Hierarchical Workload Forecasting and Reconciliation in Renewable Energy Industry

Pierre-Luc Genest

A Thesis in

The Department of

Business Analytics and Technology Management

Presented in Partial Fulfillment of the

Requirements for the Degrees of Masters of Science in

Business Analytics and Technology Management at

Concordia University

Montreal, Quebec, Canada

December 2025

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared,

By: Pierre-Luc Genest

Entitled: Hierarchical Workload Forecasting and Reconciliation in Renewable Energy Industry

and submitted in partial fulfillment of the requirements for the degree of

Master of Business Analytics and Technology Management

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final Examining Committee:

_____ Chair

Chair's Name

_____ Examiner

Dr. Dongliang Sheng

_____ Examiner

Dr. Danielle Morin

Supervisor

_____

Dr. Salim Lahmiri

Approved by: Date:

_____

Chair of Department or Graduate Program Director

Date:

_____

Dean of Facult

Abstract

Hierarchical Workload Forecasting and Reconciliation in Renewable Energy Industry

Pierre-Luc Genest

Having an accurate workload forecast is important for workload capacity planning, since the goal of capacity planning is to optimally allocate resources to current and future demand requirements. Capacity planning alongside workload forecasting determines employee headcount, backlog levels, and scheduling requirements. There are a growing number of studies in recent years that show that machine learning (ML) outperforms traditional statistical benchmarks. The thesis evaluates whether Light Gradient Boosting Machines (LightGBM), K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and feedforward neural networks (NN) consistently outperform simple statistical benchmarks for short-term and long-term workload forecasting. Based on the findings of the experiments, one recommendation to managers is that workload forecasting should be based on simple statical methods rather than ML models. For both the budget (long-term) and schedule (short-term) forecasting task, simple statistical methods outperformed KNN, LightGBM, SVM, and NN. The high level of noise that is present in workload time-series makes it unlikely that ML models will outperform simple statistical benchmarks. A second recommendation to managers is to incorporate hierarchical reconciliation using minimum trace ordinary least squares to improve forecasting accuracy while making the forecasts coherent.

Key Words: workload, hierarchical reconciliation, budget forecasting, schedule forecasting, LightGBM, Neural Networks, K-Nearest Neighbors, Support Vector Machines, workload capacity planning, forecasting benchmarks

Acknowledgment

I am grateful to my friends and family who have supported me. During my time in graduate studies, I had the chance to form friendships with fellow students that extend beyond academia. The courses that I have taken in data management and data mining at Concordia have allowed me to improve my skills as a business intelligence analyst. I thank Salim Lahmiri who has inspired me to push forward with my career in business analytics. The graduate program of John Molson School of Business enabled me to build on the competencies that I have gained through my work experience.

Table of Contents

List of Figures

List of Tables

Acronyms

Acronyms	*Description*
ACF	autocorrelation function
ARIMA	autoregressive integrated moving average
EMH	efficient market hypothesis
ERM	empirical risk minimization
IoT	internet of things
KNN	k-nearest neighbors
LightGBM	light gradient boosting machines
MAPE	mean absolute percent error
MinT	minimum trace
ML	machine learning
MSE	mean squared error
NN	neural networks
OLS	ordinary least squares
PACF	partial autocorrelation function
RMSE	root mean squared error
NN	neural network
SAP	system applications and products in data processing
SARIMA	seasonal autoregressive integrated moving average
SCADA	supervisory control and data acquisition
SVM	support vector machines
SVD	singular value decomposition
WLS	weighed least squares
WRMSSE	weighted root means squared scale error
XGBoost	extreme gradient boosting

1. Introduction

The introduction begins by providing an overview of the importance of data mining in supporting data driven decision-making. Workload forecasting and capacity planning are then discussed. The third section of the introduction examines notable developments in the renewable energy sector, since the research problem is focused on workload forecasting in the wind energy sector. The intent of the thesis is not only to address gaps in literature, but also to improve existing workload forecasting techniques for a firm who manufactures and services windfarms. The fourth section of the introduction provides context on the market that the firm is operating under. The introduction then states the research questions and highlights the contributions that the research intends to bring forth. Lastly, an overview of how the thesis is structured is provided.

1.1 Enabling Data Driven Decision-Making Through Data Mining

In recent years, there has been a growing shift towards data driven decision-making. While traditional decision-making heavily relies on experience and intuition to guide decisions, data driven decision-making exploits data and statistics to support evidence based decision-making. Studies have shown that although data driven decision-making generally leads to improved decision-making, decision-making based on experience and intuition is still common practice (Goodwin et al., 2023; Sanders & Manrodt, 2003). As technological such as Internet of Things (IoT), cloud computing, and artificial intelligence continue to improve, it is becoming increasingly important for organizations to employ business analytics to generate objective insight used for data driven decision-making. Business analytics are practices and technologies that bring quantitative data to bear on decision-making (Shmueli et al., 2017). When properly used, business analytics can provide superior foresight when compared to relying solely on intuition and experience. Foresight is "the ability to judge correctly what is going to happen in the future and plan actions based on this knowledge" (*Foresight*, 2025). Organizations with superior foresight over their competitors have the potential of achieving a sustainable competitive advantage within an industry, since the organization is in a better position to anticipate future outcomes and plan accordingly.

The first form of analytics consists of descriptive analytics, and it attempts to answer the questions "what happened" (Silva et al., 2021). Descriptive analytics involve examining historical data to identify trends and patterns. Tracking key performance indicators (KPIs) such as profit margins and market growth are applications of descriptive analytics. The second form of analytics is diagnostic analytics and is based on the results of descriptive analytics. Diagnostic analytics attempts to answer the question, "why did this happen" (Silva et al., 2021). Undertaking a root cause analysis with the support of descriptive statistics is an application of diagnostic analytics. The third form of business analytics is based on diagnostic analytics, which attempts to answer the question "what will happen" (Silva et al., 2021). Predictive analytics uses historical data, statistical models, and machine learning (ML) to predict future outcomes. Forecasting turbine failure rates, customer retention, and equipment downtime are applications of predictive analytics. Once descriptive, diagnostic, and predictive analytics have been performed, prescriptive analytics can then be used to support data driven decision-making. Prescriptive analytics answers the question "what should we do", by using the insight that was gained from the previous 3 forms of analytics

(Silva et al., 2021). Linear programming and heuristic approaches to inventory optimization are applications of prescriptive analytics.

Data mining is the process of analyzing data to extract valuable information from data. Data mining frameworks such as "SAS Sample, Explore, Modifying, Modeling, and Assessing (SEMMA) Methodology" and "IBM SPSS Modeler: Cross-Industry Standard Process for Data Mining (CRISP-DM)", offer guidance to extract value from business analytics (*CRISP-DM Help Overview*, 2021; *Introduction to SEMMA*, 2017). Typically, the data mining process begins by defining the business problem that the data mining project should address. Understanding how stakeholders will use the results of the data mining project assists in ensuring that the data mining objectives will provide business value. Once the business problem is properly understood, the next step is to obtain data to be used in the analysis. Obtaining data may require extracting data from multiple sources and consolidating it into one location.

Once all the raw data is collected, the next step generally is to explore and clean the raw data. Exploration may consist of examining summary statistics (e.g. standard deviation) or creating interactive dashboards that allow users to drill and filter various charts. Cleaning raw data may involve imputations and removing outliers. Another preprocessing step that is often employed in data mining projects is reducing the dimension of data. Dimension reduction techniques can be used to reduce the number of records or fields. Aggregating records such as taking the monthly sum of workload from daily workload, reduces the number of records to process. Feature reduction techniques reduce the number of features instead of records by either merging fields together or selecting the most important features via some predefined search criteria. Data dimension techniques such as Principal Component Analysis (PCA) can be used to improve model performance and decrease computational requirements. Feature engineering which applies transformations to existing features with the aim of improving model performance is also a common preprocessing step when developing models.

When developing models for data mining projects, it is important to understand the data mining task. Data mining tasks can be broadly divided into supervised and unsupervised learning. Supervised learning consists of classification, prediction, and time-series forecasting models. When using supervised learning models, the target variable is known, and the output of the model can be compared against the actual value to evaluate the performance of the model. While prediction models predict continuous variables, classification models predict categorical variables. Time-series models are a special application of prediction and classification models in that it uses temporal data to predict future values based on past patterns. Linear regression, neural networks, and logistic regressions are examples of supervised learning.

Unsupervised learning differs from supervised learning, in that the model learns patterns from unlabeled data, thus making it more difficult to evaluate model performance. Segmentation, association rules, and collaborative filtering are applications of unsupervised learning. Once the data mining task and techniques have been selected, data partitioning should be performed for supervised tasks. At this point, the models that have been selected for training can be optimized via hyper-parameter tuning. For supervised learning, the models should be evaluated against a validation set using performance metrics such as root mean squared error. Based on the results of

the experiments, the best performing model should be deployed for the clustering, classification, or prediction task. General data mining procedures such as the one previously described are used to facilitate the process of extracting value from data to support data driven decision-making.

1.2 Workload Modeling

A systematic literature review of 275 workload modeling articles found that workload modeling can be categorized into 5 broad categories (Safarishahrbijari, 2018). The first category are *qualitative models*, which rely on expert opinion and professional judgment. Expert opinion-based forecasting and the Delphi techniques are examples of models that fall within this category. Expert opinion-based forecasting involves gathering insight from industry experts to predict workforce needs based on their experience and knowledge. The Delphi technique consists of structured communication methods where a panel of experts answer questions in multiple rounds to reach a consensus. *Time-series models* are the second category of workload models. *Time-series models* utilize historical data to identify patterns and trends to project future workload requirements (Safarishahrbijari, 2018). Naive, autoregressive integrated moving average (ARIMA), and exponential smoothing are examples of *time-series models*.

The third category of workload models are *regression models* that aims to quantify the relationship between the dependent variable and one more independent variable(s) (Safarishahrbijari, 2018). While *time-series models* are primarily focused on temporal dependencies, *regression models* are primarily focused on quantifying relationships between variables. Multiple linear regression, polynomial regression, and logistic regressions are examples of *regression models*. The fourth category of workforce models are *optimization models*, that use mathematical techniques to find the best allocation of resources to maximize or minimize an objective function (Safarishahrbijari, 2018). Linear programming and integer programming are types of models that fall under *optimization models*. *Simulation models* are the last category of models that were identified in the literature review (Safarishahrbijari, 2018). *Simulation models* create a virtual representation of the interaction between system components and examine alternative scenarios over time.

Sanders and Manrod examined the adoption of qualitative and quantitative forecasting among various firms. Data from self-reported surveys was collected from 240 U.S. companies (Sanders & Manrodt, 2003). The study found that only 28% of firms primarily used quantitative methods, while 30% of firms primarily used qualitative methods. The remaining firms interviewed relied on a combination of qualitative and quantitative methods. Qualitative focused firms typically had lower access to time-series data.

A more recent study supports the findings of Sanders and Manrodt in that there is a relatively low adoption of systematic forecasting methods despite their benefits (Goodwin et al., 2023). The study utilized survey data from 370 managers across industries and conducted 20 in-depth semi-structured interviews. The survey found that less than 30% of firms reported regularly using systematic forecasting methods, while a significant portion of firms (44%) rarely or never use them. Lack of familiarity with statistical methods and distrust among statistical software + techniques were among the most cited reasons why the implementation of systematic forecasting methods in business is substantially behind academia. The authors emphasized that simplifying quantitative forecast implementation, enhancing familiarity of statistical methods among decision makers, and

customizing solutions to fit the specific needs of individual firms lead to higher adoption of systematic forecasting methods. The thesis will focus on *regression models and time-series models* since they are best suited in predicting future workload requirements. The United States leads in workload modeling publications, followed by Canada, and the United Kingdom (Safarishahrbijari, 2018). Since the 1980s, there has been a steady increase in workload modeling publications.

1.3 Workload Forecasting & Capacity Planning

Modeling is the process of creating a simplified representation of a real-world system to analyze, understand, and predict behavior. Workload modeling incorporates aspects of planning and forecasting. While planning and forecasting are closely related concepts in organizational strategy, they serve distinct purposes and involve different processes. Planning is a process of setting goals, developing strategies, and determining the actions and resources needed in achieving desired outcomes. The focus of planning is on "what to do" and on "how to do it" to meet future objectives. Forecasting on the other hand is the process of predicting future conditions or events based on historical data (Letmanyi, 1985). Forecasting focuses on "what is likely to happen" in the future. In short, planning focuses on achieving specific objectives and forecasting focuses on anticipating future events.

Workload modeling helps organizations estimate future workload and optimize allocations of resources to meet workload requirements. "The main objective in capacity planning is to assign fixed maintenance capacity (resources) to meet fluctuating maintenance workload in order to achieve the best utilization of limited resources (Al-Fares & Duffuaa, 2009)". Capacity planning is used to determine employee headcount, backlog levels, overtime workload, and subcontracting requirements. Capacity planning and workload forecasting are also used to support an organization's maintenance strategy. A maintenance strategy is a structured approach to managing equipment to ensure reliability and minimize downtime. The three main components of a maintenance strategy are design-out maintenance, preventive maintenance, and corrective maintenance (El-Naggar et al., 2023). Design-out maintenance consists of designing equipment to reduce the causes of failure, thus decreasing maintenance requirements. An application of design-out maintenance would be manufacturing equipment with durable material which is resistant to deterioration. Preventive maintenance is the second component of a maintenance strategy, and it consists of performing an action before a failure occurs, to reduce the likelihood of breakdown.

Preventive maintenance can either be based on condition-based maintenance or time-based maintenance (El-Naggar et al., 2023). Condition-based maintenance uses real-time data from sensors and inspections to determine when maintenance is needed. An application of condition-based maintenance would be monitoring the temperature of a bearing to determine when the bearing should be replaced. Time-based maintenance on the other hand is performed at scheduled intervals, regardless the condition of the component. The maintenance interval is often based on manufacturer recommendations and historical failure rates. Using historical failure rates to determine that a gearbox should be replaced every 10 years is an example of using time-based maintenance. Changing the grease of a vehicle on a fixed interval based on the manufacturer's recommendations is another example of time-based maintenance.

Corrective maintenance is the third component of a maintenance strategy and it consists of performing maintenance after a failure occurs. Preventive maintenance is associated with planned maintenance, while corrective maintenance is associated with unplanned maintenance. When trying to reduce downtime, planned maintenance is often more desirable than unplanned maintenance. Being able to take proactive steps before failure occurs reduces the response time associated with unplanned failure. For example, being aware that a component is approaching failure would allow staff members to order the required material to replace the component, prior to the occurrence of failure. In this case, the time it takes for the material to arrive on site would be eliminated from downtime. Capacity planning and workload forecasting can be used together to assist in optimizing resource allocation to support condition-based, preventive-based, and corrective-based maintenance.

Time-series forecasting methods can be grouped into traditional statistical approaches and ML approaches (Sandhya Arora, 2024). Linear regression, logistic regression, and ARIMA are among some of the traditional statistical approaches used in time-series forecasting. Traditional statistical methods start with a predefined hypothesis that describes the relationship between variables. These types of models commonly rely on strong assumptions to draw inferences about the population. For instance, confidence intervals and p-values generated from linear regression assume that the residuals are normally distributed. Traditional statistical methods are inference-focused in that they emphasize a) understanding relationships, b) quantifying uncertainty & c) testing hypothesis. Simple statistical techniques such as rolling averages and exponential smoothing, while not inference-focused, are also considered part of traditional time-series forecasting.

ML approaches generally emphasize prediction accuracy over understanding the underlying mechanisms. These types of models learn patterns directly from data without predefined hypotheses. Decision trees, support vector machines (SVM), and k-nearest neighbors (KNN) are examples of ML approaches. Since these approaches have few assumptions about the characteristics of the underlying data, ML approaches are better equipped at handling complex, non-linear relationships.

1.4 Renewable Energy Industry

The section provides notable developments in the renewable energy industry that has motivated the author to conduct workload forecasting research in the wind energy sector.

Overdependence on fossil fuels is leading to environmental degradation and energy insecurity. Wind energy is among the most promising technologies to reduce carbon emissions and meet growing energy demands. Despite its long history, wind energy adoption has been limited until the start of the 21st century (Kaldellis & Zafirakis, 2011). The global markets leading wind energy capacity expansion are EU, USA, and China. Technological improvements, governmental policy shifts, and decreasing costs of implementation are factors that have contributed to the drastic rise of wind energy in recent years. Denmark and Germany are technological leaders in the wind energy industry (Glowik et al., 2023). Emerging markets such as Brazil and Vietnam have also experienced increased wind power adoption in recent years.

Lawrence Berkely National Laboratory for Wind Energy Technologies Office projects significant increases in wind energy capacity in the U.S. for the upcoming years (*Land-Based Wind Market Report: 2023 Edition*, 2023). The projections are based on trends summarized in **Table 1.**

Table 1: Wind Energy Trends

Trend Categories	*Trend Description*
Installation	• U.S. added 8.5 GW of wind power capacity in 2022, totaling $12 billion of investment. • Cumulative wind capacity grew to more than 144 gigawatts (GW) by the end of 2022
Technology	• Average rated capacity of newly installed wind turbines in the United States for 2022 was 3.2 MW, up 7% from the previous year and 350% since 1998−1999.
Cost	• Projects installed over the past 16 years have, on average, incurred lower operations and maintenance (O&M) costs than older projects.

The findings of the Land-Based Wind Market Report: 2023 are primarily centered around the U.S. market which is the country with the second biggest wind energy market, behind China. In terms of global trends, 7.4% of the total electricity produced in the world was met from renewable energy in 2016, while in 2021 it increased to 12.8% (Yolcan, 2023). Among the renewable technologies, wind energy occupies the second largest market share, behind photovoltaics. The renewable energy industry is operating under an industry 4.0 environment, where digital innovation is critical to remain competitive. Digital innovation consists of "the combination of advanced technological services and processes with the goal of improving existing products, processes, and supply chain" (Gupta & Jauhar, 2023). An important outcome of industry 4.0 is that new technologies are leading to shifts towards data driven decision-making rather than intuition-based decision-making. Companies operating under an industry 4.0 environment must be able to capture intelligence from big data to remain competitive.

Online transactional processing (OLTP) and internet of things (IoT) are the primary platforms that are used by companies operating in industry 4.0 to collect data to make data-driven decisions. OLTP traditionally consists of operational databases and applications used to be able to execute a business process (Reddy et al., 2010), while IoT refers to a network of devices that exchange data. A common OLTP used to process plan and actual workload is System, Application, and Products in Data Processing (SAP), while a common IoT platform to collect sensor data from turbines is Supervisory Control and Data Acquisition (SCADA). SAP is generally used to capture human inputs such as technician hours, inventory movements, and work order. SCADA on the other hand generally captures automated machine inputs such as temperatures, windspeed, and energy production. Features collected from both SAP and SCADA can be used for workload forecasting, as is the case in the experiments conducted for this thesis.

1.5 Contributions to Literature

Having an accurate workload forecast is important for workload capacity planning, since the goal of capacity planning is to optimally allocate resources to current and future demand

requirements. Capacity planning alongside workload forecasting determines employee headcount, backlog levels, overtime workload, and scheduling requirements. Increasing the accuracy of workload forecasting leads to better planning since there is less uncertainty about future workload demands. Workload forecasting can be used to anticipate short-term and long-term workload requirements. Short-term workload capacity planning consists of scheduling workload to maintain daily operations and is often conducted on a weekly basis. Long-term workload capacity planning assists in setting the budget of an organization by providing the planned employee headcount. Both short-term and long-term capacity planning are critical to successfully implementing a maintenance strategy to minimize downtime and operational costs.

Workload scheduling forecasting supports workload scheduling planning that addresses questions such as:

- How many employees do we need to complete the demanded workload for the next few weeks?
- Are overtime hours required to complete the anticipated workload for next week?
- Is it necessary to dispatch technicians from one windfarms to support the workload requirements of another windfarm?

Workload budget forecasting supports workload budget planning to answer questions such as:

- What are acceptable backlog levels?
- How many employees should be employed for next year?
- To what extent should external sub-contractors be used to fulfill workload requirements?

There are a growing number of studies in recent years that show that machine learning (ML) outperforms statistical benchmarks. The thesis evaluates whether Light Gradient Boosting Machines (LightGBM), K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and feedforward neural networks (NN) consistently outperform simple statistical benchmarks for short-term and long-term workload forecasting. One goal of the study is to evaluate whether ML outperforms simple statistical benchmarks such as seasonal naive and rolling averages. The dataset for the short-term forecast consists of weekly data, while the dataset for the long-term forecast consists of monthly data. For the short-term forecast, a one-step ahead forecast is performed, while for the long-term forecasts, a 12 step-ahead forecast is performed. The dependent variable for the short-term and long-term forecasts are technician workload hours booked to 316 Canadian wind turbines. Workload hours are the duration of time that technicians spend on performing corrective and preventive maintenance. The short-term forecast consists of forecasting workload for the following week, while the long-term forecasts consist of forecasting workload for the following 12 months.

