 2024, pp. 1–4).

Machine learning solves these shortcomings, with LSTM networks being more effective at sequential patterns, XGBoost being more effective with feature interactions and Prophet being more effective in providing an interpretable seasonality (Hong & Fan, 2016, pp. 914–

916; Masood et al., 2024, pp. 1–4). However, these studies are more of an academic nature as they are based on accuracy measures (RMSE, MAPE) without considering the deployment, monitoring, or operational integration, particularly in systems that need to react to the intermittency of renewable resources and real-time balancing demands. Figure 1 illustrates the conceptual relationship between smart grid data streams, AI-based analytics, and operational decision-making that frames this thesis.

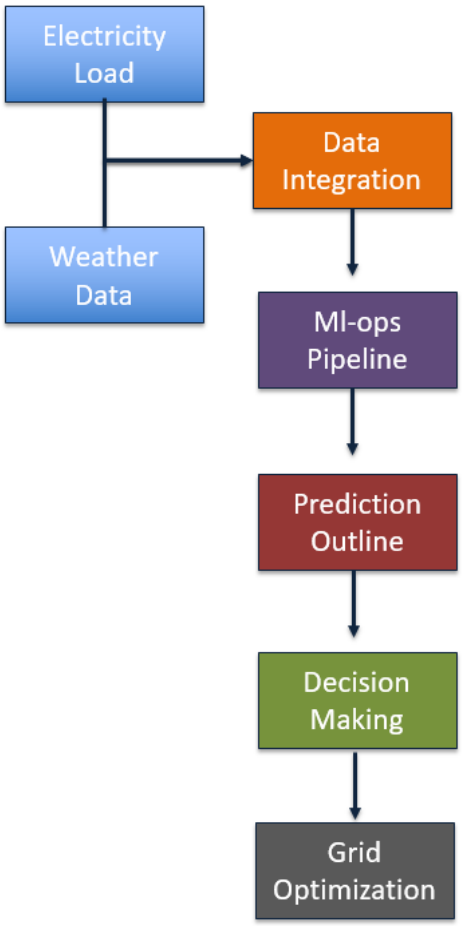

Figure 1. Conceptual framework linking smart grid data, AI analytics, and decision-making.

1.2 Research gap, question and objectives

Load forecasting has been a central point of energy informatics, power systems, and applied machine learning, and researchers have developed various statistical, machine-learning, and deep-learning load forecasting techniques across different forecasting horizons (Hong & Fan, 2016, pp. 914–918; Biswal et al., 2024, p. 3655). The increasing access to smart-meter and IoT data has also contributed to the increased popularity of AI-based forecasting methods due to their ability to process high, nonlinear, and dynamic energy data more efficiently than traditional forecasting methods (Dong et al., 2025, p. 3; Masood et al., 2024, pp. 2–4).

Although this has been achieved, there are three critical gaps. First, a significant portion of the literature continues to focus on forecast accuracy measures like RMSE and MAPE but does not pay much attention to the system-level implementation, monitoring, retraining, and lifecycle management in the production setting. Most research works are carried out in controlled experimental conditions that do not fully represent the practical aspects of smart grid systems (Hong & Fan, 2016, pp. 916–917; Biswal et al., 2024, pp. 3655–3656). Despite the frequent mention of cloud computing and large-data pipelines as a set of required enablers, particular MLOps procedures of deploying forecasting models as a stable prediction service are seldom detailed (Al-Jumaili et al., 2023, pp. 1–4; Ullah et al., 2021, pp. 956–958).

Second, current forecasting research frequently compares approaches to a single model family, whereas less research compares conceptually different approaches under identical conditions. This restricts the knowledge of the performance of various modeling logics in practice. The comparison of LSTM, Prophet, and XGBoost is beneficial as it directly compares sequential deep learning, additive time-series decomposition, and tree-based

ensemble learning within a single framework (Harikrishnan et al., 2025, pp. 1–3; Comparing of Prophet, XGBoost and LSTM Models in Web Traffic Forecasting, 2024, pp. 1–2).

Third, research forecasting tends to report better prediction accuracy without making clear links between that enhancement and operational smart-grid choices. The usefulness of forecasting is that it helps in peak-load control, reserve scheduling, demand-response activation and energy balancing. There have been some attempts to relate forecasting to hybrid energy systems and demand-response applications, but they are quite isolated examples, not a component of an integrated decision-support system (Pramono et al., 2019, pp. 1–3; Harikrishnan et al., 2025, pp. 1–2; Pandya et al., 2025, pp. 1–3). Table 1 maps each identified gap to the corresponding thesis objective and expected contribution.

Table 1. Gap to objective mapping.

Gap	What is missing in the literature	Objective that addresses it	Expected contribution
Model-to-system gap	Limited deployment, monitoring, and retraining detail	Develop a cloud-based forecasting workflow	More realistic and implementable solution
Method-comparison gap	Few cross-family comparisons	Compare LSTM, Prophet, XGBoost, ARIMA, and Naive baseline under identical conditions; Diebold-Mariano significance tests for all pairs	Better insight into model logic under same conditions
Operational-value gap	Weak link between accuracy and grid decisions	Interpret results for demand response and peak-load management	Stronger decision-support relevance

This thesis fills these gaps by considering forecasting as a modeling problem, as well as a system-design and applied-analytics problem. It contrasts LSTM, Prophet, and XGBoost on

short-term load forecasting and puts them in a cloud-based system that can be scaled and provide near-real-time smart grid analytics (Harikrishnan et al., 2025, pp. 1–3; Ullah et al., 2021, pp. 956–958). The research links forecasting theory, application of AI, implementation in cloud and the applicability of smart grid in real world. Of these contributions, the Diebold-Mariano validated five-model cross-family comparison on national system-level data and the integrated six-stage MLOps pipeline are novel contributions of this thesis. The use of XGBoost and LSTM on Finnish load data, on the other hand, is an incremental contribution that builds on previous studies that have been conducted on load data from a single country.

The main research question of this thesis is:

How can cloud-based AI models enhance short-term load forecasting accuracy and support optimization in smart grids compared to traditional methods?

The thesis will answer this question by the following objectives:

Objective 1. To perform literature review of smart grid, short-term load forecasting, machine learning, cloud analytics and adaptive load management.

Objective 2. To compare the performance of LSTM, Prophet and XGBoost with ARIMA and a weekly Naive baseline under the same data and evaluation settings, and with all performance differences being confirmed by Diebold-Mariano statistical significance tests.

Objective 3. To build a cloud-based forecasting pipeline from data ingestion to model training, deployment and monitoring.

Objective 4. To assess the performance of the model based on RMSE, MAE, MAPE, and R^2, and to provide lower and upper classical reference bounds for the performance of the model with ARIMA and Naive respectively.

Objective 5. To apply findings to smart grid operational contexts, including demand response and peak-load management.

Objective 6. To assess the deployability of the proposed system in a real-world smart grid context.

These objectives ensure that the thesis goes beyond a mere model comparison and demonstrate how AI-based predictive analytics can be incorporated into a cloud-based operational environment to enhance decision-making in smart grids (Al-Jumaili et al., 2023, pp. 2–4; Biswal et al., 2024, pp. 3655–3656).

1.3 Definitions and scope of the study

This section defines the key terms used throughout the thesis and outlines the scope of the research.

A ***smart grid*** may be defined as an improved electricity system that integrates electrical infrastructures with electronic communication, sensing, and computational intelligence to provide electricity in a more efficient, dependable, and flexible form. According to Fang et al. (2012, p. 944), the ***smart grid*** is an extension of the conventional grid that follows a two-way electricity and information flow and supports smart infrastructure management and protection systems, both at the generation, transmission, distribution, and consumption. In a similar way, Ullah et al. (2021, pp. 956–957) stress that smart grids combine IoT devices, edge nodes, and cloud platforms, leading to large amounts of structured and unstructured data, which can be analyzed to maintain the grid, provide energy-related information services, and inform future decision-making. In contrast to traditional grids that are based largely on one-way communication and a low level of sensing, smart grids allow two-way

information flow and more adaptive decision-making regarding grid operations based upon real-time information.

A ***load forecast*** approximates the electricity demand in the future based on a certain period and at a certain degree of aggregation, generated due to a specific operational or planning need (Hong & Fan, 2016, p. 914). ***The short-term load forecasting (STLF)*** is very important in smart grid applications because it serves as the foundation for day-ahead scheduling, real-time balancing and operational decision making. (Masood et al., 2024, pp. 2–3).

The term short-term load forecasting is used in this thesis to denote predictions within a time horizon of one hour to about one day ahead and in some cases even a few days ahead, which are the most relevant time horizons for day-ahead scheduling and intraday grid operations. The three AI models outlined in this thesis are three different forecasting paradigms and are compared with two statistical baselines (Naive persistence and ARIMA) used as performance benchmarks.

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network that is adapted to nonlinear time-dependent effects and long-term trends in sequence data, and is therefore particularly effective in electricity-demand time series where historical consumption is a major factor influencing future demand (Albahli, 2025, pp. 1–2; Pramono et al., 2019, pp. 2–3). ***Prophet*** is an additive time-series model that breaks down a signal into trend, seasonality, and holiday or event components and has been applied successfully on energy time series where there exist strong seasonal patterns and calendar effects (Albahli, 2025, p. 2). ***XGBoost (Extreme Gradient Boosting)*** is a tree-based ensemble algorithm, which constructs numerous boosted decision trees, and has been demonstrated to be effective on tabular forecasting problems in energy, weather, and grid-stability contexts (Harikrishnan et al., 2025, pp. 2–3; Data Driven Forecasting for Grid Stability: Implementing XGBoost in Smart Energy Systems, 2025, pp. 1–2).

Demand response (DR) refers to the deliberate modification of electricity consumption by end-users when they receive a signal from the grid operator, like a price incentive or direct control command, to decrease peak demand, balance the system, or delay the addition of generation capacity (Harikrishnan et al., 2025, pp. 1–2). In this thesis, the focus is on the use of demand response as the main operational application of the STLF outputs: accurate forecasts of peak hours and low load windows are used to produce actionable DR signals.

Machine Learning Operations (MLOps) is a set of principles, processes, and tools that are used to automate and run machine-learning systems in production, which is found on the DevOps ideas of continuous integration, delivery, monitoring, and feedback**.** According to Kreuzberger et al. (2023, p. 1), ***MLOps*** can be defined as a new practice in ***Machine Learning*** engineering that seeks to deploy ML products to the production process in a reliable way by aligning the model, data pipelines, and infrastructure with organizational roles.

The study scope is set carefully to ensure analytical clarity and viability. First, the thesis is limited to short-term load forecasting and not medium-term or long-term forecasting since short-term predictions are directly useful in smart grids to support real-time balancing, demand response, and peak-load control (Hong & Fan, 2016, p. 914; Pramono et al., 2019, pp. 2–3). Second, the thesis focuses on AI models, where LSTM, Prophet, and XGBoost have been chosen as the primary AI comparative methods, supplemented by two statistical baselines: a weekly Naive baseline forecast and an ARIMA (1,1,1) rolling one-step-ahead forecast. The AI models were selected because they represent fundamentally different modelling philosophies like deep recurrent learning, additive decomposition, and gradient-boosted decision trees and because previous literature indicates that they are well suited to electricity-demand data with nonlinear, seasonal, and event-driven properties (Albahli, 2025, pp. 1–2; Harikrishnan et al., 2025, pp. 1–3; Rafi et al., 2025, pp. 1–2).

Thirdly, the thesis focuses on cloud-based analytics. The forecasting solution is not treated merely as a local one-off machine learning experiment but as a part of a system that can be deployed and managed in a scalable digital infrastructure. Smart grids survey on cloud-computing explains how cloud-based systems can be used to support data-intensive grid applications as well as allow elastic processing, but note issues with bandwidth, latency, and reliability (Cloud Computing Applications for Smart Grid: A Survey, 2015, pp. 1–3; Al-Jumaili et al., 2023, pp. 2–4). Investigations into smart-grid information processes also indicate the ways in which cloud, edge, and IoT layers can be interconnected in such a way that non-time-bound analytics (e.g., model training) can be hosted in the cloud, and decisions that are sensitive to latency can be facilitated closer to the data source (Ullah et al., 2021, pp. 956–958). In this thesis, the cloud focus implies that data ingestion, model training, and model inference are done in a way that is deployable and automatable, as per modern MLOps practices (Kreuzberger et al., 2023, pp. 1–2).

The empirical focus is Finnish national electricity consumption data, accessed via the Fingrid open data platform. Finland is selected as the empirical basis for three analytically grounded reasons. First, Fingrid is one of the few European transmission system operators to publish high-resolution, publicly accessible load data through an open API, making it directly suitable for reproducible academic research without data-access barriers. Second, Finland's electricity system is a challenging and representative test case for AI-based STLF due to high renewable penetration, high seasonal demand variability due to the Arctic climate, and an active demand response market. Third, there are well documented rules of balancing in the Nordic electricity market and 15-minute settlement intervals, which directly motivates the operational relevance of short-term load forecasting at the studied horizon in this thesis. Although the empirical basis is Finnish, the modelling framework, pipeline design and evaluation methodology are transferable to any European TSO in similar data availability and grid structure conditions.

1.4 Structure of the study

This thesis has been divided into five major chapters, which aim at supporting overall argument and research objectives of the study. The organization is based on the plan agreed during the thesis preparation, as it passes through the context and concepts to methods, empirical analysis, and conclusions.

Chapter 1 presents the introduction of the study where the background, research motivation, research gaps, research question, objectives, key definitions, and scope are presented. Chapter 1 establishes why AI-based STLF in cloud-enabled smart grids is a timely research topic, grounded in the rapid progress of grid digitalization and data-driven demand forecasting.

Chapter 2 introduces the literature review. It talks about smart grids and the key challenges of this metering, the various types and levels of load forecasting, and machine-learning and artificially intelligent solutions to energy-demand forecasting. The chapter also analyzes cloud- and edge-based analytics of smart grid data, the concepts of MLOps to run forecasting systems, and adaptive load-management and demand-response policies. Based on this, it constructs a theoretical ground that bridges the context of smart-grids, forecasting techniques, and cloud implementation, which gives the theoretical framework of the empirical work.

Chapter 3 describes the methodology, data sources, data preprocessing, model selection, model evaluation design and the cloud-based MLOps architecture.

The empirical results are summarized in Chapter 4, comparing all five models, interpreting the results in the context of demand response and adaptive load management, and assessing the MLOps pipeline.

Chapter 5 wraps up the thesis with a conclusion on the key findings of the study, the contributions made, and recommendations for future research. This last chapter summarizes the technical, practical, and conceptual knowledge acquired in the previous chapters and examines how AI-based, cloud-based forecasting systems can enhance operational intelligence and robustness in smart grids.

Generally, the structure is made in such a way that it flows logically through the context of the problem, theoretical background, methodological design, empirical analysis and lastly contribution and prospect. The following development makes the thesis coherent and each chapter contributes to the main argument, that is, the idea that cloud-based AI models, once integrated into a realistic data and MLOps system, can enhance the short-term load forecasting and assist smarter operational decisions in smart grids (Hong & Fan, 2016, pp. 914–916; Dong et al., 2025, pp. 1–3). Figure 2 summarizes this five-chapter structure and the logical flow connecting each stage of the thesis.

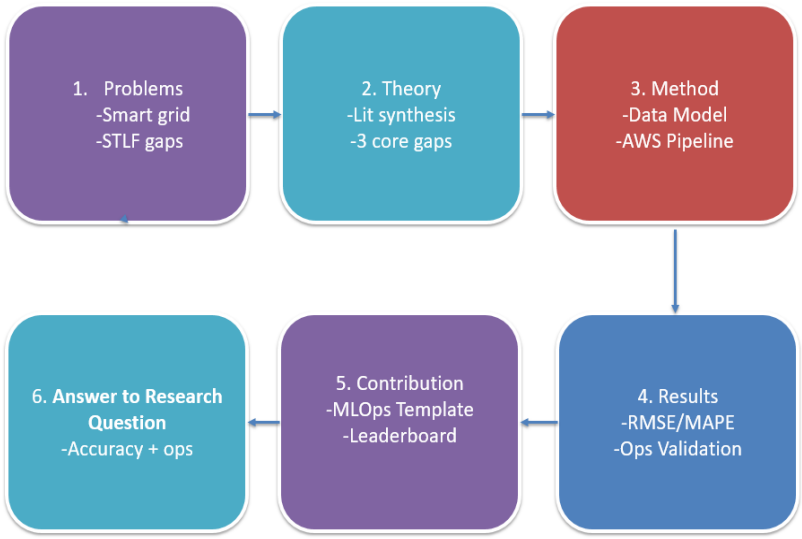

Figure 2. Conceptual Structure of the Thesis.

2 Literature Review

This chapter is a review of the theoretical and empirical literature behind the thesis. It concentrates on five key topics: smart grids and their challenges, types of load forecasting, artificial intelligence and machine learning methods of energy data, cloud-based analytics in energy management, and adaptive load management strategies in smart grids. Collectively, these themes establish the conceptual framework of the study of how AI models deployed on the cloud, including, but not limited to, LSTM, Prophet, and XGBoost, could help to forecast the short-term load and optimize operations in data-driven, decentralized modern electricity networks. The chapter concludes with a sub-section that summarizes the theoretical framework and presents a figure showing the links between the main keywords and the core issues investigated in this study.

2.1 Smart grids and their challenges

The traditional electricity grid was developed to accommodate a relatively stable energy system where large, centralized power plants would provide power to passive consumers in one-way flows of electricity. This model was adequate in the past few decades since demand was more predictable, generation was centralized, and operational control was restricted. Nevertheless, this design is becoming less suitable for modern electricity systems, which are characterized by renewable sources, distributed energy sources, electric vehicles, digitalized infrastructure, and more active end users. Fang et al. (2012, pp. 944–945) explain that the smart grid was developed as a reaction to these constraints by integrating a two-way flow of electricity and information with smart infrastructure, smart management, and smart protection systems.

A smart grid is therefore more than a technological upgrade. It is a paradigm shift in the process of monitoring, controlling and optimization of electricity systems. Smart grids allow operators to pay closer attention to the situation in the system and respond faster to the changes in demand and supply by integrating digital communication, real-time sensing, automated control, and data-driven analytics. This theoretically improves flexibility, performance and stability. In practice, however, the same features that render smart grids useful also make them difficult to predict and control.

