Technische Universität Dresden

"Friedrich List" Faculty of Transport and Traffic Sciences

Institute of Transport and Economics
Chair of Econometrics and Statistics, esp. in the Transport Sector

Master Thesis

Applied Time Series-Based Load Forecasting and Backcasting: A Comparative Study

Bhavay Singhal
Born on: 17th April 1998 in Delhi, India
Matriculation number: 5126210
Email: bhavay.singhal@mailbox.tu-dresden.de

to achieve the academic degree

Master of Science (M.Sc.)
Transportation Economics

Supervisor
Ms. Jing Zou

Supervising Professor
Prof. Dr. Ostap Okhrin

Submitted on: 08 January 2026

Abstract

This master's thesis investigates applied time series methods for short-term load forecasting and backcasting in electric power systems, using the GEFCom2012 load forecasting dataset. Motivated by the critical role of accurate load forecasting in maintaining grid stability, optimizing resource allocation, and supporting transportation infrastructure reliant on reliable electricity, the study addresses challenges in high-frequency load data under realistic constraints like limited training periods and data scarcity. The primary objective is to compare classical time series models—such as exponential smoothing, harmonic regression, piecewise linear regression, seasonal ARIMA (SARIMA), and SARIMA with exogenous variables (SARIMAX)—against naïve baseline and competition benchmark to identify efficient approaches that achieve at least 30% error reduction relative to competition benchmarks, while emphasizing model simplicity for practical deployment.

The methodology begins with exploratory data analysis (EDA) on 4.5 years of hourly load data from 20 U.S. utility zones and temperature readings from 11 stations, revealing strong daily and monthly periodicities, non-linear load-temperature relationships, and high correlations among zones and stations. Principal Component Analysis (PCA) is employed to derive a single temperature signal capturing system-wide variance. Models are trained on approximately three weeks of data per session (last weeks of selected 2004 months as test sets), evaluated using Root Mean Squared Error (RMSE) and R^2 scores. Naïve methods include repeating prior cycles or averaging loads. Advanced techniques incorporate harmonics for seasonality, piecewise segments for non-linearity, and ARIMA components for serial dependencies. Backcasting is tested by reversing temporal order in secondary training sets to combine with forecasts.

Key findings demonstrate that simpler methods for load estimation outperform complex ones under constraints. Exponential smoothing performs poorly, while simple linear regression improves load estimation slightly. Piecewise linear regression with two harmonics is found to be the best load estimation approach. It successfully captures non-linearities, without resulting in overfitting. Combining forecasting and backcasting shows no net gain, over simple forecasting. For the full grid forecast horizon, piecewise linear regression achieves 37% RMSE reduction vs. naïve (10804 vs. 17128) and 30% RMSE reduction vs. benchmarks (15513).

Contributions include validating simple, interpretable models like piecewise harmonic regression for high-frequency STLF in data-limited scenarios, akin to findings in related works on data scarcity and recency effects. This gives strong support to traditional statistical methods in realistic utility settings, promoting efficiency for emerging smart grids and transportation electrification.

Table of Contents

Chapter 1: Introduction

1.1 Background

Power utilities are companies and organizations that are responsible for generating, transmitting, distributing and maintaining electric power reliably. Such companies are considered essential infrastructure because they enable all other critical services such as healthcare systems, water and waste treatment plants, electronic financial transactions, food preservation and storage, telecommunication networks, emergency services, etc. (National Academies of Sciences, Engineering, and Medicine, 2017) Considering the scale at which electric power is consumed, disruption in power has the potential to make multiple systems fail at once since there is no absolute long-duration substitute available.

In particular, electric power functions as a systemic enabler across many facets of the modern transportation systems (Mohamed, 2019):

- Rail transportation - electric trains, trams and high-speed rail are directly powered by grid electricity. Even diesel rail relies on electric signaling, switching, and yard operations.
- Road transportation - traffic signals, adaptive traffic control, tunnel ventilation, street lighting, toll systems, and weigh stations all depend on grid power. Electric vehicle charging infrastructure is entirely dependent on power grids.
- Air transportation - airports depend massively on electricity for runway lighting, radar, air traffic control systems, baggage handling, fueling automation and terminal operations.
- Water transportation - the availability of "Shore power" allows ships to shut engines off while docked, reducing operational costs and emissions.

In addition, intelligent transportation systems are increasingly relying on digital scheduling, algorithmic optimization and continuous active monitoring for improved coordination and efficiency gains. Lack of reliable electric power supply can cause malfunctions and weaken the effectiveness of dispatch centers, fleet tracking companies, passenger information systems and electronic ticketing solutions that rely on these technologies for their operations.

The examples above illustrate how reliable electric power facilitates physical movement, information communication and infrastructure operations in the transportation sector. Recognizing the dependence on electric power, a subsequent point of curiosity could be to know how this power is acquired. Power utilities function as the primary provider and transportation operators lie downstream of these utilities in a power distribution network. In other words, even when these operators have short-duration power generation capabilities, electric power does not originate from transportation operators but is made available to them via transmission networks and distribution channels.

The focus of this section is to emphasize why balancing supply and demand of electric load in real time is critical for entities lying both up and down the power distribution network. To put it into simpler terms, balancing supply and demand here means electric load must be generated and consumed almost simultaneously. If power supply exceeds demand, frequency in electricity grids rises, and if supply falls short, frequency drops. Both scenarios result in unstable operating frequency that can damage equipment or trigger blackouts. It could be argued that electricity can be stored on a large scale in case of imbalance. However, storing energy at the scale of a city or larger spaces for long periods of time is cost prohibitive (Baran, 2017). This is because converting electricity to storage and back involves energy losses from 15% to 30%, and building storage that can handle multi-day

demand peaks requires enormous physical, financial, and environmental resources. Precise load demand forecasting is a very effective planning tool to estimate the right amount of electricity to be produced, avoiding imbalance (Karaduman, 2021). Generating units, especially thermal and nuclear plants, have ramping limits and start-up costs. Accurate forecasts allow utilities to schedule generation efficiently, avoid load shedding, and minimize fuel costs and wear-and-tear.

1.2 Project's Goal: Gefcom 2012 Competition

Gefcom 2012 was a global energy load forecasting competition hosted on Kaggle from 1 September 2012 to 31 October 2012, meant to overcome forecasting challenges in smart grids and promote analytics in power engineering (Hong et al, 2014). The host of the competition was Dr. Tao Hong, Professor of Systems Engineering and Engineering Management (SEEM) and Associate of Energy Production Infrastructure Center (EPIC) at University of North Carolina at Charlotte (UNCC). The competition required participants to submit their solutions on forecasting hourly loads (in kW) for a US utility with 20 zones. While the competition's task was to perform short-term hierarchical load forecasting that provides load estimates for the entire grid (system level) in addition to forecasting load for each individual zone (zonal level), this paper focuses only on performing short-term load forecasting (STLF) at a zonal level. Forecasting performance for the entire grid is then interpreted as the average performance across all 20 zones. Other than temperature and holiday information, no other kind of information such as geographic, socio-economic or demographic information was provided.

The goal of this paper is to explore time series dynamics of electric load and compare forecasting performance of some of the most popular classical time series models with simple naïve methods, using the load dataset provided for this competition, to determine which approach works best for high-frequency load data. Some examples of classical time series models are Exponential Smoothing, harmonic regression, SARIMAX, etc. An additional task under this competition is also to fill in eight short-term time periods of missing load values (at zonal level - per zone) present in the dataset provided using the same approach used for forecasting. Here again, the chosen approach is compared with a naïve method to see imputation improvement in terms of error reduction. Instead of developing an approach that is extremely complex and difficult to deploy in real world applications, this paper attempts to find out if a simpler approach can be developed that is capable of coming close or at par with competition winning forecasting performance by using the least amount of training data. Model simplicity and minimal training are meant to be realistic constraints for electric load forecasting witnessed in modern-day practice.

As a footnote, the host conveys that there is no way to fully restore how the competition was set up on Kaggle and directly compare the error score with a model that was developed by an independent user who didn't take part in the competition at the time it was being held. However, the competition does provide a benchmark on the dataset that can be used for comparison purposes. The winning teams in the competition managed to make a 30% error reduction compared to these benchmark load values (Hong, 2016). Hence, this paper aims to meet this error reduction target for short-term load forecasting on the Gefcom 2012 dataset.

Chapter 2: Related Works

2.1 Electricity load forecasting: a systematic review

The paper by Nti et al (2020) provides a comprehensive overview of the challenges and advancements in predicting electricity demand. As global reliance on electricity intensifies for residential, commercial, and industrial purposes, accurate forecasting becomes crucial for efficient grid management, cost reduction, and sustainable energy planning. The authors conduct a systematic review of 77 relevant studies from 2010 to 2020, emphasizing that despite technological progress, load forecasting remains a persistent problem in the power industry due to its inherent complexity and the multifaceted factors influencing consumption patterns. This review underscores the need for improved methodologies to address data-driven uncertainties, highlighting how forecasting inaccuracies can disrupt economic stability and energy security worldwide.

At its heart, the paper identifies the core problem as the difficulty in achieving precise electricity load forecasting, a longstanding issue since the inception of electric power systems. Electricity demand is inherently volatile, influenced by unpredictable variables such as weather fluctuations, economic shifts, demographic changes, and consumer behaviors. Traditional mathematical formulas fall short in capturing these dynamics, leading to models that struggle with non-linear patterns and sudden variations. The authors note that forecasting is essential for planning generation, transmission, and distribution, yet it is complicated by the high costs of infrastructure and the inability of supply to always match demand. For instance, overestimation results in wasteful resource allocation, while underestimation causes shortages, exacerbating grid instability.

The problem is framed within two primary forecasting approaches: data-driven (artificial intelligence-based) methods and engineering (correlation and extrapolation) methods. Data-driven models, which dominate 90% of the top algorithms reviewed, excel in handling complex patterns but require vast datasets and computational resources. Engineering methods, relying on economic and demographic correlations, are simpler but often inaccurate due to the challenge of predicting interrelated factors like population growth or GDP. The review classifies forecasting into time horizons—very short-term (minutes to hours), short-term (hours to weeks), medium-term (weeks to a year), and long-term (over a year)—each presenting unique hurdles. Short-term load forecasting (STLF), which constitutes 80% of the studies, is particularly problematic due to rapid fluctuations, while long-term load forecasting (LTLF) grapples with broader uncertainties like policy changes.

The extent of the load forecasting problem is global and pervasive, affecting both developed and developing economies. The authors analyzed 77 papers, revealing a research concentration in Europe (31.34%), Asia (19.40%), and Africa (17.91%), with Africa underrepresented despite severe energy crises. In Africa, for example, South Africa (41.67% of African studies) and Ghana (25%) face frequent power shortages, termed "dumsor" in Ghana, due to inadequate forecasting and planning. Globally, electricity demand has surged with economic growth, but generation capacity lags, leading to price volatility and blackouts. The review highlights that residential demand, the basic unit of consumption, is rising due to low-quality appliances and lifestyle changes, yet only 11.94% of studies focus on it, indicating a research gap.

Quantitatively, the problem's scale is evident in the diversity of factors: 50% of studies incorporate weather and economic parameters (e.g., temperature, humidity, GDP), 38.33% use historical consumption data, 8.33% consider household lifestyles (e.g., appliance usage, family composition),

and 3.33% include stock indices. However, divergent views on factors like temperature's impact—positive in some regions, negligible in others—stem from territorial and cultural differences, complicating universal models. Error metrics underscore the extent: Root Mean Square Error (RMSE) is used in 38% of studies, Mean Absolute Percentage Error (MAPE) in 35%, and Mean Absolute Error (MAE) frequently, yet no single metric consistently excels across datasets.

The impact is profound, linking directly to economic and social consequences. Inaccurate forecasts lead to inefficient grid management, resulting in over-provisioning (increased costs and emissions) or under-provisioning (outages disrupting industries and daily life). For developing countries with low electrification rates, poor forecasting hinders infrastructure expansion, perpetuating poverty and stunting growth. In Ghana, limited studies contribute to ongoing shortages, while in Europe, the 2008 energy crisis spurred more research. Globally, the problem exacerbates climate challenges by inefficiently integrating renewables, which require precise balancing against variable loads. The authors emphasize that without better forecasting, utilities face financial losses, consumers endure higher bills, and societies risk energy insecurity, with residential sectors particularly vulnerable due to understudied lifestyle influences.

Key recommendations of the paper include benchmarking AI against traditional methods to clarify superiorities, particularly in underrepresented regions like Africa and Ghana. The authors urge more focus on residential forecasting, incorporating lifestyle variables (e.g., socio-economic factors, appliance usage) to improve accuracy, as these are underexplored yet influential. They propose clarifying divergent impacts of weather parameters through contextual studies, given global variations. For metrics, RMSE and MAPE are recommended for STLF, with calls for standardized evaluations. Challenges identified in the paper include data quality issues, model overfitting in limited datasets, and regional research imbalances.