The thesis also assesses whether there are reconciliation approaches that offer superior accuracy at various cross-sectional aggregation levels in the context of workload forecasting. Forecasting reconciliation methods are used on time-series that exhibit a hierarchical structure to form coherent forecasts. Hierarchical forecasting structures occur when time-series are organized

in multiple levels in a hierarchy, where disaggregated time-series can only belong to the parent of one time-series. Reconciliation methods ensure that taking the sum of the child time-series forecasts equals the forecast produced by the parent time-series. Identifying reconciliation methods that offer superior accuracy has the potential to reduce workload demand uncertainty while keeping the forecasts coherent.

The dataset used for the short-term and long-term forecasts contains time-series for 316 wind turbines operating in Canada. Each of the time-series have been aggregated to its respective windfarm. In total, the dataset contains 10 windfarms, with the biggest windfarm consisting of 50 turbines, and the smallest windfarm consisting of 22 turbines. Each of the windfarms is associated with one and only one platform. The platform that the windfarm is associated with is based on the turbine model. In the dataset there is a total of 3 platforms, with platform B containing the most turbines at 176. The highest level of the hierarchy consists of the total workload per period for all the turbines. **Figure 1** provides a visual representation of the relationship between 3 levels of cross-sectional aggregation. In total, there are 14 time-series that were forecasted (10 windfarms, 3 platforms, 1 fleet).

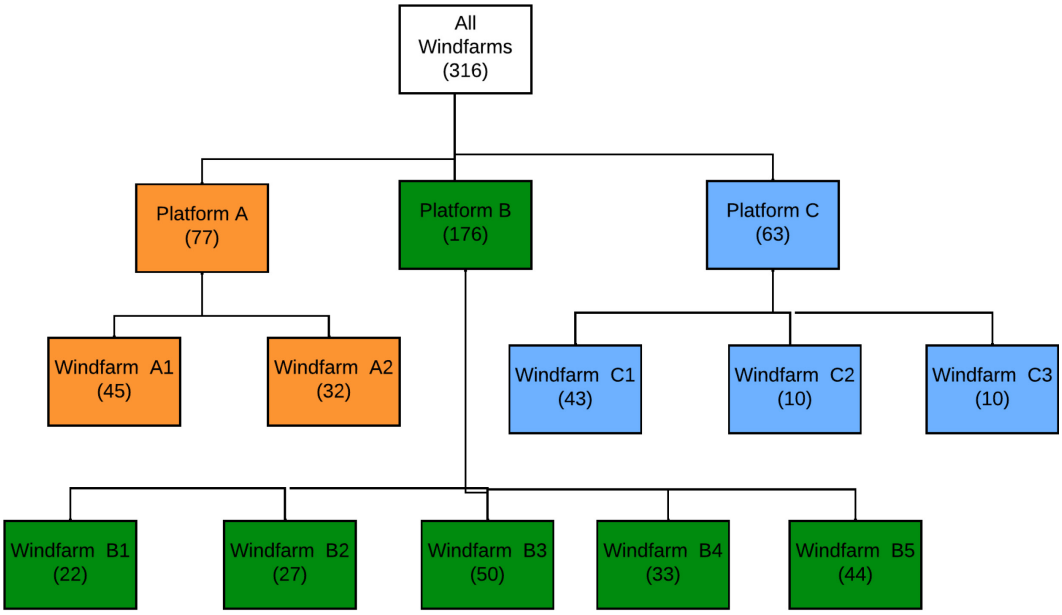


Figure 1: *Windfarm Hierarchical Structure*

While a growing body of research has shown that ML models can outperform traditional statistical methods for time-series forecasting, most existing studies focus on domains such as retail sales and energy demand. In contrast, industrial maintenance workload forecasting with ML remains largely unexplored. Moreover, hierarchical reconciliation methods have emerged as an approach for improving forecasting accuracy while keeping the forecasts coherent between aggregation levels. There has been little application of hierarchal reconciliation methods on industrial workload, especially within the renewable energy industry.

This thesis contributes to literature by:

- Evaluating LightGBM, KNN, SVM, and NN performance against simple statistical benchmarks to assess whether ML offers superior accuracy in the context of workload forecasting.
- Identifying which hierarchical reconciliation approach offers the highest accuracy at various cross-sectional aggregation levels in the context of workload forecasting when using LightGBM models.

1.6 Thesis Structure

The thesis begins by introducing workload modeling, workload forecasting, and capacity planning. The research question and contribution to literature is also examined in the introduction. The second section of the thesis consists of a literature review. The literature review begins by examining the results of various forecasting competitions that have been performed in the past. Forecasting principles and workload forecasting studies are then examined to assist in identifying research gaps. The third section of the thesis consists of defining the methodology used in the experiments to address the research questions. The methodology examines procedures related to forecasting pre-processing, processing, and post-processing. The fourth section of the thesis presents the results of the experiments, while the fifth section discusses the results and associated implications. The thesis concludes by a) reviewing the contributions that the results of experiments bring to literature, b) identifying limitations of experiments, c) and discussing future research directions.

2. Literature Review

The literature review examines the methodology and results of commonly referenced forecasting competitions. Forecasting principles are then examined to assist in establishing a framework to compare results between forecasting competitions and studies. Workload studies are also examined with the intention of identifying research opportunities in workload hierarchical forecasting.

2.1 Forecasting Competitions

The Makridakis (M) and Kaggle competitions are assessed to assist in identifying design attributes that can contribute to existing forecasting literature.

2.1.1 M Forecasting Competitions

The M- Competitions are open challenges designed to evaluate the performance of different forecasting methods. Initiated by Sypros Makridakis, the 6 competitions held between 1982 to 2024 significantly influenced the field of forecasting by providing empirical evidence on the performance of different forecasting techniques.

M1-Competition (1982)

One-thousand-and-one real-life time-series covering a wide range of data types (e.g. micro, macro, demographic) and data frequency (e.g. monthly, quarterly, yearly) were forecasted during the first competition (Makridakis et al., 1982). Various models such as naive methods, exponential smoothing, and regression were used in the competition. The competition consisted of 24 participants who used simple and sophisticated methods to forecast 6 to 18 time horizons for each

of the 1,001 time-series. Among the performance metrics used were mean absolute percent error (MAPE) and mean squared error (MSE). The results of the M1-Competition contradicted theoretical expectations at the time, which was that complex models should outperform simple models (Hyndman, 2020). Moreover, the study found that model performance is heavily reliant on the performance metric used. The implication of the study is that there is no "best" forecasting method for all time-series. The forecasting techniques that should be used depends on the specific characteristics of the time-series (e.g. data frequency, data type).

M2-Competition (1987 to 1988)

The second competition was held to determine whether adjusting quantitative forecasts based on judgment improves forecasting accuracy (Makridakis et al., 1993). Five participants used similar forecasting techniques to the M1-Competition, however participants had the opportunity to use judgment to adjust the quantitative forecast based on external information (e.g. expectations of market decline). Twenty-nine time-series from macroeconomics and specific companies were used. Exponential smoothing methods, particularly dampen and single smoothing methods provided the most accurate forecasts overall. When comparing purely quantitative forecasting methods to quantitative forecasting methods adjusted by judgment, there was no significant difference in forecasting performance. Overall, the findings of the M2-Competitions are similar to the findings of the M1-Competition. The M2-Competition adds to the findings of the M1-Competition by contradicting conventional wisdom at the time that judgmental information is an effective method to improving quantitative forecasting.

M3-Competition (2000)

The objective of the 3rd competition was to evaluate and compare the accuracy of 24 forecasting methods for 3,0003 time-series, providing insight on the relative effectiveness of different forecasting techniques (Makridakis & Hibon, 2000). The M3-Competition includes many of the models that were evaluated during the M1/M2-Competitions, while also including newer models such as neural networks. The forecasting techniques were divided into six categories which were 1) naive/simple methods 2) exponential smoothing 3) autoregressive integrated moving average (ARIMA) 4) decomposition method 5) expert systems & 6) neural networks. The study found that combining results of forecasting methods (e.g. simple average of selected models) consistently improved performance accuracy. It also found that newly introduced Theta model, which was a simple model demonstrated robust performance across most categories and time horizons. The overall results aligned with earlier competitions, emphasizing simpler approaches often outperform complex models.

M4-Competition (2018)

The 4th competition aimed to generalize the findings from previous M-Competitions by evaluating various forecasting techniques against 100,000 time-series (Makridakis et al., 2018). Unlike previous competitions, the M4-Competition evaluated model performance for point forecasts and prediction intervals. Out of the 50 point forecast submissions, 18 of them also included prediction intervals. Hybrid method combining statistical and ML techniques experienced improved results over non-hybrid methods. The six pure ML methods performed poorly, with only

one outperforming naive forecasting. The top 2 methods in reducing the symmetric mean absolute percentage error, also performed well in specifying accurate prediction intervals. Contrary to the conclusion made in the previous M competitions, the authors acknowledged that complex forecasting models have the potential to considerably outperform simple ones. The authors recommend focusing on hybrid approaches that combine the strengths of both statistical and ML methods for improved forecasting accuracy.

M5-Competition (2022)

The objective of the 5th competition was to forecasts 42,840 time-series, representing unit Sales of Walmart products (Makridakis, Spiliotis, et al., 2022). The time-series consisted of a hierarchical structure of 12 aggregation levels, which included region, store, product category, and product. The time-series covered 3,049 products sold in the USA. Exogenous variables such as promotions, prices, and calendar events were included in the dataset. The weighted root means squared scale error was used to evaluate the performance of the models. The winning methods using ML significantly outperformed statistical benchmarks, with the top 5 teams achieving over 20% improved compared to the best benchmark which was exponential smoothing. Only 415 out of 5,507 teams managed to outperform the top benchmark. Exponential smoothing remained competitive, especially at lower aggregation levels. It is worth noting that the top 5 performing teams all used variations of LightGBM models. The top performing model, which achieved 22.4% improvement over the best benchmark model, used an ensemble model of 220 LightGBM.

A common characteristic among top performing models is the use of cross-learning. All top 5 models leveraged information across multiple time-series, allowing them to capture broader patterns and relationships. Many top methods also trained separate models for different aggregation levels in the hierarchy structure. For example, the winning submission trained models at the store, store-category, and store-department level. The hierarchical modeling approach helped capture both granular and aggregated trends. Feature engineering such as using rolling statistics (e.g. moving average) were also used by top performing models. The M5-Competition showed that the combination of ensemble models, feature engineering, and cross-learning have the potential of significantly improving forecasting accuracy compared to benchmarks. The complexity of the top-performing models, however, may present challenges for practical implementation in some business context.

M6-Competition (2024)

The M6 forecasting competition aimed at contributing to the debate surrounding the Efficient Market Hypothesis (EMH) by examining whether forecasting can be used to provide stakeholders with a competitive advantage in investment returns (Makridakis et al., 2024). The EMH states that asset price fully reflects all available information, making it impossible to consistently outperform the market on a risk-adjusted basis. The key objectives of the M6-Competitions were to:

1. Investigate forecasting accuracy and investment performance for 100 publicly traded assets. (50 USA stocks and 50 international ETFs).
2. Determine whether some participants can consistently outperform the market, thus challenging the EMH framework.

3. Explore the relationship between forecast accuracy and investment performance to better understand the link between prediction and market efficiency.

The forecasting task was to forecast the relative performance of fifty S&P 500 stocks and fifty international exchange-traded funds (ETFs) (Makridakis et al., 2024). The competitions lasted for 12 months and consisted of 12 rolling submissions. Adjusted closing prices of stocks & ETFs were used to calculate returns. Participants had the option to collect the features that they desired to help improve the forecast (e.g. economic metrics, social media, & accounting data). For each of the point forecasts, participants were also required to provide probabilistic interval forecasts. Moreover, the investment weight of the asset was also required. Thirty-eight out of 163 teams managed to provide forecasts superior to a simple benchmark (Makridakis et al., 2024). However, the link between forecasting accuracy and investment performance was limited overall. Teams that were able to form superior point forecasts were not necessarily in a better position to improve the investment returns of their portfolios. The study highlights the importance of using utility metrics (e.g. returns on portfolio) rather than solely focusing on forecasting accuracy (e.g. RMSE). Overall, the study was unable to show that complex models significantly outperform simple models in forecasting asset returns.

2.1.2 Kaggle Forecasting Competitions

The findings of literature review on past Kaggle forecasting competitions are examined to identify lessons that can be applied to future forecasting competitions and research. Kaggle is an open platform for data scientists, offering a collaborative environment to share predictive modeling approaches, datasets, and code. Kaggle forecasting competitions often involves solving real-world business problems with large datasets. The winner of the competition is the team that designs the best performing model given the requirements of the competition. The literature review of Kaggle competitions builds on many of the findings of the M competitions. **Table 2** provides an overview of the forecasting tasks and dataset characteristics of the 6 Kaggle competitions under examination (Bojer & Meldgaard, 2021).

A short description of the column headers is shown below for **Table 2**:

- Competition: competition name and year
- Time-Unit: aggregation level of time-series
- Forecast Unit: target variable
- #Obs.: number of observations per time-series
- #Time-series: total number of time-series
- Forecast Horizon: number of time steps in the future

Table 2: Kaggle Forecasting Competition Overview

#	*Competition*	*Time Unit*	*Forecast Unit*	*#Obs.*	*#Time-series*	*Forecast Horizon*
1st	Walmart Store Sales (2014)	Weekly	$ sales per department	143	3331	1-29
2nd	Walmart Stormy Weather (2015)	Daily	Unit sales by product & store	851-1011	255	1-7

3rd	Rossmann (2015)	Daily	$ sales by store	942	1115	1-48
4th	Wikipedia (2017)	Daily	Views by page and traffic type	970	Around 145K	12-42
5th	Corporacion Favorita (2018)	Daily	Unit sales by product and store	1684	Around 210K	1-16
6th	Recruit Restaurant (2018)	Daily	Visits by restaurant	478	821	1-39

Walmart Store Sales (2014)

The objective of the 1st competition was to forecast weekly department sales for Walmart, to optimize inventory levels and promotions (Bojer & Meldgaard, 2021). The datasets covered 81 departments and 45 stores. The best performing model used 33 months of weekly sales data to train the model. Statistical models using singular value decomposition (SVD) and seasonal-trend decomposition (STD) led to the lowest weighted mean absolute error. An ensemble model using ARIMA, exponential smoothing, seasonal naive, and linear trend after applying SVD and STD led to the lowest errors. It is worth noting though that using a single exponential smoothing model after applying SVD and STD led to similar results.

Walmart Stormy Weather (2015)

The 2nd competition consisted of forecasting the deviation in Walmart sales caused by extreme weather events (Bojer & Meldgaard, 2021). The purpose of the forecasting competition was to evaluate whether weather conditions significantly impact store-level sales and explore how models can capture this relationship. The best performing model used projection pursuit regression with only time as an input to predict baseline sales for each time-series. The baseline sales represent the expected normal sales levels for each product and store, assuming no significant impact from extreme weather events. Deviations from baseline were then modeled using L1-regularized linear regression. The goal was to model these deviations separately to isolate the impact of weather and other external variables. Storm presence indicators and interaction terms were used to capture the impact of extreme weather on sales. Complex models such as gradient-boosted decision trees, random forest, and support vector machines were also tested however, they were unable to outperform traditional statistical models.

Rossmann (2015)

The 3rd competition consisted of forecasting sales by stores using 32 months of daily data (Bojer & Meldgaard, 2021). The dataset consisted of 1,115 distinct time-series of the target variables and exogenous variables such as holiday and promotions. The best performing model used extreme gradient boosted decision tree (XGBoost), feature engineering, cross-validation, and hyperparameter tuning. Feature engineering was shown to play a critical role in increasing the performance of ML based forecasting models. Features such as the rolling average and number of days until the next holiday were among some features that improved the performance of the XGBoost model.

Wikipedia (2017)

The forecasting objective for the 4th competition was to predict daily web traffic over a forecasting horizon of 12 to 42 days (Bojer & Meldgaard, 2021). The competition aimed to identify effective methods for forecasting high-dimensional, noisy time-series with shared patterns across series. The winning model was based on recurrent neural network architecture. Cross-learning was shown to improve model performance. This approach allows the model to learn shared patterns (e.g. seasonality, trends) that are common across multiple time-series.

Corporacion Favorita (2018)

The 5th forecasting competition aimed to predict daily unit sales for a large-scale grocery retailer (Bojer & Meldgaard, 2021). The forecast horizon of the competition was 1 to 16 days. With over 210,000 time-series, the dataset was one of the largest among Kaggle forecasting competitions. The contestants had access to 55 months of historical data for 54 stores and 3,901 products. The best performing model was an ensemble model composed of light gradient-boosting machine (LightGBM) and feedforward neural network. Notably, the winner trained one model per forecast horizon rather than using a single model for all horizons. The rolling average and standard deviation improved model performance by capturing short-term trends. A common feature among top performing models is using only recent data (e.g. 1 to 5 months) to train the models. The competition highlights how feature engineering, horizon-specific modeling approaches, and ensemble modeling can be used together to outperform traditional statistical models.

Recruit Restaurant (2018)

The 6th forecasting competition forecasted daily restaurant visits for 821 restaurants (Bojer & Meldgaard, 2021). The winning team used an ensemble approach consisting of LightGBM, XGBoost, and feedforward neural network. Exogenous variables such as the number of reservations made in advance and holidays contributed to increasing forecasting performance. The competition also confirmed the results of previous studies that neural networks and gradient boosting decision trees perform well in forecasting various types of time-series. Similarly to the previous 3 Kaggle competitions discussed, ML and deep learning outperformed traditional statistical models. As the field of forecasting continues to mature, the results of forecasting experiments are progressively advocating for the use of complex ensemble models, with the use of feature engineering such as rolling statistics (e.g. rolling average of exogenous variables).

2.2 Forecasting Guidelines & Principles

The section begins by describing common mistakes that data scientists make when forecasting. Once common forecasting pitfalls are identified, forecasting design principles are examined for various forecasting competitions to identify research gaps in the literature.

2.2.1 Common Pitfalls when Forecasting

Using inadequate evaluation metrics and benchmarks are common mistakes performed by data scientists when forecasting. The error metrics that are selected should be based on the characteristics of the dataset and forecasting task. Common error metrics include mean absolute error (MAE), root mean squared error (RMSE) and mean absolute percent error (MAPE). Although these metrics are commonly used, one must use caution in interpretation these performance metrics. For instance, MAPE should not be used when the target variable is close to 0, since even small

deviations between forecast and actual values may lead to large MAPE values. Another common pitfall is that forecasting studies often fail to compare forecasting accuracy of the models under investigation against benchmarks (Hewamalage et al., 2023). Studies will often compare the performance of complex models against the performance of other complex models, even though a simple traditional model such as naive methods yield better results. For the experiments that will be conducted for the thesis, special care will be taken for selecting adequate performance metrics and benchmarks.

2.2.2 Defining Design Attributes of Forecasting Experiments

The article "Future of Forecasting Competition: Desing Attributes and Principles", identifies design attributes that must be well defined for forecasting experiments (e.g. competitions & studies) to advance forecasting theory and practices (Makridakis, Fry, et al., 2022). The article is in response to inadequate documentation and standardization of methodologies among forecasting experiments. The authors argue that forecasting experiments that properly define 10 design attributes are more likely to lead to replicable results. Moreover, using a standard framework facilitates the process of comparing the results between various forecasting experiments. Eight of the ten forecasting design attributes are discussed in the literature review. Following the examination of the existing forecasting design attributes, two new design attributes are proposed in section **2.5 Research Opportunities**.

The 1st design attribute is *scope* which consists of three elements that are *focus,* submission type, and format (Makridakis, Fry, et al., 2022). The focus of the experiment can either be generic, specific, or semi-specific. Generic competitions feature time-series from various domains (e.g. micro, macro, demographic). Specific forecasting experiments on the other hand deal with datasets in a particular domain. The value gained from specific forecasting experiments is limited to the context of domain. Semi-specific competitions involve a set of time-series that fall within a specific domain however, the application of the forecasting tasks varies. Examples of this would be forecasting energy consumption and production. Although these two forecasting tasks are within the energy sector, benchmark models for demand forecasting may be quite different from benchmark models for supply forecasting. The *scope* of the forecast also deals with the submission type that categorizes the forecasting task as either qualitative or quantitative. The 1st design attributes also consider the format of the submission. The format of the submission consists of determining whether the forecasting task is to produce point forecasts, uncertainty estimates, and/or quantify the utility of forecasts.

The *scope* of the experiments can be linked to the reproducibility, replicability, and generalizability of forecasting studies. Reproducibility is the ability to reproduce the results using the same methods and data. Reproducibility in forecasting research is often inadequate due to incomplete documentation of methods and assumptions, which hinders the ability to validate findings (Makridakis, Fry, et al., 2022). Replicability builds on reproducibility by validating findings on different datasets while maintaining the same methods. In other words, replicability is the ability to achieve consistent results when the study is repeated with different datasets, while applying the same methods. Generalizability is the extent to which findings from a study can be applied to a broader context. The generalizability of forecasting experiments depends on its replicability and replicability is dependent on reproducibility. It is therefore important that adequate

measures are taken to ensure the reproducibility of forecasting experiments. **Table 3** provides tasks that can be performed to help ensure reproducibility of forecasting experiments.