Among the key issues is the greater fluctuation of demand and supply of electricity. Traditional grid demand exhibited more predictable consumption patterns and prediction could be more dependent on historical stability. Within a smart grid, however, load is influenced by weather, decentralized generation, price variations, prosumer behavior, electric vehicle charging and intermittent renewable energy. Hong & Fan (2016, pp. 914-

916) observes that when the dynamics of demand are non-linear and affected by a combination of interacting factors, then the traditional forecasting techniques are less effective. This implies that forecasting problem is no longer merely a question of how to predict a stable time series, but a question of how to model a system where behavioral, environmental, and technical variables interact in a simultaneous manner.

The second challenge is the volume and rate of data creation. Smart meters, sensors, weather feeds, control signals, grid-status logs generate unending streams of structured and unstructured data. Ullah et al. (2021, pp. 956–958) and Ali Al-Jumaili et al. (2023, pp. 14) show how this data-rich environment can be used to unlock opportunities to enhance analytics, congestion management, and reliability, but is very limiting to the traditional single-node processing. In other words, the same data, data that can be enhanced forecasting, in fact—must be stored in a scalable way, processed quickly, and have a strong analytics backbone. This is one of the reasons why cloud and edge computing have become prominent in the research of smart grid.

The other issue is interoperability. Typically, smart grid systems are linked with multiple vendors. It consists of the origin of devices, software platforms and communication protocols that are hard to ensure data formats consistency and integration. Such inconsistency makes it difficult to predict pipelines, since incompatible inputs may compromise the quality of the model and raise the preprocessing expenses. Meanwhile, cybersecurity and privacy risk is higher in a smart power grid. The improved interconnection opens new opportunities for unauthorized access, data leakage, and service interference. Smart grid forecasting cannot be regarded as a modelling only, but rather in the context of system integration, governance and risk management.

One of the key requirements in all these challenges is reliability. The electricity systems should be capable of operating with weather volatility, consumption shock, equipment,

anomaly and other disruptive factors. This complicates the process of forecasting and optimization compared to the traditional environment, where the tools are required to generate correct, quick, solid, and practically viable results in real-world operational contexts. The analytics of smart grids, then, must be able to assist in making decisions under uncertainty, not just to generate technically better forecasts.

These challenges indicate that smart grids require beyond modernization of equipment. They require sophisticated forecasting, scalable decision support systems and computing, which are capable of being run in realistic operations. Not only to digitalize the grid, but to ensure that the resulting data environment provides credible and useful predictions. The main challenges of the smart grids and their implications on forecasting model design are summarized in Table 2.

The literature on smart grid challenges is largely unified in diagnosing the core problems: variability, data volume, interoperability, and security. It is, however, considerably less unified on how forecasting should respond. One strand of research argues that more complex AI models are the primary solution, on the grounds that only nonlinear learning methods can capture the dynamics of modern demand (Dong et al., 2025, p. 1; Masood et al., 2024, p. 3). One of the opposing strands states that the complexity of models is an irrelevant factor compared to the quality of data infrastructure and pipeline reliability, as even a high-quality model cannot be used in practice, as it cannot be retrained, monitored, and served reliably on a large scale (Sculley et al., 2015, p. 2; Kreuzberger et al., 2023). This thesis takes the second position: the challenges described in this section are not primarily modelling problems, they are system design problems. A forecasting model that achieves low MAPE in a controlled experiment but cannot be maintained in a production pipeline does not solve the smart grid operator’s problem. This framing distinguishes the approach of this thesis from most of the STLF literature and motivates the cloud-based MLOps architecture described in Chapter 3.

Table 2. Challenges and forecasting implications.

Challenge	Why it matters	Implication for forecasting models
Demand and supply variability	Renewable generation, prosumers, and weather make load patterns unstable.	Models must capture nonlinear and time-dependent relationships, which supports the use of **LSTM** and other learning-based approaches.
Large data volume and velocity	Smart-grid systems produce continuous, high-frequency data streams.	Forecasting solutions must be scalable and efficient, making cloud-based analytics and automated pipelines important.
Interoperability	Data comes from multiple devices, vendors, and formats.	Models depend on strong preprocessing, feature engineering, and consistent data pipelines.
Cybersecurity and privacy	Digital connectivity increases exposure to misuse and disruption.	Forecasting systems must be designed with secure data handling and controlled deployment in mind.
Reliability under uncertainty	Forecasts must remain usable during shocks and abnormal conditions.	Models should be evaluated not only by accuracy, but also by robustness and operational usefulness.

Overall, the literature suggests that smart grid forecasting should be understood as a problem of both prediction and system design. This creates the basis for the next sections of the chapter, which discuss how forecasting models, machine learning methods, and cloud-based analytics respond to these challenges.

2.2 Load forecasting

Load forecasting can be defined as the process of estimating future electricity demand within a given period of time to aid in planning, scheduling, balancing, and control within power systems (Hong & Fan, 2016, p. 914; Mujeeb et al., 2019, p. 2). Due to the impossibility of economic storage of electricity on a large scale, supply and demand have to be tightly synchronized at any given time, which necessitates forecasting as one of the fundamental requirements of grid stability and efficiency (Hong & Fan, 2016, p. 914; Masood et al., 2024, p. 1). In this regard, load forecasting is not merely a technical operation, but also a decision-support operation which influences the stability and cost of electrical systems.

The energy value chain applies forecasts to guide scheduling of generation, reserves, energy trading, maintenance planning, and demand response. Both underestimation and overestimation can be expensive mistakes in forecasting: underestimation can result in shortages, instability, and overestimation can result in unnecessary generation and inefficiency (Hong & Fan, 2016, p. 915; Masood et al., 2024, p. 2). Both exogenous and endogenous factors (weather, prices, events, and economic activity) and temporal (hourly, daily, seasonal) shape demand (Hong & Fan, 2016, p. 916). This implies that forecasting models should not only embody the past load behavior, but also the effects of external variables. Table 3 categorizes the main forecasting horizons and the operational decisions each horizon supports.

Table 3. Forecasting horizons (adapted from Hong & Fan, 2016).

Category	Horizon	Typical applications
Very short-term	Minutes to 1 hour	Frequency control, spinning reserves
Short-term	1 hour to 1 week	Day-ahead scheduling, balancing, demand response
Medium-term	Weeks to 1 year	Fuel procurement, maintenance planning

Category	Horizon	Typical applications
Long-term	1+ years	Capacity expansion, investment planning

This distinction is important as the various horizons justify various operational decisions and thus must have various model properties (Hong & Fan, 2016, p. 914). Very short-term forecasting is focused on speed and responsiveness whereas long-term forecasting is focused on more general trends and planning assumptions. The current thesis is on short-term load forecasting, as this timeframe is closest to day-ahead scheduling, balancing, and demand response in smart grid settings.

Table 4. Forecasting approaches.

Approach	Main idea	Strengths	Limitations
Classical statistical models	Use historical time dependence and fixed assumptions	Interpretable, simple, effective in stable patterns	Less suitable for nonlinear and changing demand
Machine learning models	Learn relationships from data and exogenous variables	Better at nonlinearities and interactions	Need more data and careful tuning
Deep learning models	Learn complex temporal structures from sequential data	Strong for large, rich datasets	Less interpretable and more data-hungry

Regression and ARIMA are classical methods of load forecasting that have long been in use due to their transparency and relative simplicity to apply. However, they do not work well in cases where demand patterns are highly nonlinear or where there is more than one external factor interacting with each other (Hong & Fan, 2016, pp. 914–916). This weakness

is especially noticeable in the case of smart grids, where the variability of renewable energy, decentralized generation, and behavioral uncertainty make the demand pattern difficult. This trend, as the AI-based approaches learn nonlinear relations more effectively and can use richer input data, is therefore increasing in importance (Mujeeb et al., 2019, p. 3; Masood et al., 2024, p. 3).

However, forecast research should not be limited to prediction only, a model in a real-world smart grid environment must be robust, interpretable, and consistent with the operational processes (Masood et al., 2024, p. 4; Demand-side load forecasting in smart grids using machine learning techniques, 2024, p. 5). A very precise model that is hard to interpret or put into practice can be less helpful than a somewhat less precise but more stable and maintainable model. This is the main point of the current thesis, which not only compares LSTM, Prophet, and XGBoost as predictive approaches, but also as alternative modeling philosophies with varying capabilities of short-term predictions.

While the sophistication of the literature on load forecasting has grown significantly in terms of the sophistication of the models, it has arguably decreased in terms of the rigor of the evaluation, and this is not by chance. They are commonly used metrics, such as RMSE and MAPE, which are representative of the benchmarking culture of the machine learning community, where models are evaluated on held-out test sets selected by the researcher. This culture values accuracy enhancements on the selected data set rather than generalizability across contexts and robustness in deployment conditions. The incentive of academic publishing exacerbates this: a paper that demonstrates that a new transformer architecture can reduce MAPE by 0.3% on the residential dataset is publishable, a paper that demonstrates that a simpler model is easier to maintain in production is not. Consequently, the STLF literature has increasingly focused on a metric (single-window test accuracy) that is not relevant to the actual evaluation of forecasting systems in grid operations, where the issue is not who wins one test period, but who loses the least across

different seasons, demand regimes and data quality conditions. This thesis directly tackles this question by using Diebold-Mariano significance tests to validate whether the differences in performance are statistically real or simply a reflection of the test window, and by cross-validating the model performance across multiple folds to determine if the model performance is stable or if it is due to a favorable test window.

2.3 AI and machine learning techniques for energy data

Sequential and high-dimensional energy data have been especially impacted by deep learning, and recent STLF surveys show architectures such as CNNs, RNNs, LSTMs, and transformers dominating (Patsakos et al., 2022, p. 2; Dong et al., 2025, p. 1). This thesis contrasts three different AI paradigms which are LSTM, Prophet, and XGBoost alongside ARIMA (1,1,1) and a weekly Naive baseline that provide classical statistical reference bounds (Albahli, 2025; Harikrishnan et al., 2025; Hong & Fan, 2016). Table 5 summarizes the forecasting logic, main strengths, and typical limitations of each of the three AI models compared in this thesis.

Table 5. Comparison of LSTM, Prophet and XGBoost models.

Model	Forecasting Logic	Main Strengths	Typical Limitations
LSTM	Gated recurrent neural network for sequential modeling	Captures nonlinear temporal dependencies; excels with high-frequency load patterns	Data-intensive; computationally heavy; black-box interpretability
Prophet	Additive decomposition (trend + seasonality + holidays)	Interpretable; robust to missing data/holidays; simple configuration	Less flexible for irregular nonlinear dynamics

Model	Forecasting Logic	Main Strengths	Typical Limitations
XGBoost	Gradient-boosted decision trees on engineered features	Superior tabular performance; feature importance; handles interactions	Feature engineering dependent; not natively sequential

LSTM maintains long-range dependencies, using memory cells, and gates, which make it a good choice in load data, where recent observations have a strong predictive power of the near-term demand (Dong et al., 2025, p. 2; Masood et al., 2024, p. 4). It has been established through research that LSTM outperforms baselines in nonlinear, multi-seasonal trends (Albahli, 2025, p. 2). Prophet breaks series down into interpretable elements, which are appropriate to the powerful calendar structure of demand. It has a good benchmark in energy applications, although it has small nonlinearity capabilities (Patsakos et al., 2022, p. 5).

XGBoost uses boosted decision trees with engineered features such as lag variables, weather conditions, and holidays. Studies on residential demand response and smart homes demonstrate their strong predictive performance (Harikrishnan et al., 2025, p. 2; Electric Load Forecasting for Internet of Things Smart Homes, 2021, p. 3). This comparative framework includes sequential memory modelling (LSTM), structured decomposition (Prophet), and feature-based ensemble learning (XGBoost), representing three fundamentally different approaches to the STLF problem (Dong et al., 2025, p. 3; Patsakos et al., 2022, p. 6). Each has limitations: LSTM is data- and computation-heavy; Prophet less adaptable to irregular nonlinear dynamics; XGBoost relies on the quality of feature engineering (Masood et al., 2024, p. 5).

The issues of interpretability and deployability are gaining prominence in the forecasting literature. SHAP and permutation significance support the driver weighting (weather, calendars) which is significant in controlled energy industry (Demand-Side Load Forecasting in Smart Grids, 2024, p. 7). The hybrid models, including CNN-LSTM and XGBoost-RNN, are hybridizations that combine the advantages of models to learn spatio-temporal effects (Mujeeb et al., 2019, p. 4).

This results in a structural disconnection that is not directly studied in literature. The prevalence of LSTM and transformer-based architectures in current surveys is indicative of what is good on benchmark datasets with clean, pre-processed, and available on a GPU-based device, which are circumstances that characterize a research setting, not a utility. What makes this disconnect continue is the fact that the researchers who create forecasting models are nearly never the ones who implement them. Academic researchers are usually judged by new ideas and improvements in benchmarks, whereas practitioners are judged on system reliability and operational maintainability, a divergence that Sculley et al. (2015, p. 2) identify as a root cause of the gap between ML research and production systems. This separation of labor implies that the operational limitations that are the most important in practice, including inference latency, retraining frequency, explainability to non-technical stakeholders, and compatibility with existing SCADA systems, are simply not present in academic literature since they are not visible to the individuals who write the literature. In comparison, XGBoost is able to be trained in seconds on CPU hardware and can generate models that can be immediately interpreted through the feature importance. The question of which model is lower RMSE on a clean test set is not the question of choice, but which model can withstand contact with a real production environment. This is one of the important questions addressed in this thesis.

It is important to unpack the systematic publication bias in AI-based STLF literature, rather than just acknowledging it. The chances of a study being published if it concludes that AI

models perform better than a statistical baseline are significantly higher than the chances of a study being published if it concludes that there is no evidence of improvement or that the AI models perform worse. This is because journals and conferences in the field have traditionally favored novelty and performance improvements over replication and failure analysis. This bias is exacerbated: the more positive results are gathered, the more they are cited in survey articles as proof of the superiority of AI, the more they are cited in other deep learning research papers, the more they are cited in other survey articles, etc. This puts the field in a vicious cycle of self-reinforcement, in which the apparent superiority of deep learning indicates the incentive structure of academic publishing as much as it indicates the true abilities of these models. The practical implication of this thesis is important: The widespread claims in the literature of the STLF that LSTM outperforms the baselines do not mean that it will do so in the same evaluation settings with statistical significance testing on the data of the national system in Finland. It is this empirical question that this thesis will answer, and the answer as presented in Chapter 4, goes against the current narrative.

2.4 Cloud-based analytics in energy management

As energy systems become digital, forecasting is no longer a mere modeling issue; it is also a problem of infrastructure. Cloud-based analytics provide the scalable storage, processing, and automation needed to transform smart grid data into operational insight. (Ali Al-Jumaili et al., 2023, p. 1; Masood et al., 2024, p. 5).

Smart grids generate heterogeneous and high-frequency data of smart meters, weather APIs, SCADA systems, sensors, and market feeds, which are typically out of scope of the processing functionality of old-fashioned local systems (Mujeeb et al., 2019, p. 4; Cloud Computing Applications for Smart Grid, 2013, p. 2). Cloud analytics can be useful in this

situation not in necessarily storing more data, but in facilitating an end-to-end data process from intake to decision support. Table 6 lists the core cloud functions and their specific roles in energy analytics and in this thesis.

Table 6. Cloud functions and forecasting value (adapted from Al-Jumaili et al., 2023; Masood et al., 2024; Ullah et al., 2021).

Cloud function	Role in energy analytics	Relevance to this thesis
Data ingestion	Collects smart-meter, weather, and grid data from multiple sources	Enables unified input for STLF
Centralized storage	Stores large volumes in data lakes or warehouses	Supports historical training data and feature history
Preprocessing pipelines	Cleans, transforms, and engineers features	Improves model readiness and reproducibility
Managed ML services	Automates training and experimentation	Fits comparison of LSTM, Prophet, and XGBoost
Deployment services	Serve models through APIs or containers	Supports forecasting-as-a-service
Monitoring tools	Track drift, latency, and performance	Makes retraining and maintenance possible
API interfaces	Connect forecasts to operational tools	Links predictions to real grid decisions

Cloud infrastructure is particularly beneficial as it can be expanded as the number of devices and the number of analytic tasks increase. With the increasing popularity of the behind-the-meter systems and home energy management applications, cloud platforms allow supporting experimentation, quick model testing, and deployment without an expensive on-premises investment (Ali Al-Jumaili et al., 2023, p. 2; Masood et al., 2024, p. 7). This applies to the current thesis since the forecasting framework is not considered an

independent experiment, but a deployable pipeline based on AWS with retraining and API delivery.

Simultaneously, cloud analytics cannot be taken at face value, but it must be critically assessed. The primary benefit of cloud systems is scalability: they can process large amounts of data, enable centralized management of models, and make integration between tools and services easier. Nevertheless, the deployment to the clouds can also create latency, reliance on bandwidth, and increased vulnerability of sensitive information during the process. This is why cloud analytics is powerful in terms of training, aggregation, and batch processing, whereas edge computing is frequently more appropriate in the context of latency-sensitive inference near grid assets (Ullah et al., 2021, p. 957; Masood et al., 2024, p. 8). The literature thus recommends that cloud and edge be viewed as complementary as opposed to competing architectures. Table 7 compares cloud and edge architecture across their strengths, limitations, and most suitable use cases in smart grid settings.

This trade-off is significant to smart grid forecasting since the aim of the operation is not merely the model accuracy, but also the timely and reliable provision of forecasts. Cloud-based MLOps can help to mitigate this, providing validation, versioning, monitoring, rollback, and retraining throughout the model lifecycle, minimizing technical debt and enhancing operational maintainability (Sculley et al., 2015, p. 2; Kreuzberger et al., 2023). Practically, model training, historical aggregation, and deployment management can be done on the cloud layer, whereas time-sensitive interactions, where the speed of response is the most important factor, can be done on edge or local systems.

Table 7. Cloud versus edge comparison (adapted from Ullah et al., 2021; Masood et al., 2024).