2.2 Few-Shot Load Forecasting Under Data Scarcity in Smart Grids: A Meta-Learning Approach

The paper by Tsoumplekas et al (2024) tackles a pressing challenge in modern energy systems: the scarcity of data for accurate short-term load forecasting (STLF) in smart grids. As smart grids evolve with the integration of renewable energy sources, electric vehicles, and diverse consumer behaviors, reliable load forecasting becomes essential for optimizing energy distribution, reducing costs, and ensuring grid stability. However, the authors argue that despite the proliferation of advanced metering infrastructure (AMI) generating vast amounts of data at the individual consumer level, there remain numerous scenarios where sufficient historical data is unavailable or inadequate for training robust forecasting models. This data scarcity problem is not merely a minor inconvenience but a fundamental barrier that undermines the effectiveness of traditional machine learning and deep learning approaches, which typically require large datasets to achieve high accuracy.

Data scarcity in load forecasting manifests when historical consumption records are limited, incomplete, or disrupted. In smart grids, this issue arises from several practical realities. For instance, newly connected consumers or buildings may have no prior data history, making it impossible to build personalized models from scratch. Meter failures or data corruption can erase segments of records, while shifts in consumption patterns—such as the installation of photovoltaic panels, heat pumps, or electric vehicle charging stations—can render existing data obsolete by introducing new variability. The paper emphasizes that even with the rapid expansion of smart

grids, "there are still various cases where adequate data collection to train accurate load forecasting models is challenging or even impossible." This is particularly acute in short-term forecasting, where predictions are needed at fine granularities like 15-minute to 1-hour intervals over horizons of days or weeks.

The extent of this problem is widespread and multifaceted. In the context of the paper's evaluation, the authors use real-world datasets from 50 European consumers spanning October 2020 to April 2022, with time series lengths as short as 1-3 months—far below the volumes typically required for deep learning models like Long Short-Term Memory (LSTM) networks. Globally, data scarcity affects emerging smart grid deployments in developing regions, where infrastructure rollout is uneven, or in privacy-sensitive environments where data sharing is restricted. It also impacts dynamic systems with frequent changes, such as microgrids or demand-response programs, where consumption patterns evolve rapidly. Quantitatively, traditional models struggle with time series shorter than 6-8 months, leading to overfitting, poor generalization, and inaccurate predictions. The authors note that standard deep learning methods demand "large datasets," but in scarce scenarios, this leads to models that fail to capture underlying patterns, exacerbating inefficiencies in energy management.

The impact of data scarcity is profound and far-reaching. Inaccurate load forecasts can lead to over- or under-provisioning of energy, resulting in increased operational costs, higher carbon emissions from reliance on fossil fuel backups, and potential grid instabilities like blackouts or voltage fluctuations. For utilities, this translates to financial losses; for consumers, it means unreliable service and higher bills due to inefficient pricing. On a broader scale, it hinders the transition to sustainable energy systems by limiting the integration of renewables, which require precise balancing against variable loads. The paper highlights that without addressing this, smart grids cannot fully realize their potential for demand-side management, where forecasts enable real-time adjustments to consumption. Moreover, in few-shot scenarios—where only a handful of data points are available—the reliance on data-hungry models perpetuates a cycle of poor performance, discouraging adoption of advanced analytics in data-limited settings. This impact is amplified in critical applications, such as hospitals or industrial facilities, where forecasting errors could have safety implications.

To resolve the challenges of data scarcity in short-term load forecasting, the paper proposes a meta-learning framework based on an adapted Model-Agnostic Meta-Learning++ (MAML++) algorithm. This enables a base LSTM model to quickly adapt to new, unseen load time series using only a few gradient updates on limited support data. The authors deliberately choose a simple and lightweight architecture to ensure effectiveness in data-scarce environments. The base learner is a modest LSTM recurrent neural network consisting of one LSTM layer followed by a single linear layer with just 32 hidden units. This configuration is used consistently across the proposed method and all baselines to ensure fair comparisons. A key insight from the paper is the preference for simpler models with fewer parameters over more advanced or complicated architectures, particularly under limited data and meta-training tasks. In ablation studies, the authors test increasing the model's depth by adding more linear layers. They find that adding a third linear layer "greatly harms the model’s performance," attributing this to overfitting at the meta-level due to the small number of training tasks, which overwhelms the model's increased capacity. Crucially, they note that this performance drop poses no significant issue because "using fewer linear layers is desirable as it decreases the computational complexity of the final model." This explicitly highlights their encouragement of simpler designs that reduce parameters and computational overhead.

2.3 IISE PG&E Energy Analytics Challenge 2025: Hourly-Binned Regression Models Beat Transformers in Load Forecasting

The paper by Roy et al (2025), focuses on electricity load forecasting—a critical task in the energy sector for maintaining grid stability, integrating renewables, and optimizing capacity planning. The dataset provided includes only two years of historical hourly data on electricity load, temperature, and global horizontal irradiance (GHI) from five sites, along with day-ahead exogenous variables for the test year. Notably, the test set's load values are undisclosed, preventing the use of autoregressive (AR) modeling where past load predictions inform future ones. This setup simulates realistic long-term forecasting challenges, emphasizing a one-year-ahead horizon framed as a regression problem rather than traditional time-series autoregression.

The broader context is the ongoing debate in time-series forecasting, particularly in energy analytics, about the efficacy of advanced deep learning (DL) models versus simpler statistical or machine learning approaches. While DL architectures like transformers have gained popularity for handling complex patterns in large datasets, the authors highlight that real-world energy forecasting often operates under constraints: limited historical data (here, just two years), sparse exogenous features (only temperature and GHI), multicollinearity among variables, and the absence of AR capabilities due to undisclosed test targets. These constraints mimic "realistic data scarcity" in practical deployments, such as emerging smart grids or regions with incomplete metering infrastructure, where vast datasets for training DL models are unavailable. The paper addresses this by reformulating the problem into 24 independent hourly-binned regression models, using Principal Component Analysis (PCA) to reduce dimensionality of exogenous variables (capturing 99% variance with the first component), and incorporating temporal dummies like months, holidays, and weekends.

The study compares a range of models: simpler ones including piecewise linear regression, polynomial regression, Random Forest (RF), and XGBoost; and advanced DL models such as Multi-Layer Perceptrons (MLP), Long Short-Term Memory (LSTM) networks, Gaussian Processes, Transformers, Neural Hierarchical Interpolation for Time Series (NHITS), Temporal Convolutional Networks (TCN), Temporal Fusion Transformers (TFT), and the pre-trained TimeGPT (a large language model adapted for time-series). Experiments involve training on one year and testing on another (cross-year holdouts), as well as 5-fold cross-validation, with metrics like R^2 , Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and symmetric MAPE (sMAPE). This setup tests model robustness under data limitations, including variability between years (e.g., higher skewness in Year 1 load) and the impact of adding lagged or leading features.

The paper strongly supports simpler methods by demonstrating through empirical results that, in scenarios with limited training data and features, simpler models like XGBoost not only match but often surpass advanced DL techniques in accuracy, efficiency, and generalization. This is evidenced in several ways, directly tying back to the constraints of data scarcity.

First, the results show clear performance gaps favoring simplicity. XGBoost achieves the lowest errors across all test cases: R^2 scores up to 0.9, RMSE around 159 – 178, MAPE of 5.5 – 5.9%, and sMAPE of 5.5 – 5.9% in year-specific holdouts. In contrast, advanced DL models like TimeGPT, TFT, NHITS, and TCN exhibit higher errors—RMSE often exceeding 220 and MAPE over 7%—with predictions becoming erratic over long horizons due to error accumulation. For instance, TimeGPT's long-horizon "all-at-once" forecasts degrade significantly by November in the test year, illustrating how DL's reliance on sequential dependencies falters without AR access or abundant data to learn robust patterns. Even pre-trained models like TimeGPT, designed for few-shot

learning, underperform, suggesting that their advantages diminish in highly constrained, domain-specific tasks like this.

The authors attribute this to DL's inherent vulnerabilities under data scarcity: these models require extensive datasets for effective training and are prone to overfitting or poor generalization when data is sparse (only two years here, far below the thousands of samples often needed for transformers). Simpler methods, conversely, thrive with minimal feature engineering—XGBoost uses PCA-transformed features and basic lags (e.g., Lag1 for marginal gains)—and avoid the computational overhead of DL. Training XGBoost takes about one hour, versus hours to days for DL models, making it more practical for real-world applications where resources are limited. Regression baselines like linear models struggle with outliers in cross-validation but still outperform DL in holdouts, reinforcing that complexity does not guarantee better results.

Furthermore, the paper's experiments simulate realistic scarcity by restricting to non-AR setups and testing cross-year shifts, where data distribution changes (e.g., varying load variability) expose DL's sensitivity. Adding features like lags provides only slight improvements (R^2 ~0.85 – 0.86), indicating that even with enhancements, simpler models suffice without the bloat of DL architectures. This supports the argument by showing that in energy forecasting—where data is often incomplete or variable—defaulting to advanced methods can lead to suboptimal outcomes, while tailored, efficient approaches like XGBoost deliver superior accuracy and deployability.

2.4 Electric load forecasting with recency effect: A big data approach

The paper by Hong et al (2016) explores how to enhance traditional regression-based load forecasting by incorporating the "recency effect"—the idea, borrowed from psychology, that recent temperatures have a disproportionate influence on electricity demand beyond current-hour weather. Building on prior work establishing simple multiple linear regression as a robust benchmark for load forecasting, the authors use modern computational power to address a longstanding limitation: how many lagged hourly temperatures and daily moving average temperatures should be included to fully capture this effect without degrading accuracy due to overfitting or noise.

The study is grounded in real-world constraints of electric load forecasting, where temperature is the dominant driver of demand variability, but historical computing restrictions often capped the number of temperature-related variables in models. By treating this as a "big data" problem—testing thousands of potential variables—the paper demonstrates that carefully extending a simple regression framework yields substantial accuracy gains, particularly in hierarchical forecasting scenarios like those in utility operations.

The research extends Tao's Vanilla Benchmark Model, a straightforward multiple linear regression that includes trend, calendar effects (e.g., hour-of-day, day-of-week, holidays), and basic temperature polynomials (e.g., current temperature, squared, and cubed terms). This benchmark has proven competitive in competitions like GEFCom2012, but it overlooks the recency effect: demand responds not just to immediate weather but to temperatures from preceding hours or days (e.g., lingering heat buildup affecting cooling needs).

Historically, practitioners avoided large numbers of lagged variables due to computational costs and risks of multicollinearity or overfitting. The paper poses a direct question: With today's computing

capabilities, what is the optimal number of lagged hourly temperatures (e.g., T-1 to T-n) and 24-hour moving average temperatures to include for maximum recency capture?

The experimental setup uses data from the GEFCom2012 hierarchical load forecasting track: 4.5 years of hourly load and temperature data from 21 zones of a U.S. utility (20 individual zones plus one aggregated total). The first 4 years train models, with rolling forecasts evaluated at both aggregated (top-level) and disaggregated (zone-level) hierarchies. Weather stations are pre-selected based on prior methodology, and parameter estimation uses a 730-day moving window for realism.

Without modeling recency, standard benchmarks like the Vanilla Model miss significant demand drivers, leading to systematic errors—especially during weather transitions or extreme events where recent conditions amplify load (e.g., multi-day heat waves). In hierarchical grids, this compounds: aggregated forecasts may mask zone-specific patterns, resulting in inefficient generation scheduling, higher costs, or reliability risks.

The paper quantifies this gap empirically. At the aggregated level, the Vanilla Benchmark alone achieves reasonable accuracy, but omitting recency leaves 18%–21% relative error reduction untapped. At lower hierarchy levels (individual zones), the shortfall is 12%–15% on average, highlighting how disaggregated forecasting—common in smart grids for localized management—suffers more from unmodeled recent weather influences.

The core solution is a "big data" extension of simple multiple linear regression: systematically add lagged hourly temperatures and daily moving averages, then select the optimal count via trial-and-error validation to balance recency capture and accuracy.

The paper explicitly advocates for “simplicity in model structure”. Enhancements come from data-driven variable inclusion in regression, not advanced architectures (e.g., no neural networks or exponential smoothing discussed in depth). This aligns with realistic constraints: regression is interpretable, computationally efficient (even with thousands of variables tested), and robust to the noisy, hierarchical data typical in utilities. The paper concludes that in GEFCom2012's constrained setup (limited features, noisy real-world data, hierarchical reconciliation needs), adding recency via many variables in regression yields 12%–21% error reductions without resorting to machine learning black boxes. This conclusion echoes broader themes from the competition: traditional regression, with targeted engineering, excels in practical energy forecasting where data is constrained in quality or exogenous richness.