Table 3: Factors Contributing to Reproducible Forecasting	
Factors	*Description*
Comprehensive Documentation	Provide detailed description of all methods, algorithms, and parameter setting. This includes pre-processing (e.g. removal of outliers) and post-processing steps (e.g. undue transformations to target variable).
Open Data & Code Sharing	Make the dataset(s) and code publicly available on a platform such as GitHub.
Detail Optimization Algorithms	Specify the optimization algorithm, stopping criteria, and initialization parameters used.

The 2nd design attributes being *data structures* examines the degree to which data is interrelated (Makridakis, Fry, et al., 2022). One element of *data structures* examines whether the forecasting task uses explanatory variables. The use of explanatory variable(s) in forecasting experiments can address questions such as, can predictive performance be improved by incorporating independent variables? The design attribute *data structure* also consists of assessing whether the time-series in the dataset form hierarchical structures. An example of a dataset where time-series form hierarchical structures, is a dataset that contains sales by store, province, and country. The relationship between these hierarchical structures can potentially be exploited to improved forecasting accuracy.

Data granularity and *data availability* are the 3rd and 4th design attributes (Makridakis, Fry, et al., 2022). *Data granularity* refers to the most disaggregated level at which data is available. There has been a growing trend for forecasting experiments to use higher frequency data such as daily and hourly data, however high frequency data is largely limited to operational forecasting and has not shown to perform well in strategic forecasting. *Data availability* refers to the amount of information obtained to produce the requested forecast. This includes the number of time-series made available to the forecasters and the number of historical values included for each of the time-series. When dealing with seasonal data, generally a minimum of three seasonal periods is required to capture the seasonal component of a time-series.

The *forecasting horizon* is another important design attribute when undertaking forecasting experiments (Makridakis, Fry, et al., 2022). It refers to the length of time into the future the predictions are to be made. The number of evaluations rounds the forecasting competition undertakes is referred to as the *evaluation setup* and is the 6th design attribute (Makridakis, Fry, et al., 2022). An *experimental setup* where participants provide forecasts for a single time window, with no feedback on their performance is known as a single-origin evaluation. A rolling-origin evaluation on the other hand evaluates forecasts across multiple rounds.

Selecting the correct *performance measurements* for the forecasting task is the 7th design attribute (Makridakis, Fry, et al., 2022). Ideally performance measures should measure the uncertainty, utility, and costs associated with the forecasting models under consideration. Metrics that measure uncertainty are error metrics such as root mean squared error (RMSE) and symmetric mean absolute percentage error (sMAPE). Examples of metrics that measure the costs associated

with forecasting models are model run time and hardware utilization requirements. As for utility metrics, they should quantify the value that improved forecasts generate. For instance, model A decreased RMSE by 10%, which saves the company $15,000 a year. The last design attribute is *benchmarks* (Makridakis, Fry, et al., 2022). *Benchmarks* are standards of comparison used for assessing the performance of improvement. Typically *benchmarks* consist of forecasting methods that perform well in previous experiments and are considered standard approaches for the forecasting task at hand. An example of a benchmark is the random walk for S&P 500. Incorporating the 8 design attributes into forecasting methodology facilitates the advancement of knowledge by providing a standard framework to compare results of various forecasting experiments.

2.2.3 Mapping Design Attributes of Past Competitions

Except for the M5 and M6 competitions, the M competitions focus on generic forecasting problems, covering a broad range of time-series from different domains. Kaggle forecasting competitions on the other hand, tend to target particular industries and specific business applications. As shown in **Table 4** only two of the ten forecasting competition examined evaluate performance based on uncertainty estimates. This highlights a gap in the adoption of probabilistic forecasting approaches, which are critical for decision-making in dynamic environments. While point forecast provide a single best estimate, they fail to capture the range of possible outcomes. The gap suggests an opportunity for future forecasting research and competitions to place greater emphasis on producing prediction intervals to improve decision-making.

Table 4: Scope – Mapping Design Attribute of Past Competitions

	1. Scope		
Competition (year)	*1.a Focus*	*1.b Submission Type*	*1.c Format*
M1-Competition (1982)	generic	quantitative	point forecast
M2-Competition (1993)	generic	quantitative/qualitative	point forecast
M3-Competition (2000)	generic	quantitative	point forecast
M4-Competition (2018)	generic	quantitative	point forecast /uncertainty estimate
M5-Competition (2020)	specific	quantitative	point forecast /uncertainty estimate
Walmart Store Sales Forecast (2014)	specific	quantitative	point forecasts
Rossmann Store Sales (2015)	specific	quantitative	point forecasts
Wikipedia Web Traffic (2017)	specific	quantitative	point forecasts
Corporate Favorita Grocery Sales (2019)	specific	quantitative	point forecasts
Recruit Restaurant Visitor (2018)	specific	quantitative	point forecasts

Table 5 shows whether exogenous variables and hierarchical time-series are present in the forecasting task. Hierarchical time-series was absent in earlier M competitions but was present for the M5-Competition. Among Kaggle competitions, hierarchical time-series was present in many cases, particularly in sales forecasting datasets. The inclusion of hierarchies is particularly

beneficial in business applications, as decisions often need to be made at different levels of granularity. For example, in the M5-Competition and Walmart Store Sales Forecast, models needed to predict sales at various hierarchical levels. In contrast, earlier M Competitions did not explicitly account for hierarchies, limiting the models' ability to leverage structured dependencies within the data. It is also worth mentioning that exogenous variables are increasingly being applied across forecasting competitions.

Table 5: Data Structures – Mapping Design Attribute of Past Competitions

Competition (year)	*2. Data Structures*	
	2.a Hierarchies	*2.b Exogenous Variables*
M1-Competition (1982)	false	false
M2-Competition (1993)	false	true
M3-Competition (2000)	false	false
M4-Competition (2018)	false	false
M5-Competition (2020)	true	true
Walmart Store Sales Forecast (2014)	true	true
Rossmann Store Sales (2015)	true	true
Wikipedia Web Traffic (2017)	true	false
Corporate Favorita Grocery Sales (2019)	true	true
Recruit Restaurant Visitor (2018)	true	true

As shown in **Table 6**, forecasting competitions differ significantly in terms of the granularity of data and the availability of historical observations. Data granularity refers to the level of detail in the cross-sectional and temporal dimensions of the dataset. Earlier M competitions primarily focused on highly aggregated temporal granularity, often using monthly, or quarterly observations. In contrast, later competitions such as M4 and M5 shifted towards more granular datasets, incorporating daily and weekly frequencies to better capture short-term variations and trends. A finer level of detail allows models to detect localized trends and seasonality effects that may not be visible in aggregated data. However, higher granularity also introduces challenges such as increased noise and data sparsity.

Data availability measured by the number of time-series and the number of observations per series also plays a crucial role in forecasting performance. Earlier M competitions featured relatively small datasets. This constrained the ability to apply ML methods, favoring statistical models requiring less observations in the training set. In contrast, later competitions such as M4 and M5 significantly expanded the number of time-series, reaching up to 100,000 series in M4. Kaggle competitions tend to feature large-scale datasets, with competitions such as Wikipedia Web Traffic and Corporate Favorita Grocery Sales containing hundreds of thousands of observations.

Table 6: Data Granularity & Availability – Mapping Design Attribute of Past Competitions				
	3. Data Granularity		*4. Data Availability*	
Competition (year)	*3.a Cross-Sectional*	*3.b Temporal*	*4.a # of Series*	*4.b Obs. per Time-series*
M1-Competition (1982)	country/company	monthly, quarterly, yearly	1,001	monthly (66), quarterly (40), yearly (15)
M2-Competition (1993)	country/company	monthly, quarterly	29	monthly (82), quarterly (167)
M3-Competition (200)	country/company	monthly, quarterly, yearly	3,003	monthly (115), quarterly (44), yearly (19)
M4-Competition (2018)	country/company	daily, weekly, monthly, etc.	100,000	daily (2,490), weekly (943), monthly (202)
M5-Competition (2020)	store/product	daily	30,490	1,782
Walmart Store Sales Forecast (2014)	store/department	weekly	143	3,331
Rossmann Store Sales (2015)	store	daily	942	1,115
Wikipedia Web Traffic (2017)	page/traffic type	daily	970	around 145K
Corporate Favorita Grocery Sales (2019)	store/product	daily	1,684	around 210K
Recruit Restaurant Visitor (2018)	restaurant	daily	478	821

As shown in **Table 7**, forecasting competitions also vary in the length of their forecast horizons and in the way that models are evaluated. Earlier M Competitions primarily used longer-term horizons (monthly, quarterly, and yearly forecasts), while later competitions such as M4 and M5 shifted towards shorter-term horizons, with daily and weekly forecasts becoming more prevalent. Kaggle competitions tend to favor short to medium term forecasting horizons, with most tasks requiring predictions for only a few weeks or months ahead. This aligns with the practical needs of operational decision-making that are based on short to medium term forecasts. However, longer forecasting horizons remain important in strategic planning scenarios, such as market expansion decisions.

Evaluation setup is another important aspect of forecasting competitions. Most competitions, including M1, M3, M4, and M5, rely on a single evaluation round, meaning that models are tested on a fixed holdout set of unseen data. However, competitions such as M2 and M6 introduced multiple evaluation rounds, allowing participants to iteratively refine their models based on updated information. Live evaluation, where forecasts are assessed in real time as new data becomes available, remains rare but provides a more realistic representation of real-world forecasting challenges. These findings highlight the need to carefully consider forecasting horizon and evaluation methodology when designing forecasting experiments.

While short-term forecasts are essential for many business applications, long-term forecasting remains crucial for strategic decision-making. Additionally, incorporating multiple evaluation rounds or live forecasting can enhance the practical utility of forecasting models, ensuring that they remain robust under changing conditions.

Table 7: Forecasting Horizon & Evaluation Step – Mapping Design Attribute of Past Competitions

	5. Forecasting Horizon	*5. Evaluation Setup*	
Competition (year)		*5.a Live*	*5.b Rounds*
M1-Competition (1982)	monthly (1-18), quarterly (1-8), yearly (1-6)	false	1
M2-Competition (1993)	monthly (1-15), quarterly (1-5)	true	2
M3-Competition (2000)	monthly (1-18), quarterly (1-8), yearly (1-6)	false	1
M4-Competition (2018)	daily (1-14), weekly (1-13), monthly (1-18)	false	1
M5-Competition (2020)	daily (1-28)	false	1
Walmart Store Sales Forecast (2014)	weekly (1-29)	false	1
Rossmann Store Sales (2015)	daily 1-48	false	1
Wikipedia Web Traffic (2017)	daily 12-42	true	1
Corporate Favorita Grocery Sales (2019)	daily 1-16	false	1
Recruit Restaurant Visitor (2018)	daily 1-39	false	1

As shown in **Table 8**, forecasting competitions differ significantly in how performance is measured and benchmarks are used. Performance measures vary across competitions, with earlier M competitions primarily using traditional error metrics such as MAPE and MSE, while later competitions introduced more sophisticated evaluation criteria such as overall weighted average (OWA) and weighted root mean squared scaled error (WRMSSE) to account for scaling and distributional differences. M competitions mainly compared forecasting models against naive, exponential smoothing, and ARIMA; whereas Kaggle competitions tend to use overly simplistic benchmarks such as median and mean.

Table 8: Performance Measures, Benchmarks, & Learning – Mapping Design Attribute of Past Competitions		
Competition (year)	*7. Performance Measure*	*8. Benchmarks*
M1-Competition (1982)	MAPE, MSE, AR, MdAPE, PB	Naive, ES
M2-Competition (1993)	MAPE	Naive, ES, ARIMA
M3-Competition (2000)	sMAPE	Naive, ES, ARIMA
M4-Competition (2018)	OWA and MSIS	Naive, ES, ARIMA
M5-Competition (2020)	WRMSSE/WSPL	Naive, ES, ARIMA
Walmart Store Sales Forecast (2014)	WMAE	median
Rossmann Store Sales (2015)	RMSPE	median
Wikipedia Web Traffic (2017)	sMAPE	median
Corporate Favorita Grocery Sales (2019)	NWRMSLE	mean
Recruit Restaurant Visitor (2018)	RMSLE	median
Performance Metric Description		
1. MAPE: mean absolute percent error 2. MSE: mean squared error 3. AR: average ranking 4. PB: percentage best 5. MdAPE: median absolute percent error 6. sMAPE: symmetric mean absolute percent error 7. OWA: overall weighted average 8. MISIS: mean scaled interval score 9. WRMSSE: weighted root mean squared scaled error		10. WSPL: weighted scaled pinball loss 11. WMAE: weighted mean absolute error 12. RMSPE: root mean square percentage error 13. RMSLE: root mean squared logarithmic error 14. NWRMSLE: normalized weighted root mean squared logarithmic error

2.3 Workload Forecasting Studies

The studies that are discussed in this section are presented in chronological order to facilitate the discussion regarding the evolution of workload forecasting. The first study examined workload forecasting for outdoor weather equipment in Alaska. The study found that isolating repetitive patterns in the data can be used to form indicator variables that improve model performance (Salman, 2004). The researchers used spectral analysis techniques to decompose the detrended series into sine and cosine waves with different amplitudes and frequencies. The most dominant signals were selected to form cyclical indicator variables used for casual and statistical models. The casual model consisted of multiple linear regression using ordinary least squares (OLS), while the statistical model consisted of SARIMX. The seasonal ARIMA model with cyclical indicator variables provided the best overall goodness-of-fit and forecasting accuracy (Salman, 2004). The training set consisted of 90 monthly observations, while the testing set consisted of the last five observations. The MAPE of the five withheld observations was 5.9%.

In the second study that is examined, the authors use leading indicators in combination with an additive technique to improve existing workload forecasting, for a company operating within the manufacturing sector of the refractory industry (Haberleitner et al., 2010). Leading indicators

provide early warnings of upcoming changes and have been shown to improve short-to-mid-term forecasts when used with the additive technique. The study varies from prior research which focus primarily on using lagging indicators to forecast workload. Leading indicators are features that occur before the event of interest occurs, while lagging indicators are features that occur after the event of interest has occurred. The study used historical actual workload hours as the lagging indicator and anticipated planned hours as leading indicator.

The dataset that was used in the study is based on actual and plan workload collected by an enterprise resource planning system (Haberleitner et al., 2010). The planned hours of work orders are based on incoming customer demand thus making the feature a leading indicator of actual workload. The additive technique is used to merge a forecast of not yet known workload to the already known workload. Known workload is based on orders that have already been created in the enterprise resource planning system, which have planned hours associated with them. Unknown workload are orders that are expected based on historical trends. The additive technique starts by identifying the already booked orders for future periods. It then uses forecasting techniques to predict the volume of unknown orders that are likely to be received in the future.

An advantage associated with using the additive technique is that it is effective in predicting demand shocks. During sudden changes in market conditions, the forecast can quickly adapt because a significant portion of the forecast is based on already booked orders. The study shows that the additive algorithm significantly outperformed short term forecasts that rely only on historical data (Haberleitner et al., 2010). A limitation of the additive technique is that as the percentage of booked orders decreases, the performance of the forecast decreases relative to conventional forecasting methods. The implication is that as the time horizon increases, the superior performance that the additive technique has over conventional forecasting methods decreases. The results of the study show that using leading indicators can be used in improving demand planning by significantly increasing forecasting performance.

The study by Van Gils et al. (2017) investigates the use of time-series forecasting in zone order picking system to predict workload (Van Gils et al., 2017). The purpose of the study is to explore forecasting methods to improve workforce planning, reduce labor costs, and maintain high service levels. The research compares top-down and bottom-up hierarchial forecasting approaches to evaluate their performance in predicting aggregated and disaggregated demand. The dataset consists of historical daily order data from a Belgian warehouse specializing in automotive spare parts. The warehouse is divided into seven zones (A-G), with significant variability in order volume across zones. Strong weekly seasonal patterns are observed, with peaks on Mondays and Thursdays. Some zones exhibit clear season cycles, while others show more erratic demand. The dataset spans two years, with 2013 data used to train the model, and 2014 orders used to evaluate the model. The study evaluates 12 time-series forecasting models, which included naive, exponential smoothing, SARIMA, ARIMAX, and composite models that combine outputs of multiple models (Van Gils et al., 2017). A composite model consisting of ARIMA, SARIMA, and exponential smoothing obtained the lowest MASE. Top-down and bottom-up approaches were used when evaluating the 12 models.

Both top-down and bottom-up approaches yielded accurate forecasts for total workload (Van Gils et al., 2017). Composite forecasts which combined outputs from multiple models outperformed other methods. Bottom-up approaches significantly outperformed the top-down approaches in predicting zone-level demand for most zones. By modeling each zone's unique demand patterns directly, the bottom-up approach accounts for variability across zones more effectively than the proportional allocation used in the top-down methods. The study suggests that while both approaches are effective for aggregated demand forecasting, the bottom-up approach is superior to disaggregated zone-level predictions due to its ability to model individual zone patterns.

A study published in the "Journal of Quality in Maintenance Engineering" addresses the challenges of accurately estimating maintenance work hours in oil and gas industrial setting (Khalid et al., 2021). Traditional methods for estimating work hours rely heavily on expert judgment and outdated techniques, which lead to prolonged downtime and increased operational cost. The purpose of the study is to propose a ML based methodology to improve the accuracy of work hour predictions for preventive maintenance tasks. The data collected for the study consists of 804 work orders between 2011 to 2017 for an oil and gas company. The scope of the study is limited to forecasting maintenance work on two types of equipment which are emergency shutdown valves and blow-down valves.

Initially, 54 attributes related to order and equipment characteristics were considered as potential exogenous variables (Khalid et al., 2021). One-hot encoding was used to create features from categorical variables. In total 9 algorithms were tested which included Random Forest, Gradient Boosting, Support Vector Regression, AdaBoost, and Bayesian Ridge Regression. Hyper-parameter tuning and 3-fold cross-validation were applied to the models that were evaluated. Prior to fitting the models, principal component analysis was used to reduce the number of features. Normalization was also used as a preprocessing step to scale the features. When comparing the various models, random forests emerged as the best-performing algorithm. Each of the 9 models were evaluated in terms of mean absolute error (MAE) and mean squared error (MSE).

When comparing forecasts based on random forest to judgment, the accuracy of median-term workload increased by 83%.(Khalid et al., 2021). Median-term workload are orders with workload between 20 to 60 hours. It is worth noting that for short-term workload, judgmental forecasting was superior to the best performing ML model. A major limitation of the study was the quality of the data. For instance, 60% of the orders had the exact same estimated work hours as the actual work hours, showing bias in the data. Obtaining a cleaner dataset or using techniques to remove the bias from the data is likely to significantly improve the results of the study. Despite its limitations, the study provides a framework for improving maintenance planning through ML techniques.

A recent study conducted by Amazon Science undertook developing workload forecasting models for 4 e-commerce warehouse facilities (Purwar & Reimherr, 2023). The goal of the study was to improve efficient planning and resource allocation by reducing future uncertainty of workload demands. The models were based on historical work orders that were categorized into planned work, unplanned work, and training. As a preprocessing step, the Augmented Dicky-Fuller (ADF) test was performed to check whether the times-series was stationary. Transformations were

applied to time-series that were found to be non-stationary. Another preprocessing step was aggregating daily data to weekly data to reduce random variation of the time-series caused by inconsistent daily logging practices. The analysts also removed anomalies from the data caused by specific events such as warehouse shutdowns.

Exponential smoothing, ARIMA, SARIMA, and SARIMAX were among the candidate models (Purwar & Reimherr, 2023). Hyper-parameter tuning was performed to find the optimal parameters for the seasonal and non-seasonal components of the ARIMA models. Optimal ARIMA parameters were selected based on Akaike Information Criterion (AIC) and model significance. A rolling window rather than an expanding window was used to train the forecasting models.

SARIMAX outperformed the other candidate models in terms of MAPE (Purwar & Reimherr, 2023). U.S. holidays were used as an exogenous variable to improve model performance. Moreover, fitting models to each warehouse and workload classification (e.g. unplanned work) significantly improved model performance. As a post-processing step, the analysts also ensured that the results of the models could not exceed the workforce capacity of the warehouse, which further improved model performance. The study shows how modern data preparation, model refinement, and post-processing methods can be used to improve planning by reducing workload uncertainty.

The study by Li et al. (2024) develops a supervised learning framework to forecast the workload of non-routine tasks in aircraft maintenance (Li et al., 2024). Non-routine maintenance significantly contributes to increased costs, workload, and operational uncertainty of airlines. Current estimation methods which rely on averaging past work hours fail to account for the variability of non-routine tasks, thus limiting capacity planning. The dataset used in the study consists of maintenance records for 30 wide-body aircraft. Numerous features associated with the aircraft and work orders were used as exogenous variables. Aircraft age, historical work hours of non-routine tasks, and historical frequency of non-routine tasks were among the most important features for predicting future workload. The methodology of the study involves a two-step prediction process using random forest models. A random forest classification model first predicts the total number of non-routine tasks. The results of the first model are then fed into a random forest regression model to predict the workload requirements. The dataset was split into 80% training and 20% testing. Task prediction accuracy was assessed using classification accuracy, whereas workload predictions were evaluated using MAE. When compared to taking historical averages, tasks prediction frequency accuracy improved by 21%, while labor hours prediction MAE decreased by 20%.