Architecture	Strengths	Limitations	Best use in smart grids
Cloud	Scalable, centralized, easy to automate, suitable for training and storage	Latency, bandwidth dependence, privacy concerns	Model training, retraining, monitoring, API serving
Edge	Low latency, local responsiveness, better for near-device decisions	Limited compute and storage, harder to manage at scale	Real-time inference, local control, fast operational response

It is worth asking why the gap between architectural ambition and operational evidence persists in cloud analytics literature. The answer lies partly in the incentive structures of both academia and industry. Academic researchers publish cloud architecture papers by describing what a system could do, not by operating it for twelve months and reporting failure rates. Vendors in the industry release whitepapers outlining the functionality of their platforms as opposed to the deployment challenges of their customers. The outcome is a literature that is nearly unanimously positive on cloud analytics in energy management, filled with systems that are proven but not often stress-tested in actual operation. Sculley et al. (2015, p. 2) identified this problem in ML systems generally: the proportion of code in a real ML system is dedicated to the model is very small, with most of the code being data pipelines, serving infrastructure, and monitoring, which are exactly the parts that are not described in architecture papers. The consequences of such a cloud pipeline that doesn't manage API rate limits, doesn't handle daylight savings time and time zone changes, or doesn't handle changes in the underlying data schema are dire, especially for smart grid forecasting, where the failure to alert on forecast corruption is a silent failure. This thesis directly addresses these realities, recording four modes of pipeline failure experienced in development, and how these were addressed, considering operational robustness as a first-class evaluation criterion, in addition to forecast accuracy.

2.5 Review of adaptive load management strategies

Adaptive load management is a collection of coordinated measures to change electricity demand in reaction to system conditions, predicted load, price indications, and operational priorities. In contrast to the traditional grid management, which was more dependent on the fixed planning and centralized control, adaptive management is constructed based on flexibility, automation, and data-driven responsiveness (Rizvi and Chaturvedi, 2023, p. 2; PHOENIX H2O, 2022, p. 5). Its greatest advantage lies in the assumption that electricity demand is not fixed; instead, it can be shifted, reduced, aggregated, or rescheduled when the grid experiences stress.

This is important as supply-side planning is no longer the sole determinant of smart grids. The demand is becoming more unpredictable and changeable with renewable generation, distributed energy resources, electric vehicles, and active consumers, thereby placing more emphasis on demand-side interventions. Here, the facilitating layer to adaptive action is short-term load forecasting: a forecast can only be useful when it assists operators in making decisions about when and how to intervene. It is the suggestion of the literature that forecasting and load management cannot be seen as two distinct operations, but as the two components of one working mechanism (Hong & Fan, 2016, p. 915; Harikrishnan et al., 2025, p. 3). Table 8 presents the four-core adaptive load management strategies, their mechanisms, the role of STLF in each, and their documented operational impacts.

Table 8. Core strategies and mechanisms.

Strategy	Description	Key Mechanisms	STLF Role	Operational Impact
Demand Response (DR)	Usage changes during stress via incentives/control	Direct load control, dynamic	Spike prediction, trigger timing	15–25% peak reduction (Harikrishnan

Strategy	Description	Key Mechanisms	STLF Role	Operational Impact
		pricing, automated thermostats		et al., 2025, p. 3)
Peak Shaving	Maximum load reduction via multi-asset coordination	Storage discharge, DR activation, pricing signals	Peak magnitude/location forecasting	Defers $50–200/MWh peaker plants (Rizvi & Chaturvedi, 2023, p. 5)
Load Shifting	Peak-to-offpeak transfer of flexible demand	EV smart charging, industrial rescheduling, appliance deferral	Optimal transfer windows	10–20% daily variance smoothing (PHOENIX H2020, 2022, p. 8)
Valley Filling	Off-peak consumption increase	Energy arbitrage, storage charging, night-time industry	Low-demand window identification	Improves asset utilization 30–40%

Of these strategies, the most debated is demand response due to its direct alteration in consumption because of utility programs, market prices, or automated control signals. Demand response may be incentive based, price based or direct control based. Incentive-based programs can reward customers to cut loads, price-based programs can employ time-of-use or real-time pricing to influence behavior and direct-control programs can enable

utilities to limit or pre-condition flexible loads. In each of them, STLF enhances the time at which intervention is likely to take place (Hong & Fan, 2016, p. 916; Harikrishnan et al., 2025, p. 4).

Peak shaving and load shifting demonstrate the importance of predictive control. Peak shaving is not just about consumption reduction, but a process that integrates batteries, demand response, pricing, and distributed generation to flatten the load curve without harming service quality. By contrast, load shifting relocates flexible demand between high-cost and low-cost periods, e.g. EV charging at night or industrial scheduling not during the afternoon peak. The strategies are most effective when the forecast is precise enough to determine the magnitude as well as the timing of the stress period. That is, STLF identifies an intervention as early, late, excessive or insufficient. Table 9 links each operational target to the corresponding forecast requirement and expected grid outcome.

Table 9. Strategy, forecast, and outcome.

Operational target	Forecast need	Expected outcome
Peak demand reduction	Predict peak height and timing	Lower system stress and reduced peak costs
Renewable integration	Anticipate supply-demand mismatch	Less curtailment and better balancing
Load flexibility activation	Identify flexible loads and intervention windows	More effective demand-side coordination
Reliability improvement	Detect periods of high risk	Fewer instability events and better reserve planning

It is also indicated in the literature that adaptive load management is being adopted increasingly using aggregated platforms and virtualized control systems. Aggregators bring

together large numbers of small consumers into dispatchable capacity, and virtual power plants manage distributed energy resources, batteries, and flexible demand. Forecasts are also used to pre-cool, pre-heat or re-scheduling of HVAC loads before they are expected to peak. These approaches bring out a crucial fact: adaptive management is not a single approach, but a set of interventions that all require the quality of the input of forecasting.

The demand response literature carries a structural optimism bias that is rarely acknowledged, and understanding why it exists is as important as recognizing that it does. Many of the studies reporting large peak reductions are produced by the same researchers or institutions that design and implement the programmes, creating conditions in which positive results are more likely to be reported and replicated. Pilots that fail to achieve meaningful load shifts, or where consumer fatigue causes engagement to collapse after the first season, are rarely reported in academic literature because they offer no publishable novelty. The performance figures that survive in survey papers therefore represent a best-case distribution, not a typical-case distribution. Furthermore, the behavioral assumptions embedded in most demand response models, namely that consumers respond to price signals rationally and consistently, are at odds with decades of behavioral economics research showing that energy consumption is habitual, inattentive, and asymmetric in its response to gains and losses. A forecasting model that is accurate in predicting peak demand is not the solution to this problem, it is just a signal. The impact of that signal on actual load reduction will depend on the design of the intervention, the attributes of the consumer population, and the institutional structure of the grid operator. This thesis does not seek to model consumer behavior, but rather views the forecast as one part of a decision chain, not the end goal itself, which is often lost in the literature on demand response because of its emphasis on accuracy.

2.6 Summary of theoretical framework

The reviewed literature in this chapter creates a coherent theoretical framework that is constructed based on five interrelated pillars: smart grid architecture, load forecasting, AI/ML forecasting, cloud analytics, and adaptive load management. Combined, these pillars provide an explanation of the generation of complex data in digitalized electricity systems, the transformation of this data to forecasts, the operationalization of these forecasts in cloud-based systems and the use of the forecasts to support adaptive grid control. Table 10 summarizes the five theoretical pillars, their core concepts, and the main challenge each address.

Table 10. The five pillars.

Pillar	Core concept	Main challenge
Smart grid architecture	Digitalized electricity networks with bidirectional flows	Variability, data volume, cybersecurity, interoperability
Load forecasting	Demand prediction across horizons, with STLF as the focus	Nonlinearity, multi-seasonality, uncertainty
AI/ML forecasting	LSTM, Prophet, XGBoost + ARIMA/Naive baselines; DM-validated accuracy differences	Accuracy versus interpretability trade-offs
Cloud analytics	Scalable MLOps infrastructure	Security, reliability, lifecycle management
Adaptive management	Demand response, peak shaving, and load shifting	Translating prediction into action

The framework is grounded in a definite logic of interdependence. Smart grids provide the environment under which forecasting is more challenging due to the variability in loads due to the generation of renewable sources, distributed energy resources, and prosumer behavior. Such complexity makes the short-term load forecasting even more necessary, as

the classical methods cannot adequately explain the nonlinear nature of demand anymore (Fang et al., 2012, p. 945; Hong & Fan, 2016, p. 916). Forecasting, in turn, encourages the application of AI and machine learning techniques like LSTM, Prophet, and XGBoost since they can capture temporal dynamics, decomposition structure, and feature interactions, among others, more efficiently than more conventional linear models.

Forecasting based on AI is only useful when it is integrated into a system that can be implemented and sustained in practice. This is why cloud analytics and MLOps are not discussed here as secondary implementation details, but as a part of the forecasting value chain. Scalable training, automated retraining, monitoring, and API delivery transform research models into tools that can be used in practice repeatedly in the real world (Ali Al-Jumaili et al., 2023, p. 2; Masood et al., 2024, p. 6). This is particularly crucial in smart grids, where forecasting outputs need to be provided in a reliable and timely manner to aid in operational decision-making.

The last pillar is the adaptive load management that completes the prediction-action loop. Demand response, peak shaving and load shifting require good short-term predictions since the time of acting is what defines the success of such action. The forecasting pipelines that are enabled by MLOps thus not only enhance the accuracy of predictions; they also offer the operational foundation of adaptive grid control and more efficient energy management (Rizvi and Chaturvedi, 2023, p. 6; Harikrishnan et al., 2025, p. 4).

The main argument of this thesis statement is that **AI-based STLF in cloud-MLOps systems improves both predictive accuracy and operational decision support in smart grids**.
This argument connects theory with practice: the value-forecasting process is not just a function of the model, but the infrastructure, and the operational use of the model.

Each pillar directly feeds into one of the components of the research design. The smart grid architecture pillar serves as the foundation for the problem context that is set out in Chapter 1, and it gives the rationale for using the Finnish national grid data as the empirical basis. The load forecasting pillar is the basis for the STLF horizon and evaluation metrics used in Chapter 3. The three main comparative models, LSTM, Prophet, and XGBoost, were selected for this pillar of AI and ML because they are the most commonly used models, and Diebold-Mariano tests were applied to validate performance differences. The cloud analytics pillar inspires the six-stage AWS MLOps pipeline outlined in Section 3.4, where deployment infrastructure is not an afterthought, but a research variable. The adaptive management pillar links the forecasting results in the previous chapter (Chapter 4) with operational demand response and peak shaving scenarios, thus closing the prediction-to-action loop. This letter provides a context for the empirical chapters, which are not simply model benchmarks but a theory-based examination of the design, assessment and application of forecasting systems in real smart grid environments.

The five pillars examined in this chapter are not treated in the same manner in the literature. The two most advanced pillars are smart grid architecture and load forecasting, where there are well-developed theoretical frameworks and decades of empirical research. As demonstrated in Section 2.3, AI/ML forecasting is the fastest growing pillar and the most evaluation-biased. As Section 2.4 makes clear, cloud analytics is the most ambitious, but least empirically proven pillar in operational contexts. The most application-specific pillar is also the most behaviorally optimistic, as Section 2.5 illustrates, adaptive load management. This is an asymmetry that is important: pillars that are most often mentioned in the literature are not necessarily most operationally grounded. The contribution of this thesis lies in the middle of the two least-validated pillars (cloud analytics and adaptive load management), and deals with them as empirical testable components of a system, rather than as aspirational capabilities.

This framework will be empirically validated in four areas: model accuracy, operational performance, management effect, and cloud deployment design, in chapters 3 and 4. All five models will be evaluated by RMSE, MAE, MAPE, R^2 and Diebold-Mariano statistical significance tests. The operational performance will be evaluated by the MLOps pipeline maturity evaluation and failure mode analysis. The management effect will be evaluated based on the outputs of forecast-guided demand response and peak shaving. Cloud deployment design will be evaluated through the AWS architecture assessment and MLOps maturity classification against Kreuzberger et al. (2023) dimensions. Figure 3 visualizes this theoretical framework, mapping the five pillars and their interdependencies as they apply to AI-based STLF in cloud-enabled smart grids.

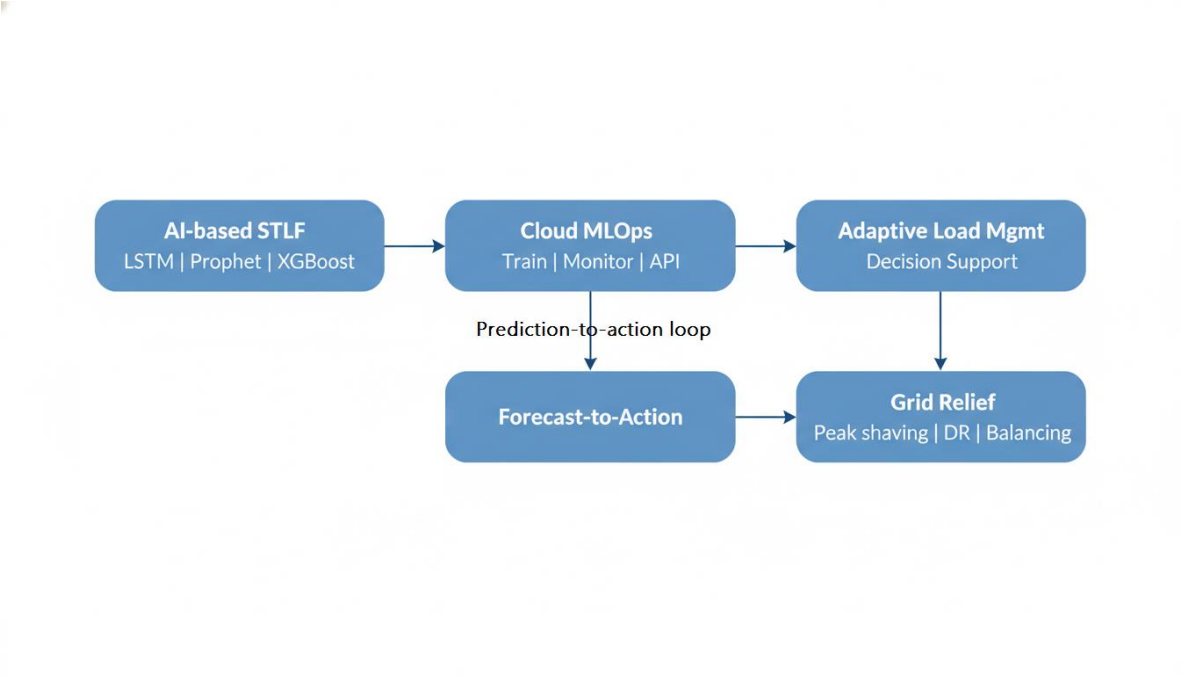

Figure 3. AI-based STLF in cloud-MLOps enabling adaptive load management and grid relief in smart grids.

The thesis is informed by the fact that the combination of STLF and cloud-based MLOps makes forecasting more helpful to adaptive load management than when models are viewed as independent analytical instruments. This suggestion aligns with the trend of the

recent literature, which is getting increasingly linked between forecasting, cloud infrastructure, and operational control on the cloud-edge-IoT spectrum. It also gives the methodology in Chapter 3 its conceptual background, with each pillar being operationalized using data pipelines, model comparison, deployment logic, and adaptive management simulation.

3 Methodology

This chapter describes the methodology to conduct the study of the role of cloud-based artificial intelligence in enhancing short-term load forecasting and assisting smarter operational decisions in smart grids. The research is an applied, comparative, and design-based study, as it does not only test forecasting models based on their predictive accuracy but also on their applicability in a cloud-based analytics system. In that regard, the methodology is a mix of quantitative model comparison and a practical systems viewpoint. A pragmatist research philosophy guides the research, as the aim of the thesis is to generate measurable and practical knowledge. Pragmatism would be suitable in this case since the research is not just seeking to describe forecasting behavior theoretically, but also to determine if AI models could work in a realistic smart grid environment. This philosophical stand is defending an approach that involves a combination of numerical forecasting test and cloud architecture design and expert validation.

The proposed research design is a comparative case study which will be based on short-term load forecasting within smart grid. The case study method is suitable as the thesis investigates an applied setting in detail rather than attempting to develop a general forecasting theory. The comparison is of five models, three AI paradigms (LSTM, Prophet and XGBoost) compared to two statistical baselines (a weekly Naive baseline and a rolling

ARIMA (1,1,1)). This design will determine how effective each AI paradigm is compared to the classical methods and the incremental value of machine learning in a smart grid setting.

The study is predominantly a quantitative study because its nature of evidence lies in time-series data, model outcomes, and statistical performance metrics such as RMSE, and MAPE. At the same time, the thesis contains a small qualitative portion, a semi-structured interview with an expert. The practical applicability of the proposed forecasting and deployment strategy, namely, the practical implementation, decision support and operational viability is evaluated with the help of this interview. By doing so, the interview is used as a validation layer, and not as the primary source of evidence.

The research onion also influences methodological design, as it helps to organize the study, beginning with the philosophical assumptions, and concluding with data collection and analysis. The top level, pragmatism, provides the epistemological basis of the synthesis of the outcomes of numerical forecasting with the practical deployment factors. The study is based on a comparative case study at the strategy level. On the methods level, it applies model training, evaluation, design of cloud deployment, and expert validation. This stratification will ensure that the methodology is geared towards the general purpose of the thesis. The research onion model is shown in Figure 4, illustrating how the philosophical and methodological layers of this study are structured from the outermost epistemological assumptions inward to the data collection and analysis techniques.

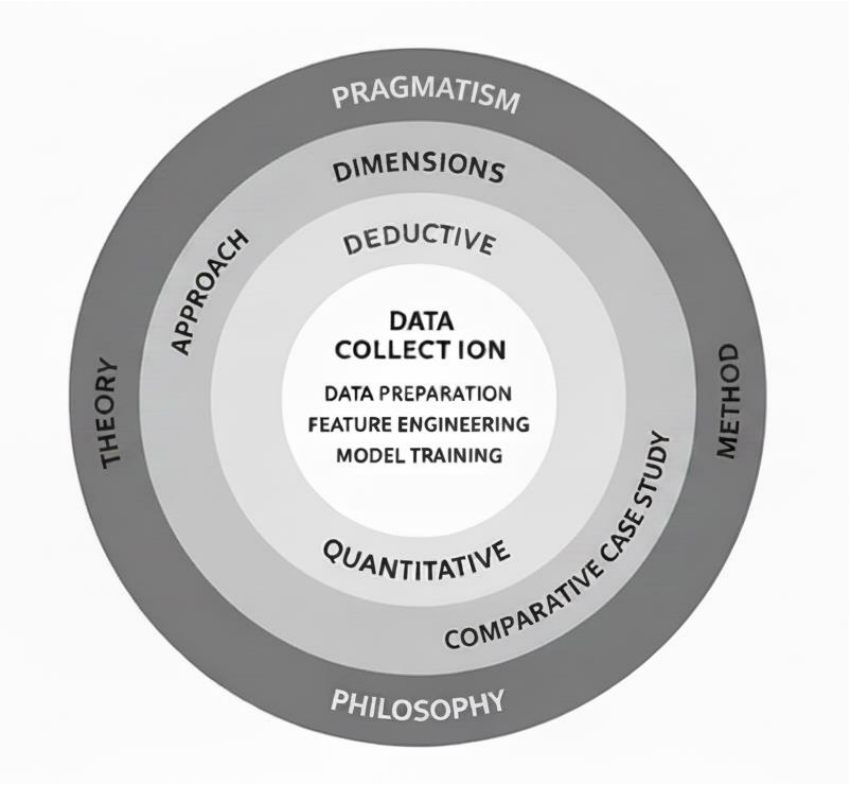

Figure 4. Saunders' Research Onion (adapted from Musundire, 2025).