Chapter 3: Fundamentals

This chapter provides brief explanations of concepts related to time series analysis relevant to this paper. Mathematical equations and technical workings behind time series models used are also discussed here. The next chapter builds on the explanations provided here and contains the results of the techniques covered in this chapter after applying them on the Gefcom 2012 dataset.

3.1 Principal Component Analysis (PCA)

Principal Component Analysis is an unsupervised linear mathematical technique that transforms a dataset with many possibly correlated variables into a new set of variables that together represent a new coordinate system (Jolliffe, 2002). The axes of this system, called principal components, satisfy three properties:

1. They are linear combinations of the original variables
2. They are mutually orthogonal directions in the feature space
3. They are ordered by the amount of variance they explain

The goal of PCA is to represent the essential structure of the data using fewer dimensions, while preserving as much variability as possible. PCA assumes the data matrix X is mean-centered and studies how variables vary together by forming the covariance matrix (Σ):

$$\Sigma = \frac{1}{n} X^T X$$

where X is the data matrix.

PCA looks for directions (ω) such that, when all data points are projected onto that direction, the resulting one-dimensional values have maximum variance:

$$\max_{||\omega||=1} \omega^T \Sigma \omega$$

Optimizing this expression leads to an eigenvalue problem in which each eigenvector of Σ defines a principal component direction, and each corresponding eigenvalue measures how much variance that component explains (Hastie et al, 2009). Each principal component captures variance not explained by the previous ones, and the components are ordered from largest to smallest eigenvalue.

Once the principal components are found, data points are expressed in this new coordinate system:

$$Z = X\,V$$

where V is the matrix, whose columns are the principal component directions.

This transformation decorrelates the variables in the transformed space and concentrates most of the variation into the first few components.

Instead of describing each point using the original axes, PCA rotates the coordinate system to better align with the natural shape of the data. In this paper PCA is applied to reduce dimensionality from eleven weather stations, so that a few principal components can capture most of the total variation in temperature.

3.2 Root Mean Squared Error

Gefcom 2012 uses root mean squared error (or RMSE) as the default approach to measure prediction accuracy. The same has been used in this paper as the primary metric to determine forecast error. A forecast "error" is defined as the gap between an observed value and its forecast (Montgomery et al, 2015). In forecasting, a key difference between "residuals" and "error" is that residuals are calculated on the *training* set while errors are calculated on the *test* set. For a τ-step ahead forecast made at time *T*, the forecast error is expressed as:

$$e_{T+\tau} = \; y_{T+\tau} - \hat{y}_{T+\tau|T}$$

RMSE is a scale dependent error, but it lies on the same scale as the series or the forecast variable. In general, scale-dependent errors cannot be used to compare performance between series that involve different units. RMSE for *n* observations is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_T - \hat{y}_T)^2}$$

The squared deviation component particularly makes RMSE sensitive to large outliers.

3.3 Time Series Decomposition

Time series decomposition is a statistical technique used to break down a single time series into its three underlying, distinct components. The general mathematical model for this decomposition is

$$y_t \; = \; f(S_t, T_t, \varepsilon_t)$$

Where,
t (in subscript): index that denotes a specific point in time
S_t: the seasonal component,
T_t: the trend effect and
ε_t: the random error component

Trend (T_t) is the long-term direction of the data. For example, mean of a time series E(y_t) can vary linearly with time (i.e. linear trend):

$$\mathrm{E}(y_t) = \beta_0 + \beta_1 t$$

Seasonality (S_t) in a time series (y_t) represents a periodic pattern that is repeated in the series at fixed intervals in time, such as daily, weekly, or yearly. For example, daily seasonality existing in hourly time series data can be represented for any t as:

$$y_t = y_{t+24}$$

For the time series decomposition described there are usually two forms for the function *f*:

an additive model $y_t = S_t + T_t + \varepsilon_t$ and
a multiplicative model $y_t = S_t T_t \varepsilon_t$

The additive model is seen appropriate if the amplitude of the seasonal variation doesn't vary with the level of the series, while the multiplicative version is seen appropriate if the amplitude of the seasonal variation changes with the average level of the series (Montgomery et al, 2015). The additive model was found appropriate for this paper, as discussed in the next chapter.

3.4 Exponential Smoothing

The kind of Exponential Smoothing used here is also known as additive Holt-Winters' seasonal exponential smoothing or triple exponential smoothing. Since the Holt-Winters' is an additive model, the time series is decomposed as:

$$y_t = S_t + L_t + \varepsilon_t$$

Where,

L_t is the linear trend component (representing the level of the series),

S_t represents seasonal adjustment ($S_t = S_{t+s} = S_{t+2s} \ldots .$ for $t = 1, \ldots, s-1$, where *s* is the length of the seasonal period) and,

the ε_t are assumed to be uncorrelated with mean 0 and constant variance σ_ε^2.

A usual restriction imposed on this model is seasonal adjustments sum to zero in the duration of one season:

$$\sum_{t=1}^{s} S_t = 0$$

Holt Winters' Exponential Smoothing model learns about the series via the procedure of recursive updating. When a new data point (y_t) arrives, the model updates its estimates for the three components using three smoothing parameters: λ_1 (level), λ_2 (trend), and λ_3 (seasonality). Recursive updating happens as follows:

Step 1: Estimate of L_t gets updated using

$$\hat{L}_T = \lambda_1(y_T - \hat{S}_{T-s}) + (1 - \lambda_1)(\hat{L}_{T-1} + \hat{\beta}_{1,T-1})$$

where $0 < \lambda_1 < 1$. The initial part can be seen as the "current" value for L_t (after seasonal adjustment) and the second part as the forecast of L_t based on the estimates made at $T - 1$.

Step 2: Estimate of β_1 gets updated using

$$\hat{\beta}_{1,T} = \lambda_2(\hat{L}_T - \hat{L}_{T-1}) + (1 - \lambda_2)\,\hat{\beta}_{1,T-1}$$

where $0 < \lambda_2 < 1$. Similar to the previous step, the initial part can be seen as the "current" value for of β_1 and the second part as the forecast made for it at $T - 1$.

Step 3: Estimate of S_t gets updated using

$$\hat{S}_T = \lambda_3(y_T - \hat{L}_T) + (1 - \lambda_3)\,\hat{S}_{T-s}$$

where $0 < \lambda_3 < 1$.

Step 4: The τ-step-ahead forecast, $\hat{y}_{T+\tau}(T)$, is made using

$$\hat{y}_{T+\tau}(T) = \hat{L}_T + \hat{\beta}_{1,T}\tau + \hat{S}_T(\tau - \mathrm{s}).$$

Looking at the update equations, one might observe and question how the individual components are determined at $T = 0$. There are multiple ways in which the initial values can be chosen. Generally, the initial values for the level and the trend terms can be obtained by fitting a linear regression model on the data such that time *t* is placed as a regressor. The intercept and the slope can then be used as the initial values of L_t and T_t respectively (Montgomery et al, 2015). However, optimization-based estimation solvers and decomposition heuristics are often favored by software packages such as *statsmodels* (statistical library in Python) for better accuracy.

This paper performs log-transformation on the series, which is discussed in the exploratory data analysis part of the next chapter. This makes:

$$y_t{}' = \log(y_t)$$

and the additive decomposition as,

$$y_t{}' = S_t' + L_t' + \varepsilon_t{}'$$

The recursive updating procedure is similarly applied, with respect to the transformed decomposition equation. Applying the logarithmic identity *log(AB) = log(A+B),* the additive decomposition equation gets reverted back to the original (untransformed) space as:

$$y_t = e^{S'_t}.e^{L'_t}.e^{\varepsilon'_t}$$

3.5 Time Series Regression

Linear Regression Model

A simple linear regression model allows for a linear relationship between the forecast variable *y* and a single predictor variable *x*. Multiple linear regression model is an extension of simple linear regression model in that it allows for more than one or multiple regressors to be part of the model equation:

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_k x_{k,t} + \varepsilon_t$$

The intercept β_0 is the predicted value of *y* at $(x_1, x_2, \ldots, x_k) = 0$ and the slope β_1 represents the average predicted change in *y* that results from increasing x_1 by 1 unit (ceteris paribus). "$\beta_0 + \beta_1 x_t + \cdots + \beta_k x_{k,t}$" represents the explained part of the model and "ε_t" represents the deviation from the systematic part of the model (Hyndman & Athanasopoulos, 2021).

Using a linear regression model, some assumptions about the variables are implicitly made:

- Firstly, the relationship between the forecast variable and the regressors must be linear in parameters and the functional form of the model should reasonably approximate the relationship between forecast variable and regressors in reality.

- Secondly, errors $(\varepsilon_1, \ldots, \varepsilon_T)$ should have zero mean, no autocorrelation, and must be unrelated to predictor variables.

- Errors are ideally normally distributed with a constant variance σ^2

The coefficients $(\beta_0, \beta_1, \ldots, \beta_k)$ are determined using least squares estimation:

$$\min \sum_{t=1}^{T} \varepsilon_t^2$$

With the estimated regression coefficients, the equation for the fitted value at time *t* for the forecast variable is represented as:

$$\hat{y}_t = \hat{\beta}_0 + \hat{\beta}_1 x_{1,t} + \hat{\beta}_2 x_{2,t} + \cdots + \hat{\beta}_k x_{k,t}$$

A common way to summarize how well a linear regression model fits the data (used generally to test fit on training set) is via the coefficient of determination, or R^2:

$$R^2 = \frac{\sum(\hat{y}_t - \bar{y})^2}{\sum(y_t - \bar{y})^2}$$

Where, $0 \leq R^2 \leq 1$ and the summations are done over all observations.
R^2 reflects the proportion of variation in the forecast variable that is explained by the regression model used to fit the data. R^2 being close to 1 indicates that the fitted values are very close to the actual data.

Harmonic Regression

With respect to the time series decomposition discussed earlier, regression models can also be used to capture seasonal and trend components in terms of a sine wave for time series data (Montgomery et al, 2015):

$$\begin{aligned} E(y_t) &= \beta \sin \omega(t+\theta) \\ &= \beta \cos \omega\theta \sin \omega t + \beta \sin \omega\theta \cos \omega t \\ &= \beta_1 \sin \omega t + \beta_2 \cos \omega t \end{aligned}$$

Where,

β: amplitude of the wave
θ: phase angle of the wave
ω: cycle length of the wave
$\beta_1 = \beta \cos \omega\theta$ and,
$\beta_2 = \beta \sin \omega\theta$

For representing the same model into a regression style equation with linear trend, it can be modified by adding an intercept and defining the frequency $\omega = \frac{2\pi}{d}$:

$$E(y_t) = \beta_0 + \beta_t t + \beta_1 \sin\frac{2\pi}{d} t + \beta_2 \cos\frac{2\pi}{d} t$$

Where d is the length of the season and $\frac{2\pi}{d}$ is expressed in radians.

A single sine wave as described here can sufficiently capture single, symmetric seasonal pattern in the data. However, alongside the chosen method to model trend, more complex seasonal pattern can also be captured by adding harmonics of the same fundamental (base) frequency to the core wave. A second harmonic with double the frequency of previous harmonic, added on top of the core wave with linear trend, would appear as:

$$E(y_t) = \beta_0 + \beta_t t + \sum_{j=1}^{2} \left(\beta_j \sin\frac{2\pi j}{d} t + \beta_{2+j} \cos\frac{2\pi j}{d} t\right)$$

The entire idea behind harmonic regression is for added harmonics to interfere with each other. While the core wave provides the basic "up and down" movement, the 2nd harmonic allows the model to "sharpen" or "flatten" those peaks to match the true time signal.

Non-Linear Regression

A foundational assumption in linear regression is that the relationship between forecast variable and predictors should be linear in parameters. If transformation of variables is not able to meet this assumption adequately, piecewise linear regression can be used to break down the non-linear relationship into smaller segments where localized relationship becomes approximately linear.

This is achieved by defining individual *knots*, which are specific points introduced on the predictor variable in question that allow the slope of the linear segments to change as we move across these points. This paper identifies two knots on the temperature variable that help simplify the non-linear relationship between load (y_t) and temperature (*temp*) into localized linear patterns. Regression equation that comprises temperature variable as predictor with two knots is:

$$E(y_t) = \beta_0 + \beta_t t + \beta_1 \max(0, T_h - temp_t) + \beta_2 \max(0, temp_t - T_c)$$

Where T_h is 55°F (1st temperature knot) and T_c is 65°F (2nd temperature knot)

The two knots divide the temperature variable into three portions with their own slope coefficients. For the region where temperature lies in the range: 55 < *temp* < 65, $E(y_t)$ is estimated to be constant value of "$\beta_0 + \beta_t t$".