2.4 Hierarchical Forecasting

Hierarchical forecasts can be reconciled using bottom-up, top-down, middle-out, or optimal-combination approaches (Abolghasemi et al., 2019). The bottom-up approach starts from the most disaggregated level and aggregates forecasts up the time-series hierarchy. The bottom-up approach tends to provide higher accuracy at lower levels of the hierarchy, making it particularly useful for operational decisions. The bottom-up approach however is prone to capturing noise at the bottom levels of the hierarchy. Moreover, bottom-up approaches can be computationally expensive and labor intensive. In contrast, top-down reconciliation approaches generate forecasts

at an aggregate level and disaggregate forecasts down the hierarchy. One of the main limitations of the top-down approach is that the accuracy of the forecasts tends to decrease at lower levels of the hierarchy due to its reliance on disaggregation techniques.

The middle-out approach attempts to balance the strengths of the bottom-up and top-down approaches by forecasting at an intermediate level. (Abolghasemi et al., 2019). Middle-out approaches aggregate upwards and disaggregate downwards to obtain reconciled forecasts. Middle-out approaches can provide a compromise between strategic and operational accuracy. The last hierarchical reconciliation method is known as the optimal-combination approach. The optimal-combination approach reconciles forecasts at all levels of the hierarchy, typically by minimizing a forecast variance under certain assumptions. The optimal-combination approach is also known as the regression-based reconciliation approach. The approach is commonly cited as having superior overall accuracy over top-down, bottom-up, and middle-out approaches. A limitation of the optimal-combination approach is that it relies on assumptions about the error variance that may not hold in practice.

Silveira Gontijo and Azevedo Costa (2020) investigated the application of hierarchical time-series forecasting in the Brazilian power generation sector. The study evaluates the performance of 3 reconciliation approaches. Specifically, the bottom-up, top-down, and optimal-combination reconciliation approaches are examined to determine whether an approach is significantly superior to others. The study found that the optimal-combination reconciliation approach showed the best average performance when compared to the bottom-up or top-down approach (Silveira Gontijo & Azevedo Costa, 2020). The results reinforce the theoretical advantage that optimal-combination approaches tend to lead to superior performance when compared to bottom-up or top-down approaches.

2.5 Research Opportunities

Prior to examining the research opportunities in workload forecasting, the distinction between traditional statistics and ML should be made. Traditional statistical methods often begin with a predefined hypothesis that specifies the relationship between variables. Techniques such as linear regression, logistic regression, and ARIMA are commonly used in time-series forecasting. These methods are typically inference-focused, emphasizing (a) understanding relationships, (b) quantifying uncertainty, and (c) testing hypotheses. Simple statistical techniques such as rolling averages and exponential smoothing, while not inference-focused, are also considered part of traditional time-series forecasting. Compared to traditional statistical methods, ML approaches generally emphasize prediction accuracy over understanding the underlying mechanisms. These types of models learn patterns directly from data without predefined hypotheses. Decision trees, support vector machines, and k-nearest neighbors are examples of ML approaches.

Both the M and Kaggle competitions were examined within the framework of forecasting design attributes developed by Makridakis among others. The intention of using Makridakis's design attributes in examining past forecasting competitions is to identify missing design attributes that can be incorporated in future forecasting studies to advance the field of workload forecasting. Based on the findings of the literature review, there are two new forecasting design attributes that the author of the thesis proposes to add to Makridakis's forecasting design attribute framework.

The first new design attribute is named “ML Vs. Traditional Statistical Benchmarks”. This design attribute examines whether ML offers superior accuracy to a competitive traditional benchmark such as moving average. A common theme in the Kaggle and workload studies is that ML and traditional statistical models are studied in isolation of each other. In other words, most studies examined in the literature review fail to compare the accuracy of ML models (e.g. NN, SVM, etc..) to the accuracy of traditional statistical benchmarks (e.g. moving averages, exponential smoothing, ARIMA etc..).

Forecasting studies that fail to compare the accuracy of ML models with traditional statistical models make it difficult to evaluate the overall performance of ML models, since the accuracy of the ML models are not being evaluated against a competitive benchmark. Conversely, studies that focus only on examining traditional statistical models without considering ML models fail to examine whether ML has the potential of outperforming traditional statistical benchmarks. For forecasting studies to be able to determine whether ML models should be used for a specific forecasting task, ML models should be compared to a competitive benchmark. Exponential smoothing and rolling averages are considered competitive benchmarks in a variety of forecasting tasks such as workload forecasting.

The first three M competitions either did not explore ML models or were unable to demonstrate that ML models outperform traditional statistical models. ARIMA and exponential smoothing yielded the highest overall accuracy for the first two M competitions, while the theta model yielded the highest overall accuracy for the 3rd M competition. The 4th M competition was the first M competition that showed that ML has the potential of outperforming traditional statistical methods. The forecasting model with the highest accuracy for the M4 competition was an ensemble model composed of a recurrent neural network and exponential smoothing. As for the M5 competition, LightGBM models significantly outperformed traditional statistical benchmarks. Lastly, for the M6 competition, no forecasting model was able to consistently outperform the random walk approach.

With regards to the Kaggle competitions examined in the literature review, ML is used for all the forecasting competitions, however the competitions fail to compare the performance of ML models against traditional statistical benchmarks. The Kaggle competition uses mean or median as the benchmark, which is insufficient in determining whether ML outperforms traditional statistical models. The workload studies examined in the literature review are similar to the Kaggle forecasting studies, in that ML is not evaluated against competitive statistical benchmarks. For each of the six workload forecasting studies examined, the study either exclusively focuses on ML or traditional statistical models. The failure of past workload forecasting studies to compare ML accuracy to traditional statistical benchmarks has led to a lack of empirical evidence that ML outperform traditional statistical benchmarks. Out of the eighteen studies examined, only three of the studies compared the accuracy of ML models against competitive benchmarks. As ML models become ever more complex, there is an increasing need to determine whether the increased complexity of the model justifies the potential accuracy gains. There is a growing trend towards comparing complex ML models against other ML models without considering whether simpler approaches yield similar or improved results. The design attribute “ML Vs. Traditional Statistical Benchmark” encourages future forecasting research to compare new ML against traditional

statistical benchmarks before commenting on the utility of the ML model being considered in the study.

The second proposed design attribute named “Hierarchical Reconciliation”, encourages future studies to perform hierarchical reconciliation when there are hierarchical structures between time-series. Hierarchical reconciliation leads to coherent forecasts at various cross-sectional aggregation levels, while potentially improving forecasting accuracy. Only one of the workload forecasting studies examined in the literature review implements hierarchical reconciliation. Seeing that workload forecasting often requires forecasts at various cross-sectional aggregation levels, where the aggregations are dependent on hierarchical relationships between time-series, hierarchical reconciliation has the potential of greatly improving workload forecasting. The experiments conducted for this thesis incorporate the two newly introduced design attributes to address the gaps that exist in workload forecasting literature.

3. Methodology

The methodology of the experiments consists of three main sections which are model pre-processing, processing, and post-processing. In the model pre-processing phase, data exploration is performed to obtain an understanding of the characteristics of the datasets. During the exploration phase outliers are identified and missing values are imputed. The pre-processing phase also involves examining the correlation between variables to select features used to fit the ML models. During the pre-processing phase, ACF and PACF plots are used to guide feature engineering tasks. During the model processing phase, six simple forecasting methods are evaluated to identify a benchmark to evaluate the ML models against. It is worth noting that the six simple forecasting methods are used for each the scheduling and budget forecasts. Once a baseline model has been established, three LightGBM, KNN, SVM, and NN models are fitted for each the scheduling and budgeting forecast. Hyper-parameter tuning and time-series cross-fold validation are used to fit the models. Once the models have been trained, the models are evaluated in the post-processing phase. During the post-processing phase of the experiments, the RMSE of the fitted models are examined to evaluate the predictive performance of the models. Lastly, hierarchical reconciliation is performed on the LightGBM model.

3.1 Model Pre-Processing

In this sections, Pearson correlation coefficient, ACF, PACF, and feature engineering used in the experiments are described.

3.1.1 Pearson Correlation Coefficient

The Pearson Correlation Coefficient is used to measure the linear relationship between two variables (James et al., 2023). The correlation coefficient ranges from -1 to 1. An absolute value of 1 implies a perfect linear equation between x and y. The sign of the correlation coefficient is determined by the slope of the line. When the sign of the correlation coefficient is positive, an increase in x leads to an increase in y, while when the sign of the correlation coefficient is negative, the inverse relationship exists. When the correlation coefficient is 0, it implies that there is no linear relationship between x and y. The expanded form of the correlation coefficient is shown in **Figure 2.**

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})\,(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}\quad(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}\quad(y_i - \overline{y})^2}}$$

where:

- *n: the sample size*
- *xi: individual sample points indexed with i*
- *yi: individual sample points indexes with i*
- *$\bar{x}$: mean of x*
- *$\bar{y}$: mean of y*

Figure 2: *Pearson Correlation Coefficient Expanded Formula*

The numerator of the equation consists of the covariance between x and y. It determines how much two variables change together from their mean values. Covariance assists in determining the relationship between two variables, however it does not quantify the strength of the relationship between two variables. The denominator of the equation consists of multiplying the standard deviation of x with y. Dividing the variation explained by x and y by the total variance quantifies the strength of the linear relationship between x and y. **Figure 3** provides an alternative way of expressing the pearson correlation coefficient. Using elements of the compact form of the equation will facilitate understanding of autocorrelation function (ACF) and partial autocorrelation function (PACF).

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y}$$

where:

- *Cov(X,Y): covariance of X and Y*
- *σ_x: standard deviation of x*
- *σ_y: standard deviation of y*

Figure 3: *Pearson Correlation Coefficient Condensed Formula*

3.1.2 ACF & PACF

While correlation measures the linear relationship between two variables, autocorrelation measures the linear relationship between time-series and its lag values. Autocorrelation measures the degree to which the current value of a series depends on its past values (Lazzeri, 2020). Autocorrelation helps identify dependencies in the time-series that can be used to improve forecasting accuracy.

$$\rho_k = \frac{\mathrm{Cov}(X_t, X_{t-k})}{\sqrt{\mathrm{Var}(X_t) \cdot \mathrm{Var}(X_{t-k})}}$$

where:

- *Xt : value of time-series at time t*

- *k : the number of lags from t*
- *X_{t-k} : value of time-series at time t-k*
- *$Cov(X_t X_{t-k})$: covariance of X_t and X_{t-k}*
- *$Var(X_t)$: variance of X at t*
- *$Var(X_{t-k})$: variance of X at t-k*

Figure 4: *Autocorrelation Function (ACF) Formula*

A positive ACF at lag k represents a positive correlation between the current observation and observation at lag k, while a negative ACF indicates a negative correlation between the current observation and the observation at lag k. The decay in autocorrelation over increasing values of k may reveal trend and seasonality in the time-series. Significant ACF values at certain lags may reveal patterns that can be exploited in forecasting such as identifying the order of moving average (MA) terms. Partial autocorrelation (PACF) may be used alongside ACF to identify relationship between lag values of a time-series. PACF removes the influence of intermediate lags, which measures the direct relationship between a variable and its past values.

$$\phi_k = \frac{\mathrm{Cov}(X_t, X_{t-k} \mid X_{t-1}, X_{t-2}, \dots, X_{t-k+1})}{\sqrt{\mathrm{Var}(X_t \mid X_{t-1}, X_{t-2}, \dots, X_{t-k+1}) \cdot \mathrm{Var}(X_{t-k} \mid X_{t-1}, X_{t-2}, \dots, X_{t-k+1})}}$$

where:

- X_t: value of the time-series at t
- X_{t-k}: value of the time-series at t-k
- $\mathrm{Cov}(X_t, X_{t-k} \mid X_{t-1}, X_{t-2}, \dots, X_{t-k+1})$: conditional covariance between X_t and X_{t-k} given the values of the intermediate lags
- $\mathrm{Var}(X_t \mid X_{t-1}, X_{t-2}, \dots, X_{t-k+1})$: conditional variance of X_t given the values of the intermediate lags
- $\mathrm{Var}(X_{t-k} \mid X_{t-1}, X_{t-2}, \dots, X_{t-k+1})$: conditional variance of X_{t-k} given the values of the intermediate lags

Figure 5: *Partial Autocorrelation Function (PACF) Formula*

While autocorrelation depicts the overall correlation structure of the time-series, partial autocorrelation examines the direct relationship between the lag values of a time-series (Lazzeri, 2020). Significant PACF values indicate autoregressive terms that most capture the direct linear relationship in the lag values of the time-series.

3.1.3 Dependent & Independent Variables

The dependent variable for the short-term and long-term forecasts are technician workload hours booked to 316 Canadian wind turbines. Workload hours are the duration of time that technicians spend on performing corrective and preventive maintenance. The short-term forecast consists of forecasting workload for the following week, while the long-term forecasts consist of forecasting workload for the following 12 months. For the short-term forecast, a one-step ahead forecast is performed, while for the long-term forecasts, a 12 step-ahead forecast is performed. The correlation between candidate variables and the dependent variable are examined to identify an

exogenous variable with the potential of improving workload forecasting accuracy. **Table 9** provides a description of each of the variables that are examined during the model pre-processing phase. The exogenous variable that is selected for the ML models is based on identifying the variable that has the highest correlation to the target variable across all the windfarms.

Table 9: Candidate Exogenous Variables

Feature	*Feature Label*	*Feature Description*
F0	Availability	% of time that turbine is operational
F1	Operational	period of time that turbine is ready to produce electricity
F2	Data Acquisition	data acquisition time interval under consideration
F3	Grid Connection	duration grid connection is not available
F4	Customer	duration turbine is not operation at request of customer
F5	Disconnected	duration turbine is disconnected from SCADA
F6	Downtime	duration that turbine is down due to fault or maintenance

Feature engineering is performed for both the budgeting and scheduling forecasting tasks. For each of the ML models evaluated, three different combinations of features sets are used to fit the models. ***Table 10*** provides the feature sets used to train the ML models for the budget forecasting task, while **Appendix A** provides the feature sets used to train the ML models for the schedule forecasting task. For the budget forecasting task, feature set 1 uses lag values of the dependent variable. Both a short-term and long-term component are included in each of the three feature sets. The short-term component consists of taking the seasonal naive after applying a 3-month centered rolling average on the original workload series. The long-term component consists of taking a 3-year seasonal trailing rolling average after applying a 3-month centered rolling average on the original workload series.

Feature set 2 uses the same variables as feature set 1 but also includes the exogenous variable downtime. The transformations that are applied to the exogenous feature downtime are the same as the transformations applied to the lag values of workload for feature set 1. For feature set 3, there are a total of six variables that are used. The first four variables are the same ones as in feature set 2. The two new variables that are added in feature set 3 are the same as the ones in feature set 1, except instead of using the centered rolling averages of workload, the centered rolling standard deviation is used. The intention of adding variables based on workload standard deviation is to provide variables to the ML models that quantify the short-term and long-term volatility of series over time. **Appendix A** provides the features set used to train the ML models for the workload schedule forecasting task. The variables used to train the ML models for workload schedule forecasting are similar to the variables used to train the ML models for workload budget forecasting. The variables of the three feature sets are normalized by using min-max scaling.

Table 10: Budgeting Forecast Task Feature Sets			
Feature Set	*Feature Characteristics*	*Component*	*Variable Description*
Feature Set 1	Centrality of seasonal lag values of target variable	Short-term	Seasonal naive after applying 3-month centered rolling average to original workload series
		Long-term	Seasonal trailing 3-year rolling average after applying 3-month centered rolling average to original workload series
Feature Set 2	Centrality of seasonal lag values of target variable	Short-term	Seasonal naive after applying 3-month centered rolling average to original workload series
		Long-term	Seasonal trailing 3-year rolling average after applying 3-month centered rolling average to original workload series
	Centrality of seasonal lag values of exogenous variable	Short-term	Seasonal naive after applying 3-month centered rolling average to original downtime series
		Long-term	Seasonal trailing 3-year rolling average after applying 3-month centered rolling average to original downtime series
Feature Set 3	Centrality of seasonal lag values of target variable	Short-term	Seasonal naive after applying 3-month centered rolling average to original workload series
		Long-term	Seasonal trailing 3-year rolling average after applying 3-month centered rolling average to original workload series
	Centrality of seasonal lag values of exogenous variable	Short- term	Seasonal naive after applying 3-month centered rolling average to original downtime series
		Long-term	Seasonal trailing 3-year rolling average after applying 3-month centered rolling average to original downtime series
	volatility of seasonal lag values of target variable	Short-term	Seasonal naive after applying 3-month centered rolling standard deviation to original workload series
		Long-term	Seasonal trialing 3-year rolling average after applying 3-month centered rolling standard deviation to original workload series

3.2 Model Processing

This section discusses the forecasting models used for the monthly budget forecast and weekly schedule forecast. Simple forecasting models are compared to LightGBM, KNN, SVM, and NN to see whether ML algorithms can outperform simple statistical benchmarks.

3.2.1 Simple Statistical Forecasting Models

For the budget and schedule forecasting task, six simple forecasts are computer. The RMSE of each of the six models are evaluated by using the testing dataset. The best performing simple model in terms of RMSE is selected as the benchmark model. The RMSE of the selected benchmark model for the budget and schedule forecasting tasks are then compared to RMSE produced by the ML models.

The first simple model consists of the naive method. The naive method consists of taking the previous value available to predict the future value (Joseph, 2022). The second simple model used is the expanding average, which uses an expanding trailing rolling average to forecast workload. The third simple model consists of a lagging rolling average that uses a fixed window size of 3 periods to calculate the moving average. The periods for the budget forecasting task are months, while the periods for schedule forecasting tasks are weeks.

The fourth model consists of the seasonal naive. The seasonal naive approach takes the lag value corresponding to the same season in the previous cycle. For the 5th and 6th models, the first step is to take a 3-period centered rolling average of historical workload. The 5th model then takes the seasonal naive value of the 3-period centered rolling average. As for the 6th model, the 3-period seasonal average is taken from the 3-period centered rolling average. ACF, PACF, and domain knowledge were used to determine the window size of the rolling averages. **Table 11** provides an overview of the simple traditional models used with the intention of setting a competitive benchmark to compare against ML models.

Table 10: Simple Model Abbreviation & Description

#	*Model Abbreviation*	*Description*
M1	M1_naive	Model 1 (Naive)
M2	M2_expAvg	Model 2 (Expanding Average)
M3	M3_roll3Avg	Model 3 (3-period Lagging Rolling Average)
M4	M4_sNaive	Model 4 (Seasonal Naive)
M5	M5_roll3Avg_s1	Model 5 (3 Period Centered Rolling Average of Previous Season)
M6	M6_roll3Avg_s3	Model 6 (3 Period Centered Rolling Average of Previous 3 Seasons)

3.2.2 LightGBM Forecasting Models

LightGBM fits an ensemble of decision trees through a sequential training process, where each new tree is designed to correct the mistakes of the one before it (Monteiro et al., 2024). The process starts with training an initial decision tree. The predictions from the first tree are compared to the actual labels. These residuals are then used as the target value for the next tree. The cycle of calculating residuals and fitting new trees is repeated for a set number of iterations or until a stopping criteria is reached. At every iteration, the model seeks to minimize a loss function that

measures the difference between predicted and actual values. One of model's key strengths is its ability to efficiently handle large datasets with numerous features. Bayes hyper-parameter tuning along with 10-fold cross-validation is used when training the LightGBM models. Hyper-parameter tuning was used to identify the optimal values for the maximum number of leaves per tree and maximum tree depth per tree.

3.2.3 KNN, SVM, and NN Forecasting Models

The K-Nearest Neighbors (KNN) uses a simple algorithm that searches for the closest data points in the training set based on a distance metric (Shmueli et al., 2020). The choice of parameter K determines how many neighbors are considered. A small K value captures local patterns while a large K value captures global patterns. Small K values are prone to capturing noise which may lead to overfitting, while large K values are at risk of ignoring local patterns.

Support Vector Machines (SVM) is another commonly used ML model. The terminology used to describe how SVM function are summarized below by Raquel Rodríguez-Pérez and Jürgen Bajorath.

> A *hyperplane* is defined as a subspace with one dimension less than the N-dimensional feature space in which it is formed. In SVM modeling, the hyperplane represents a classification boundary. The *margin* of the hyperplane is the distance between two object classes in feature space separated by the hyperplane for SVM classification. *Support vectors (SVs)* represent data samples of one class that are closest to the other class and thus used to define the margin of the hyperplane. *Kernel function* is a similarity function that takes as input vectors in original feature space and calculates a modified inner product in a higher-dimensional space. The *kernel trick* refers to a strategy for generating a non-linear SVM using a kernel function instead of computing an explicit mapping of data into a higher-dimensional space. The *ε-insensitive tube* in SVR is equivalent to the margin in SVM classification and indicates the deviations that are tolerated in the prediction of numerical values. Deviations larger than ε are penalized. Support vectors in SVR correspond to data points falling outside the ε-tube (Rodríguez-Pérez & Bajorath, 2022).