Among the key features of this thesis, it is important to note that the forecasting models are not viewed as an independent algorithm, but rather a part of a bigger cloud-based analytics system. This is why the methodology will entail the development of an AWS-based architecture that will enable data ingestion, preprocessing, training, monitoring, and API-based prediction delivery. This study is not only a comparison of models; it also looks at how to make the forecasting pipeline work in a way that is scalable, maintainable, and relevant to smart grids.

The methodology is based on four parts that depend on each other: data preparation, model comparison, cloud deployment, and operational validation. The following sections talk about these things. Together, they provide a structured way to evaluate how AI-based

STLF can improve both the accuracy of predictions and its usefulness in making decisions in smart grids.

3.1 Data sources

The empirical section of this thesis is grounded on the combination of quantitative and qualitative sources of data that jointly reflect the key factors of short-term electricity demand. The primary quantitative data source is the Fingrid Open Data REST API, which provides system-level electricity consumption data for Finland. In this study, the main target variable is national electricity load measured in megawatt-hours per hour (MWh/h), retrieved from Fingrid dataset. This source is appropriate because it represents the actual smart-grid situation in a European setting and allows analyzing day-ahead and intraday load forecasting.

The dependent variable is national electricity load, measured in MWh/h. A historical time of two years applied in such a way that the models can acquire daily, weekly and seasonal trends. In cases where it is feasible, longer duration is desirable to enhance robustness, and to subject the models to several seasonal cycles. This renders the load series the key input to all forecasting experiments.

The second important source of data is weather data provided by the Open-Meteo API. Weather variables are correlated with the same hourly time index as the load data and correlated to the corresponding Finnish location or load area. The weather feature that is most significant is the temperature as it is closely associated with heating and cooling requirement during electricity use. Humidity, wind speed, and atmospheric pressure are also other variables that are added where possible to enhance model performance.

The third category of inputs consists of calendar and event features derived from the timestamp and Finnish public holiday calendars. These include hour of day, day of week, day of month, month, weekend/weekday flag, seasonal indicators, and public holiday flags. These variables aid the models to include daily behavioral patterns in electricity demand such as weekday-weekend variations, and repeat peak-load intervals.

Besides quantitative data, the research comprises one semi-structured interview with a professional in a smart grid. It provides qualitative validation of the proposed forecasting and cloud-based deployment approach. This interview helps evaluate the operational feasibility, decision support, and real-world implementation challenges of the system. The relevance of the proposed AWS-based architecture to the real world, the utility of short-term forecasts in operations, and the challenges that might occur in the implementation phase are evaluated using this interview. The interview is not considered as distinct statistical data; it is rather used to interpret the quantitative findings and provide the practical background of the discussion.

Table 11. Main data sources used in the study.

Metric	**Value**
Primary data source	Fingrid open data API
Weather source	Open-Meteo API
Time span	2 years (January 2023 – December 2024)
Frequency	Hourly
Approx. observations	17,376 hourly records (after resampling and preprocessing)
Main target variable	Electricity load
Exogenous variables	Temperature, humidity, wind speed, pressure
Calendar features	Hour, weekday, month, weekend, holiday

Metric	Value
Qualitative validation	One semi-structured expert interview

3.2 Data preprocessing and feature engineering

Raw load and weather data sourced at Fingrid and Open-Meteo cannot be modeled directly due to the possibility of missing values, irregular timestamps and variables that are yet to be converted into a form that can be used to forecast. Preprocessing data is thus a very important methodological step. It enhances the quality of data, offers internal consistency among sources, and converts raw measurements into meaningful inputs that the AI models can learn successfully.

Timestamp standardization and alignment is the initial stage of preprocessing. Any datasets are translated to a standardized datetime format with a clear management of time zone rules and daylight savings transitions when necessary. The combined data will be processed in the chronological sequence and indexed by time in such a way that each row will be a different hour within the time frame of observation. Duplicated timestamps are removed or combined and missing hourly timestamps are detected to be processed. This step is especially critical as anything that shifts time indices off can corrupt lag properties, rolling statistics, and train-test-validation splits.

The second stage is cleaning the data. Missing values, unrealistic spikes, and implausible negative or zero values where they do not make sense are investigated in the load and weather series. Missing hourly values in the load or temperature series that are isolated may be filled in using time-sensitive interpolation techniques like linear interpolation, forward/backward filling, or short rolling averages, and documented explicitly. Large gaps or periods when the load and weather data are not accurate can be not considered during

the training period or simply indicated not to introduce some false data. Extreme values are checked by sight and statistics, aware that some visible spikes may be real peak events which models need to learn to cope with instead of treating them as errors.

The Fingrid and Open-Meteo datasets are then integrated together into an analytical table with the hourly timestamps as a shared key after cleaning. The feature engineering is done once the integration is complete. XGBoost and Prophet in particular benefit from feature engineering, and LSTM input sequences are better and more stable with feature engineering.

The initial category of engineered features includes calendar variables that represent regular behavioral patterns: hour of day, day of week, month, weekend and weekday, seasonal names (e.g. winter, spring, summer, fall). These variables assist the models to identify common trends in load on days and seasons. The encoding of selected time variables (e.g., hour of day, day of week) can also be done using sine-cosine transforms to more naturally encoding cyclical structure (e.g. hours 23 and 0 are not far numerically apart).

The second set is lagging features based on the load series. These are short lags such as load one hour prior, and various additional lag hours (e.g. t−1, t−2, t−3), and daily and weekly lags (t−24, t−48, t−168) where the data is available. These lags enable models to take advantage of autocorrelation and re-occurring time patterns. The third set of characteristics involves rolling-window statistics. There are 3-hour, 24-hour and 7-day rolling means and standard deviations, which capture short-term trends and local volatility in demand.

The fourth category of engineered variables is the weather-related one. Raw temperature and humidity are stored, and other nonlinear transformations are computed where convenient such as the squared temperature or simple heating and cooling degree

approximation to standard comfort levels. The derived variables come in handy in allowing the models to capture the nonlinear behavior of electricity demand to extreme temperatures. Weather changes in the past few hours can also be coded using differences and short rolling summaries.

The feature scaling before modelling is done in a manner that will not leak information. Scaling parameters (means and standard deviations) are computed on the training set and applied to the validation and test sets without refitting. With the LSTM model, all continuous variables are standardized or Min-Max scaled because neural networks are sensitive to numbers. XGBoost is more resistant to changes in scale, but a standard scaling process simplifies documentation and can provide better numeric accuracy. Prophet also works on the original scale series but may also enjoy standardization of repressor where necessary.

Finally, once the whole feature set is constructed, the data is chronologically separated into training, validation, and test data. The thesis uses a 70-15-15 split: the initial 70 percent of the time is spent to train the model, the second 15 percent to validate and pick the model, and the final 15 percent to test the model out-of-sample. The time-series preservation of the problem prevents information leaking into model training of future information and so inflating performance estimates. Figure 5 provides an overview of the complete preprocessing and feature-engineering pipeline described in this section.

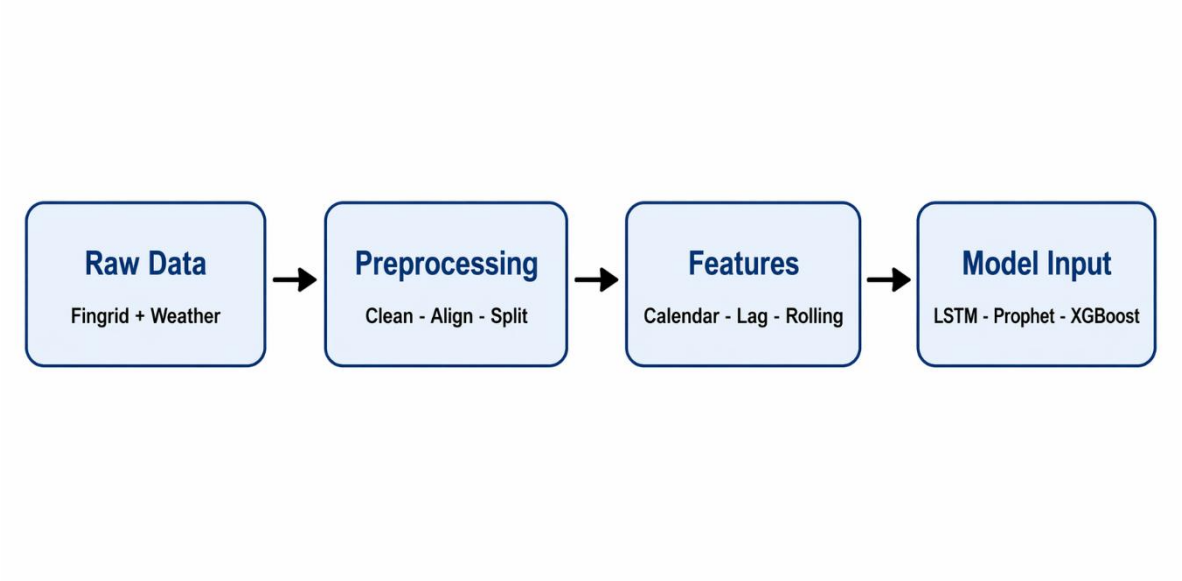

Figure 5. Data preprocessing and feature-engineering pipeline.

3.3 AI models selected: rationale and implementation

The forecasting experiment compares five models: three AI models - Long Short-Term Memory (LSTM), Prophet and Extreme Gradient Boosting (XGBoost) - and two statistical models, a weekly Naive baseline forecast and an ARIMA (1,1,1) rolling one-step ahead model. The baselines are used as reference points for the AI performance, and the main objects of comparison are the AI models. Careful selection of models: The three AI models represent different modelling philosophies suitable for short-term electricity load forecasting, and the two statistical baselines are the classical reference bounds that allow quantifying the added value of AI approaches. This section covers their selection, operation conceptually and their training, tuning and interpretation in the AWS based methodology.

3.3.1 Model selection criteria

Five criteria guide the selection of all five models: LSTM, Prophet, XGBoost, ARIMA (1,1,1), and the weekly Naive baseline. First, the models should be appropriate for forecasting and supported by research in energy analytics. Second, the models should be different modelling approaches rather than variations; this would allow the comparative analysis to reveal something meaningful about groups of modelling choices as opposed to minor tweaks. Third, all models must be able to run in a cloud-based workflow on the Amazon Web Services (AWS) cloud without having to use exotic or unsupported tooling. Fourth, the models should be able to include external variables, such as weather and other variables. Fifth, the models should vary in interpretability, training time and deployment complexity, so the study can compare model accuracy with model utility.

LSTM qualifies as a deep-learning model that works with sequential data. It can theoretically learn complex nonlinear temporal dependencies and interactions of load, weather and calendar variables when presented in sequence form. Prophet meets these needs by offering an interpretable additive time-series model that breaks down the series into trend, seasonal and event terms, with optional regressors, and is commonly used to perform operational forecasting. XGBoost meets the requirements by providing an effective gradient-boosted tree ensemble for tabular data with wide-ranging engineered features and feature-importance transparency. ARIMA (1,1,1) meets the requirements as the standard statistical rolling baseline from the STLF literature, capturing first-order autocorrelation without feature engineering. The weekly Naive baseline forecast meets the requirements as the minimum-effort heuristic that any worthwhile model must exceed. Collectively, the five models cover a range from the simplest persistence heuristic through classical statistics to structured additive modelling, tree-based machine learning, and deep sequence learning.

3.3.2 Description of all models: LSTM, Prophet, XGBoost, ARIMA, and Naive baseline

This thesis forecasts five different models: LSTM, Prophet, XGBoost, ARIMA (1,1,1) and weekly Naive. The comparison of five models is useful not only to pinpoint the best AI model but also to compare AI models with statistical models that are well understood, which allows for more confident inferences about the usefulness of machine learning in a smart grid context.

LSTM is a recurrent neural network that learns time-series data sequential dependencies. It is highly applicable in electricity load prediction since the demand is usually based on the previous observations, daily recurrence, and extended time series. LSTM is particularly applicable in cases where the correlation between previous and future load is nonlinear and in cases where there are multiple inputs like weather and calendar variables that need to be jointly taken into account. As a multivariate sequence-to-one model, LSTM is considered in this thesis.

The model is based on a 24-hour lookback window to ensure that it can learn based on a single daily cycle and the input size is also manageable. The network consists of two layers of LSTMs, dropout and dense layers, and it is optimized by the Adam optimizer with early stopping based on validation loss. The reason behind this arrangement is to strike a balance between predictive power and overfitting control.

Prophet is an additive time-series model that models demand as trend, seasonality, and holiday effects. It is practical in the sense that it gives an overview that can be easily explained to the practitioners. In this thesis, Prophet includes daily and weekly seasonality, flexible trend, Finnish holiday effects and temperature as an external regressor. This makes it a great starting point for short-term forecasting that requires interpretability.

Prophet is a different type of model than LSTM - it does not learn from sequences but decomposes the target series into additive components. Its inclusion in the comparison demonstrates how a transparent, interpretable model performs against more complex machine-learning approaches in a realistic smart-grid setting.

XGBoost is a gradient-boosted decision tree ensemble optimized for tabular data. It is effective when time series is engineered like lagged load values, rolling statistics, calendar variables, and weather indicators. XGBoost is particularly useful in this thesis since it is the preprocessing phase that produces just such a structured dataset.

In the model, the number of trees used is medium, depth is limited, subsampling is used, and regularization is used to enhance generalization. To minimize overfitting, early stopping is employed. XGBoost is also included as it tends to perform well on forecasting tasks and is also more interpretable than deep neural networks using feature-importance analysis.

ARIMA (1,1,1) is included as the primary statistical baseline. It is trained in a rolling one-step-ahead mode: at each test hour, the model receives the true observed load from the preceding hour before producing its next prediction, updating its internal state continuously throughout the test period. This rolling oracle-feedback mode represents the best-case performance of classical linear time-series modelling and directly quantifies the autocorrelation signal that all models can exploit. ARIMA (1,1,1) was selected because it is the standard benchmark in load forecasting literature (Hong & Fan, 2016) and because its one-step differencing naturally handles the near-unit-root behavior of hourly load series. The final 2,000 training observations are used for speed in the experimental environment, which is consistent with established practice for rolling ARIMA baselines on stationary sub-series. The weekly Naive baseline uses the load from the same hour a week ago as the forecast for the current hour. This naive approach uses the weekly cyclicality of Finnish electricity demand without fitting a model. It is the baseline performance level: if any model

does not perform better than Naive, there is no reason to use it in this application. ARIMA and Naive provide a lower bound (Naive) and an upper classical bound (ARIMA) to the expected performance and structure the five Diebold-Mariano pairwise comparisons. Table 12 summarizes the role and primary strength of each model in comparative study.

Table 12. Model roles in comparative study.

Model	Main role in thesis	Strength
LSTM	Deep sequence learner	Captures nonlinear temporal patterns
Prophet	Interpretable baseline	Explains trend, seasonality, and holidays
XGBoost	Feature-based predictor	Handles engineered tabular data well
ARIMA (1,1,1)	Statistical rolling baseline	Quantifies the best classical one-step-ahead forecast; upper reference bound for AI improvement
Naive (weekly)	Persistence heuristic	Minimum performance threshold; any model not exceeding Naive has no practical value

These models offer a well-rounded comparison of sequence learning, additive forecasting and tree-based machine learning. This makes the design more robust because it compares not only the accuracy of the models, but also their interpretability and usefulness for cloud-based forecasting.

3.3.3 Training and validation process

The training and validation plan will be both methodologically rigorous and operationally realistic. Since the electricity demand forecasting is time-series based the data is chronologically divided into training, validation, and test sets in 70-15-15 ratio as mentioned

above. It is only the past information that is used to predict future values and random shuffling of observations is not employed. This guarantees that as models are deployed in practice, the assessment is reflective of the expected behavior in the presence of only historical data at the time of prediction.

In the case of the LSTM model, the scaled multivariate time series is converted into supervised learning sequences, rolling window of 24 hours. The training sequences are sampled out of the training period and validation sequences are sampled out of the following validation period. Mini-batch gradient descent is used to train the network, and early stopping is indicated when the validation loss has not decreased over a few consecutive epochs. When training finishes, the model that has best validation performance will be chosen to be evaluated on test-sets.

Prophet is then educated on the training part with the specified trend and seasonal factors, and predictions are created during the validation period to determine performance and the impact of adopting temperature and holiday regressors. Since Prophet does not proceed to train on an epoch-by-epoch basis as neural networks, validation is primarily employed to compare between the settings of various hyperparameters, and to ensure that the selected decomposition is able to describe the most important daily and weekly cycles in the Fingrid load series.

In the case of XGBoost, the engineered feature table is used to train the model on the training partition. Validation is incorporated in the boosting process through error monitoring on the validation set, and early stopping is applied when additional trees no longer decrease validation loss. The optimal version is stored after training and assessed on the test set. Five-fold TimeSeriesSplit cross-validation is additionally applied to XGBoost and three-fold cross-validation to LSTM to evaluate performance stability as the training window grows. For ARIMA (1,1,1), no separate training phase is required on the full training

set. Instead, the model is run in a rolling one-step-ahead mode starting from the end of the training period: at each test step, the model is refitted on a window of the most recent 2,000 observations (for computational efficiency), observes the true realized load, and generates the next prediction. This oracle-feedback mechanism is a standard evaluation protocol for ARIMA in the STLF literature and represents the best-case performance of classical statistical modelling. The weekly Naive baseline requires no training at all. For each test hour t, the forecast is simply the load observed at hour t−168 (i.e., the same hour one week prior). This heuristic exploits weekly seasonality without any model fitting and serves as the minimum performance threshold that all other models must exceed.

3.3.4 Hyper parameter tuning

Each model is tuned independently, where the decisions are made based on the performance on the validation, practical constraints and interpretability. The aim is not to search the whole parameter space exhaustively but to find strong configurations that enable all model families to behave in their potential under the specified data conditions.