Complete Time Series Regression Model

This paper uses temperature as one of the primary predictor variables and uses log-transformation on the forecast variable. The final time series regression model developed in this paper combines multiple regression with linear trend, harmonics and piece-wise segmentation of the temperature variable.

The fitted value generated by the final time series regression model can be expressed as:

$$E(\log(y_t)) = \beta_0 + \beta_t t + \beta_h \max(0, T_h - temp_t) + \beta_c \max(0, temp_t - T_c) + \sum_{j=1}^{2} \left(\beta_j \sin \frac{2\pi j}{d} t + \beta_{2+j} \cos \frac{2\pi j}{d} t \right)$$

where,

y_t = Load at time *t*
$temp_t$: temperature (in °F) at time *t*
T_h : 55°F (1st temperature knot)
T_c : 65°F (2nd temperature knot)
β_t : slope coefficient for linear trend
β_h : slope coefficient for region where *temp* < T_h
β_c : slope coefficient for region where *temp* > T_c
β_1, β_3 : coefficients for the sine and cosine components of 1st harmonic
β_2, β_4 : coefficients for the sine and cosine components of 2nd harmonic

3.6 ACFs and PACFs

In time series analysis, ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) are diagnostic tools used to identify the internal structure of the time series. They help determine how past values influence the present.

The Autocorrelation Function (ACF) measures the total correlation between a time series (y_t) and a lagged version of itself (y_{t-k}), where *k* can take up multiple positive values representing different lags. If we are looking at the ACF at $k = 3$, it measures the correlation between y_t and y_{t-3}. The correlation measured here includes both the direct effect and the indirect effect. For example, y_{t-3} might affect y_{t-2} which might affect y_{t-1} and ultimately y_t.

Partial Autocorrelation Function (PACF) specifically measures the direct correlation between y_t and y_{t-k} after removing the effects of all shorter lags ($0 < k < 3$, for the mentioned example). PACF *controls* for the influences from other lags, leaving only the *pure* relationship between y_t and y_{t-k}.

Before determining ACF and PACF plots, it is often better to check if the underlying series is stationary. Non-stationary series renders distorted ACF and PACF plots, making it difficult to properly identify the underlying Autoregressive or Moving Average processes (Montgomery et al, 2015).

The autocorrelation coefficient at lag *k* for a stationary time series is:

$$\rho_k = \frac{\mathrm{Cov}(y_t, y_{t+k})}{\mathrm{Var}(y_t)}$$

Typical behavior of the ACF and PACF for stationary ARMA processes are summarized in Table 3.1. This table can be helpful in correctly identifying the orders of the AR and MA components, according to Montgomery et al (2015).

Model	**ACF**	**PACF**
MA(q)	Cuts off after lag q	Exponential decay and/or damped sinusoid
AR(p)	Exponential decay and/or damped sinusoid	Cuts off after lag p
ARMA(p,q)	Exponential decay and/or damped sinusoid	Exponential decay and/or damped sinusoid

Table 3.1: Behavior of ACF and PACF for stationary ARMA processes

3.7 Stationarity and Augmented Dickey-Fuller Test

A series exhibiting stationarity (or 'weak' stationarity) behaves in the two following ways (Montgomery et al, 2015):

- The expected value of the time series is independent of time
- $\mathrm{Cov}(y_t, y_{t+k})$ for any lag k is only a function of k and not time

While stationarity is defined by the time-invariant properties of the mean and autocovariance (Montgomery et al., 2015), it is formally tested using unit root tests such as the Augmented Dickey-Fuller (ADF) test. The ADF test evaluates the null hypothesis (H_0) that the series possesses a unit root (i.e. non-stationary and non-seasonal structure).
The Dickey-Fuller (DF) test only works if the noise is "white noise". The "Augmented" (ADF) test is used to cater to the problem of autocorrelation found in real world data. In other words, the augmented version of the test accounts for higher-order serial correlation by including lagged differences of the series, ensuring that the resulting test statistic is not biased by residual autocorrelation (The MathWorks, Inc., 2025).

For a sufficiently small p-value of the ADF test statistic (such as < 0.05 at 95% level of significance) 'H_0' can be safely rejected, indicating that the series under consideration does not contain a unit root.

3.8 Seasonal ARIMA (SARIMAX)

Time series models are generally expressed as a sum of two distinct components, deterministic component and stochastic component. The deterministic part captures predictable patterns such as trends and seasonality, while the stochastic part accounts for random noise added on to the deterministic signal. An assumption often made in time series modelling is that random noise arises out of *independent* shocks to the process i.e. errors are assumed to be 'white noise'. However, this assumption is often not met in practice due to the presence of serial dependence in successive observations (Montgomery et al, 2015). ARIMA models are specifically designed to model this serial dependence to improve forecasting. ARIMA is broken down into Autoregressive (AR) and Moving Average (MA) components that cater to serial dependence in observations, while the Integration (I) component handles the deterministic part of the model.

<u>Moving Average Models</u>

Moving average (MA) process is appropriate in cases of a stationary series exhibiting autocorrelation, especially the type of autocorrelation where the serial dependence becomes abruptly non-significant after some lag q. Moving Average process expresses the current value of forecast variable (y_t) to be linearly dependent on the mean of the series, the current error term, and past error terms.

$$MA(q): \qquad y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$

where,

q: order of MA process
μ: mean of the series
ε_t: current error term
ε_{t-q}: past error term
θ: coefficient of past error term

The error terms are assumed to be 'white noise' and be mutually independent and normally distributed. The order q of the MA process determines how many past error terms influence the present value of the forecast variable (Peixeiro, 2022). MA(q) process is always *weakly* stationary, for any value of θ (Montgomery et al, 2015). An extension of MA(q), known as infinite Moving Average or MA(∞), where $q \to \infty$ also exists. However, it implicitly requires estimation of infinite number of weights which is practically infeasible.

Autoregressive Models

In a MA(q) process, only a finite q number of past errors contribute to the current value of the time series. However, there might also be series where the underlying structure is better explained by error terms that occur way back in the past, similar to the MA(∞) case. Autoregressive processes are a way to model such series without requiring estimation of infinitely many weights.

Autoregressive process is appropriate in cases of a stationary series exhibiting autocorrelation, especially the type of autocorrelation where the serial dependence on past lags lingers much longer than a MA process and follows a distinct pattern. AR process expresses the present value of forecast variable (y_t) to be linearly dependent on its past values.

$$AR(p): \qquad y_t = C + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \epsilon_t$$

where,

p: order of AR process
C: constant term
ε_t: current error term
y_{t-p}: past value of series
φ: coefficient of past value of series

The error terms are assumed to be 'white noise' and be mutually independent and normally distributed. Similar to MA(q), the order p of the AR process determines how many past values influence the present value of the forecast variable.

The way AR process is able to model lingering serial dependencies with only a handful of parameters can be illustrated by showing the mathematical equivalence of a stationary AR(1) process and a MA(∞) process.

MA(∞) process generated from AR(1) can be depicted as:

$$y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_\infty \varepsilon_{t-\infty}$$

Following Montgomery et al. (2015), θ is used here to illustrate the geometric decay implied by a stationary AR(1) process. If $|\theta| < 1$, such that θ follows an exponential decay pattern, MA(∞) can be re-written as:

$$y_t = \mu + \varepsilon_t + \underbrace{\theta\varepsilon_{t-1} + \theta^2\varepsilon_{t-2} + \cdots + \theta^{\infty}\varepsilon_{t-\infty}}_{\theta y_{t-1} - \theta\mu}$$

$$= \underbrace{\mu - \theta\mu}_{C} + \theta y_{t-1} + \varepsilon_t$$

$$= C + \theta y_{t-1} + \varepsilon_t \qquad \text{(Which is nothing but AR(1) process)}$$

The $|\theta| < 1$ condition ensures that the AR(1) process is stationary and θ decay in a distinct exponential pattern (Montgomery et al, 2015). The stationarity condition in AR models is essential because it ensures that model parameters don't explode with time but remain stable. Higher order autoregressive models (with additional stationarity conditions) can also be depicted in terms of MA(∞) process.

This paper makes use of AR(2) process that makes up a dominant component of the SARIMA model discussed further. AR (2) process can be expressed as:

$$y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \epsilon_t$$

or,

$$\underbrace{(1 - \phi_1 B - \phi_2 B^2) y_t}_{\Phi(B) y_t} = C + \epsilon_t$$

where B is a linear backshift operator such that $By_t = y_{t-1}$ and $\Phi(B)$ is the lag polynomial.

This process can be represented as a MA(∞) process through recursive substitution, similar to what was seen in the AR(1) process. AR(2) one period back can be written as:

$$y_{t-1} = C + \phi_1 y_{t-2} + \phi_2 y_{t-3} + \epsilon_{t-1}$$

$$\implies y_t = C + \phi_1(C + \phi_1 y_{t-2} + \phi_2 y_{t-3} + \epsilon_{t-1}) + \phi_2 y_{t-2} + \epsilon_t$$

It can now be seen that y_t depends on both ϵ_t and ϵ_{t-1}. Substituting every past value of series in terms of a linear combination of older past values will eventually replace every single y term with an infinite string of past ϵ shocks. A more formal representation for the same idea is to consider the AR(2) expression with the backshift operator:

$$\underbrace{(1 - \phi_1 B - \phi_2 B^2) y_t}_{\Phi(B) y_t} = C + \epsilon_t$$

$$\implies y_t = \underbrace{\Phi(B)^{-1} C}_{=C'} + \underbrace{\Phi(B)^{-1}}_{=\Psi(B)} \epsilon_t$$

$$\implies y_t = C' + \Psi(B)\epsilon_t$$

$$\implies y_t = C' + \sum_{i=0}^{\infty} \psi_i \, \epsilon_{t-i}$$

$$\Longrightarrow y_t = C' + \sum_{i=0}^{\infty} \psi_i B^i \epsilon_t$$

The weights ψ_i are determined by the roots of the equation $\Phi(B) = 0$. For the autoregressive process to be mathematically stable, the impact of past errors must eventually disappear, and the sum of weights applied to past errors (ψ_i) must converge to a finite number. Otherwise, the variance of the process would not be finite, resulting in the series exploding to infinity (Montgomery et al, 2015). The requirements for desired ψ_i behavior are met when the roots (r_1, r_2) of the aforementioned equation $1 - \phi_1 B - \phi_2 B^2 = 0$ lie outside the unit circle (i.e. $|r| > 1$). The following three stationary conditions on model parameters ϕ_1, ϕ_2 are sufficient to ensure that $|r| > 1$:

$$\phi_1 + \phi_2 < 1,$$

$$\phi_2 - \phi_1 < 1, \text{ and}$$

$$|\phi_2| < 1$$

Mixed Autoregressive-Moving Average (ARMA) process

The primary reason to use a mixed ARMA process instead of a high-order pure AR or MA process is parsimony i.e. the principle of achieving a good fit with the fewest parameters possible (Montgomery et al, 2015). High-order models are prone to overfitting and tend to carry penalties in terms of lower AIC scores. An ARMA(p,q) be generally denoted as:

$$y_t = C + \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t - \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

To see why AR-MA mix is useful, consider the AR(1) model. In this model, the coefficients of the infinite moving average representation are constrained to an exponential decay, with ϕ_1 determining the rate of decline. However, if the actual weights (ψ_i) of past errors do not always follow a simple exponential pattern, while still remaining square-summable for stationarity, a basic AR(1) model will fail to capture the underlying structure. For example, the first weight of past error (ψ_1) could be an anomaly that deviates from the distinct decay pattern. To address this, one could increase the autoregressive order (p) to better approximate the unique behavior of ψ_1. Alternatively, adding a single MA(1) term allows adjustment for that specific anomaly without disrupting the overall decay pattern. This approach is more parsimonious because it allows for flexible adjustments to the exponential decay without the need for an excessively high-order model.

According to Montgomery et al, the AR component of an ARMA process determines the stationarity of the ARMA process. Similar to the AR(2) case, if all roots of the associated lag polynomial equation $\Phi(B) = 0$ lie outside the unit circle ($| r | > 1$), then ARMA(p, q) is stationary and be expressed in terms of MA(∞) process.

Integration Component ("I") of ARIMA process

Standard ARMA models assume the data is stationary. However, many real-world time series are non-stationary for specific reasons. For example, the underlying series may exhibit homogeneous

behavior over time but may not have a constant average level due to the presence of a trend component. A time series y_t can be called a homogenous nonstationary series if it's not stationary, but its first difference $\{w_t = y_t - y_{t-1} = (1-B)y_t\}$ or higher order differences $\{w_t = (1-B)^d y_t\}$ produce stationary time series. Differencing transforms homogenous nonstationary series into a stationary form, allowing the autoregressive (AR) and moving average (MA) parts to model patterns effectively. A homogenous nonstationary series is known as AutoRegressive Integrated Moving Average process of orders *p,d* and *q*, i.e. ARIMA(*p,d,q*), if its d^{th} difference produces a stationary ARMA(p,q) process.