In the terminology provided by Raquel Rodríguez-Pérez and Jürgen Bajorath, the goal of SVM is to separate data into different classes, while SVR aims to predict continuous values. Raquel Rodríguez-Pérez and Jürgen Bajorath define SVM and SVR based on how it was traditionally cited, however the term SVM is increasingly being used to include both classification and regression problems. The traditional term SVM that describes an algorithm which attempts to solve a classification problem is increasingly being replaced by Support Vector Classification (SVC). Moreover, SVM is increasingly being used to describe an overarching algorithm that encompasses both SVC and SVR. In this thesis, SVM is used to describe an algorithm that addresses both classification and regression problems.

Neural networks (NN) are the 4^{th} type of ML model that are used for the forecasting experiments conducted for this thesis. NN consists of an input layer, hidden layer(s), and an output layer (Shmueli et al., 2020). The input layer is where the features of the dataset enter the network. Each node in the input layers represents one predictor variable. The hidden layer consists of one or

more layers with nodes utilizing an activation function such as a rectified linear unit activation function. The output layer consists of a single layer with one or mode nodes that produce the final predictions. The strength of the relationship between the nodes is determined by the weights that are assigned through a process called backpropagation. The weights between the nodes are initialized at a random value and then iteratively updated during training to minimize error.

Bayes hyper-parameter tuning along with 10-fold cross-validation is used when training KNN, SVR, and NN models. For KNN, hyper-parameter tuning was used to determine the number of nearest neighbors. For SVR, hyper-parameter tuning was performed on the regularization parameter, kernel function type, and kernel coefficient. For NN, hyper-parameter tuning was performed on the activation function and learning rate.

3.3 Model Post-Processing

3.3.1 Model Evaluation Metrics

Mean Absolute Error (MAE), Mean Squared Error (MSE), Mean Absolute Percent Error (MAPE), and Root Mean Squared Erro (RMSE) are the most widely used metrics when evaluating model accuracy. The performance metric used for these experiments are RMSE and Normalized RMSE. **Figure 6** shows the formula used to evaluate the accuracy of the various models that are built during the experiments (Lazzeri, 2020).

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

where:

- $\hat{y}_i$: predicted value at index i
- y_i: actual value at index i
- n: total number of data points

Figure 6: *Root Mean Squared Error (RMSE) Formula*

When comparing the performance of different platforms and windfarms against each other, it is important to recognize RMSE should be normalized. The reason for this is because different platforms and windfarms do not have the same number of turbines. Dividing RMSE by the number of turbines for a particular windfarm or platform enables a better comparison of model accuracy between categories and hierarchies.

Normalized RMSE = RMSE/ Number of turbines for selected group & hierarchical level

Figure 7: *Normalized RMSE Formula*

3.3.2 Hierarchical Reconciliation Forecasting Techniques

In this section, the button-up, top-down, middle-out, and minimum trace approaches to hierarchical forecasting are described. **Table 12** provides an overview of the six reconciliation

techniques that are assessed. The Python HierarchicalForecast library developed by Nixtla is used to perform the hierarchical reconciliation (Olivares et al., 2024)..

Table 11: Reconciliation Techniques Examined	
Technique Code	*Technique Name*
BottomUp	Bottom Up
TopDownAvgProp	Top Down - Average Historical Proportions
TopDownPropAvg	Top Down - Proportions of Historical Average
MiddleOutAvgProp	Middle Out - Average Historical Proportions
MiddleOutPropAvg	Middel Out - Proportions of Historical Average
OptimalCombination-ols	Optimal Combination - Minimum Trace (OLS)
OptimalCombination-wls	Optimal Combination – Minimum Trace (WLS)

3.3.2.1 General Formulation to Hierarchical Forecasting

The various approaches to hierarchical forecasting can be generalized by the following equation (A. H. Mohamed, 2023):

$$\tilde{Y}_n(h) = SP\hat{Y}_n(h)$$

Where:

- $\tilde{Y}_n(h)$: reconciled forecast at horizon h for all n series in the hierarchy
- P: projection matrix that distributes the forecasts depending on the reconciliation approach used (e.g. top-down forecasting)
- S: summation matrix that encodes the hierarchical structure of time-series at various aggregation levels
- $\hat{Y}_n(h)$: base forecast vector at horizon h for series at different levels of the hierarchy

Figure 8: *Generalized Hierarchical Reconciliation Forecasting Formulation*

While the summation matrix remains constant across the application of various hierarchical reconciliation forecasting approaches, the projection matrix depends on the approach used. The summation matrix used in the experiments is shown in **Figure 9.** There is a row for each time-series and the number of columns is equivalent to the number of base forecasts.

$$\mathbf{S} = \begin{bmatrix} \text{All Windfarms} \\ \text{Platform A} \\ \text{Platform B} \\ \text{Platform C} \\ \text{A1} \\ \text{A2} \\ \text{B1} \\ \text{B2} \\ \text{B3} \\ \text{B4} \\ \text{B5} \\ \text{C1} \\ \text{C2} \\ \text{C3} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 9: *Application of Summation Matrix*

The procedure to calculate the reconciled forecast starts with calculating the base forecast of each of the 14 series to produce a base forecast vector. The projection matrix is then applied to the base forecast vector. The components of the projection matrix for the bottom-up and top-down approach are discussed in the following two sections. Once the projection matrix is applied to the base forecast vector, the summation matrix is then used to produce the reconciled forecast for all n series at time horizon h. In the generalized hierarchical reconciliation forecasting formulation, the summation matrix and base forecast share the same property, in that they remain fixed irrespective of the hierarchical reconciliation approach used. ***Figure 10*** shows the base forecast vector that is used in the experiments of evaluating various hierarchical forecasting reconciliation approaches.

$$\hat{\mathbf{Y}}_n(h) = \begin{bmatrix} \hat{y}_{\text{Total}} \\ \hat{y}_{\text{Platform A}} \\ \hat{y}_{\text{Platform B}} \\ \hat{y}_{\text{Platform C}} \\ \hat{y}_{A1} \\ \hat{y}_{A2} \\ \hat{y}_{B1} \\ \hat{y}_{B2} \\ \hat{y}_{B3} \\ \hat{y}_{B4} \\ \hat{y}_{B5} \\ \hat{y}_{C1} \\ \hat{y}_{C2} \\ \hat{y}_{C3} \end{bmatrix}$$

Figure 10: *Application of Base Forecast Vector*

3.3.2.2 Bottom-Up Approach

The bottom-up approach consists of taking the forecast of each of the time-series at the lowest level of aggregation and adding up the results to higher level of aggregation. In the context of the dataset that is being used, this equates to first forecasting the time-series of 10 windfarms. The forecast of the platforms would then be calculated by taking the sum of the individual windfarms, and the total will be calculated by taking the sum of the 3 platforms. An advantage of

using this method, compared to the top-down approach, is that no information is lost due to aggregation. A disadvantage of the bottom-up approach compared to more advanced approaches is that it disregards the relationships between different time-series at the same level in the hierarchy. For instance, it does not take into consideration interactions between the workload of platform A and B. When compared to other approaches, the bottom-up approach tends to provide superior forecast for the bottom levels of the hierarchy, however its performance tends to degrade for each level of aggregation. The implication to the business problem at hand is that it is expected that the bottom-up approach will perform well in forecasting workload for specific windfarms, however will perform poorly in forecasting total workload when compared to other hierarchical forecasting approaches.

Figure 11 represents the projection matrix for the bottom-up approach for hierarchical reconciliation forecasting (A. H. Mohamed, 2023).

$$P = [0_{mk\times(m-mk)} \mid I_{mk}]$$

where:

- P: projection matrix for bottom-up approach
- 0: null matrix, used to remove forecasts that are not at bottom-level of hierarchy
- I: identify matrix, used to extract forecasts at the bottom-level of hierarchy
- $_{m}$: total number of series at bottom level of hierarchy
- $_{mk}$: total number of series

Figure 11: *Projection Matrix for Bottom-Up Forecasting Reconciliation Approach*

3.3.3.3 Top-Down Approach

The top-down approach can be seen as the opposite of the bottom-up approach in that it first creates a forecast for the highest level of the hierarchy and then applies weights to propagate the forecasts down the tree. In the context of the dataset being used for the experiments, this means that a single forecast in made for the total workload, and weights are then applied to derive the platform and windfarm workload forecasts from the total workload forecast. Two common weights that used to derive the disaggregated forecast are average historical proportions and proportions of historical averages. The average historical proportions equation captures the average historical proportions of the bottom-level series over the period relative to the total aggregation. The proportions of the historical average on the other hand, captures the average historical value of the bottom-level, relative to the average value of the top-level of the hierarchy. The equations to calculate both weights are provided below (A. H. Mohamed, 2023).

$$p_j = \frac{1}{T}\sum_{t=1}^{T} \frac{y_{j,t}}{y_t}$$

where:

- t: time index, running from 1 to T where T is the period of time that has elapsed

- j: bottom-level index, running from 1 to m, where m is the number of elements at lowest level of hierarchy
- y_t: value of the total (top-level) series at time t
- $y_{j,t}$: value of bottom-level series j at time t

Figure 12: *Average Historical Proportions Formula*

$$p_j = \sum_{t=1}^{T} \frac{y_{j,t}}{T} / \sum_{t=1}^{T} \frac{y_t}{T}$$

where:

- t: time index, running from 1 to T where T is the period of time that has elapsed
- j: bottom-level index, running from 1 to m, where m is the number of elements at lowest level of hierarchy
- $\sum_{t=1}^{T} \frac{y_{j,t}}{T}$: mean of bottom-level series for j at t
- $\sum_{t=1}^{T} \frac{y_t}{T}$: mean of total series at t

Figure 13: *Proportions of the Historical Average Formula*

Figure 14 represents the projection matrix for the top-down approach for hierarchical reconciliation forecasting.

$$P = [p \mid 0_{mk \times (m-1)}]$$

where:

- P: projection matrix for bottom up-approach
- 0: null matrix, used to disregard all forecasts except the top level of hierarchy
- p: the proportion matrix used to assign weights at different levels of hierarchy
- $_m$: total number of series at bottom level of hierarchy
- $_{mk}$: total number of series

Figure 14: *Projection Matrix for Top-Down Forecasting Reconciliation Approach*

3.3.2.3 Middle-Out Approach

The middle-out approach uses both the bottom-up approach and top-down approach (A. H. Mohamed, 2023). This approach first starts by calculating the base forecast at a middle level of a hierarchy. In the context of the dataset being used, this would entail generating the forecasts for each of the 3 platforms. The approach then calculates the forecasts of higher levels of the hierarchy by using the bottom-up approach and calculates the forecasts of lower levels of the hierarchy by using the top-down approach. Consequently, by using this approach, the total workload would be calculated by using the bottom-up approach, while the workload for the windfarms would be calculated by using the top-down approach. Both the average historical proportions and proportions of historical averages methods are examined when performing the experiments for the middle-out approach.

3.3.2.4 Optimal-Combination Approach

The procedure for the optimal-combinations approach starts by generating a base forecast for each level of the hierarchy (A. H. Mohamed, 2023). Methods such as the minimum trace (MinT) or empirical risk minimization (ERM) are then used to reconcile the forecasts at each aggregation level. The thesis only focuses on examining the optimal-combination reconciliation approach using minim trance. The MinT reconciles the base forecasts by creating a regression model that attempts to minimize forecast error across all base forecasts by either using ordinary least squares (OLS) or weighted least squares (WLS). In the case of OLS, it is assumed that all base forecast errors are uncorrelated and have the same variance. MinT OLS aims to reconcile forecast while minimizing errors by setting equal weights to each of the base forecasts. Although OLS is simpler than methods such as WLS, it ignores the fact that some base forecasts may be less accurate than others. Past studies have shown MinT WLS to be superior to MinT OLS, since base forecasts with lower errors contribute more to the reconciled forecasts than base forecasts with greater error. The experiments conducted for this thesis use MinT WLS and MinT OLS for evaluating the optimal-combination reconciliation approach.

4.3 Mapping Design Attributes to Experiments

Table 13 maps the design attributes discussed in the literature review to the experiments conducted for this thesis. The scope of the study is defined as specific, with a quantitative point forecast submission. The experiments also incorporate hierarchical structures with exogenous variables. Cross-sectional hierarchies and temporal aggregation are incorporated in the experiments. In terms of data availability, the study uses 14 series consisting of 6 years of data, for each the budget and schedule forecasting task. The forecasting horizon is split between long-term (12 step-ahead for budget) and short-term (1 step-ahead for schedule). The short-term forecast predicts workload for the following week, while the long-term forecast predicts workload for the following year. The evaluation step adopts a single-round approach. Benchmarking is carefully incorporated, with seasonal rolling average selected for budgeting and rolling average selected for scheduling.

Table 12: Mapping Design Attributes to Experiments

Design Attributes	*Elements*	*Assigned Values*
1. Scope	1.a Focus	Specific
	1.b Submission Type	Quantitative
	1.c Format	Point forecast
2. Diversity	2.a Diversity	Low
3. Data Structures	3.a Hierarchies	True
	3.b Exogenous	True
4. Data Granularity	4.a Cross-Sectional	windfarm/platform/total
	4.b Temporal	weekly/monthly
5. Data Availability	5.a Number of Series	14
	5.b Observations per Tiem Series	Budget: 72 Scheduling: 312
6. Forecasting Horizon	6. Forecasting Horizon	Budget: 12 steps-ahead Scheduling: 1 step-ahead

7. Evaluation Setup	7.a Live	False
	7.b Rounds	1
8. Performance Measure	8.a Performance Measure	RMSE & Standardized RMSE
9. Benchmarks	9.a Benchmarks	Budget: seasonal rolling average Scheduling: rolling average
10. Learning	10.a Learning	True

4. Results of Experiments

This section presents the results of experiments that were conducted to address the research questions. The section begins by describing the characteristics of the dataset used for forecasting and reconciliation experiments. Details on dataset cleaning and feature selection are provided. The results of the forecasting models are then examined by comparing simple statistical benchmarks to ML models. The section concludes by examining the performance of various reconciliation methods at minimizing RMSE at 3 cross-sectional aggregation levels. The experiments are performed for both short-term workload scheduling and long-term workload budgeting.

4.1 Data Exploration & Preparation

The section discusses dataset characteristics, outliers, missing values, feature selection and feature engineering.

4.1.1 Dataset Characteristics

The dataset contains data from two sources which are Systems, Applications, and Products in Data Processing (SAP) & Supervisory Control and Data Acquisition (SCADA). The SAP extraction consists of employee hours booked to turbines for maintenance, while the SCADA extraction consists of IoT data produced by sensors installed on turbines. The technician hours booked to turbines is the dependent variable. The features extracted from SCADA were examined to see whether forecasting performance can be improved by using an exogenous variable. The dataset contains time-series for 316 turbines, that were aggregated to windfarm level, platform level, and fleet level using the hierarchical structure of the dataset. The decision was made to aggregate the lowest level of the hierarchy to the windfarm level since a) little business values is provided in being able to forecast workload at a specific turbine and b) to remove noise from the time-series.

Table 14 shows the total number of turbines installed by year for each of the windfarms. The non-aggregated daily dataset consists of 2,192 datapoint for each of the 10 windfarms. The rollup summation function was used to obtain the time-series for platform and fleet workload. In total there are 14 time-series: 10 windfarms, 3 platforms, and 1 fleet.

Table 13: Number of Turbines by Installation Year									
Windfarms	*2010*	*2011*	*2012*	*2013*	*2014*	*2015*	*2016*	*2017*	*Total*
A1							45		45
A2							32		32
B1	22								22
B2	9	18							27
B3				50					50
B4			15	18					33
B5			41	3					44
C1					42	1			43
C2					10				10
C3						4	3	3	10
Total	31	18	56	71	52	5	80	3	316

Table 15 provides details on how the dataset was partitioned for the experiments. The dataset contains data for 6 years, starting from 2019-01-01 and ending at 2024-12-31. The first 5 years are used for training which consists of 83% of the entire dataset. The last year is used as the testing partition.

Table 14: Dataset Characteristics & Partitions		
Partition	*Training*	*Testing*
Start Date	2019-01-01	2024-01-01
End Date	2023-12-31	2024-12-31
Number of Time-series	14	14
Number of Levels	3	3
Number of Daily Records	25,564	5,124
Number of Weekly Records	3,640	742
Number of Monthly Records	840	168
% of Total Dataset	83.3%	16.7%

Figure 15 shows the weekly workload for the 3 aggregation levels. **Appendix B** shows similar line-charts to **Figure 15**; however the time-series are aggregated by month instead of week.

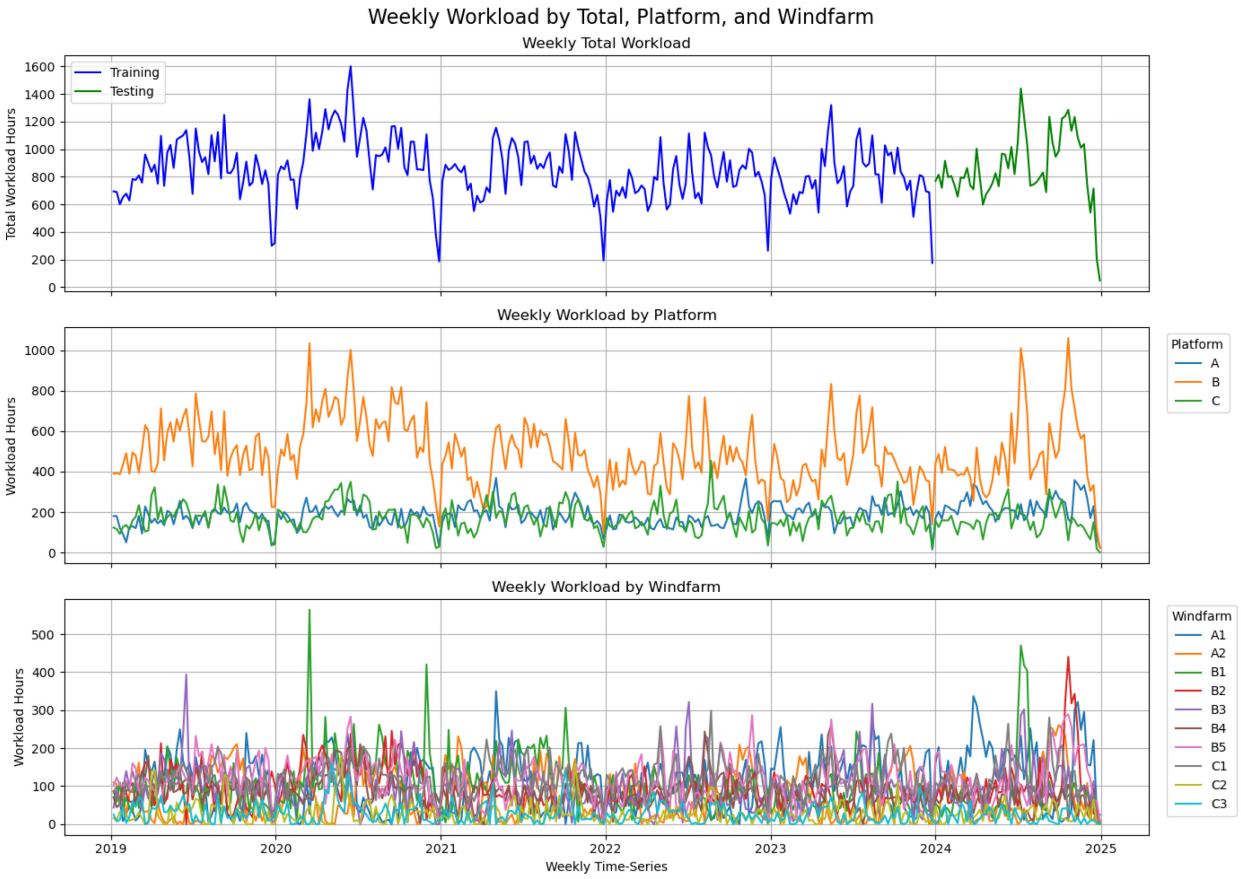


***Figure 15**: Weekly Workload by Total, Platform, and Windfarm (Line Charts)*

4.1.2 Outliers & Missing Values

The datasets from SAP and SCADA were merged into a single table. The source SCADA dataset consisted of daily time-series, while the source SAP dataset consists of events. The SAP dataset was transformed into a daily time-series before being merged with the SCADA dataset. ***Table 17*** provides details regarding:

- Record Count - number of days in time-series
- Downtime Record Count - number of records for downtime (aka. feature 6)
- Downtime % of Record Avail. – percentage of records available for downtime
- Workload % of Record Avail. – percents of records where workload value is not 0

Table 15: Dataset Record Availability Characteristics at Daily Aggregation

Windfarm	*Record Count*	*Downtime Record Count*	*Downtime % of Record Avail.*	*Workload % of Record Avail.*
A1	2,192	2,192	100.0%	4.0%
A2	2,192	2,190	99.9%	4.5%
B1	2,192	2,187	99.8%	5.9%
B2	2,192	2,189	99.9%	5.8%
B3	2,192	2,192	100.0%	10.7%

B4	2,192	2,192	100.0%	17.4%
B5	2,192	2,188	99.8%	6.6%
C1	2,192	2,176	99.3%	9.9%
C2	2,192	2,192	100.0%	46.3%
C3	2,192	2,192	100.0%	34.3%

The number of records available for downtime is shown in **Table 17** since it's the candidate feature that is used as the exogenous variable for the ML models. **Table 17** also shows that most of the values for daily workload are 0. Since maintenance is only performed on a turbine when it has faulted or is part of scheduled maintenance, it is expected that workload is 0 for most days for smaller windfarms. **Figure 16** consists of a histogram that illustrates the univariate distribution of monthly workload. Moreover, **Figure 16** contains box-plots that show the distribution of historical workload by platform and windfarm. Although the box-plots show outliers, outliers were not removed from the dataset. The outliers can be caused by corrective maintenance that leads to unusually high levels of workloads, such as replacing a major component of a turbine. Missing values for feature downtime were imputed by using the average value of the corresponding windfarm.