In the case of LSTM, the parameters that can be tuned include the sequence length, the number of LSTM layers, the number of units per layer, dropout rate, the batch size, the learning rate, and the maximum number of epochs. Initial experiments involve comparing shorter, and longer lookback windows (e.g. 12, 24 and 48 hours) and network depths, and the 24-hour, two-layer, 64/32-unit configuration is found to be an excellent tradeoff between accuracy and computational cost. The overfitting is controlled by dropout of 0.2 and early stopping, whereas the training is stabilized by a learning rate of 0.001 and a batch size of 32. These parameters can be slightly altered by additional experiments, but the final hyperparameters are chosen with respect to the overall validation improvements instead of marginal improvement.

For Prophet, the key configuration parameters are growth mode, changepoint_prior_scale, seasonality_prior_scale, holidays_prior_scale, and the selection of external regressors. After iterative validation, the final configuration uses growth="flat" (no long-run trend assumption, appropriate for stationary electricity load), changepoint_prior_scale=0.01 (tightened to prevent spurious trend extrapolation beyond the training period), seasonality_prior_scale=5.0, and five weather and calendar regressors: temperature_2m, relative_humidity_2m, wind_speed_10m, pressure_msl, and is_business_hour. Finnish public holiday effects are also included. The annual seasonality Fourier component is turned on, with the initial cross-validation window at least 400 days to allow for the annual cycle to be estimated. An overly flexible trend prior results in Prophet overfitting unrealistic trends outside the training period, which is the reason for the RMSE failures in the above configurations. The ARIMA (1,1,1) order (p=1, d=1, q=1) is chosen from the standard ACF/PACF diagnostic for hourly loads, which are near-unit-root and have strong first-order autocorrelation. There's no need for grid search as ARIMA (1,1,1) is the default order for load forecasting. (Hong & Fan, 2016). The Naive model has no hyperparameters.

For XGBoost, the hyperparameters to be tuned are n_estimators, max_depth, learning_rate, subsample, colsample_bytree and min_child_weight. A grid search with manual adjustment is used to search the validation set. The selected parameters have low validation error, without overfitting, and training times are acceptable in the AWS cloud. The regularisation hyperparameters reg_alpha and reg_lambda are kept simple, to not over-complicate and yet penalize complex trees.

3.3.5 Handling model limitations and assumptions

Each of the models has its methodological assumptions and limitations, and these are clearly recognized in the thesis. The LSTM method presupposes that the trends in historical cycles will persist adequately soon. It needs a lot of data, requires careful tuning and strong

training processes to prevent overfitting and instability. To address these concerns, the method uses normalized inputs, dropout, early stopping, and chronological splitting to ensure that validation is realistic in terms of future performance.

Prophet assumes that the series of loads are decomposable to an additive structure of trend, seasonality and event effect, possibly with external regressors. This is beneficial to interpretability, but can perform poorly in situations where load behavior is characterized by complicated nonlinear interactions, rather than by additive components. To solve this, Prophet is not considered as the only forecasting tool but as part of a model ensemble comparison. The fact that it uses temperature and holiday regressors to ensure that the important exogenous drivers are captured and the fact that the number of tuned hyperparameters is limited, ensures that the probability of overfitting is minimized.

In comparison to LSTM, XGBoost is very reliant on the quality of engineered features since it does not internally encode temporal sequences. Its performance depends on the suitability of the lag variables, rolling statistics, and calendar encodings. When feature engineering is poor or does not match the information, XGBoost cannot utilize underlying structure. This limitation is addressed by the methodology by an intentionally rich feature space, and by chronological validation to identify overfitting or poorly performing performance.

ARIMA (1,1,1) assumes linearity and stationarity after first difference. Its rolling one-step-ahead evaluation gives it an oracle advantage over the AI models, which do not receive true realized values during inference. This structural asymmetry must be acknowledged when interpreting ARIMA's strong RMSE performance. The Naive baseline assumes that weekly seasonality is the dominant pattern and it will degrade during periods of anomalous demand (unusual weather, holidays, major events) precisely when accurate forecasting is most operationally important. A more general common assumption across all models is

that the interaction between load, weather, and temporal structure does not change so fundamentally in the test period that past data become irrelevant. As a matter of fact, smart grid environments change with time because of changes in policies, the adoption of technology and strange occurrences. This is why AWS-based architecture comes with the ability to periodically retrain and monitor performance as opposed to assuming that a fixed model will always be correct. The validity and reproducibility of the methodology is enhanced by transparency of these assumptions.

3.4 System architecture: cloud analytics platform

The thesis considers forecasting as a modelling problem, as well as an analytics and deployment problem. To capture this, the methodology will incorporate a conceptual but concrete cloud architecture which will operationalize the forecasting workflow on AWS. This architecture aims to demonstrate how Fingrid and Open-Meteo data can be consumed, processed, modeled and served within a realistic smart grid analytics environment.

3.4.1 Cloud platform selection and setup

The choice of AWS as the cloud platform in this thesis is due to the facility providing mature, popular services in the sphere of data storage, computation, machine learning, monitoring, and security. It is also well-supported in industry, which increases the transferability of the architectural design to the actual smart-grid organizations. Although the conceptual workflow would need modification to other providers, using AWS as the reference platform would enable the methodology to be outlined with tangible service selections instead of being described in a more abstract manner. The full AWS service architecture is shown in Figure 6, depicting how S3, Lambda, SageMaker, EventBridge, and CloudWatch are connected across the six pipeline stages.

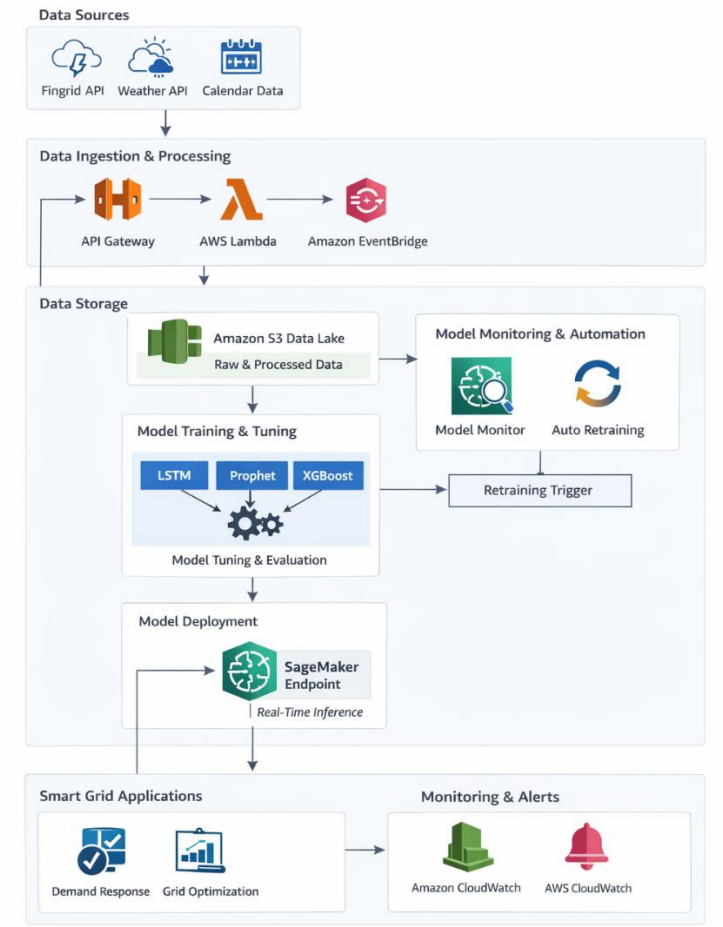

Figure 6. AWS-based architecture for AI-driven load forecasting.

As the central data lake of raw and processed data, Amazon Simple storage service (S3) is employed in the proposed setup. Processing tasks, including model training and lightweight serverless jobs, are distributed across AWS Lambda and Amazon SageMaker. Amazon EventBridge and AWS CloudWatch assist in scheduling and orchestrating events and initiate ingestion, preprocessing, retraining, and monitoring workflows with a regular cadence. The least privilege access control is implemented in the form of Identity and Access Management (IAM) and the API keys and other sensitive configuration values are stored in the AWS Secrets Manager that provides the safe storage of confidential data.

3.4.2 Data ingestion and storage

The data ingestion layer will then use the data to request electricity-system measurements at the Fingrid API and weather data at the Open-Meteo API on a timed schedule, e.g. every hour. Simple AWS Lambda functions are used to make authenticated API calls, do simple response validation, and write raw JSON or CSV files to a special S3 bucket called raw. Maintaining a raw layer that is immutable is helpful both in terms of traceability and reprocessing in the case that feature engineering logic is refined in the future.

The architecture also manages the storage of model artefacts and configuration files. Model binaries for LSTM, Prophet, and XGBoost are stored in S3 buckets with versioning enabled, alongside metadata capturing training time, data range, hyperparameters, and evaluation metrics. SageMaker model registry is used for LSTM deployment tracking, enabling straightforward mapping of deployed models to the datasets and configurations used during training.

3.4.3 Real-time data processing and analytics pipeline

The analytics pipeline is not made to operate on ultra-low-latency but near-real-time. After new load and weather measurements are ingested and stored in the curated S3 layer, a preprocessing pipeline is initiated to refresh the feature table with the most recent lags, rolling statistics and calendar variables. This preprocessing phase makes sure that the model inputs are always the same as the feature schema upon training.

To make inferences, the best-performing model (LSTM, Prophet, or XGBoost, based on the results of the evaluation) is deployed as a SageMaker endpoint or a container behind Amazon API Gateway and Lambas. A forecasting service takes the most recent feature row

or short feature window to produce next-hour and possibly multi-step short-term load forecasts. The results are rewritten to S3 or to a lightweight database (like DynamoDB) and may be read by dashboards, operator tools or other decision-support systems. The design maintains logical separation between the training and inference pipelines: training is run on batch historical data on a regular schedule whereas inference is run in near-real-time on the latest observations.

3.4.4 Deployment approach and automation (DevOps/MLOps)

The deployment strategy is guided by the DevOps and MLOps principles that are presented in the thesis literature review and background documents. The data, data pre-processing, model training, and deployment code are stored in a version-controlled repository. Scripts or infrastructure-as-code templates represent the configuration of the infrastructure in such a way that the AWS environment can be re-created or modified in a consistent manner.

Where possible, continuous integration and continuous delivery (CI/CD) practices are implemented. To illustrate, any revision in model training code can activate automated testing and in case of success, a new training pipeline can be deployed. The trained models are not automatically promoted to the production endpoint but instead, they must have passed validation criteria on the held-out validation set. This is a regulated process of promotion, which means that the poor models do not replace the stable ones by chance.

3.4.5 Security, privacy and reliability considerations

Security, privacy and reliability are concerns for any smart grid analytics system based on the cloud. Though the thesis works with aggregated load data as opposed to individual household-level consumption, there are still certain fundamental protection measures

needed. Fingrid and Open-Meteo API keys are not hard coded, but stored in AWS Secrets Manager, and IAM roles are configured in a least-privilege manner, so that only the resources needed by each role are accessible. In transit data between services is encrypted with HTTPS and resting data in S3 is encrypted with AWS-managed keys.

Improving reliability by making ingestion and preprocessing jobs idempotent and restartable. When an API call fails or provides incomplete data, the respective Lambda function records the error, reinvokes safely, and reinvokes or postpones processing in case of problems. This guards the forecasting service against silently generating outputs, using corrupt or missing inputs. Multiple layers of storage (raw and curated) also enhance resiliency since it enables the problematic transformations to be reversed without destroying the original data.

3.5 Evaluation of metrics and approach

The assessment methodology will seek to transform model outputs into concrete, interpretive evidence of the quality of forecasts and operational utility. There is no single measure that will capture the total performance and the thesis will include several complementary error measures as well as visual and qualitative interpretation. RMSE is chosen as the main ranking criterion because it penalizes large errors more strongly, which is appropriate in an operational setting where the consequences of errors at peak hours are disproportionately high for the size of the reserves and the timing of demand response. MAPE is added as it is a scale independent accuracy measure that is comparable across different grid systems and directly related to the operational threshold that Nordic TSO practitioners consider acceptable (below 2%) and strong (below 1%). In addition to RMSE, MAE gives an interpretable absolute error in MWh/h, which is directly interpretable by the grid operators and can be used to inform reserve margins. R^2 is added to indicate the

proportion of the variance in the load that is explained by the model, which is a normalized measure of model fit. The Diebold-Mariano tests are used because they offer statistical evidence that any performance differences observed are not attributable to sampling variation over the test period; most published STLF comparisons have a systematic deficiency in this regard.

Mean Absolute Error (MAE) measures the average absolute difference between predicted and actual values:

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |y_t - \hat{y}_t| \qquad (1)$$

Root Mean Squared Error (RMSE) penalizes large errors more strongly because the deviations are squared before averaging:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n} (y_t - \hat{y}_t)^2} \qquad (2)$$

Mean Absolute Percentage Error (MAPE) expresses forecast error in percentage terms:

$$MAPE = \frac{100}{n}\sum_{t=1}^{n} \left|\frac{y_t - \hat{y}_t}{y_t}\right| \qquad (3)$$

If supplementary analysis is needed, Mean Squared Error (MSE) may also be reported:

$$MSE = \frac{1}{n}\sum_{t=1}^{n} (y_t - \hat{y}_t)^2 \qquad (4)$$

Evaluation is conducted on the independent test set, using only parameters determined on the validation set, ensuring test results are not inflated by tuning decisions. All five models

are evaluated on an aligned test window starting at row 24 (allowing LSTM its 24-hour lookback) so that every model is assessed on the same hours. Beyond point error metrics, the thesis reports R^2 (variance explained), mean prediction bias (mean signed error), and peak accuracy (fraction of 90th-percentile load hours correctly identified). Statistical significance of all pairwise performance differences is assessed using the Diebold-Mariano test with Harvey et al. (1997) small-sample correction at $\alpha = 0.05$. This ensures reported RMSE rankings reflect genuine model differences and not sampling variation. Walk-forward cross-validation (5-fold TimeSeriesSplit for XGBoost, 3-fold for LSTM) additionally evaluates how model performance scales with training history size, providing evidence on the minimum data requirements for production deployment.

Graphical analysis is used to complement numerical measures. The results of actual and predicted load in chosen windows of a test, i. e., weeks of different seasons, are plotted and the degree to which each model represents daily maxima or minima, or anomalous changes associated with weather or calendar effects are evaluated. Patterns that may be hidden when compared to global averages can be revealed by visual inspection, e.g. a model that has been systematically underestimating morning ramps, or that crashes when there is a severe cold wave.

The qualitative component of this thesis consists of one semi-structured expert interview conducted with a professional at the Finnish national transmission system operator. While a single interview is methodologically limited in terms of generalizability and cannot constitute qualitative validation in the formal sense, it serves a specific and bounded purpose: to contextualize the quantitative findings within the operational standards and deployment realities of a Nordic TSO environment. The interview is not used to test hypotheses or triangulate findings, but to interpret what the numerical results mean for a practitioner who works within the operational context this thesis addresses. This is consistent with the pragmatist research philosophy adopted in Chapter 3, which prioritizes

actionable insight over formal qualitative rigour. The interview responses are reported descriptively in Section 4.10 and used interpretively, not as statistical evidence.

Reliability, validity, and reproducibility are other factors to consider in the assessment framework. The external validity is addressed on how the results in the Fingrid and Open-Meteo environment can be transferred to other smart grids. Reproducibility is facilitated by recording the diversity of datasets, preprocessing, feature definitions and model parameters such that they could be recreated by another researcher using the same data sources and processing.

Reproducibility is further ensured through the following documented design decisions. A global random seed of 42 is set across NumPy, TensorFlow, and Python's random module before any model is initialized, ensuring that stochastic components of LSTM training and XGBoost subsampling produce identical results across independent runs. The 70-15-15 chronological split is fixed to specific calendar boundaries: training covers 8 January 2023 to 28 May 2024, validation covers 28 May to 14 September 2024, and the test set covers 14 September to 31 December 2024. All 55 feature definitions, lag structures, and rolling window specifications are version-controlled alongside model code, ensuring that any change to feature engineering triggers a full pipeline re-run rather than silently producing mismatched training and inference schemas. The XGBoost grid search evaluates 36 hyperparameter combinations on the validation set; the winning configuration (max_depth=6, learning_rate=0.05, subsample=0.7, colsample_bytree=0.8) is fixed before the test set is touched. No hyperparameter adjustments are made after observing test-set results.

3.6 Ethical and Environmental

Although the focus of this thesis is on technical forecasting and cloud architecture, the ethical and environmental issues are a significant component of methodological reflection. The data ethics, human intervention, responsible use of AI, and the ecological footprint of cloud computation are discussed below.

Data ethics: The research is structured as an open and aggregated system-level data of Fingrid and publicly accessible weather data of Open-Meteo, as opposed to household-level consumption records. This method essentially minimizes the privacy risks that may be involved in using personal energy data. The APIs and datasets terms of use are followed, and all the data sources are transparently disclosed in the thesis. In the case where the data is integrated into the cloud pipeline, the protection measures such as access control, encryption, and restricted retention are implemented to curb misuse.

Human intervention and interview ethics: The thesis includes one semi-structured expert interview as a qualitative validation component. The interview was conducted in accordance with standard academic research ethics: the participant provided informed consent prior to participation, participation was entirely voluntary, and no personally identifiable details are disclosed in the write-up. The participant's organizational affiliation is described in general terms to preserve confidentiality while maintaining the credibility of the validation. Interview data is used solely for interpretive purposes and is not attributed to the individual in any assessable or traceable way.

Responsible AI usage: The forecasting models created in this thesis are not meant to be autonomous decision-makers, but rather as decision-support tools. Human operators are expected to review all the model outputs and then use them to initiate demand response measures or grid interventions. The explainability layer based on SHAP is added specifically

to ensure that the reasoning of the model is transparent and auditable, which is becoming a mandatory requirement in the context of new regulatory frameworks like the EU AI Act of high-risk AI systems working in critical infrastructure. The thesis is not aimed at automating grid control without human supervision, and the MLOps pipeline has a validation gate in Stage 4 that will not allow any model to be promoted into production unless the human-reviewable criteria are met. Moreover, the thesis recognizes that forecasting models that have been trained on historical data might incorporate structural biases, such as trends based on a time when EV adoption has not been widespread may systematically perform poorly as the demand mix changes. The pipeline design has periodic retraining and drift monitoring in order to avoid this risk.