Seasonal ARIMA (SARIMA)

SARIMA(p,d,q)$\times$(P,D,Q)$_s$ builds on standard ARIMA by adding seasonal components to handle recurring seasonal patterns at period s (e.g., s=24 for daily periodicity).

D (Seasonal Differencing):
Analogous to the non-seasonal "I" (*d*), seasonal differencing removes seasonal non-stationarity. It applies $(1-B^s)^D$ to the series y_t. This subtracts values from *s* periods ago (e.g., $y_t - y_{t-24}$) stabilizing the seasonal average level while preserving underlying homogeneous short-term behavior, just as regular differencing does for trends.

P (Seasonal Autoregressive Order):
The seasonal AR component captures dependence on past values at seasonal lags (multiples of *s*). Similar to the non-seasonal AR(*p*) part in the ARMA process, which determines long-term stationarity via roots of the characteristic lag polynomial, the seasonal AR(*P*) imposes structure on seasonal autocorrelations, ensuring the overall process remains stationary after differencing.

Q (Seasonal Moving Average Order):
The seasonal MA component models dependence on past errors at seasonal lags. Like the non-seasonal MA(*q*), which provides parsimony by allowing flexible adjustment to anomalies in error weights without requiring high AR orders, seasonal MA(*Q*) efficiently captures deviations from simple seasonal patterns (e.g., unusual shocks repeating daily) while keeping the model parameter-efficient.

Together, after non-seasonal (*d*) and seasonal (*D*) differencing produce a stationary series, the combined ARMA(*p,q*) and seasonal ARMA(*P,Q*)$_s$ terms parsimoniously model both short-term and seasonal dynamics in a unified mixed framework.

SARIMAX: Seasonal ARIMA with Exogenous Regressors

SARIMAX extends the Seasonal ARIMA framework to incorporate exogenous variables (such as temperature) that influence the time series but are independent of its own past values or errors. Just as Seasonal ARIMA builds on standard ARIMA by adding seasonal AR(*P*), differencing (*D*), and MA(*Q*) components to handle periodic non-stationarity, SARIMAX integrates these with regression terms for the exogenous inputs:

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_k x_{k,t} + (SARIMA\ terms)$$

This allows the model to add linear dependencies on exogenous variables to explain additional deterministic variation, while maintaining the parsimonious mixed ARMA approach for stochastic serial dependence. The errors remain white noise, improving forecast accuracy in real-world applications where the series is influenced by both its past and known external drivers.

3.9 Auto ARIMA and Akaike Information Criterion (AIC)

Auto ARIMA is a function provided by *pmdarima* library in Python that automatically selects the optimal parameters for the SARIMA model. This automation eliminates the manual, time-consuming process of using statistical plots (such as ACF and PACF plots) and tests to determine the best parameters (Smith, 2022).

For a SARIMA(p,d,q)(P,D,Q)[m] model, Auto ARIMA works as follows:

- It performs an exhaustive grid search or stepwise selection across a range of potential p, d, q and P, D, Q values that are user-defined.
- For each combination of parameters, it fits a SARIMA model and calculates the Akaike Information Criterion (AIC)
- It then selects the model that yields the lowest AIC value, as lower values indicate a better fit that balances model's goodness of fit with model complexity.

The Akaike information criterion (AIC) is a model selection method that measures the quality of one model in relation to the other (Peixeiro, 2022). AIC is a function of the number of parameters k in a model and the maximum value of the likelihood function $\hat{L}$:

$$\text{AIC} = 2k - 2\log(\hat{L})$$

The higher orders of (p,q) or (P,Q) directly lead to an increase in k (model complexity). A higher value of the likelihood function indicates a better goodness of fit on the data. A model quality is evaluated to be better if increased complexity (potentially increasing AIC) results in such an increase in the goodness of fit on the data that the overall AIC reduces. This is the reason why a model with a lower AIC value is preferred.

Chapter 4: Methodology & Results

4.1 Gefcom 2012 Load Forecasting Dataset

The competition provides load history data (hourly frequency in kW) ranging from the 1st hour of 2004/1/1 to the 6th hour of 2008/6/30 for 20 zones. To help with forecasting, hourly temperature readings (in Fahrenheit) from 11 weather stations have also been provided, covering the same time range as the load history data.

Given the actual temperature readings, the 8 weeks of data belonging to the time periods below (format: yy-mm-dd) for load history are set to be missing and are required to be filled-in using a load estimation approach:

- 2005/3/6 - 2005/3/12;
- 2005/6/20 - 2005/6/26;
- 2005/9/10 - 2005/9/16;
- 2005/12/25 - 2005/12/31;
- 2006/2/13 - 2006/2/19;
- 2006/5/25 - 2006/5/31;
- 2006/8/2 - 2006/8/8;
- 2006/11/22 - 2006/11/28;

For the forecasting challenge, participants need to forecast hourly loads at a zonal level from 2008/7/1 to 2008/7/7. For this time duration, no actual temperature readings are provided. Hence, they must also be estimated in case participant's modelling approach relies on temperature readings for hourly load estimation.

For the entire duration of time that the dataset covers, no information on spatial proximity either related to zones or temperature stations has been provided. As additional information, the competition provides a list of public holidays in the US for the covered time duration that can be used by the participant if it helps improve error reduction. Finally, the benchmark load values for the time periods corresponding to missing load values as well as the forecasting horizon are present in the dataset as well.

4.2 Exploratory Data Analysis

4.2.1 Periodic Pattern in Load Values, Monthly basis

To get an understanding of how load values are spread across time for all the 20 zones, the load history is plotted at a zonal level. The plots for the first three zones are plotted in figure 4.1 below, while plots for remaining zones can be seen in appendix A.1. Complete time duration from 1st hour

of 2004/1/1 to the 6th hour of 2008/6/30 is covered. Each segment between two vertical columns in the plot grid represents a duration of six months. Loess smoothed curve is also added to obtain a visual depiction about underlying load movement across time.

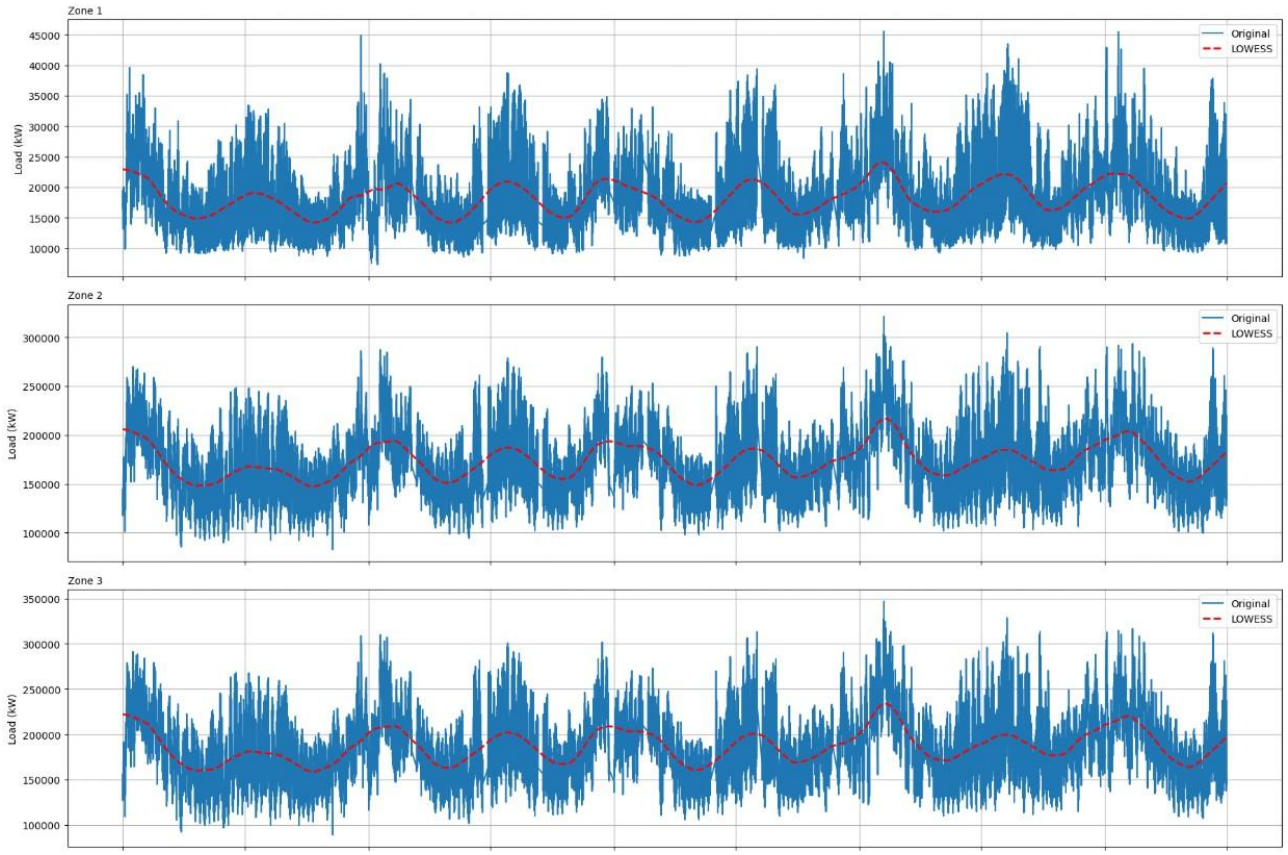


Figure 4.1: Hourly Load from 2004 — 2008 (in kW), by Zone

There appears to be no significant trend in the data throughout most zones. Changes in the average level of the series appear to arise from seasonality, hence the series are non-stationary. Considering variance, there does not appear to be a clear increase or decrease of load variance with time. However, the vertical spread is noticeably larger during the seasonal peaks, and it is noticeably smaller during the seasonal troughs, suggesting heterogeneity in variance. While seasonality can exist at multiple levels, a consistent pattern exists across zones where load values continue rising at the beginning of every year till the end of the first month or the beginning of the second month. After this peak, they begin to continuously drop down to reach a trough around the month of March-April. They then begin to continually rise again from this period onwards to reach a second peak around the month of July-August. From this month load values again continually drop to reach a second trough around the end of September. Finally, the values continually rise past the end of the year. This consistent pattern is repeated year-on-year and is more clearly visible in figure 4.2 which compares the average monthly load values across years, for Zone 1.