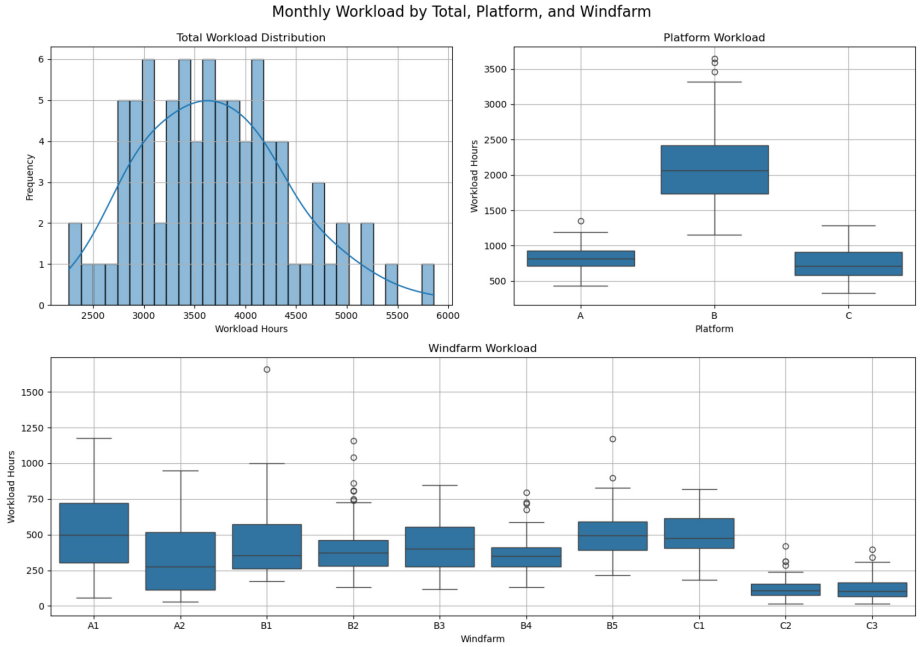


Figure 16: *Monthly Workload by Total, Platform, and Windfarm (Histograms & Boxplots)*

4.1.3 Feature Selection

Table 17 shows the correlation between workload and each of the features considered as a potential exogenous variable. When looking at the correlation between downtime and workload for each of the windfarms, one can see that downtime on average has a higher correlation than any other feature under consideration. Downtime represents the duration that the turbine is not operating due to a fault or maintenance. Turbines that have historical higher levels of downtime

typically require more corrective maintenance, thus explaining the correlation between downtime and maintenance time. Downtime is the candidate feature that is selected as the exogenous variable since it has considerably higher correlation for each of the 10 windfarms when compared to the 5 other features under consideration. **Appendix C** illustrates the relationship between monthly workload and downtime in the form of scatter-plots at various cross-sectional levels of aggregation.

Table 16: Workload Exogenous Variables Correlation Matrix by Windfarm

	F6	*F5*	*F4*	*F3*	*F2*	*F1*	*F0*
A1	0.57	0.01	-0.09	0.04	0.00	-0.09	-0.20
A2	0.60	-0.04	0.09	0.00	0.01	-0.08	-0.22
B1	0.27	-0.02		0.04	-0.01	-0.10	-0.17
B2	0.33	0.00	0.00	-0.02	-0.01	-0.14	-0.31
B3	0.56	0.09	0.09	0.09	0.00	-0.14	-0.11
B4	0.64	0.06	-0.03	0.07	0.00	-0.15	-0.28
B5	0.50	0.08	0.00	0.05	-0.01	-0.14	-0.15
C1	0.27	0.10		0.11	0.01	-0.16	-0.20
C2	0.46	0.07	0.00	0.04	0.00	-0.20	-0.24
C3	0.45	0.06		0.05	0.00	-0.22	-0.31
Average	0.46	0.04	0.01	0.05	0.00	-0.14	-0.22

4.1.4 Feature Engineering

The ACF and PACF plots were used to assist in identifying the lag values to be used with ML and benchmark models. **Figure 17** shows the ACF and PACF values for daily, weekly, and monthly workload for the 14 time-series under consideration. The time-series exhibits significant monthly and weekly seasonality, thus features capturing seasonality were used when fitting ML and benchmark models. The monthly seasonal plot in **Appendix D** corresponds to the findings of the ACF plot in **Figure 17**, in that it shows that there is high monthly seasonality in the time-series. Moreover, the PACF value of lag 1 for both weekly and monthly workload series is significant, suggesting that short-term rolling average windows of the target variables should be used as a feature for the ML models. Features derived for the ML models are based on findings from ACF, PACF, and correlation matrix, as well as domain knowledge.

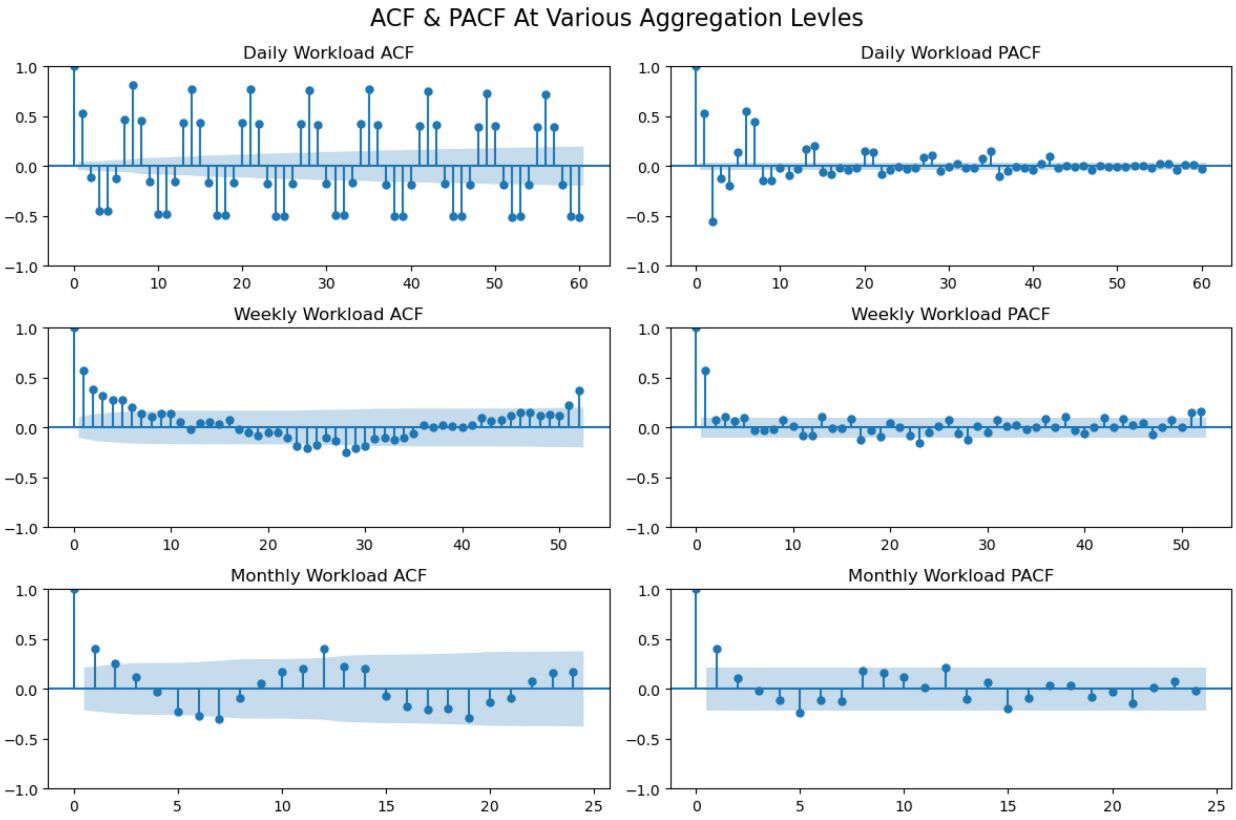


***Figure 17**: ACF & PACF At Various Aggregation Levels*

4.2 Simple Statistical Model Forecasting Results

The first step in the model processing phase of the experiments was to create six simple budget and schedule forecasts. The purpose of creating these simple forecasting models is to identify a baseline model that can be compared against the ML models. The baseline model selected for the budgeting forecast is the 3-year seasonal average after taking the 3-month centered rolling average of workload. In other words, the 3-month centered rolling average was first applied to the target variable. The 3-year seasonal average was then taken from the transformed target variable to form the budgeting forecast. The selected baseline model for the scheduling forecast is the 3-week rolling average. ***Table 11*** provides an overview for each of the six simple statistical models created for the budgeting and scheduling forecasts.

4.2.1 Monthly Budget Forecasting Results of Simple Models

Table 18 presents the standardized RMSE for the 14 time-series across the six simple statistical forecasting methods under consideration. As expected, the forecasting accuracy improves with higher levels of aggregation. For example, fleet level forecasts achieve the lowest errors across all models (ranging from 1.3 to 1.8 hours), while disaggregated windfarm level series such as B1 exhibit much higher standardized RMSE values (ranging from 14.4 to 18.5). The results confirm the stabilizing effects of aggregation, where variability at the windfarm level is smoothen when combined into larger groups. Among the models considered, model 6 exhibits on average the lowest standardized RMSE. As elaborated in section **3.2.1 Simple Statistical Forecasting Models**,

model 6 is formed by taking the 3-year seasonal average from the 3-month centered rolling average of workload. Model 6 was selected as the benchmark model for the budget forecast because as indicated by the color coding, model 6 on average outperforms the other models under consideration.

Table 17: Standardized RMSE- Simple Budget Forecast for 2024 (Monthly)

Level	*M1*	*M2*	*M3*	*M4*	*M5*	*M6*	*Turbine Count*
A	2.3	**3.3**	**3.2**	**2.7**	**2.6**	**3.1**	**77**
A1	10	8.1	8	8	7.5	7.1	45
A2	11.8	9	9	7.5	7	6.3	32
B	**5.3**	**4.1**	**4.2**	**4.3**	**4.4**	**3.8**	**176**
B1	18.5	18	17.9	14.4	16.2	16.3	22
B2	11.5	10	10.9	11.7	11.7	10.7	27
B3	5	3.2	3.4	2.2	2.5	2.5	50
B4	2.5	3.5	3	4.7	3.4	3	33
B5	6.6	5.8	6	7.2	6.7	5.5	44
C	**3.4**	**3.2**	**3.1**	**2.4**	**2.5**	**2.5**	**63**
C1	4	3.8	3.8	2.9	3.2	3.3	43
C2	7.2	4.1	4.2	6.4	3.6	3.5	10
C3	4.8	6	5.2	8.2	5.7	4.9	10
Fleet	**1.8**	**1.4**	**1.5**	**1.4**	**1.4**	**1.3**	**632**

Notes: M1 (Naive); M2 (Expanding Average); M3 (3-month Lagging Rolling Average); M4 (Seasonal Naive); M5 (3-month Centered Rolling Average of Previous Season); M6 (3-month Centered Rolling Average of Previous 3 Seasons);

Figure 18 highlights how the six models perform over the twelve month forecasting horizon. In the early months, all models produce similar cumulative RMSE, reflecting limited divergence in accuracy for short-horizon forecasts. As the forecasting horizon is extended however, difference between models become more pronounced. The naive model (M1) consistently accumulates error at a faster rate, eventually producing the highest cumulative RMSE by the end of year. In contrast, the benchmark model (M6) maintains the lowest cumulative RMSE for most of the forecasting horizons. Other rolling average models such as M3 and M5 track closely behind M6, though as the forecasting horizon increases, the divergence between the models increases. The cumulative RMSE results reinforce the findings of **Table 18** in that model 6 is superior to the other simple statistical models in terms of forecasting accuracy.

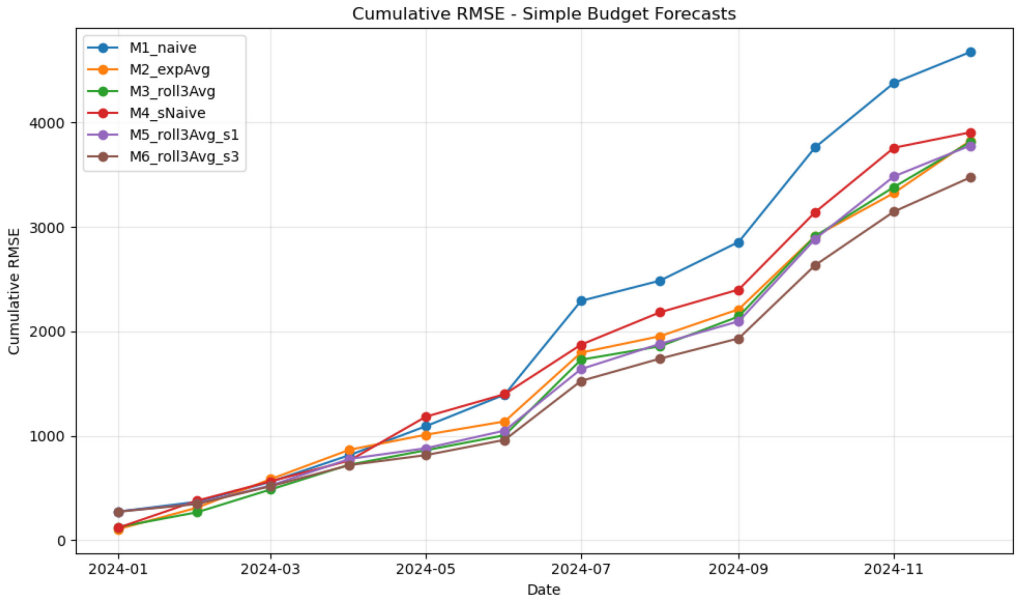


Figure 18: *Cumulative RMSE – Simple Budget Forecasting*

4.2.2 Weekly Schedule Forecasting Results of Simple Models

Table 19 presents the standardized RMSE for 14 time-series across the six simple forecasting methods under consideration for scheduling workload. As expected, the forecasting accuracy improves with higher levels of aggregation. For example, at the fleet level the standardized RMSE values are the lowest across all models. Windfarm level series such as B1 and C2 exhibit much higher standardized RMSE values (ranging from 1.6 to 4.3) when compared to fleet level standardized RMSE (ranging from 0.2 to 0.4). Among the models considered, model 3 (3 period lagging rolling average) exhibits on average the lowest standardized RMSE, thus was selected as the benchmark model for workload scheduling.

Table 18: Standardized RMSE of Simple Scheduling Forecast for 2024 (Weekly)

Level	*M1*	*M2*	*M3*	*M4*	*M5*	*M6*	*Turbine Count*
A	**0.8**	**1**	**0.5**	**0.9**	**0.9**	**0.9**	**77**
A1	1.8	2.2	1.2	2.4	2.3	1.9	45
A2	1.5	2.4	1	2.3	2.1	1.8	32
B	**0.9**	**1.1**	**0.6**	**1.2**	**1.2**	**1**	**176**
B1	3	4.3	2.2	3.6	3.4	3.7	22
B2	2.4	3.1	1.7	3.5	3.4	3.2	27
B3	1.4	1.2	0.9	1.3	1	1.1	50
B4	1.2	1.1	0.7	1.7	1.4	1.1	33
B5	1.3	1.6	0.9	2.2	2	1.6	44
C	**1**	**1**	**0.7**	**1.1**	**1**	**0.9**	**63**
C1	1.4	1.3	0.9	1.4	1.2	1.2	43
C2	2.7	1.9	1.6	2.9	2.2	2.1	10
C3	2.3	2	1.4	3.1	2.8	2.2	10

Total	**0.3**	**0.4**	**0.2**	**0.4**	**0.4**	**0.3**	**632**

Notes: M1 (Naive); M2 (Expanding Average); M3 (3-week Lagging Rolling Average); M4 (Seasonal Naive); M5 (3-week Centered Rolling Average of Previous Season); M6 (3-week Centered Rolling Average of Previous 3 Seasons);

Figure 19 presents the cumulative RMSE of the six simple statistical models used for the weekly schedule forecast across the 52 months of 2024. Note that for the budget forecast, a 12-step forecast is performed, while for the scheduling forecast only a 1-step ahead forecast is performed. At the beginning of the year, all the models perform similarly, with cumulative errors remaining close during the first quarter. However, as the forecasts approach the end of the year, the differences in cumulative error increase considerably. Model 3 (3 period lagging rolling average) exhibits a substantially lower RMSE at end of year compared to all other models under consideration, thus reinforcing the assertion that model 3 should be the benchmark model for the workload schedule forecast.

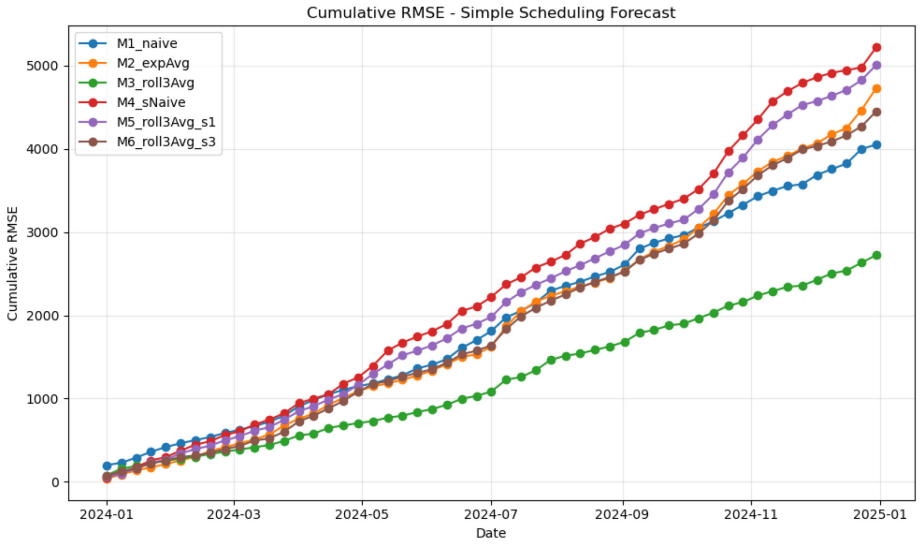


Figure 19: *Cumulative RMSE – Simple Scheduling Forecast*

4.3 LightGBM Forecasting Results

Three LightGBM models were created for each budget and schedule forecasting tasks. The models differ in the feature sets that are used to fit the ML models. **Table 10** and **Appendix A** describe the feature sets that are used for each of the LightGBM models. The 1st LightGBM model corresponds to feature set 1, the 2nd LightGBM model corresponds to feature set 2, and the 3rd LightGBM model corresponds to feature set 3. This section evaluates RMSE of the LightGBM models against the RMSE of the benchmark models. The intention of comparing LightGBM models to benchmark models is to identify whether the commonly cited LightGBM algorithm can consistently outperform simple statistical forecasting benchmarks.

4.3.1 Monthly Budget Forecasting Results of LightGBM Models

Table 20 presents the percentage difference in RMSE between the 3 LightGBM models and the statistical benchmark model for 2024 budget forecast. At the total fleet level, model 1 and

model 2 slightly improve upon the benchmark with 1.9% reduction in RMSE, while model 3 performs worse with a 4.6% increase. Adding the exogenous variable to the model does not improve performance since the results are the same between model 1 and 2. On average model 3 slightly outperforms the first two models. When compared to the benchmark model, the first two models on average have a RMSE that is 11.0% higher than the benchmark, while the 3rd model has a RMSE that is 8.4% higher than the benchmark. Although model 3 has RMSE that is slightly lower than the first two models, the 1st model is selected over the other two LightGBM models; since on average model 1 performs similarly to the other two ML models, while being the simplest model to understand since only two features are used. When comparing the benchmark model to the three LightGBM models, on average the benchmark outperforms the ML models for budget forecast.