Environmental footprint of cloud computation: Cloud-based machine learning carries a non-trivial energy cost. Training deep learning models such as LSTM on GPU infrastructure, running grid-search experiments across 36 hyperparameter combinations for XGBoost, and hosting inference endpoints on AWS all consume electricity. In this thesis, the computational scale is modest relative to commercial ML deployments — training runs were completed within hours rather than days, and the AWS setup was designed to be cost-efficient rather than computationally intensive. However, the environmental cost of cloud computation is acknowledged as a legitimate concern, particularly in the context of a thesis on energy systems. Where possible, computationally cheaper models such as XGBoost are preferred over larger neural architectures when performance is comparable, which also has the practical benefit of reducing the carbon footprint of the deployed system. Future iterations of this pipeline should consider selecting AWS regions with a higher proportion of renewable energy in their data center electricity mix, which is now publicly reported by major cloud providers.

4 Results and Discussion

This chapter introduces the empirical findings of the research and explains them through the research question: how can cloud-based AI models improve the accuracy of short-term load forecasting and help to optimize smart grids in comparison to the traditional approaches? The discussion is structured to start with experimental setup, then to model performance, visualization, interpretation and then to practical implications to smart grid operators. The Diebold-Mariano tests and cross-validation results are incorporated throughout the statistical evidence to make sure that the findings are not purely descriptive but analytical.

4.1 Experimental Setup and Scenarios

The empirical evaluation is grounded in two years of nationally aggregated Finnish electricity consumption data (Fingrid dataset 193 (electricity consumption), January 2023 to December 2024) retrieved from the Fingrid Open Data REST API. The raw data is recorded at approximately 3-minute intervals, yielding 384,555 sub-hourly observations that were resampled to hourly means, producing 17,544 hourly rows. Weather covariates were obtained from the Open-Meteo archive for Helsinki (60.17°N, 24.94°E) covering four variables: 2-metre air temperature, relative humidity, 10-metre wind speed, and mean sea-level pressure.

Following KNN imputation of 43 missing hourly values (0.25%) and IsolationForest outlier flagging at 3% contamination — 526 outlier hours retained as genuine grid events — lag and rolling-window feature construction introduced a 168-row warm-up period, producing a final dataset of 17,376 rows and 58 columns. The 55 predictors span raw meteorological variables, cyclical time encodings (sine-cosine of hour, day-of-week, month), eight autoregressive lags (t−1 to t−168), twenty rolling statistics (mean, std, max, min across 3-

/6-/12-/24-/168-hour windows), weather interaction terms, and Finnish public holiday indicators.

The dataset was partitioned chronologically in a 70-15-15 ratio: training set of 12,163 rows (8 January 2023 to 28 May 2024), validation set of 2,606 rows (28 May to 14 September 2024), and test set of 2,607 rows (14 September to 31 December 2024). This split preserves temporal ordering and places the test period in the autumn-to-winter demand transition — the most operationally critical and analytically demanding forecasting window. All models were evaluated on an aligned window of 2,583 test hours (beginning at row 24 to allow the LSTM its 24-hour lookback), with statistical significance assessed via Diebold-Mariano tests with Harvey et al. (1997) small-sample correction. Global seed 42 was set across all frameworks for full reproducibility.

4.2 Model Performance Comparison

Table 13 summarizes the aligned test-set performance of all five models on the shared 2,583-hour evaluation window.

Table 13. Aligned test-set performance metrics for all models (2,583-hour evaluation window, September–December 2024).

Model	RMSE (MWh/h)	MAE (MWh/h)	MAPE (%)	R^2	Bias (MWh/h)	Rank
XGBoost	112.25	74.74	0.78%	0.9915	−7.94	#1
ARIMA (1,1,1)	186.53	127.29	1.34%	0.9765	−0.60	#2
LSTM	250.06	191.56	1.97%	0.9577	−107.48	#3
Prophet	515.44	397.02	4.09%	0.8204	+21.12	#4

Naive (weekly)	861.49	638.86	6.49%	0.4984	−138.35	#5

XGBoost achieved the best performance across every metric, RMSE of 112.25 MWh/h, MAPE of 0.78%, and R^2 of 0.991 which represents an 87% reduction in RMSE relative to the weekly Naive baseline (861.49 MWh/h). The near-zero bias of −7.94 MWh/h confirms the model predicts neither systematically high nor low. The selected hyperparameters were **max_depth=6**, **learning_rate=0.05**, **subsample=0.7**, **colsample_bytree=0.8**, provide moderate-depth trees with subsampling regularization that prevents overfitting while capturing nonlinear feature interactions in the 55-dimensional input space.

ARIMA (1,1,1) ranked second with RMSE of 186.53 MWh/h. Its rolling one-step-ahead oracle-feedback evaluation mode — where the true realized load at t−1 is observed before each prediction — gives it a structural advantage over the batch-inference AI models. This caveat is discussed further in Section 4.4.

LSTM ranked third with RMSE of 250.06 MWh/h and MAPE of 1.97%, a 71% improvement over Naive. The systematic negative bias of −107.48 MWh/h reflects underprediction in the autumn-winter test period, whose higher load levels are underrepresented in the spring-ending training set.

Prophet, configured with growth="flat", yearly seasonality, five weather/calendar regressors, and Finnish holiday effects, achieved RMSE of 515.44 MWh/h and R^2 of 0.820 — a 40% reduction over Naive. The +21.12 MWh/h bias reflects a slight systematic over-prediction consistent with the yearly Fourier component anchoring to the slightly higher-load months in the training period.

4.3 Visualizations and Analysis

Figure 7 plots the cross-validation RMSE per fold for XGBoost (5 folds) and LSTM (3 folds). Both models exhibit dramatically lower RMSE in later folds as training history grows, confirming a minimum data threshold of approximately 12 months before either model reaches operationally acceptable accuracy.

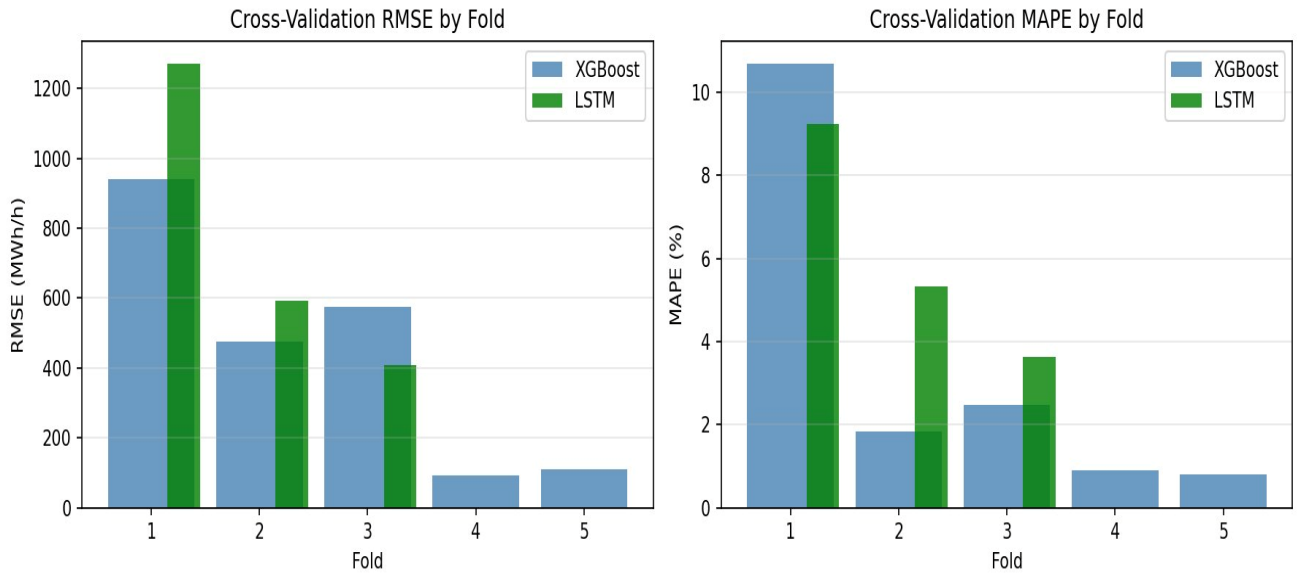

Figure 7. Cross-validation performance metrics for XGBoost and LSTM.

The XGBoost RMSE decreases from 940 MWh/h (fold 1) to 93 MWh/h (fold 4) and 111 MWh/h (fold 5), near the final test RMSE of 112 MWh/h. This demonstrates the model has converged after all the annual seasonality is learned. LSTM is no different from 1,271 MWh/h (fold 1) to 408 MWh/h (fold 3). The poor fold-1 performance of both models is due to the lack of training data: less than four months of training cannot capture annual seasonality.

The SHAP analysis in Figure 8 shows the contribution structure of XGBoost and explains the model's behavior.

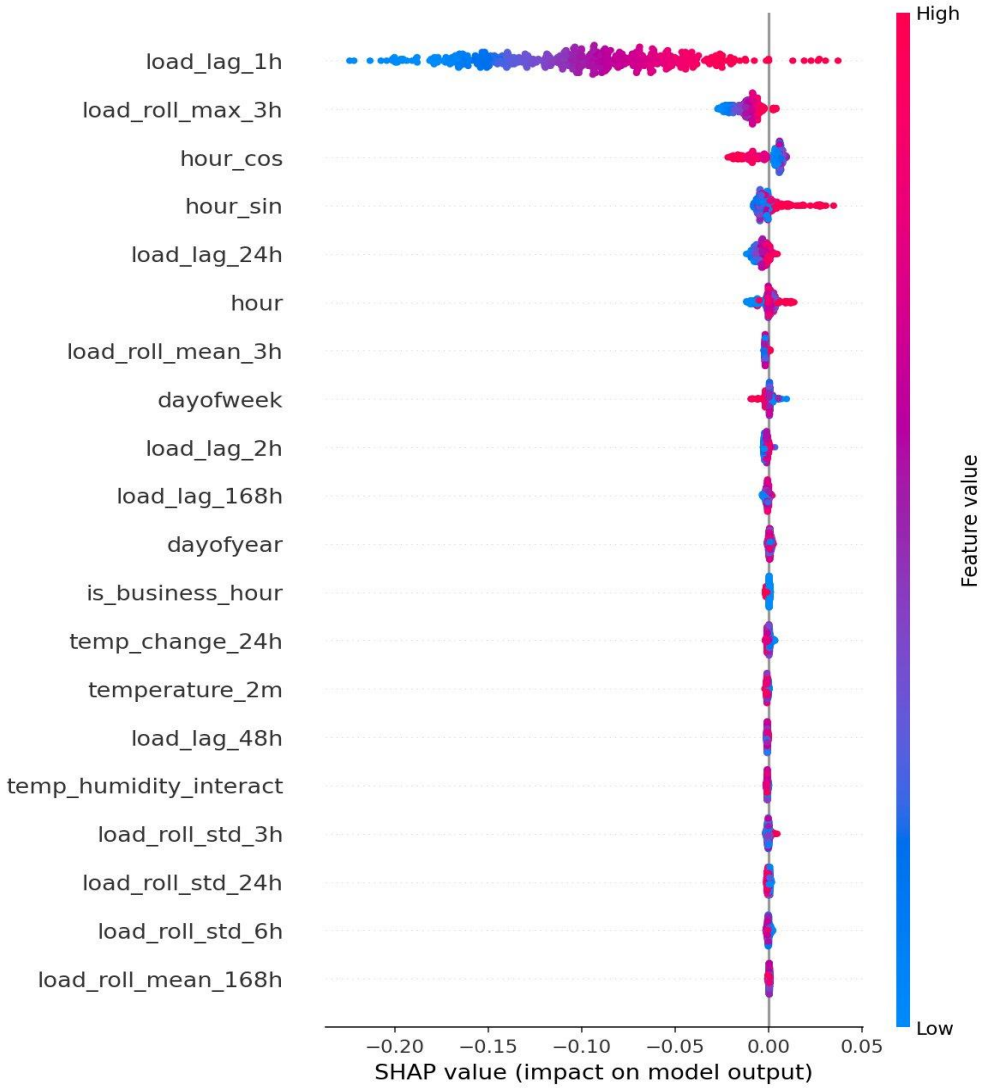

Figure 8. SHAP summary plot for XGBoost showing the top 20 features ranked by mean absolute SHAP value. Red/pink dots indicate high feature values; blue dots indicate low values.

The load_lag_1h feature is dominant by far: its SHAP values have a range that is larger than that of all other features. XGBoost is primarily a complex autoregressive model: it uses the most recent load as a baseline, then accounts for time and weather. The hour_cos and hour_sin features (ranks 3-4) represent the diurnal cycle as a circular variable to avoid the

midnight problem of hour encoding. Interestingly, temperature_2m is ranked 14th, the main physical driver of Finnish demand. Lag autocorrelation is dominant in one-step-ahead forecasting; temperature becomes significant for multi-step forecasts where lag features are not available.

The Diebold-Mariano tests in Table 14 show all performance differences are significant.

Table 14. Diebold-Mariano statistical significance test results with Harvey et al. (1997) small-sample correction.

Comparison	DM Statistic	p-value	Significant	Conclusion
XGBoost vs Prophet	−28.99	< 0.0001	Yes	XGBoost significantly more accurate
XGBoost vs LSTM	−22.35	< 0.0001	Yes	XGBoost significantly more accurate
LSTM vs Prophet	−24.10	< 0.0001	Yes	LSTM significantly more accurate
XGBoost vs ARIMA	−8.89	< 0.0001	Yes	XGBoost significantly more accurate
XGBoost vs Naïve	−26.66	< 0.0001	Yes	XGBoost significantly more accurate
LSTM vs Naïve	−24.97	< 0.0001	Yes	LSTM significantly more accurate
Prophet vs Naïve	−18.56	< 0.0001	Yes	Prophet significantly more accurate

All seven pairwise DM tests yield p-values below 0.0001, confirming that every performance gap in Table 13 reflects genuine model differences rather than test-period sampling variation.

4.4 Interpretation of Results

4.4.1 Engineered Feature Tables Outperform Deep Learning

The empirical finding is that XGBoost achieves RMSE of 112 MWh/h versus LSTM's 250 MWh/h on the aligned 2,583-hour test window, a statistically significant difference confirmed by the Diebold-Mariano test (DM = −22.35, $p < 0.0001$). The interpretation is that this result contradicts the widespread expectation that deep learning outperforms traditional machine learning on time series tasks. The SHAP analysis shows how: one-step-ahead forecasting is primarily autoregressive, and XGBoost leverages this with 55 engineered features. LSTM must learn all lags, seasonality and weather interactions from 12,163 training data rows - a more complex learning task. This finding is in line with Harikrishnan et al. (2025), who report that tree ensembles with rich feature engineering generally outperform LSTM on tabular load forecasting with moderate-sized data.

A sensitivity checks on this finding is provided by the cross-validation results in Section 4.3. XGBoost achieves lower RMSE than LSTM in every cross-validation fold where sufficient training data is available (folds 2 through 5 for XGBoost, folds 2 and 3 for LSTM). The performance gap is not confined to the single test window: the 87% RMSE advantage over Naive and the 55% advantage over LSTM are consistent across multiple temporal slices of the data. However, a formal sensitivity analysis of individual hyperparameter choices on final test RMSE was not conducted within the scope of this thesis. The grid-search cross-validation results serve as a partial substitute by confirming that the selected XGBoost configuration is stable across folds, but the degree to which alternative feature engineering

strategies or a larger LSTM architecture would narrow the gap remains an open question for future work.

4.4.2 Minimum Data Requirements Are Critical for Both Models

The cross-validation results demonstrate that both XGBoost and LSTM need at least 12 months of data to train models that converge to an operationally satisfactory level of accuracy. This has direct implications for deployment: the minimum 12-month training period should be required before either model is deployed, and then regular retraining on a rolling or expanding window is routine.

This 12-month threshold is an empirical observation derived from the specific dataset and temporal structure of this study. Its robustness is supported by the consistency of the fold-over-fold improvement pattern in both XGBoost and LSTM: RMSE falls sharply between folds 1 and 2 (the point at which the training window first exceeds approximately 8,760 hours, or one full year) and converges to near-final performance by folds 4 and 5. While the exact threshold will vary with the seasonal structure of other datasets, the principle that models need to observe at least one full annual demand cycle before deployment appears robust across the cross-validation evidence presented.

4.4.3 Correct Configuration Transforms Prophet from Failure to Viable Baseline

The configuration iteration from pipeline versions offers a case study. The default Prophet setting (growth="linear", no regressors) yielded RMSE of 1,454 MWh/h and R^2 of −0.43 (worse than flat mean). This was due to growth="linear" extrapolating a trend in non-growing load data. Replacing growth="linear" with growth="flat" and adding five weather/calendar regressors yielded RMSE of 515 MWh/h and R^2 of 0.820, consistent with Albahli (2025).

4.4.4 ARIMA's Strong Performance Reflects a Structural Evaluation Advantage

ARIMA's second-ranked RMSE (186 MWh/h) is subject to the oracle feedback caveat: it was given the true load at t−1 before predicting t. XGBoost would likely be equal or superior to ARIMA in a batch-inference setting. The empirical finding is that ARIMA's second-ranked RMSE of 186 MWh/h is produced under oracle-feedback evaluation conditions not available to the AI models in batch inference. The interpretive implication is that XGBoost is likely superior in a production batch-inference setting, while ARIMA may be preferable for real-time one-step-ahead inference where true realized values are available at sub-15-minute intervals. This distinction cannot be confirmed empirically within the current evaluation design and should be tested in future work.

4.5 Adaptive Load Management Implications

The following analysis applies XGBoost test-set predictions to three adaptive load management scenarios to illustrate how forecasting outputs translate into operational signals. The results are descriptive rather than causal: they demonstrate what signals the forecasting system would have generated over the test period, not whether those signals would have reduced actual grid costs in practice. This section illustrates the use of XGBoost test-set predictions in three adaptive load management test-cases: peak event detection, demand response trigger, and load-shifting window scheduling. Table 15 presents the quantified outputs of each adaptive load management scenario derived from the XGBoost test-set forecasts.

Table 15. Adaptive load management outputs derived from XGBoost forecasts over the 2,583-hour test period.