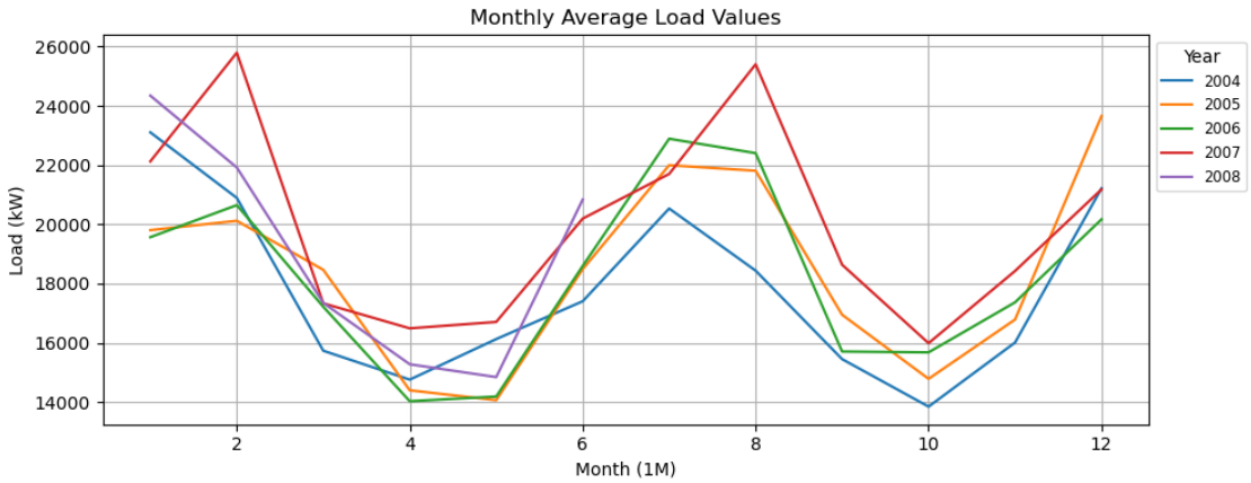


Figure 4.2: YoY Average Monthly Load (in kW), Zone 1

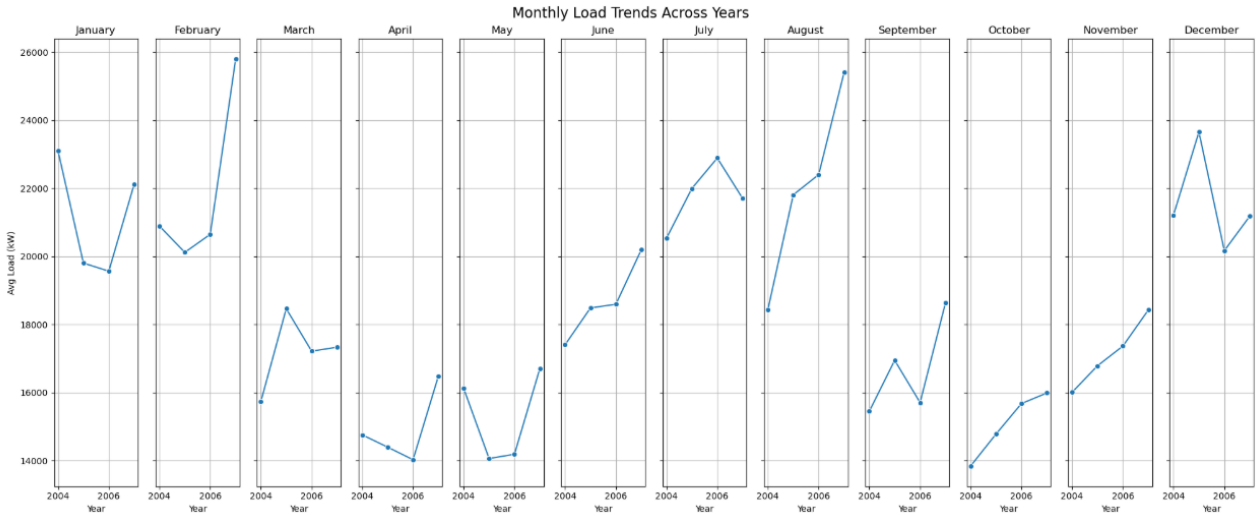


Figure 4.3: Average Monthly Load Trend by Month, Zone 1

The average monthly load values across years for all zones can be seen in appendix A.2. Seasonality pattern can also be seen from the seasonal mini plot for the monthly load trends across years (zone 1) in figure 4.3. Lowest loads consistently appear in spring months (Mar–May) while highest loads cluster in summer (Jun–Aug) and again in winter (Dec–Feb). When it comes to load variance, shoulder months (i.e. transitional periods in a year) such as April and May exhibit less volatility compared to summer months that show larger swings in load. Since peak demand appears in summer months with highest volatility, it is likely the duration of year that should be carrying the dominant planning concern.

4.2.2 Periodic Pattern in Load Values, Hourly basis

Zooming-in at load distribution on an hourly basis, differences in load shape are seen between summer and non-summer months. All summer months tend to follow a particular rhythm as depicted in figure 4.4, suggesting that load behavior is likely linked with ambient temperature:

Overnight low: ~2–5 AM,
Morning ramp: starts ~6–9 AM
Afternoon peak: ~3–6 PM
Evening drop: after ~7 PM

Across all summer months the peak always happens in the late afternoon, likely arising from high cooling demand. However, not all summer months are equal. July–August period has the highest peaks and steepest ramps while May and September (transition months) are noticeably softer.

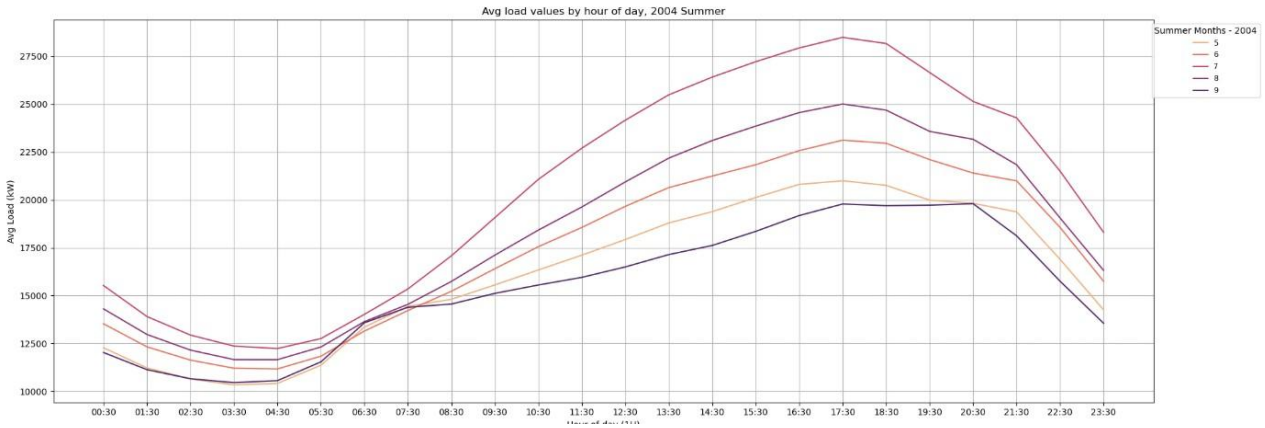


Figure 4.4: Monthly Average Hourly Load Distribution, Summer (Zone 1 - 2004)

For non-summer months on the other hand, the load shape is structurally different, as depicted in figure 4.5. Non-summer days are typically bimodal, meaning that there exists a morning peak around 7–9 AM and a later evening peak around 6–9 PM. This is unlike summer, which had a single dominant afternoon peak. Load demand for non-summer periods mimics classic workday and residential heating/lighting pattern.

Additionally, the load gap between night load and daytime peak aren't as pronounced as they are in the summer months. This suggests that load behavior, likely arising from heating demand, is less sensitive to temperature change and more sensitive to daily human schedules.

The distinct behavior of average hourly loads between summer and non-summer months is valid throughout all years present in the dataset. The plots for some other years can also be seen in appendix A.3 and A.4.

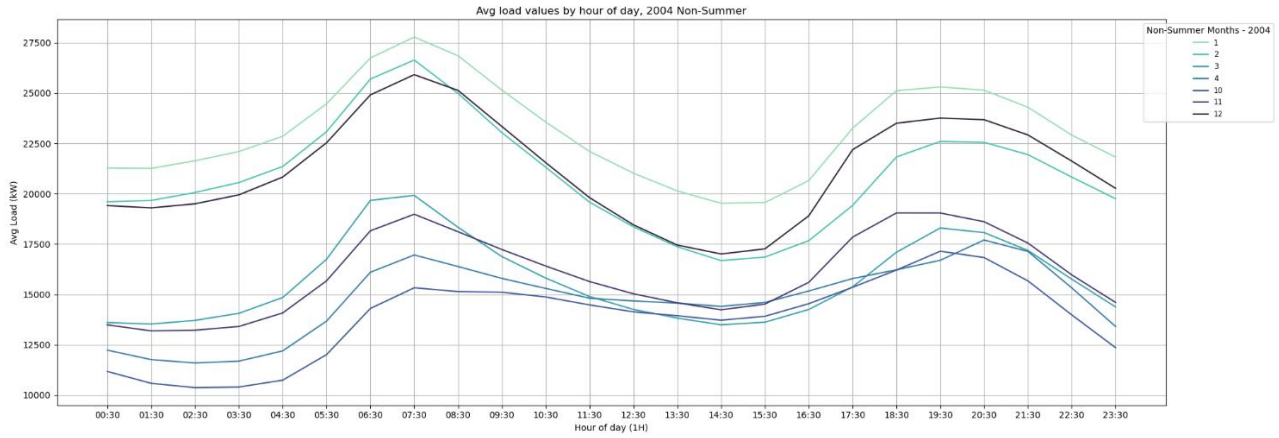


Figure 4.5: Monthly Average Hourly Load Distribution, Non-Summer (Zone 1 - 2004)

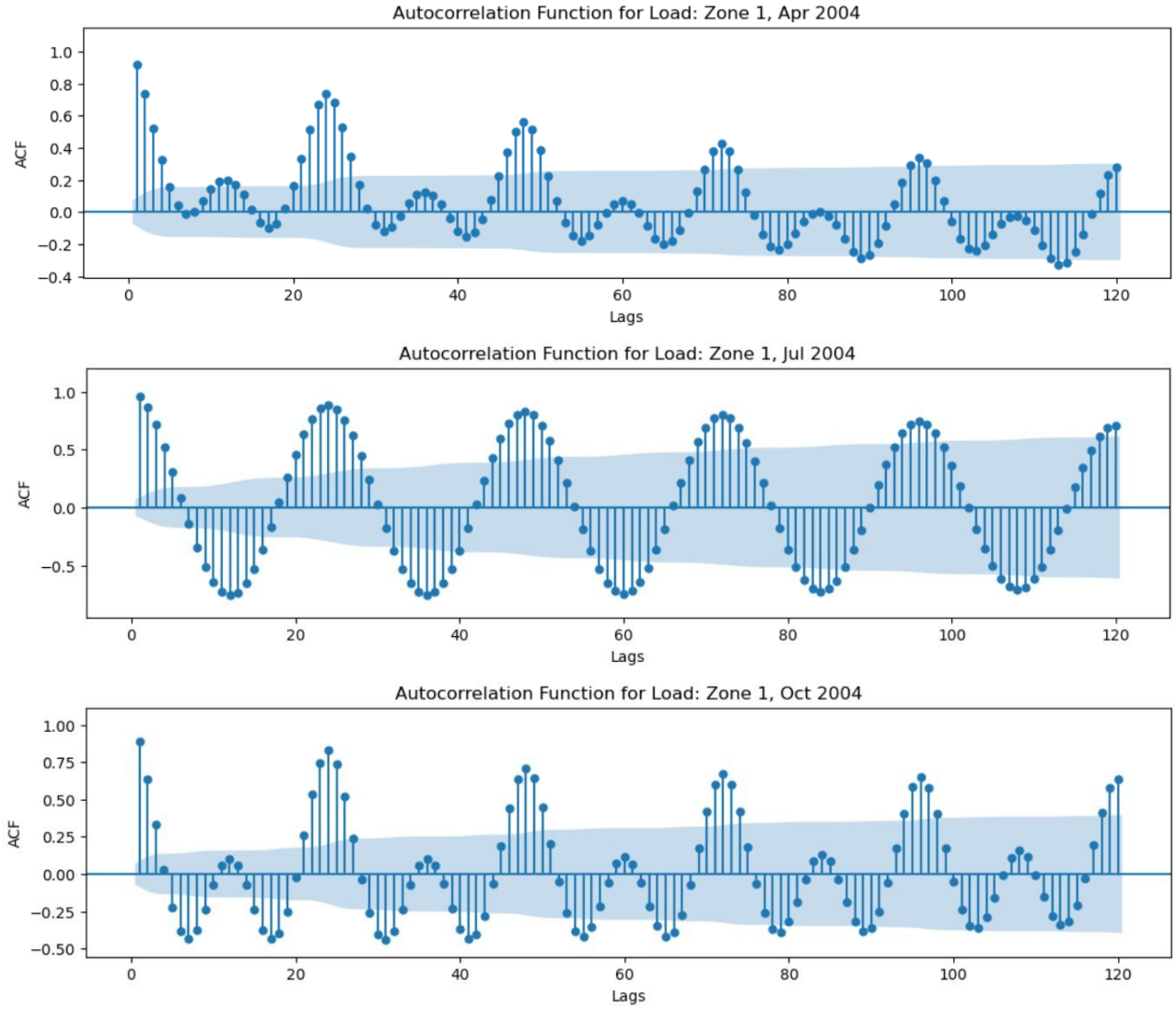


Figure 4.6: Monthly Autocorrelation Function for Zone 1, 2004

Besides load shape, it is seen that load values exhibit serial correlation with persistent 24-hour (i.e. daily) seasonality. This can be seen in the ACF plots i.e. figure 4.6 for April, July and October

months of 2004, for Zone 1. While ACF plots are more meaningful when series are first converted into a stationary series, it is nonetheless used here as a visual tool. “Zone 1” ACF plots for other portions of 2004 and 2005 can be found in appendix A.5. Daily seasonality is seen across all zones, throughout the entire range of time. It is very much possible that after the load series are made stationary, the sinusoidal shape and the correlation decay pattern of every subsequent lag end up changing. However, from the correlation spikes that are significantly outside the confidence intervals and that occur at lags being multiples of “24”, existence of daily periodic pattern can be inferred. The inconsistent autocorrelation structure for different months also indicates the original series being non-stationary.

4.2.3 Zonal Level Correlation

Correlation can be calculated to see how close load distribution in one zone of the grid is to all the other zones. This exercise is done by plotting a correlation matrix with heatmap, as shown in fig 4.7. The Pearson correlation coefficients can inform the strength and direction of the linear relationship, independent of scale.