Table 19: LightGBM Budget Forecast RMSE Percent Difference to Benchmark Model

Level	*Model 1*	*Model 2*	*Model 3*
A	13.7%	13.7%	13.5%
A1	-1.1%	-1.1%	-0.2%
A2	7.8%	7.8%	11.1%
B	7.6%	7.6%	5.8%
B1	7.5%	7.5%	8.6%
B2	-11.2%	-11.2%	-12.0%
B3	-3.2%	-3.2%	-3.5%
B4	55.5%	55.5%	26.6%
B5	-8.9%	-8.9%	5.5%
C	2.4%	2.4%	8.3%
C1	0.6%	0.6%	1.9%
C2	-8.3%	-8.3%	-11.0%
C3	18.7%	18.7%	27.6%
Fleet	-1.9%	-1.9%	4.6%
Average	11.0%	11.0%	8.4%

Notes: Model 1 (Feature Set 1); Model 2 (Feature Set 2); Model 3 (Feature Set 3)

Figure 20 compares the cumulative RMSE of the benchmark statistical models with the three LightGBM variants for the 2024 budget forecast. Across the forecasting horizon, all four models track each other closely, with little divergence in cumulative error during the early months. As the year progresses, small divergences emerge, though they remain small. Overall, the results indicate that for the budget forecast, the benchmark outperforms the LightGBM models.

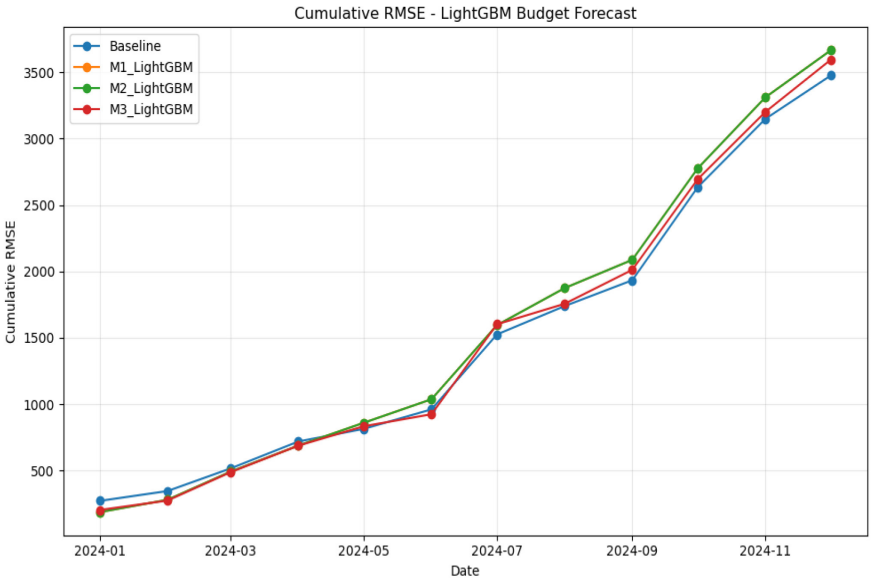


Figure 20: *Cumulative RMSE – LightGBM Budget Forecast*

4.3.2 Weekly Schedule Forecasting Results of LightGBM Models

Table 21 reports the percentage difference in RMSE between three LightGBM models and the statistical benchmark model for the 2024 schedule forecasts. The results show that across all aggregation levels, the LightGBM model performs worse than the benchmark. On average across all series, LightGBM increased RMSE by between 11.9% to 12.4% when compared to the benchmark. Unlike the budget forecast where LightGBM models occasionally achieved localized improvements, the benchmark statistical model for the schedule forecasts offers improved accuracy for all series except one. The LightGBM model with the lowest RMSE on average across all series is model 2. Model 1 which uses two instead of four features obtains a RMSE that is only 0.5% higher than model 2. Model 1 is selected as the best performing model since it has approximately the same error as model 2, while being considerably less complex than model 2 since model 1 does not use exogenous variables. Complex models should only be selected over simpler models if the increased performance justifies the use of a more complex model, which in this case it does not.

Table 20:LightGBM Scheduling Forecasting RMSE Percent Difference to Benchmark Models

Level	*Model 1*	*Model 2*	*Model 3*
A	**14.7%**	**14.8%**	**9.9%**
A1	9.5%	7.8%	6.6%
A2	20.6%	22.8%	25.9%
B	**8.7%**	**7.8%**	**5.5%**
B1	32.3%	25.8%	29.5%
B2	33.4%	34.9%	34.7%
B3	0.7%	4.5%	5.0%
B4	5.6%	2.4%	-0.3%
B5	19.3%	18.9%	32.4%

C	**11.3%**	**4.7%**	**1.8%**
C1	3.8%	5.9%	2.8%
C2	0.6%	5.3%	0.3%
C3	2.6%	1.4%	1.6%
Fleet	**10.7%**	**9.6%**	**11.8%**
Average	12.4%	11.9%	12.0%
Notes: Model 1 (Feature Set 1); Model 2 (Feature Set 2); Model 3 (Feature Set 3)			

Figure 21 presents the cumulative RMSE of the benchmark and three LightGBM variants for the 2024 schedule forecasts. The results show that while all models perform similarly in the early months, differences between benchmark and LightGBM models become more pronounced as the forecasts approaches the end of year.

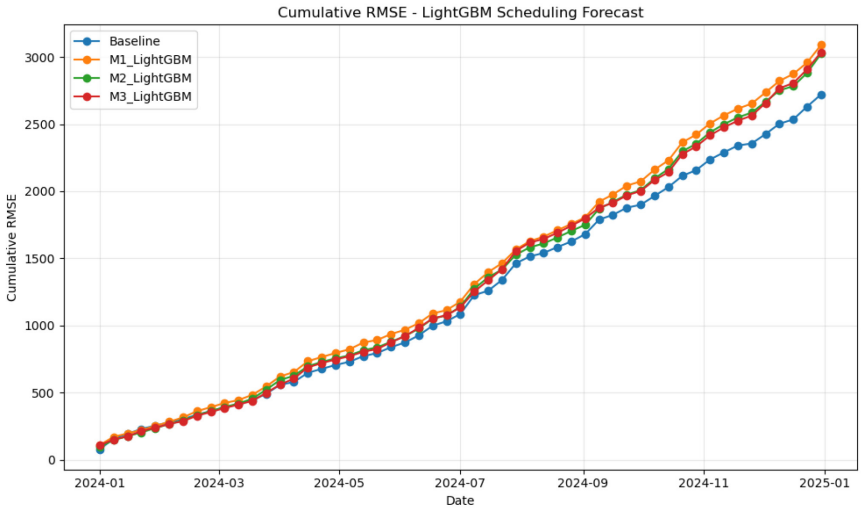


Figure 21: *Cumulative RMSE – LightGBM Scheduling Forecast*

4.4 KNN, SVM, & NN Forecasting Results

As shown in **Table 22**, KNN, SVM, and NN are compared to the benchmark models for the budget and schedule forecasting tasks. Similarly to the prior section, the RMSE % error between ML and benchmarks are examined to determine whether ML models yield improved accuracy over a simple statistical benchmark. The results in **Table 22** show that the ML models examined did not outperform the simple statistical benchmarks. Each of the ML models examined where fitted against the three features sets used for the LightGBM model. For the workload schedule forecasting task, feature set 1 yields a lower RMSE when compared to feature set 2 and 3. As for the workload budget forecasting task, feature set 1 yields the lowest RMSE for KNN, while feature set 2 and 3 yields the lowest RMSE for SVM and NN respectively. For both the budget and schedule workload forecasting task, KNN yields to the lowest RMSE when compared to LightGBM, SVM, and NN.

Table 21: RMSE % Difference Between ML & Benchmarks				
Forecasting Taks	*Model*	*Feature Set 1*	*Feature Set 2*	*Feature Set 3*
Monthly Budgeting	LightGBM	11.0%	11.0%	84.0%
	KNN	7.3%	8.0%	8.5%
	SVM	16.3%	16.3%	14.8%
	NN	11.8%	8.7%	12.8%
Weekly Scheduling	LightGBM	12.4%	11.9%	12.0%
	KNN	11.3%	70.0%	69.1%
	SVM	24.6%	69.6%	70.1%
	NN	44.1%	68.0%	70.6%
KNN (K-Nearest Neighbors); SVM (Support Vector Machines); NN (Neural Networks); LightGBM (Light Gradient Boosting Machines)				

4.5 Hierarchical Reconciliation Technique Results

Table 23 presents the results of the LightGBM budget forecasting reconciliation experiments, showing the percentage difference in RMSE between the base model and reconciliation methods outlined in **Table 12**. The negative values signify the percent improvement that the reconciled forecast has over the base model. Feature set 1 was used to train the base LightGBM model for the budget and schedule forecasting reconciliation experiments. The results indicate that the bottom-up approach delivers the greatest improvement at reducing RMSE when compared to the base model. On average the bottom-up approach leads to a 4.7% decrease in RMSE when compared to the base model, with particular notable gains at the platform level (-10.8%). The two middle-out approaches produce the same results and increase RMSE by 4.4%. Both the top-down and optimal-combination approaches decrease RMSE, with the optimal-combination weighted least squares leading to the greatest reduction in RMSE at 2.0%.

Table 22: LightGBM Budget Forecasting Reconciliation (RMSE % Difference Between Reconciliation Techniques & Base Model)				
Reconciliation Technique	*Level*			
	Fleet	*Platform*	*Windfarm*	*Avg.*
BottomUp	-3.3%	-10.9%	0.0%	-4.7%
TopDownAvgProp	0.0%	-7.1%	4.9%	-0.7%
TopDownPropAvg	0.0%	-8.0%	4.5%	-1.2%
MiddleOutAvgProp	5.9%	0.0%	7.4%	4.4%
MiddleOutPropAvg	5.9%	0.0%	7.2%	4.4%
OptimalCombination-MinT-ols	0.6%	-4.2%	2.4%	-0.4%
OptimalCombination-MinT-wls	-0.4%	-7.0%	1.4%	-2.0%
Task 1 Avg.	1.2%	-5.3%	4.0%	0.0%

Table 24 presents the same metric as **Table 23**, however the reconciliation approaches of the schedule forecast are evaluated instead of the reconciliation methods of the budget forecast. The results indicate that most reconciliation methods increase the forecast error for schedule forecasting when compared to the base model. The bottom-up approach increases RMSE by 6.0% on average, with the largest increases of RMSE at the fleet level. The top-down approaches performs worst among all techniques, with an average RMSE increasing close to 20%. Similarly, the middle-out approaches also reduces forecast accuracy, increasing RMSE by an average of 10%. In contracts, the optimal combination methods achieve a small average reductions in RMSE between 1 to 2 percent.

***Table 23**: LightGBM Schedule Forecasting Reconciliation*

(RMSE % Difference Between Reconciliation Techniques & Base Model)

Reconciliation Technique	*Level*			
	Fleet	*Platform*	*Windfarm*	*Avg.*
BottomUp	9.2%	8.7%	0.0%	6.0%
TopDownAvgProp	0.0%	19.6%	39.9%	19.8%
TopDownPropAvg	0.0%	18.1%	38.9%	19.0%
MiddleOutAvgProp	-3.0%	0.0%	34.0%	10.3%
MiddleOutPropAvg	-3.0%	0.0%	33.6%	10.2%
OptimalCombination-MinT-ols	-2.7%	-0.6%	-2.2%	-1.8%
OptimalCombination-MinT-wls	-2.2%	0.2%	-2.0%	-1.3%
Task 1 Avg.	-0.2%	6.6%	20.3%	8.9%

5. Discussion of Results and Implications

The forecasting design attributes "ML Vs. Traditional Statistical Benchmarks" and "Hierarchical Reconciliation" were both incorporated in the forecasting experiments to address the research gap identified in the literature review. The design attribute "ML Vs. Traditional Statistical Benchmark" assesses whether the use of ML outperforms simple statistical benchmarks for workload forecasting. The benchmark that was used for the budget forecast was the trailing 3-year seasonal average after applying a 3-month centered rolling average to the original series. As for the benchmark for the schedule forecast, a 3-year trailing rolling average was used. The ML models that were evaluated against the benchmarks were LightGBM, KNN, SVM, and NN.

The best performing ML model for both the budget and schedule forecasting task was KNN, which is the simplest model among all the ML models examined. Although the KNN model outperformed other ML models, it failed to outperform the simple statistical benchmark. Another interesting finding is that for most of the ML experiments, the lowest RMSE was achieved by using feature set 1. Feature set 2 and 3, which incorporated the exogenous variable downtime, was not shown to significantly improve forecasting accuracy for any of the ML models that were evaluated.

Although there is a considerable correlation between downtime and workload, downtime did not significantly improve workload forecasting for both budgeting and scheduling.

The forecasting design attribute "Hierarchical Reconciliation" was incorporated within the experiments by identifying whether reconciliation approaches offer higher accuracy compared to the base LightGBM models. The results of the reconciliation performed on the schedule forecasting task support the findings of previous reconciliation studies. The top-down approach significantly outperforms the bottom-up approach at the fleet level, while the bottom-up approach significantly outperforms the top-down approach at the windfarm level. In terms of the approaches that performed best overall, the optimal-combination methods using ordinary least squares and weighted least squares achieved the highest performance. The optimal-combination reconciliation approaches were the only methods that outperformed the schedule base forecast in terms of forecasting accuracy. The accuracy gains provided by the optimal-combination approaches are small, however it leads to coherent forecasts across aggregation levels which is a characteristic that the base forecasts do not provide to decision-makers.

The results of the hierarchical reconciliation performed for the schedule forecast conform to the expected results. At the most disaggregated level in the hierarchy (e.g. windfarm level), the bottom-up approach is more accurate than the top-down and middle-out approaches. This can be explained that both the top-down and middle-out approaches use weights to derive forecasts at lower levels of the hierarchy. The top-down approach is expected to have lower errors than the middle-out and bottom-up at the top level of the hierarchy, since the base forecasts are directly taken from the highest level of the hierarchy for the top-down approach, whereas this is not the case for the bottom-up and middle-out approaches. Lastly, the optimal combination approach using minimum trace is expected to outperform the other approaches previously mentioned because it uses the correlation between series to minimize the overall error of the forecasts.

As for the findings of the budget forecast reconciliation experiments, the optimal-combination reconciliation approaches also slightly improved forecasting accuracy compared to the base LightGBM models. Moreover, the middle-out approaches led to the least accurate forecasts among all the reconciliation approaches that were evaluated. There are however some unexpected results that were observed for the schedule forecast reconciliation experiments. The first unexpected results are the accuracy gains produced by the bottom-up approach at the fleet and platform level. Past studies have shown that the accuracy of the bottom-up approach relative to the base forecast should decrease when moving up through the aggregation hierarchy. In other words, for the bottom-up approach, the fleet level is expected to have the highest error relative to the base model, followed by the platform and windfarm levels. The negative RMSE percent difference between the top-down approach and base models at the platform level is also unexpected.

An explanation to why the scheduled forecasts yield expected results, while the budget forecast yields unexpected results is that the budget forecasts do not have a large enough testing dataset thus leading to spurious results. For both the schedule and budget forecasting task, the testing set consists of predicting workload for 2024. The dataset is partitioned into 83.3% training and 16.7% testing. Although the testing dataset contains 168 datapoint for the budget forecast, there are only 12 forecast predictions for each of the 14 series. The scheduled forecast on the other hand

makes 53 forecast predictions per series. Since the forecasting task was to evaluate forecasting models and reconciliation methods for 2024, the number of predictions made per series for the budget was set to 12 and the number of predictions made for the schedule forecast was set to 53. The schedule budget forecast requires a prediction for each month of the year, while the schedule forecast requires a prediction for each week of the year.

One explanation for why the simple rolling-average benchmarks outperformed the ML models is based on the characteristics of the time-series used in the study. Both the monthly and weekly time-series under consideration have low PACF lag values, which indicate that the time-series are driven by irregular fluctuations rather than strong predictable temporal relationships. Under these conditions, ML models such as NN, SVM, and LightGBM tend to overfit noise in the training dataset, which leads to lower out-of-sample accuracy when compared to a simple statistical benchmark such as rolling averages. Another important factor that explains why rolling averages outperformed ML models are that ML models generally require substantially more data to learn temporal patterns, often needing several hundred observations to reliably estimate nonlinear relationships. The dataset used in this research, particularly at the monthly frequency, may not have had enough observations for ML forecasting models to be competitive, despite using 6 years of data to train the models.

Although it was found that simple statistical models outperform ML for workload forecasting, one should be cautious in generalizing the findings to other firms and industries. It may be the case that the characteristics of the dataset of other firms and industries differ significantly, which may lead to experimental results that contradict the findings of this thesis. Future studies should attempt to replicate the methodology outlined for this thesis on datasets spanning across multiple firms and industries to assess the generalizability of the findings provided by this thesis. It is also important to mention that the list of ML models that are assessed against simple statistical models is not exhaustive. There is the potential that a special application of ML such as deep learning models using long short-term memory architecture outperforms simple statistical methods for workload forecasting. Future research should examine whether deep learning models can outperform simple statistical methods.

6. Conclusion

Based on the findings of the experiments, the first recommendation to managers is that workload forecasting should be based on traditional statical methods rather than ML models. For both the budget and schedule forecasting task, simple statistical methods outperformed KNN, LightGBM, SVM, and NN. The high level of noise that is present in the time-series makes it improbable that ML models will outperform simple statistical benchmarks. Traditional statistical methods also have the added benefit of yielding results that are easier to interpret than complex ML models. Another recommendation to managers is to perform feature engineering on lag values of workload. Taking the 3-year seasonal average after applying a 3-month centered rolling average to the original time-series led to the highest forecasting accuracy among all the budget forecasting models. Using domain knowledge, ACF, and PACF for feature engineering has the potential of greatly increasing the accuracy of the budget forecasts.

The last recommendation is to use optimal-combination approaches for workload hierarchical forecasting. As for schedule forecasting, the results clearly show that optimal-combination reconciliation outperform bottom-up, top-down, and middle-out approaches. The best performing reconciliation model for budget forecast was the bottom-up approach; however, it was determined that the number of predictions may have been insufficient to yield reliable results. Existing theories along with past studies have shown that the optimal-combination approach typically yield superior results when compared to the other reconciliation approaches under consideration. The two optimal-combination approaches used for the experiments were MinT with either OLS or WLS. There was not a significant difference between the forecasting accuracy of OLS and WLS. Since the results of OLS and WLS are similar, OLS is recommended since it is simpler to interpret than WLS.

With the intention of directing future studies, the limitations of the experiments are discussed. Data quality issues related to historical workload hours prevented distinguishing between scheduled maintenance and unplanned maintenance. Technician hours are booked to work orders which are supposed to detail the type of work that has been performed. Unfortunately, the information on the type of work that has been performed on the turbine is often unreliable. The inability to distinguish between scheduled and unscheduled maintenance is a major limitation of the study, since most of the value from workload forecasting comes from anticipating unplanned workload. Time-based scheduled forecasting does not need to be forecasted since it is planned ahead of time. Future studies in workload forecasting should aim to find datasets that only contain historical unplanned workload.