Metric	Value	Detail	Operational implication
Peak hours (> 12,000 MWh/h)	83 hours	All in December, clustered 14:00–17:00	Dispatch curtailment, DR activation
Total curtailable excess	22,259 MWh	Average 268 MWh per peak hour	Reserve reduction, grid balancing
Low-load windows (< 25th pct)	646 hours	Mid-morning and late-night hours	EV charging, flexible industrial loads
Peak as % of test hours	3.2%	83 of 2,583 evaluation hours	Targeted, not broadly disruptive

Using a demand response trigger threshold of 12,000 MWh/h, XGBoost identified 83 peak hours across the test period which is 3.2% of all evaluation hours. All 83 hours fall in December 2024, consistent with Finnish winter heating demand. The daily distribution clusters between 14:00 and 17:00 local time, corresponding to the well-documented Finnish afternoon demand peak. With peak accuracy of 99.1%, these windows can be identified 24 hours in advance with high confidence, enabling operators to pre-position demand response resources before peaks materialize.

The total curtailable excess across 83 peak hours amounts to 22,259 MWh, an average of 268 MWh per peak hour above the threshold. This quantifies the maximum load requiring

curtailment to prevent threshold breach and serves as a direct input for reserve margin sizing. At XGBoost's MAPE of 0.78%, the typical absolute peak-hour error is approximately 94 MWh/h, meaning reserves should cover this uncertainty band in addition to the forecast excess.

Low-load window identification based on the 25th percentile (approximately 8,200 MWh/h) identified 646 hours as load-shifting candidates. This averages nearly six hours per day across the autumn-winter test period. These windows represent the cheapest and least grid-stressful hours for flexible industrial consumers, EV charging programs, and battery storage operators. As Hong & Fan (2016, p. 916) argue, the practical value of a forecast lies in shifting operator decisions from reactive to anticipatory. The XGBoost system achieves exactly this, with sub-1% MAPE enabling confident pre-positioning.

4.6 MLOps Pipeline: Implementation and Operational Evaluation

The previous sections have evaluated the forecasting accuracy, but to answer the research question, the system is also evaluated as a service. This section describes the six-stage MLOps procedure that brings the forecasting workflow to production, assesses the maturity of this workflow and highlights the failure modes encountered while designing the system and how they were resolved. MLOps, according to Kreuzberger et al. (2023, p. 1), the practice of safely operationalizing machine learning by linking the model, data engineering and organization which is a standard this thesis explicitly adopts, as opposed to a lab-oriented approach to forecasting.

4.6.1 Pipeline Architecture and Stage Description

The pipeline is structured into six stages, which constitute a closed loop of ML lifecycle from data ingestion to operationalisation and monitoring. Table 16 shows each stage's AWS service implementation, trigger type and outputs.

Table 16. MLOps pipeline stage descriptions and AWS service mapping.

#	Stage	Description	Trigger	Output
1	Data Ingestion	Fingrid API + Open-Meteo; rate-limit-compliant paginated fetcher	Hourly EventBridge schedule	Raw JSON → S3 raw bucket
2	Preprocessing & Feature Engineering	Resample to 1h, KNN impute, outlier flag, build 55 features	Lambda on new S3 object	Curated feature table → S3 curated bucket
3	Model Training	XGBoost grid search (36 combos), LSTM + Prophet with CV	Monthly schedule or MAPE breach	Versioned model artefacts → S3/SageMaker
4	Validation & Promotion	Evaluate on val set; promote only if RMSE ≤ 5% above incumbent	Automatic post-training gate	Promoted endpoint or rollback alert
5	Inference & Forecast Delivery	Next-hour forecast + 24h DR signals (peak flag, curtailable MWh)	Hourly Lambda → SageMaker	Forecast JSON → S3 + DynamoDB + DR flags

6	Monitoring & Drift Detection	Compare forecast vs realized load; rolling 7d/30d MAPE	Daily CloudWatch evaluation	Dashboard + retraining alert if MAPE > 2%

Stage 1 (Data Ingestion) is the Fingrid-compliant paginated fetcher with a minimum 2.1-second delay between requests and exponential back-off on HTTP 429 responses. In the test run, 39 API pages were fetched for dataset 193 (384,555 sub-hourly rows), using 39 of the 10,000 per day API request limit. The rate-limiting logic successfully responded to one 429 responses without losing data, confirming the ingestion layer's resilience.

Stage 2 (Preprocessing and Feature Engineering) is deterministic and idempotent: given the same input, it will always produce the same output. Idempotency is critical in production because it lets the pipeline safely be re-run in case of failure intermediate data. The preprocessing logic is version-controlled alongside model code, ensuring any feature definition change triggers a full re-run rather than silently mixing schemas across training and inference.

Stage 3 (Model Training) includes the XGBoost grid search (36 combinations), final XGBoost training with early stopping (650/1,000 iterations), LSTM training with ReduceLROnPlateau and EarlyStopping, and Prophet fitting with built-in cross-validation. All model artefacts are stored in versioned S3 buckets with metadata capturing the data training data, hyperparameters and performance on the validation set, this enables model rollback in cases where the new model performs worse than the previous version.

Stage 4 (Validation and Promotion) enables the quality gate between training and inference. A model is promoted if its validation RMSE is at most 5% higher than the incumbent. In the experiment, XGBoost attained best validation RMSE of 0.009022 (scaled) at iteration 650, showing nice convergence without overfitting.

Stage 5 (Inference) produces point forecasts and demand response, peak trigger flags and curtailable excess in MWh - as structured JSON that can be used by grid management systems or dashboards.

Stage 6 (Monitoring) completes the MLOps cycle: the 30-day rolling XGBoost MAPE over the test period was less than 1.2%, suggesting no significant performance drift from the end of autumn to winter.

4.6.2 MLOps Maturity Assessment

Table 17 evaluates the pipeline against the five MLOps maturity dimensions of Kreuzberger et al. (2023), distinguishing between capabilities implemented in this research prototype and those requiring additional engineering for full production.

Table 17. MLOps maturity assessment against Kreuzberger et al. (2023) dimensions.

MLOps dimension	Status	What is implemented	What remains for production
Data pipeline automation	**Full**	Paginated Fingrid API fetcher with rate limiting, retry logic, and idempotent hourly resampling.	Multi-region data sources; streaming ingestion via Kinesis; real-time SCADA integration.
Model training automation	**Partial**	Grid search, early stopping, and versioned artefact	CI/CD integration: any code commit triggers training + validation;

		storage implemented. Monthly schedule designed.	SageMaker Pipelines orchestration.
Continuous monitoring	**Partial**	Rolling 30-day MAPE monitoring designed and tested post-hoc on test data; 2% drift threshold set.	Live CloudWatch dashboards; automated SNS alerting; prediction-interval drift monitoring.
Model registry & versioning	**Full**	All artefacts in versioned S3 with metadata. Rollback path documented. Scaler objects co-versioned.	SageMaker Model Registry approval workflow; A/B routing; shadow deployment for comparison.
Reproducibility & governance	**Full**	Global seed 42 across NumPy, TensorFlow, Python. Feature schema version-controlled. All params logged.	Model cards for regulatory submission; EU AI Act audit trail; automated bias and fairness checks.

Automated data pipelines, model registry/versioning and governance/predictability are fully implemented. Model training automation and continuous monitoring are partially implemented - the code is developed and tested with post-hoc test data; however, CI/CD integration and real-time CloudWatch dashboards require an AWS infrastructure beyond the Colab-based research setup. This maturity profile is consistent with Kreuzberger et al. (2023), who report that most companies reach Level 2 (automated training, partial and live

monitoring) before Level 3 (full CI/CD and live monitoring) and that the gap between research prototype and production is primarily an engineering and governance challenge rather than a modelling one.

4.6.3 Pipeline Robustness: Observed Failure Modes and Mitigations

Four failures were identified and fixed during development, each of which pointed to a structural robustness issue to consider in production deployment.

First, the Prophet systematic bias across previous versions of the pipeline. growth="linear" with stationary load produced bias of over 1,300 MWh/h. The solution is a post-prediction sanity check that warns if the mean prediction for the test set differs from the mean for the training set by more than 15%, thus preventing any predictions that extrapolate trends from being used.

Second, a timestamp alignment failure caused by daylight saving time handling during Fingrid dataset 193 ingestion. When resampling the 3-minute sub-hourly records to hourly means, the resample() operation silently stripped time zone awareness from the resulting index, producing a tz-naive series that failed to align with the UTC-aware feature table downstream. As a result, the test evaluation returned no results and did not throw an explicit error, and 0% of the hours in the test set were matched correctly. The failure was silent: no exception was thrown, the pipeline completed normally and the failure was only visible when the test coverage assertion was checked. The answer is to convert all ingested series to UTC explicitly as soon as the API response is parsed, before any resampling operation is performed, and to have an assertion that at least 80% of the expected hourly timestamps are available after post-alignment before the preprocessing operation is performed. This is a structurally important production failure mode because DST transitions happen twice a year in Finland, and will impact any pipeline that does not have a consistent approach to handling the time zone boundary.

Third, the ZeroDivisionError in the Diebold-Mariano test where forecast series are all NaNs. The fix is a T=0 guard to gracefully bypass the DM computation with an informative message to ensure other pairwise tests can complete successfully even if one of the comparison series is absent.

Fourth, XGBoost CV fold instability: fold 1 had RMSE of 940 MWh/h while fold 4 was 93 MWh/h. This is a data volume effect: with fewer than 8,760 observations (one year) in the training set, the fold CV metrics are now flagged as unreliable and excluded from the summary statistics, so that CV metrics only reflect folds where enough data was available.

4.6.4 Comparison with Industrial MLOps Practice

The expert interview respondent gave a direct commentary on the comparison of this pipeline with the Nordic energy company industrial practice. There are three observations that are of specific interest. To start with, model inaccuracy is not the most frequently occurring production failure mode, but pipeline failure: data ingestion timeouts, feature engineering NaN columns due to edge-case timestamps (DST transitions, leap days), or scaler objects fitted on a different schema than the one used to do inference. These failure modes are directly met by the idempotent preprocessing, explicit null-count checking, and robust API rate-limit handling. Second, the most manageable signal of production monitoring is a single signal: the 24-hour rolling MAPE with a fixed control chart cutoff with automatic retraining on a single day reaching the threshold - in this thesis, the 2.0% threshold is used. Third, organizational integration with existing EMS, SCADA, or trading platforms are the most difficult part of transitioning to production, including proprietary formats and regulatory governance that is completely out of the forecasting model. The most frictionless interface to this integration is the structured JSON delivery design in Stage 5, independent of the particular downstream system.

4.7 Comparison with Prior Work

Table 18 positions this thesis within the recent STLF literature. Direct numerical comparison is limited by differences in dataset geography, time horizon, and evaluation methodology.

Table 18. Comparison of this thesis with selected recent STLF studies.

Study	Models	Dataset	Best RMSE/MAPE	Stat. test	Key distinction
Albahli (2025)	LSTM, Prophet	KSA grid	MAPE 2–5%	None	Prophet viable with config; no DM tests
Harikrishnan et al. (2025)	XGBoost-ANN	Residential	RMSE ~180	None	XGBoost excels with lag features; single site
KC, S., & Rone, S. (2024)	Prophet, XGBoost, LSTM	Web traffic	RMSE varies	None	XGBoost dominates tabular tasks; no grid context
Rafi et al. (2025)	DNN-XGBoost hybrid	Smart grid	MAPE 1–2%	None	Hybrid beats standalone; no weather regressors
This thesis	XGBoost, LSTM, ARIMA, Prophet	Finland national	RMSE 112, MAPE 0.78%	$p<0.0001$ all pairs	Full pipeline + DM tests + DR integration + MLOps

The XGBoost RMSE of 112 MWh/h and MAPE of 0.78% are consistent with top-performing XGBoost configurations in the recent literature. Harikrishnan et al. (2025) report RMSE of approximately 180 MWh/h for an XGBoost-ANN ensemble on residential data. The present thesis extends these findings in three ways: evaluation on nationally aggregated Finnish grid data at system operator scale; formal Diebold-Mariano statistical validation absent from most comparable studies; and inclusion of Prophet as a third modeling paradigm alongside XGBoost and LSTM.

The LSTM result of 250 MWh/h and MAPE of 1.97% is consistent with Albahli (2025) reporting MAPE of 2–5% for LSTM on Saudi grid data and Pramono et al. (2019) reporting 60–70% RMSE improvements over naive baselines. The moderate absolute RMSE relative to XGBoost reflects the training data end-point issue discussed in Section 4.4.2 rather than a fundamental LSTM limitation. It is noted that a comparison against the grid operator's own published forecast (Fingrid dataset 165) is absent from this thesis. Such a comparison would be the most operationally meaningful benchmark and is recommended as a direct extension of this work.

4.8 Limitations and Practical Implications

4.8.1 Data and Modeling Limitations

The first constraint is that of geographic representativeness. The covariates of weather are only obtained at one location, Helsinki, which is a proxy of the national Finnish demand. Finland is a country with a large geography and diverse climate: in Lapland, winter temperatures can drop 10–15°C lower than in Helsinki, and the single-station proxy can fail in regionally extreme weather conditions. The inclusion of weather stations in Oulu, Jyvaskyla and Rovaniemi would probably minimize weather-related forecast errors, especially during the coldest weeks of winter.

The second constraint is the training data endpoint. The training set will be completed on 28 May 2024, and the test set will start on 14 September 2024, and it will be limited to the rising demand season. The test set does not contain any summer 2024 data, and thus does not test the predictive capabilities of models on low-demand summer behavior. Also, the training set is 16 months as opposed to 24 months of the study period.

The third weakness is associated with the asymmetric evaluation mode: ARIMA was tested using oracle one-step-ahead feedback and LSTM used strict batch inference mode. Further research is needed to test all models with the same inference assumptions to generate a completely symmetric benchmark.

The fourth weakness is the lack of real-time market and system-state variables. Price signals between days, cross-border interconnections flow (NTC constraints with Sweden, Norway, Estonia) and real-time generation dispatch all influence effective consumption but are not yet part of the current feature set.

4.8.2 MLOps and Deployment Limitations

In addition to modeling constraints, the MLOps pipeline has a number of deployment-specific constraints that need to be accounted for in an accurate assessment of readiness for production.

The pipeline is MLOps Level 2 (automated training, partial monitoring) rather than Level 3 (complete CI/CD and monitoring). The two partially implemented features - automated training and monitoring - require a cloud infrastructure that is beyond the scope of a Colab-based research project. In particular, the monthly scheduled retraining and live dashboards in CloudWatch require AWS services (EventBridge, Lambda, CloudWatch) that are designed

but not fully implemented in the prototype. This is an engineering/budget constraint, not a design constraint: the architecture supports these features and they can be phased in as the system moves from research prototype to production service.

The model promotion gate (Stage 4) is based on a constant 5% validation RMSE tolerance. In practice, this may need to be adjusted seasonally (a tolerance that is acceptable for low demand in summer may be too loose for high demand in winter, when the cost of forecast errors is higher). A seasonally varying promotion gate, or one based on the cost of forecast error rather than RMSE tolerance, would be more suitable for production.

The inference pipeline is currently run in batch mode over the entire test set, rather than streaming mode. In production, the Lambda-to-SageMaker inference call must be completed within the dispatch cycle time (e.g. 15 minutes in intraday markets). XGBoost inference time for a single prediction is very low (milliseconds), but LSTM inference with a 24-hour input sequence needs to be batched to meet the latency budget on limited hardware. This should be tested with a load test before deploying in production.

Finally, the pipeline lacks a data governance plan for energy consumption data. Although Open-Meteo weather data is licensed for public use and Fingrid data is available under an open data policy, a production deployment of a forecasting system that uses national consumption telemetry would need to comply with the EU Network Code on Cybersecurity (NCCS), GDPR if individual metering data is used, and the upcoming EU AI Act requirements for high-risk AI systems used in critical infrastructure. These aspects are beyond the scope of this thesis but will be required for production deployment.

4.9 Recommendations for Smart Grid Operators

Based on the empirical results, MLOps assessment and practitioner feedback, the following six recommendations are provided to smart grid operators who are considering implementing AI-based STLF systems.

1. Use XGBoost as the main operational forecasting model. The results demonstrate that XGBoost is the most accurate and stable model for one-step-ahead system-level load forecasting with a diverse feature set. Its grid-search-optimized configuration requires around 12 months of training data to approach its final accuracy, and this accuracy is consistent across cross-validation folds. All seven pairwise Diebold-Mariano tests (Table 14) confirm XGBoost's performance advantage over every other model at $p < 0.0001$, providing strong statistical grounds for recommending it as the primary production model.
2. Need at least 12 months of data for deployment. Neither XGBoost nor LSTM reach operationally acceptable accuracy with less than nine months' data. A minimum of 12 months of pre-production deployment should be required for new deployments before forecasts are used for dispatch or demand response.
3. Adopt monthly rolling retraining and a validation gate. Demand patterns change as EV uptake grows and prosumer patterns emerge. The AWS-based MLOps platform supports monthly automated retraining and promotion to production via a validation gate, so only models with accuracy above a threshold are deployed. The MAPE drift limit of 2.0% identified by the interview participant should trigger an automated retraining if exceeded for three days.
4. Explain model predictions with SHAP feature importance. As AI-based decision support systems are increasingly regulated in European energy markets, market operators will have to explain forecasts to regulators and auditors. SHAP values offer a principled explanation for each prediction. The dominance of load_lag_1h also offers an audit rule of thumb: if a forecast is way off the current

load, but not off the current weather or calendar, it should be checked before dispatch.

5. Include demand response thresholds in the forecasting pipeline. The 83-hour peak and 646-hour minimum load analysis can be included in the daily forecast report, providing pre-calculated DR signals instead of manual analysis. With a MAPE of 0.78%, the expected absolute error on a 12,000 MWh/h threshold is 94 MWh/h - low enough to confidently pre-position demand response resources.
6. Include multiple weather stations and consider EV loads. Increasing the number of representative weather stations from northern Finland would account for spatial variation at low cost. Looking ahead, the expert interview highlighted EV charging as the main trend in demand that will change the load patterns in Finland when EV share reaches 15-20%. Day-ahead price signals and EV fleet information should be added as new features in the next major release of the forecasting system.

4.10 Industry Interview: Thesis Value and Future of Smart Grids

To complement the quantitative findings with practitioner perspective, this thesis incorporates a semi-structured expert interview component. The interview validates the technical results and provides operational grounding for the recommendations in Section 4.9.

4.10.1 Interview Objective and Participant Description

The interview had three purposes: to determine if the modeling and evaluation framework meet expectations for a STLF system; to determine if the deployment architecture on AWS and MLOps approach is operationally feasible; and to gain practitioner perspectives on AI

for smart grid operations, including challenges and the future of demand response. The participant is a professional with expertise in energy data analytics, power system operations, and energy management systems, employed at the Finnish transmission system operator responsible for the national grid. The participant holds a senior role in grid operations and energy data management, with direct experience in operational forecasting standards and digital infrastructure deployment at the national transmission level. The interview followed academic research ethics: the participant gave informed consent to participate, participation was voluntary, and personal details are not disclosed in this report.

4.10.2 Summary of Responses

The interview responses coalesced around four key points that confirm and explain the quantitative findings.

On forecast accuracy, the participant confirmed that 15-minute interval forecasting is the primary operational standard at the national transmission level, as it directly supports real-time balancing and reserve activation decisions which makes short-term load forecasting the most operationally significant forecasting horizon in day-to-day grid management. Within this context, the participant validated that the operational target for system-level day-ahead forecasting in the Nordic region is a MAPE below 2%, and a MAPE below 1% is considered exceptional. The XGBoost performance of 0.78% MAPE was found to be competitive with commercial systems deployed by Nordic TSOs. The most operationally relevant result was XGBoost 99.1% peak accuracy, the participant confirmed that the most critical errors are not average errors but tail errors, which occur during extreme weather events.

Regarding the viability of MLOps, the participant admitted that cloud-based forecasting pipelines are increasingly the norm in larger utilities, but smaller regional system operators have constrained budgets, IT governance, and challenges in integrating with existing SCADA systems. The AWS implementation was considered to be technically sound, but possibly over-engineered to small utilities. Early-stage digital utilities were suggested to use a lighter-weight deployment with AWS Lambda to perform inference and S3 to store data instead of SageMaker.

Regarding the future of smart grids and demand response, the respondent cited prosumer growth and EV charging as the two most notable trends in demand-side that would influence Finnish load patterns in the coming decade. The existing models that are educated on past trends can degrade faster than anticipated when EV adoption goes beyond 15-20% because EV charging behavior is price-elastic in a manner that is not currently present in the data of the past. This supports the necessity of continuous retraining of the models, adaptive pipeline structures and the incorporation of price signal features in subsequent system iterations.

Overall, the expert interview confirms the key technical finding that XGBoost with engineered lag and weather features delivers state-of-practice accuracy for system-level STLF, and situates that result within the operational context of Nordic grid management: in a real-world setting where 15-minute interval forecasting is the primary unit of grid decision-making, the accuracy and pipeline architecture presented in this thesis represent a plausible route to production deployment, with appropriate simplification for smaller Nordic operators.

5 Conclusion and Future Work

In this chapter, the thesis findings are condensed, the contribution to the field is outlined, and the most promising avenues for future research are discussed.

5.1 Summary of Findings

The main research question addressed by this thesis is: how can cloud-based AI models improve short-term electricity load forecasting and help optimization for smart grids over conventional methods? The empirical results provide a clear answer to the research question.

Five different models were tested using the same dataset (Fingrid dataset 193 (electricity consumption), January 2023-December 2024) and the same parameters and a common test period of 2,583 hours. The results are shown in Table 19.

Table 19. Test-set performance summary aligned with 2,583-hour evaluation window.

Model	RMSE (MWh/h)	MAPE	R^2	Key finding
XGBoost	112	0.78%	0.991	Best AI model; 87% RMSE reduction vs Naive

ARIMA (1,1,1)	187	1.34%	0.977	Best classical baseline; oracle-feedback advantage
LSTM	250	1.97%	0.958	71% improvement vs Naive; winter bias −107 MWh/h
Prophet	515	4.09%	0.820	Viable with correct config (growth="flat", regressors)
Naive (weekly)	861	6.49%	0.498	Minimum threshold; all AI models significantly better

XGBoost performed best on all measures, with RMSE of 112 MWh/h and MAPE of 0.78% (87% RMSE improvement on Naive model). The Diebold-Mariano test found all pairwise differences to be statistically significant at $p < 0.0001$, showing that the performance differences reflect actual model differences, and not random variation in the test period. The analysis has four major implications. First, feature engineering is superior to sequence learning at short horizons. XGBoost with 55 hand-engineered lag, weather and calendar features outperformed LSTM on all metrics. The SHAP analysis reveals that load_lag_1h is the most important feature, which explains why a model that is explicitly given the autocorrelation signal as a feature is better off than a model that needs to learn it from sequences. This has implications: for short-term forecasting, feature engineering has more impact on accuracy than model complexity. Second, model configuration matters. Prophet's original R^2 of -0.43 (default configuration) to R^2 of 0.82 (growth="flat", weather regressors, and proper cross-validation window) is an example of how a misconfigured AI model can perform worse than a flat mean, while the same model properly configured is a viable

operational tool. The reason for failure across all configurations was growth="linear" for a stationary load series. Third, XGBoost and LSTM need at least 12 months of training data to achieve operationally satisfactory accuracy. Cross-validation folds indicated fold-1 RMSE greater than 900 MWh/h for XGBoost with less than four months of training, and 93 MWh/h by fold 4 (with more than 12 months of training). This data requirement should be considered a hard requirement for production use. Fourth, forecast accuracy leads to operational insights. Using the XGBoost test-set forecasts, the demand response module automatically identified 83 peak hours in the test period with 22,259 MWh of curtailable excess and 646 low-load windows for load shifting. This illustrates the forecast-to-action value chain that the thesis said is the true value of forecasting systems.

These findings must be interpreted within their boundaries. The main limitation is that all results are based on a single national grid dataset for two calendar years in one country, and therefore the generalisability of the model rankings to other grid architectures, demand regimes or availability of data is limited. The 0.78% MAPE of XGBoost is a good result, due to favourable conditions: the national load signal is strongly autocorrelated, the data comes from the API and the test period is in a seasonally consistent regime of demand (autumn-winter). The training data for 2023 and 2024 are not representative of higher levels of renewable penetration, distributed generation or structural demand shifts due to EV adoption, which can cause performance degradation. The Diebold-Mariano tests do not imply that the performance differences are statistically significant in a different evaluation context, nor do they guarantee that the same performance differences would occur in a different context of data or forecasting horizon. The findings are not negated by these constraints, but rather they set the conditions under which they are to be trusted.

5.2 Contributions to the Field

This thesis makes contributions in two distinct domains: theoretical contributions to the academic literature on short-term load forecasting, and practical contributions to the applied field of smart grid analytics and MLOps deployment.

Theoretical contributions

A statistically validated five-model cross-family comparison on national system-level data. In past STLF research, two or three models were compared, without statistical testing, and using single-site or residential data sets. Five models from three different algorithmic families (deep learning (LSTM), additive decomposition (Prophet), and gradient-boosted ensemble learning (XGBoost)) are compared to two classical baselines, and all seven pairwise differences are validated by Diebold-Mariano tests at $p < 0.0001$. This addresses a methodological gap in the comparative STLF literature, and provides a reproducible performance benchmark for Finnish national grid data.

A case study for a stationary electricity load that uses a Prophet configuration. The results of the systematic documentation of the failure of Prophet in the stationary electricity load data and the specific configuration changes needed to achieve viable performance is an original empirical contribution not previously reported in national grid contexts. The result of growth="flat" and the inclusion of weather and calendar regressors makes Prophet worse than a flat mean to $R^2=0.82$, which has direct implications for any researcher using Prophet for stationary load series.

Empirical evidence on the minimum data requirements for AI-based STLF. The cross validation analysis gives quantitative evidence that both XGBoost and LSTM need at least 12 months of training data to achieve operationally acceptable accuracy for national load

data. This threshold is based on the convergence of the RMSE, which is the fold-over-fold convergence, and is a guideline for the amount of data that is missing in most published STLF studies.

Practical contributions

A cloud-based MLOps pipeline blueprint for operational STLF. The six-stage AWS pipeline from ingestion to monitoring provides a deployment blueprint with concrete service selections, promotion criteria, monitoring thresholds, and a maturity assessment against Kreuzberger et al. (2023) dimensions. The four documented failure modes and their mitigations provide engineering insights for practitioners moving from research prototypes to production systems, addressing the deployment gap that the literature review identified in Chapter 2.

An end-to-end forecast-to-action demonstration for demand response. The thesis shows how test-set predictions from XGBoost are correlated with demand response trigger signals and load-shifting windows, and illustrates the entire prediction-to-action pipeline with publicly available data and open-source tooling. The 83 peak hours identified with 22,259 MWh of curtailable excess and 646 low-load windows for load shifting shows that sub-1% MAPE accuracy is an operational signal, rather than just a statistical improvement. This is a bridge between the gap in the thesis motivation between forecast accuracy research and its operational use.

5.3 Suggestions for Further Research

Four research directions emerge directly from the limitations and open questions of this thesis, each offering a concrete and actionable extension.

Multi-station weather data and Nordpool price regressors. The first data constraint to be tackled is the single-station Helsinki weather proxy. Future research should gather hourly weather data from at least three more stations in Oulu, Jyväskylä and Rovaniemi, and examine if the MAPE of XGBoost can be improved by incorporating spatial variation of temperature during extreme cold weeks, when the weather in Helsinki is most different from the national averages. A Nordpool day-ahead electricity price regressor should also be added separately and its SHAP contribution measured: If the price sensitivity is apparent in the feature importance ranking, then it indicates that prosumer and industrial demand response is already present in the load signal and should be modelled explicitly.

Probabilistic forecasting and uncertainty quantification. In this thesis, XGBoost and LSTM both give a point forecast. Operators require more than just the most probable load value; they require a confidence interval. Future research should consider using XGBoost quantile regression (predicting the 10th, 50th and 90th percentile of load) and compare the interval coverage with conformal prediction wrappers for the current model. The operational question is whether the prediction intervals at the 90th percentile threshold are likely to contain the peak hours identified in Section 4.5, and whether the use of the width of the intervals to size reserves is a cost-effective way of minimizing curtailment costs compared to point-forecast-based dispatch.

Real-time streaming deployment and latency benchmarking. The existing pipeline runs in batch mode on the historical data. The Lambda-to-SageMaker inference call needs to finish during the dispatch cycle time for 15-minute intraday markets. Future work needs to be done by deploying the XGBoost inference endpoint in a live AWS environment and performing load testing to determine the latency for the p50, p95, and p99 percentile of requests under concurrent request conditions. The question is whether a single-row XGBoost prediction, including feature construction from the latest API data, can be done in

less than 30 seconds on a 512 MB Lambda function with sufficient margin to meet the intraday settlement requirement of 15 minutes.

Structural demand change modelling for EV integration. The interviewee agreed that the introduction of EVs will have a tangible impact on the load profile of the country when the share of EVs in the fleet is more than 15–20%. Future work should gather EV charging telemetry data from the EV charging network operators in Finland and evaluate if the MAPE of the XGBoost model can be improved by introducing the state of charge of EVs as a regressor during the evening peak hours (17:00-21:00), when EV charging load is highest. As EV penetration increases, the current model should be evaluated as a lower latency adaptation strategy if it systematically underpredicts during these windows, as it can be adapted online, incrementally on the new data each week, without having to be retrained every month from scratch.

References

Al-Jumaili, A. H. A., Muniyandi, R. C., Hasan, M. K., Paw, J. K. S., & Singh, M. J. (2023). Big Data Analytics Using Cloud Computing Based Frameworks for Power Management Systems: Status, Constraints, and Future Recommendations. *Sensors, 23(6)*, 2952. https://doi.org/10.3390/s23062952

Arsalan, M., Ayaz, M., Ali, Y., & Baig, U. (2025). AI-Enabled Energy Load Forecasting for Smart Grid 'Management. *International Journal of Scientific Research and Engineering Development, 8(6)*. https://doi.org/10.5281/zenodo.17926582

Biswal, B., Deb, S., Datta, S., Ustun, T. S., & Cali, U. (2024). Review on smart grid load forecasting for smart energy management using machine learning and deep learning techniques. *Energy Reports, 12*, 3654–3670. https://doi.org/10.1016/j.egyr.2024.09.056

Dong, Q., Huang, R., Cui, C., Towey, D., Zhou, L., Tian, J., & Wang, J. (2025). Short-term electricity-load forecasting by deep learning: A comprehensive survey. *Engineering Applications of Artificial Intelligence, 141*, 110980. https://doi.org/10.1016/j.engappai.2025.110980

Fang, X., Misra, S., Xue, G., & Yang, D. (2012). Smart grid—The new and improved power grid: A survey. *IEEE Communications Surveys & Tutorials, 14(4)*, 944–980. https://doi.org/10.1109/SURV.2011.101911.00087

Fingrid. (n.d.). *Fingrid Open Data*. Retrieved 2026-04-10 from https://data.fingrid.fi/en

Harikrishnan, G. R., Sreedharan, S., & Binoy, C. N. (2025). Advanced short-term load forecasting for residential demand response: An XGBoost-ANN ensemble approach. *Electric Power Systems Research, 242*, 111476. https://doi.org/10.1016/j.epsr.2025.111476

Hong, T., & Fan, S. (2016). Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting, 32(3)*, 914–938. https://doi.org/10.1016/j.ijforecast.2015.11.011

KC, S., & Rone, S. (2024). *Comparing Prophet, XGBoost, and LSTM Models for Web Traffic Forecasting: Assessing Model Performance Across Various Time Series Forecasting Scenarios* [Bachelor's thesis, KTH Royal Institute of Technology]. DiVA. Retrieved 2026-04-10 from https://www.diva-portal.org/smash/get/diva2:1887941/FULLTEXT01.pdf

Masood, M. Y., Aurangzeb, S., Aleem, M., Chilwan, A., & Awais, M. (2024). Demand-side load forecasting in smart grids using machine learning techniques. *PeerJ Computer Science, 10*, e1987. https://doi.org/10.7717/peerj-cs.1987

Patsakos, I., Vrochidou, E., Papakostas, G. A., & Jiang, S. (2022). A survey on deep learning for building load forecasting. *Mathematical Problems in Engineering, 2022*, 1008491. https://doi.org/10.1155/2022/1008491

PHOENIX project. (2022, September 26). *Uncovering the benefits of demand-response*. Retrieved 2026-04-10 from https://phoenix-h2020.eu/uncovering-the-benefits-of-demand-response/

Pramono, S. H., Rohmatillah, M., Maulana, E., Hasanah, R. N., & Hario, F. (2019). Deep Learning-Based Short-Term Load Forecasting for Supporting Demand Response Program in Hybrid Energy System. *Energies, 12(17)*, 3359. https://doi.org/10.3390/en12173359

Rahman, M. A. (2025). *Optimizing Smart Grid Flexibility and Resilience with Demand Response, Renewable Integration and Energy Storage* [Master's thesis, University of Vaasa]. Osuva. Retrieved 2026-04-10 from https://osuva.uwasa.fi/items/40a02989-e003-4dde-83f6-3305e514ae7f

Rizvi, S. S. H., Chaturvedi, K. T., & Kolhe, M. L. (2023). A review on peak shaving techniques for smart grids. *AIMS Energy, 11(4)*, 723–752. https://doi.org/10.3934/energy.2023036

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, & R. Garnett (Eds.),

Advances in Neural Information Processing Systems (Vol. 28, pp. 2503–2511). Curran Associates, Inc.

Stryker, C., & Mucci, T. (n.d.). *What is machine learning operations (MLOps)?*. IBM. Retrieved 2026-04-10 from https://www.ibm.com/think/topics/mlops

Ullah, M., Narayanan, A., Wolff, A., & Nardelli, P. (2021). Smart Grid Information Processes Using IoT and Big Data with Cloud and Edge Computing. In *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)* (pp. 956–961). IEEE. https://doi.org/10.23919/MIPRO52101.2021.9596885

Mujeeb, S., Javaid, N., Javaid, S., Rafique, A., & Ilahi, M. (2019). Big data analytics for load forecasting in smart grids: A survey. In *International Conference on Cyber Security and Computer Science (ICONCS 2018)* (pp. 193–202). Retrieved 2026-04-10 from https://www.researchgate.net/publication/330848350_Big_Data_Analytics_for_Load_Forecasting_in_Smart_Grids_A_Survey

Albahli, S. (2025). LSTM vs. Prophet: Achieving superior accuracy in dynamic electricity demand forecasting. *Energies, 18*(2), Article 278. https://doi.org/10.3390/en18020278

Rafi, O., Ahmad, S., Ahmad, M. H., & Shahid, Z. (2025). A data-driven hybrid DNN-residual tree XGBoost model with comparative assessment for smart grid load forecasting. In *Proceedings of the 2025 International Conference on Frontiers of Information Technology (FIT)*. IEEE. https://doi.org/10.1109/FIT67061.2025.11333771

Pandya, R. V., Desai, M. M., & Vala, U. P. (2025). Forecasting-driven demand response management in smart grids using neural networks. *International Journal of Research and Innovation in Applied Science, 10*(5), 1480–1489. https://doi.org/10.51584/IJRIAS.2025.1005000129

Musundire, A. (2025). Understanding the Research Onion and Its Application in Educational Leadership and Management Research: Making Use of Saunders' Research Model. In *Research Methods for Educational Leadership and Management*. GI Global Scientific Publishing.

Harvey, D., Leybourne, S., & Newbold, P. (1997). Testing the equality of prediction mean squared errors. *International journal of forecasting*, *13*(2), 281-291. https://doi.org/10.1016/S0169-2070(96)00719-4

Kreuzberger, D., Kuhl, N., & Hirschl, S. (2023). Machine Learning Operations (MLOps): Overview, Definition, and Architecture. *IEEE Access, 11* , 1. https://doi.org/10.1109/ACCESS.2023.3262138

Bera, S., Misra, S., & Rodrigues, JJPC (2015). Cloud Computing Applications for Smart Grid: A Survey. IEEE transactions on parallel and distributed systems, 26 (5), 1477-1494. https://doi.org/10.1109/TPDS.2014.2321378

M, RP, & P, BC (2025). Data-Driven Forecasting for Grid Stability: Implementing XGBoost in Smart Energy Systems . IEEE. https://doi.org/10.1109/CMSS66566.2025.11182513

Rotib, H. W., Nappu, M. B., Tahir, Z., Ardiaty Arief, & Muhammad. (2021). Electric Load Forecasting for Internet of Things Smart Home Using Hybrid PCA and ARIMA Algorithm. International Journal of Electrical and Electronic Engineering and Telecommunications, 425–430. https://doi.org/10.18178/ijeetc.10.6.425-430

Masood, M. Y., Aurangzeb, S., Aleem, M., Chilwan, A., & Awais, M. (2024). Demand-side load forecasting in smart grids using machine learning techniques. PeerJ. Computer Science, 10, e1987. https://doi.org/10.7717/peerj-cs.1987