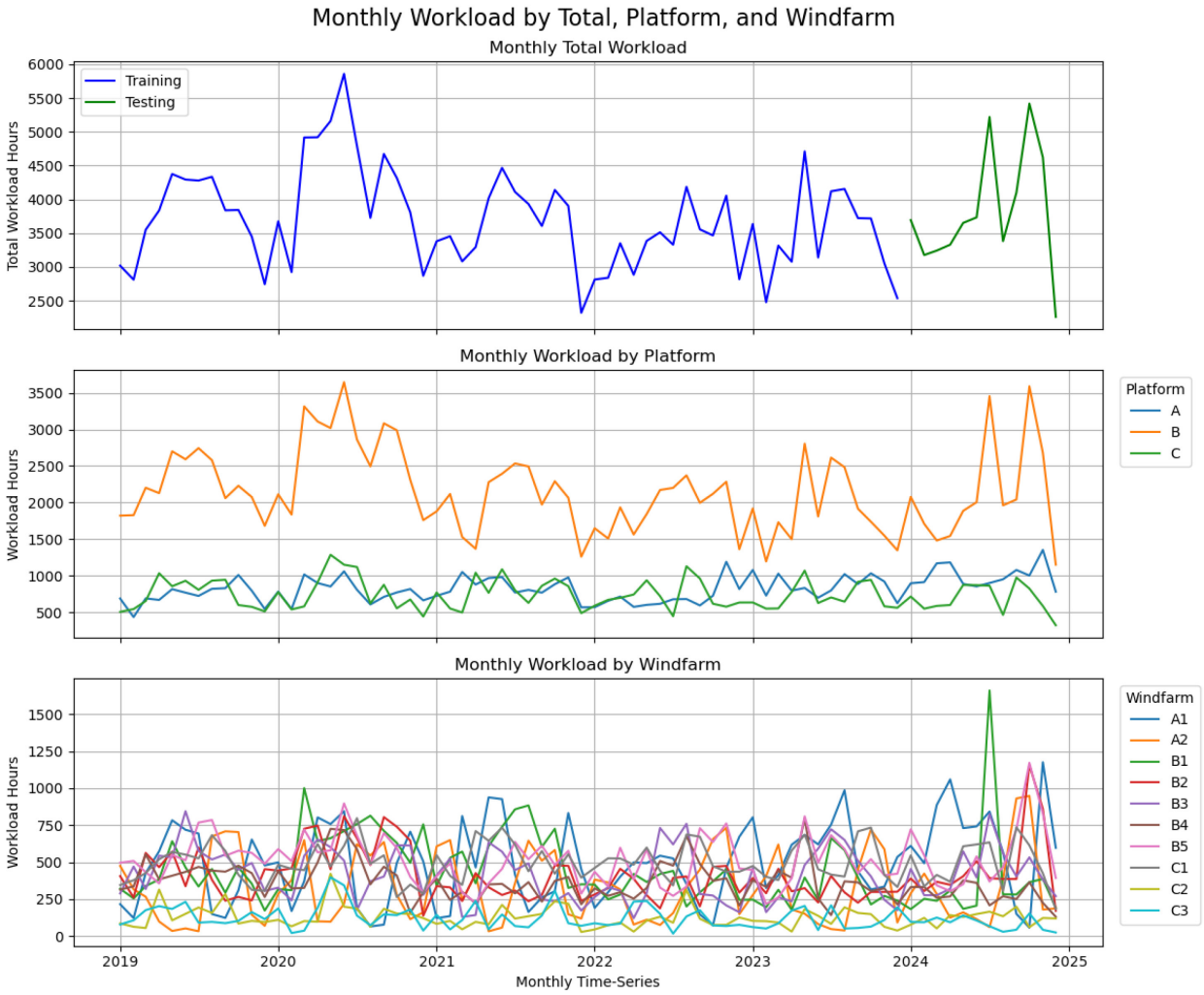
Another limitation of the study is that there were some hierarchical forecasting methods that were not evaluated. For instance, future studies can examine whether approaches such as empirical risk minimization (ERM) reconciliation outperform the other reconciliation approaches that were explored in this thesis. Another limitation of the study is that hyper-parameter tuning of ML models was not exhaustive. For example, hyper-parameter tuning was limited to 2 key parameters for the LightGBM models. Future studies should examine whether additional hyper-parameter tuning have the potential of significantly improving model performance. Lastly, although the study contains data from 316 turbines for a period of 6 years, the data comes from only one company. A similar study that collects historical workload from multiple companies operating in the wind-energy sector would improve the generalizability of the findings.

Appendices

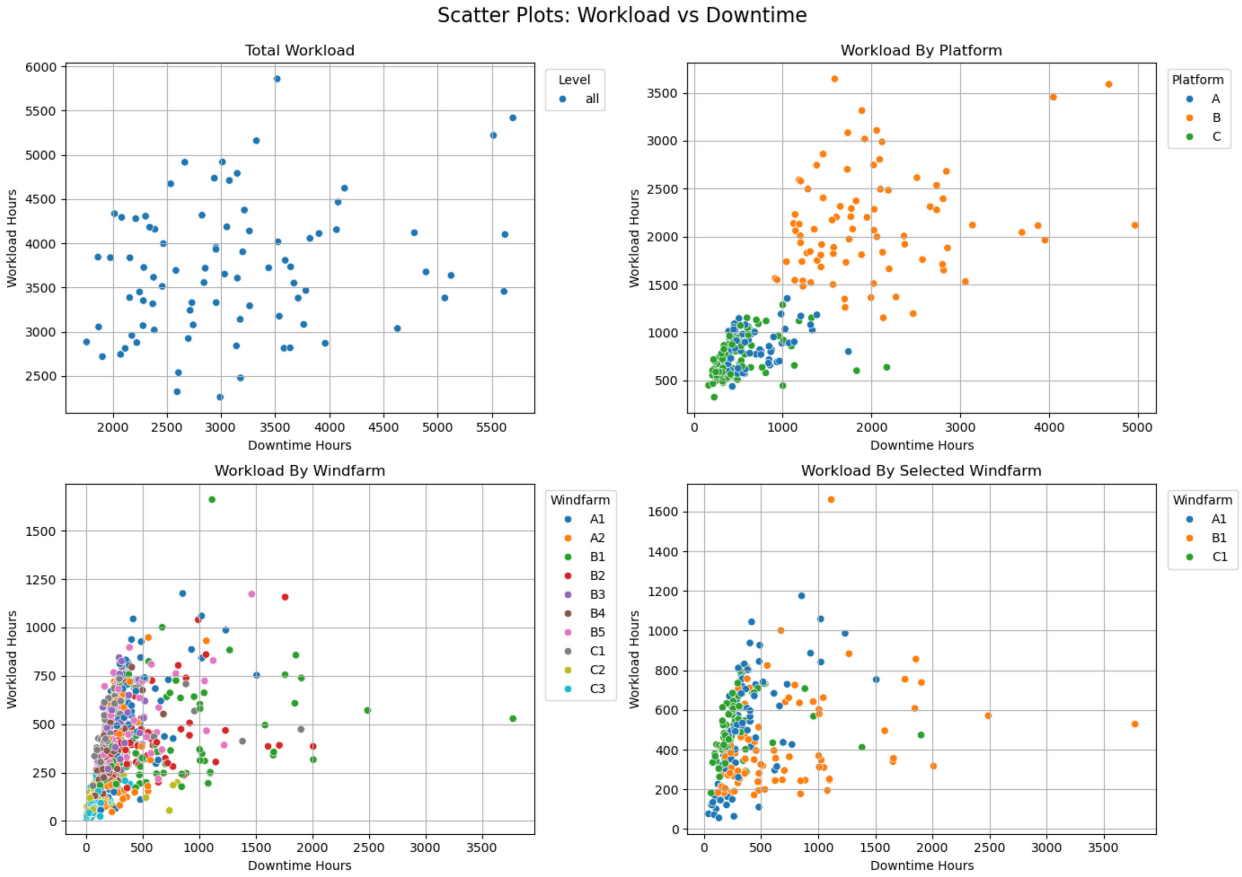
Appendix A: Scheduling Forecasting Task Feature Sets

Scheduling Forecasting Task Feature Sets			
Feature Set	*Feature Characteristics*	*Component*	*Variable Description*
Feature Set 1	Centrality of seasonal lag values of target variable	Short-term	Trailing 3-week rolling average of workload
		Long-term	Seasonal trailing 3-year rolling average after applying 3-week centered rolling average to original workload series
Feature Set 2	Centrality of seasonal lag values of target variable	Short-term	Trailing 3-week rolling average of workload
		Long-term	Seasonal trailing 3-year rolling average after applying 3-week centered rolling average to original workload series
	Centrality of seasonal lag values of exogenous variable	Short-term	Trailing 3-week rolling average of downtime
		Long-term	Seasonal trailing 3-year rolling average after applying 3-week centered rolling average to original downtime series
Feature Set 3	Centrality of seasonal lag values of target variable	Short-term	Trailing 3-week rolling average of workload
		Long-term	Seasonal trailing 3-year rolling average after applying 3-week centered rolling average to original workload series
	Centrality of seasonal lag values of exogenous variable	Short- term	Trailing 3-week rolling average of downtime
		Long-term	Seasonal trailing 3-year rolling average after applying 3-week centered rolling average to original downtime series
	volatility of seasonal lag values of target variable	Short-term	Trailing 3-week rolling standard deviation of workload
		Long-term	Seasonal trailing 3-year rolling standard deviation after applying 3-week centered rolling average to original workload series

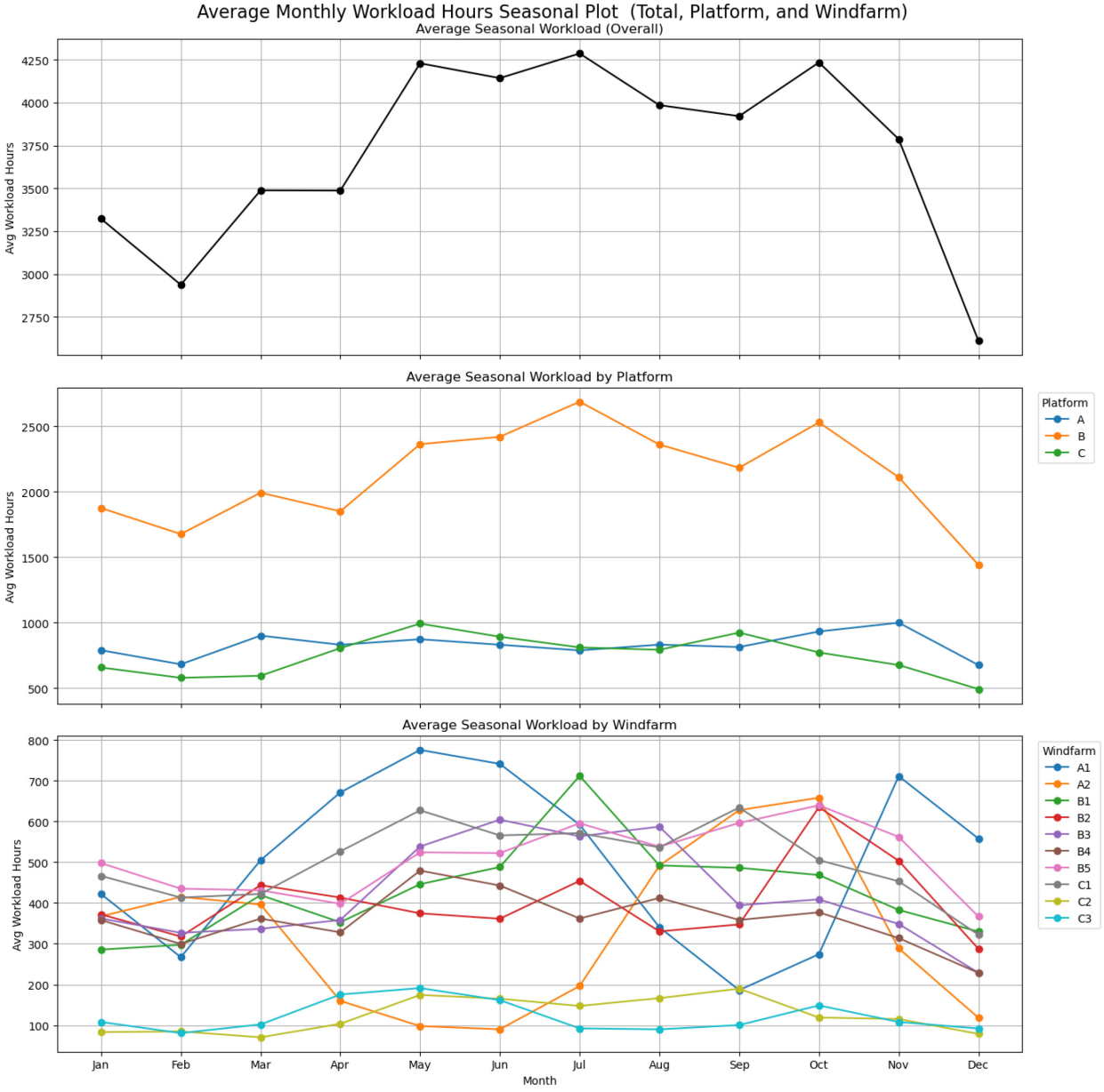
Appendix B: Monthly Workload by Total, Platform, and Windfarm



Appendix C: Scatter Plots (Windfarm Vs Downtime)



Appendix D: Average Monthly Workload Seasonal Plots



References

A. H. Mohamed, R. (2023). Enhancing forecast accuracy using combination methods for the hierarchical time series approach. *PLOS ONE*, *18*(7), e0287897. https://doi.org/10.1371/journal.pone.0287897

Abolghasemi, M., Hyndman, R. J., Tarr, G., & Bergmeir, C. (2019). *ML applications in time series hierarchical forecasting* (No. arXiv:1912.00370). arXiv. https://doi.org/10.48550/arXiv.1912.00370

Al-Fares, H. K., & Duffuaa, S. O. (2009). Maintenance Forecasting and Capacity Planning. In M. Ben-Daya, S. O. Duffuaa, A. Raouf, J. Knezevic, & D. Ait-Kadi (Eds.), *Handbook of Maintenance Management and Engineering* (pp. 157–190). Springer London. https://doi.org/10.1007/978-1-84882-472-0_8

Bojer, C. S., & Meldgaard, J. P. (2021). Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting*, *37*(2), 587–603. https://doi.org/10.1016/j.ijforecast.2020.07.007

CRISP-DM Help Overview. (2021, August 17). IBM. https://www.ibm.com/docs/en/spss-modeler/saas?topic=dm-crisp-help-overview

El-Naggar, M., Sayed, A., Elshahed, M., & EL-Shimy, M. (2023). Optimal maintenance strategy of wind turbine subassemblies to improve the overall availability. *Ain Shams Engineering Journal*, *14*(10), 102177. https://doi.org/10.1016/j.asej.2023.102177

Foresight. (2025, February 12). Cambridge Dictionary. https://dictionary.cambridge.org/us/dictionary/english/foresight

Glowik, M., Bhatti, W. A., & Chwialkowska, A. (2023). A cluster analysis of the global wind power industry: Insights for renewable energy business stakeholders and environmental policy decision makers. *Business Strategy and the Environment*, *32*(6), 2755–2766. https://doi.org/10.1002/bse.3268

Goodwin, P., Hoover, J., Makridakis, S., Petropoulos, F., & Tashman, L. (2023). Business forecasting methods: Impressive advances, lagging implementation. *PLOS ONE*, *18*(12), e0295693. https://doi.org/10.1371/journal.pone.0295693

Gupta, M., & Jauhar, S. K. (2023). Digital innovation: An essence for Industry 4.0. *Thunderbird International Business Review*, *65*(3), 279–292. https://doi.org/10.1002/tie.22337

Haberleitner, H., Meyr, H., & Taudes, A. (2010). Implementation of a demand planning system using advance order information. *International Journal of Production Economics*, *128*(2), 518–526. https://doi.org/10.1016/j.ijpe.2010.07.003

Hewamalage, H., Ackermann, K., & Bergmeir, C. (2023). Forecast evaluation for data scientists: Common pitfalls and best practices. *Data Mining and Knowledge Discovery*, *37*(2), 788–832. https://doi.org/10.1007/s10618-022-00894-5

Hyndman, R. J. (2020). A brief history of forecasting competitions. *International Journal of Forecasting*, *36*(1), 7–14. https://doi.org/10.1016/j.ijforecast.2019.03.015

Introduction to SEMMA. (2017, August 30). SAS. https://documentation.sas.com/doc/en/emref/14.3/n061bzurmej4j3n1jnj8bbjjm1a2.htm

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An Introduction to Statistical Learning: With Applications in Python*. Springer International Publishing. https://doi.org/10.1007/978-3-031-38747-0

Joseph, M. (2022). *Modern time series forecasting with python: Explore industry-ready time series forecasting using modern ML and deep learning, 1st edition* (1st ed). Packt Publishing.

Kaldellis, J. K., & Zafirakis, D. (2011). The wind energy (r)evolution: A short review of a long history. *Renewable Energy*, *36*(7), 1887–1901. https://doi.org/10.1016/j.renene.2011.01.002

Khalid, W., Albrechtsen, S. H., Sigsgaard, K. V., Mortensen, N. H., Hansen, K. B., & Soleymani, I. (2021). Predicting maintenance work hours in maintenance planning. *Journal of Quality in Maintenance Engineering*, *27*(2), 366–384. https://doi.org/10.1108/JQME-06-2019-0058

Land-Based Wind Market Report: 2023 Edition. (2023).

Lazzeri, F. (2020). *ML for Time Series Forecasting with Python®* (1st ed.). Wiley. https://doi.org/10.1002/9781119682394

Letmanyi, H. (1985). *Guide on workload forecasting* (No. NBS SP 500-123; 0 ed., p. NBS SP 500-123). National Bureau of Standards. https://doi.org/10.6028/NBS.SP.500-123

Li, H., Ribeiro, M., Santos, B., & Tseremoglou, I. (2024, January 8). Prediction of Non-Routine Tasks Workload for Aircraft Maintenance with Supervised Learning. *AIAA SCITECH 2024 Forum*. AIAA SCITECH 2024 Forum, Orlando, FL. https://doi.org/10.2514/6.2024-2529

Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E., & Winkler, R. (1982). The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of Forecasting*, *1*(2), 111–153. https://doi.org/10.1002/for.3980010202

Makridakis, S., Chatfield, C., Hibon, M., Lawrence, M., Mills, T., Ord, K., & Simmons, L. F. (1993). The M2-competition: A real-time judgmentally based forecasting study. *International Journal of Forecasting*, *9*(1), 5–22. https://doi.org/10.1016/0169-2070(93)90044-N

Makridakis, S., Fry, C., Petropoulos, F., & Spiliotis, E. (2022). The Future of Forecasting Competitions: Design Attributes and Principles. *INFORMS Journal on Data Science*, *1*(1), 96–113. https://doi.org/10.1287/ijds.2021.0003

Makridakis, S., & Hibon, M. (2000). The M3-Competition: Results, conclusions and implications. *International Journal of Forecasting*, *16*(4), 451–476. https://doi.org/10.1016/S0169-2070(00)00057-1

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). The M4 Competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, *34*(4), 802–808. https://doi.org/10.1016/j.ijforecast.2018.06.001

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, *38*(4), 1346–1364. https://doi.org/10.1016/j.ijforecast.2021.11.013

Makridakis, S., Spiliotis, E., Hollyman, R., Petropoulos, F., Swanson, N., & Gaba, A. (2024). The M6 forecasting competition: Bridging the gap between forecasting and investment decisions. *International Journal of Forecasting*, S0169207024001079. https://doi.org/10.1016/j.ijforecast.2024.11.002

Monteiro, P., Lino, J., Araújo, R. E., & Costa, L. (2024). Comparison between LightGBM and other ML algorithms in PV fault classification. *EAI Endorsed Transactions on Energy Web*, *11*. https://doi.org/10.4108/ew.4865

Olivares, K. G., Garza, A., Luo, D., Challú, C., Mergenthaler, M., Taieb, S. B., Wickramasuriya, S. L., & Dubrawski, A. (2024). *HierarchicalForecast: A Reference Framework for Hierarchical Forecasting in Python* (No. arXiv:2207.03517). arXiv. https://doi.org/10.48550/arXiv.2207.03517

Purwar, A., & Reimherr, M. (2023). ARIMAX Model for Forecasting Maintenance Work (AMFM): A Multi-Stage Seasonal ARIMAX Model for Workorder Time Series Forecasting. *Amazon Science*.

Reddy, G. S., Srinivasu, R., Rao, M. P. C., & Rikkula, S. R. (2010). *DATA WAREHOUSING, DATA MINING, OLAP AND OLTP TECHNOLOGIES ARE ESSENTIAL ELEMENTS TO SUPPORT DECISION-MAKING PROCESS IN INDUSTRIES*. *02*(09).

Rodríguez-Pérez, R., & Bajorath, J. (2022). Evolution of Support Vector Machine and Regression Modeling in Chemoinformatics and Drug Discovery. *Journal of Computer-Aided Molecular Design*, *36*(5), 355–362. https://doi.org/10.1007/s10822-022-00442-9

Safarishahrbijari, A. (2018). Workforce forecasting models: A systematic review. *Journal of Forecasting*, *37*(7), 739–753. https://doi.org/10.1002/for.2541

Salman, I. A. (2004). Forecasting models for maintenance work load with seasonal components. *Annual Symposium Reliability and Maintainability, 2004 - RAMS*, 514–520. https://doi.org/10.1109/RAMS.2004.1285499

Sanders, N. R., & Manrodt, K. B. (2003). The efficacy of using judgmental versus quantitative forecasting methods in practice. *Omega*, *31*(6), 511–522. https://doi.org/10.1016/j.omega.2003.08.007

Shmueli, G., Bruce, P., Gedeck, P., & Patel, N. (2020). *Data mining for business analytics: Concepts, techniques and applications in Python*. John Wiley & Sons, Inc.

Shmueli, G., Bruce, P., & Yahav, I. (2017). *Data Mining For Business Analytics Concepts, Techniques, and Applications in R*. John Wiley & Sons, Inc.

Silva, A. J., Cortez, P., Pereira, C., & Pilastri, A. (2021). Business analytics in Industry 4.0: A systematic review. *Expert Systems*, *38*(7), e12741. https://doi.org/10.1111/exsy.12741

Silveira Gontijo, T., & Azevedo Costa, M. (2020). Forecasting Hierarchical Time Series in Power Generation. *Energies*, *13*(14), 3722. https://doi.org/10.3390/en13143722

Van Gils, T., Ramaekers, K., Caris, A., & Cools, M. (2017). The use of time series forecasting in zone order picking systems to predict order pickers' workload. *International Journal of Production Research*, *55*(21), 6380–6393. https://doi.org/10.1080/00207543.2016.1216659

Yolcan, O. O. (2023). World energy outlook and state of renewable energy: 10-Year evaluation. *Innovation and Green Development*, *2*(4), 100070. https://doi.org/10.1016/j.igd.2023.100070