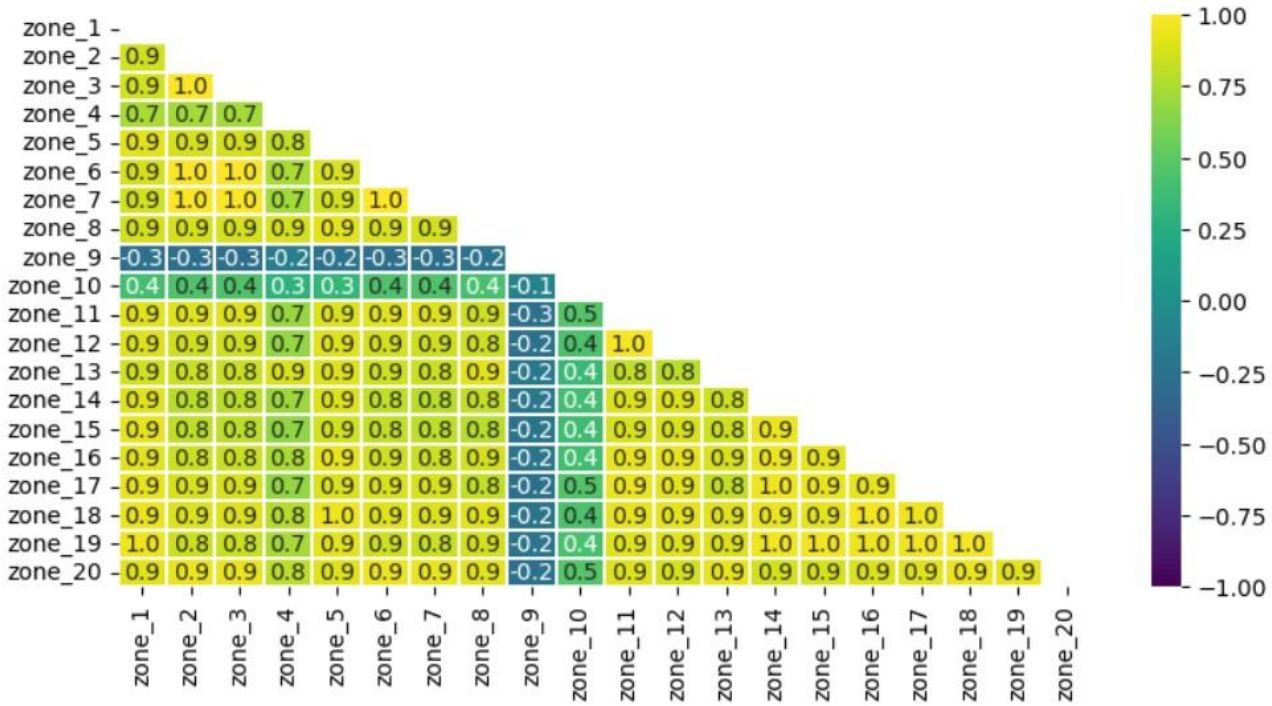


Figure 4.7: Correlation Matrix for All Zones

Except for zones 9 and 10, all zones show very high positive correlation with each other. In other words, load movement across most of the zones is highly synchronized. The most distinct load movement is seen in zone 9, which shows low negative correlation with all other zones. Zone 10 does appear to have positive correlation, but the strength of correlation isn’t as strong, suggesting load movement in this zone is not as synchronized with the rest of the system as the majority of other zones. Hence, conclusions made about the load behavior in one zone should be strongly applicable to the rest of the system.

4.2.4 Correlation Between Temperature Stations

For the entire eleven individual temperature stations, a correlation matrix is similarly plotted to observe how similar temperature distribution is one station compared to all others. This can be seen in fig 4.8. Since all stations show a perfect positive correlation with each other, there is no point in treating all stations as separate sources, since they will only provide redundant temperature information. Instead, a single common temperature signal can be extracted from all these stations.

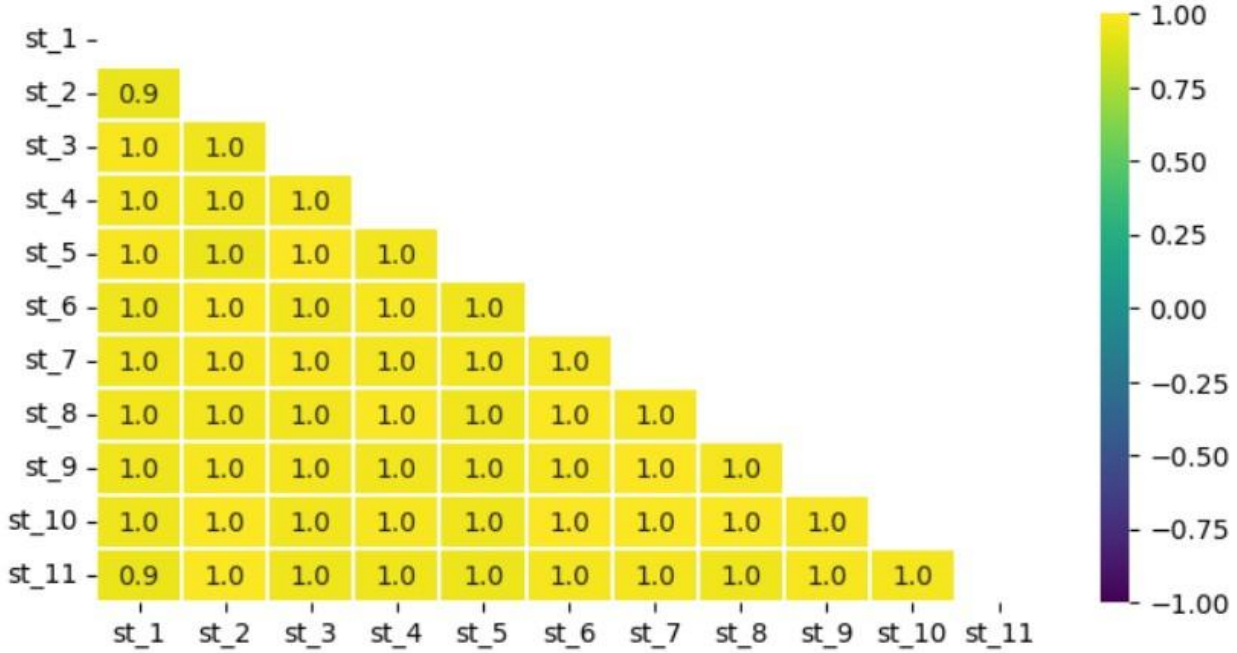


Figure 4.8: Correlation Matrix for All Temperature Stations

4.2.5 Load-Temperature Relationship

Temperature information has been provided to help understand load variation better and possibly help with generating better load predictions. Figure 4.9 depicts the non-linear load-temperature relationship in terms of a scatterplot. The U-shaped pattern occurs across zones, no matter which temperature station is picked. Disintegrating the relationship on a monthly basis reveals a systemic pattern that is persistent across years. It is found that for the duration of year between June and August, i.e. summer period, the load-temperature relationship is linearly positive (depicted in figure 4.10). On the other hand, for the periods January – March and November – December the relationship is linearly negative (depicted in figure 4.11). For the shoulder months where cold-to-hot transition (April – May) and hot-to-cold transition (September – October) occur annually, the relationship becomes ambiguous as shown in figure 4.12. Additional plots for the three distinct load-temperature patterns can be seen in appendix A.6.

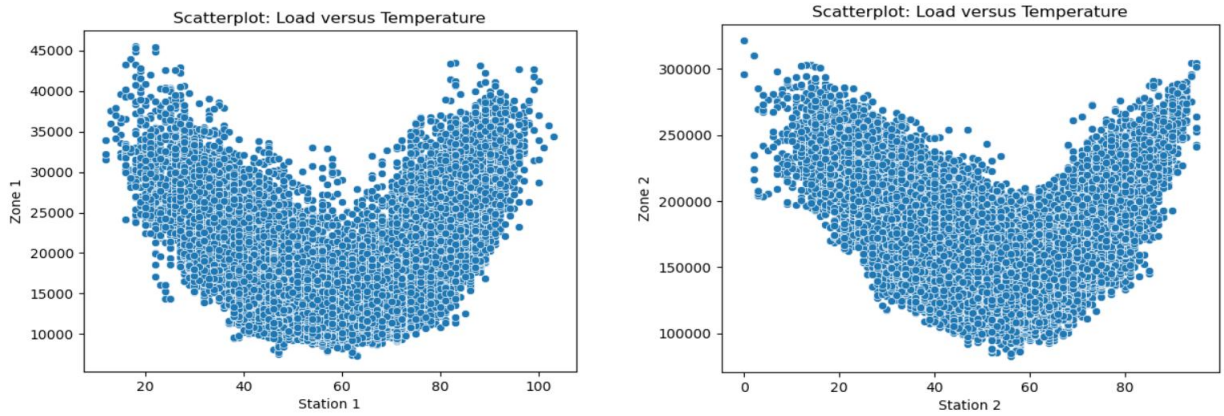


Figure 4.9: Load-Temperature relationship

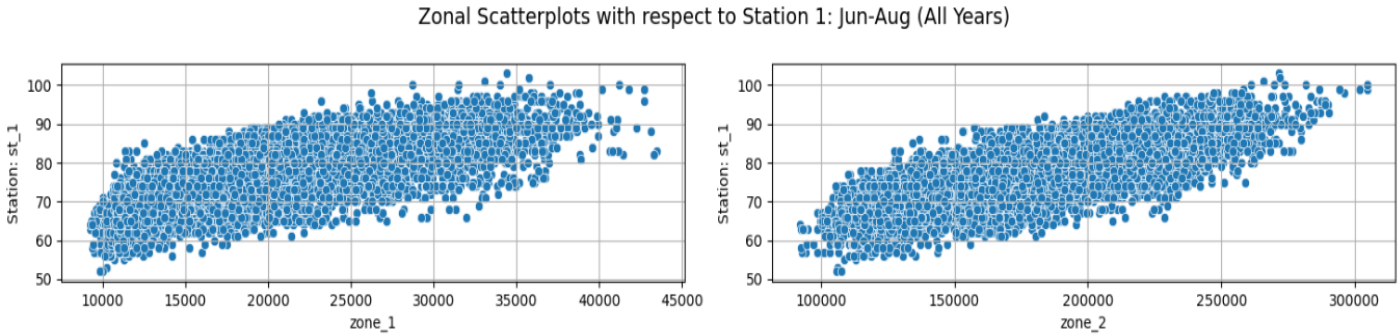


Figure 4.10: Load-Temperature relationship in Summer Period

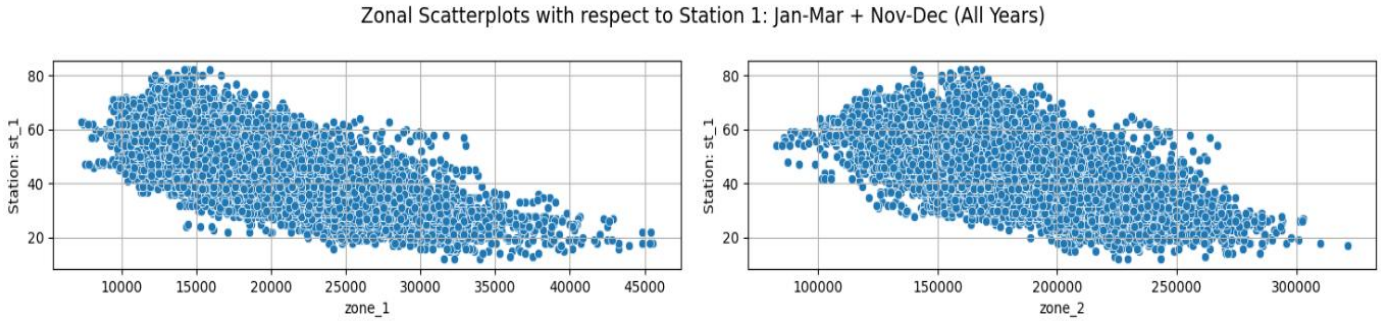


Figure 4.11: Load-Temperature relationship in Non-Summer Period

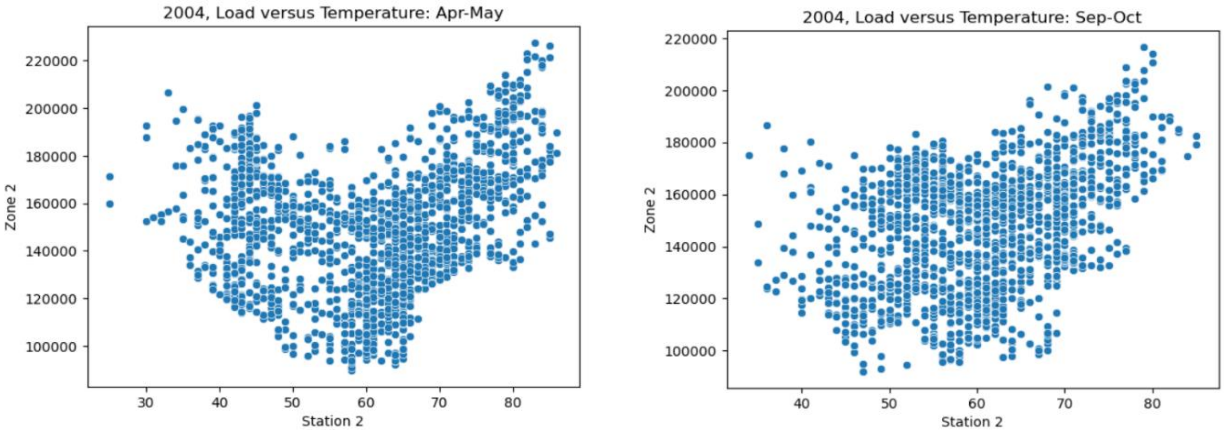


Figure 4.12: Load-Temperature relationship in Shoulder Months

4.3 Model Evaluation and Comparison

To identify the most effective load estimation techniques on the Gefcom 2012 dataset, the following classical time series models have been considered:

- Exponential Smoothing
- Time Series Regression
- SARIMA (seasonal ARIMA)
- SARIMAX (seasonal ARIMA with exogenous variables)

Each of these models are compared to a baseline model to assess whether or not there are significant improvements over considering a simple naïve forecasting technique. Out of four naïve methods implemented, the best performing one is considered to be the baseline model. These four naïve methods are discussed further in detail. Because load movement is highly synchronized across most zones, performance of load estimation techniques is compared only on Zone “1” with the assumption that the findings should hold good across the entire grid system. Also, since load movement follows a consistent annual pattern each year, the analysis is performed only for the year 2004, assuming that findings should hold good for other years lying ahead in time. In particular, load series for the months of January, April, July, Oct and December are considered individually by month as they spread evenly across the year 2004 and represent different seasons of the year in entirety.

For every aforementioned month of 2004, the duration of last 7 days of the month is kept as test set and the remaining portion of the month (approximately 3 weeks) is kept as training set. The average RMSE over these five training sets is taken to be the performance measure for each load estimation technique. However, load variable is not used in its original state. Instead, “log” transformation is performed to compress very high values, reduce outlier influence and to make the distribution more symmetric. Both the original and transformed distribution of the variable are plotted in figure 4.13.

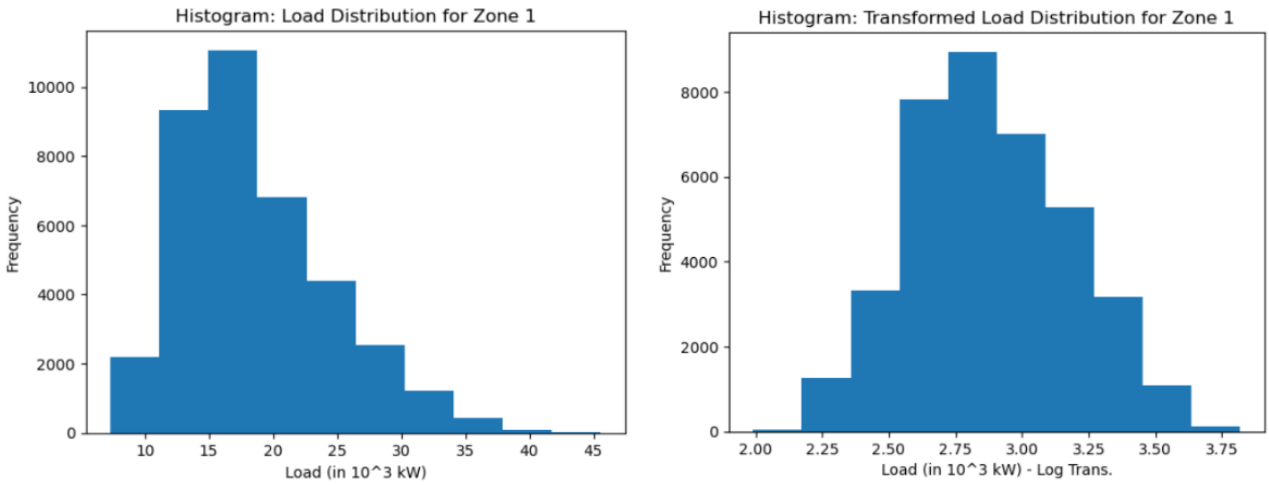


Figure 4.13: Load Distribution for Zone 1 (All Years) – Original and Log Transformed

4.3.1 Treatment for Temperature

In the previous section, almost perfect positive correlation among temperature stations was seen. It was shown that extracting a single temperature signal from the stations, instead of using station readings can be more useful. This is achieved using Principal Component Analysis (PCA) which generates linear combinations of readings from 11 stations per timestamp, i.e. principal components, that captures maximum variance in system temperature.

Station	Temperature Variance
1	279.6953842020234
2	303.0187789065871
3	309.18441857963165
4	302.2389678346821
5	309.5590831777797
6	291.3627644284706
7	322.2223836627818
8	315.2284430195869
9	320.9430675178908
10	302.8390846571321
11	320.689151646415

Table 4.1: Variance of Temperature Readings, by Station

Variables or features with exceptionally larger range can have a proportionally larger share in total variance. This mandates the use of normalization to allow each variable to contribute fairly towards building principal components. Since variance for all stations is more or less similar, lying between the range of 280-320 as shown in table 4.1, normalization is not required and is therefore not performed. Moreover, since the first principal component explains more than 97% of total temperature variance (figure 4.14), weights from the PC1 are used and other principal components are not considered.

The sum of weights from PC1 did not add to 1, so re-scaling is performed on weights to achieve this. The original and re-scaled weights can be found in appendix A.7 and A.8. Using the re-scaled weights, a common temperature signal has been extracted that is used further in models that require temperature information for load estimation.

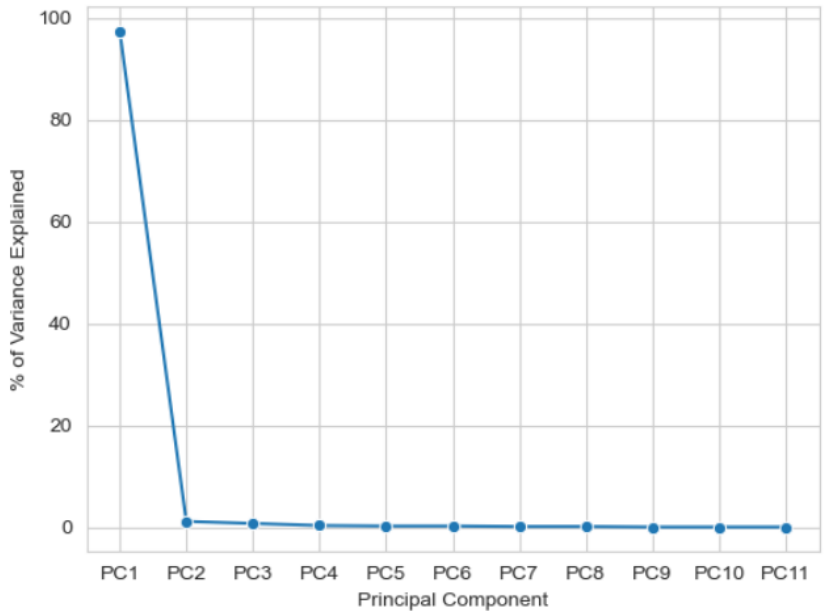


Figure 4.14: Percentage of Total Variance Explained, by Principal Components

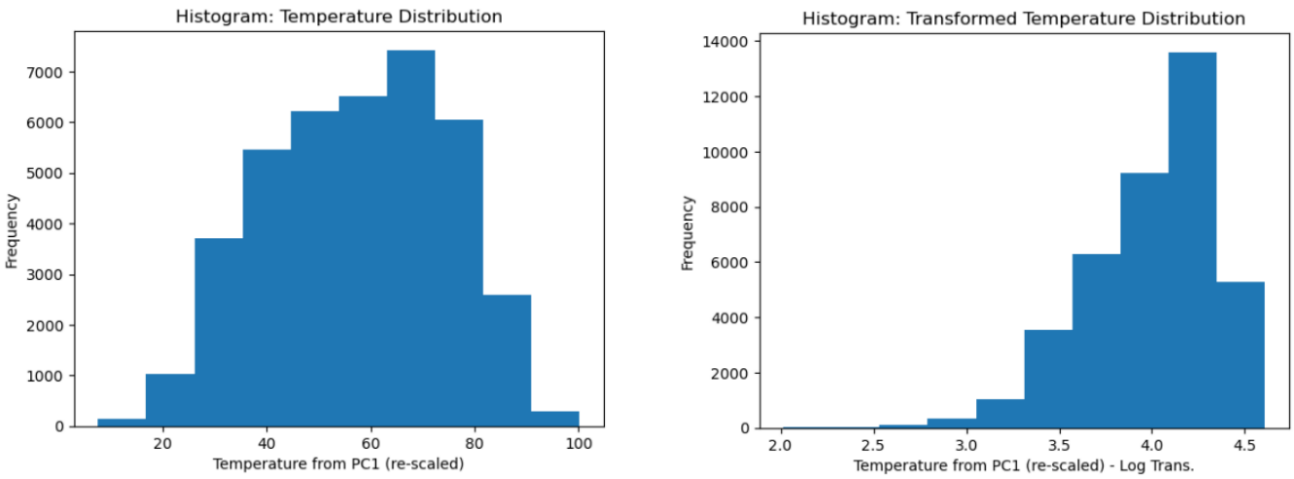


Figure 4.15: Temperature Distribution (Rescaled Weights) – Original and Log Transformed

Log transformation on the distribution of the extracted load signal was considered, to check for any improvements. Both the temperature with re-scaled weights and the log transformation of this series has been plotted in figure 4.15. Since the transformation makes the temperature distribution highly skewed, the temperature series is left untouched.

4.3.2 Naïve Methods

The first naïve method used is just to repeat the load values of the last week, prior to the testing period, as predicted values for the test set. The second method is to use the hourly load values of the day prior to the testing period as predicted values for each day of the testing period. This is also known as seasonal naïve where values of the last season (seasonal length being "24 hours") are used as predictions for the forecasting horizon. The third method is to take the mean of hourly load for the week prior to the testing period as predicted load for all hours of day in the testing period. And finally, the fourth method considered is to take average load at a specific hour to be the predicted value for that hour in testing set. The hourly average here is taken over the week prior to the testing period.

The overall performance on test set for each method can be seen in table 4.2. The best performing naïve methods are found to be the first and fourth methods, i.e. taking "hourly average" for specific hour of day or repeating the load values of previous week (last cycle). The forth method has the lowest average RMSE and the highest average R2 score on the test set, closely followed by the first method. For all four naïve methods the RMSE and R2 scores for each test set, by individual month, can be found in appendix A.9.

Method	Average RMSE	Average R2 Score
Repeating Last Cycle	0.168	0.108
Seasonal Naïve	0.184	-0.050
Overall Average Load (Week Prior)	0.217	-0.145
Hourly Average Load (Week Prior)	0.149	0.340

Table 4.2: Performance Comparison of Naïve Methods

4.3.3 Exponential Smoothing

The length of the training period here is the standard duration i.e. first three weeks of month, with the training set being the same i.e. last week of specified months of 2004. However, to define the Holt Winter's method properly, seasonal decomposition of each training session is performed.

The seasonal decomposition plot for the first training session in January 2004 is shown in figure 4.16. The seasonal component shows a very consistent oscillation and constant amplitude over time, indicating the Holt–Winters with seasonality to be the appropriate model. The amplitude of seasonality does not appear to be proportional to the average level of the series, so an additive Holt Winter's model should be preferred. Additionally, the seasonal pattern repeats cleanly and consistently with each repeating cycle comprising 24-hour periods. The trend component shows the

existence of gradually changing trend with no sharp breaks. However, there is no clear signal of the trend being particularly strong or significant. Hence, a damped trend should be added to avoid over-extrapolation. The residual plot aims to explain part of the series that is left uncaptured by the trend and the seasonal components. It is seen that residuals are roughly centered around zero and are roughly exhibiting a constant variance. There appears to be no systematic structure or patterns such as funnel or curve shapes in the residuals plot. It can be inferred that trend and seasonality are capturing the underlying structure in the data.

The findings from the seasonal decomposition plot of the first training session appear to hold good for the remaining training sessions. To have a deeper look at load behavior for each training session, seasonal decomposition plots for the remaining training sessions can be found in appendix A.10.

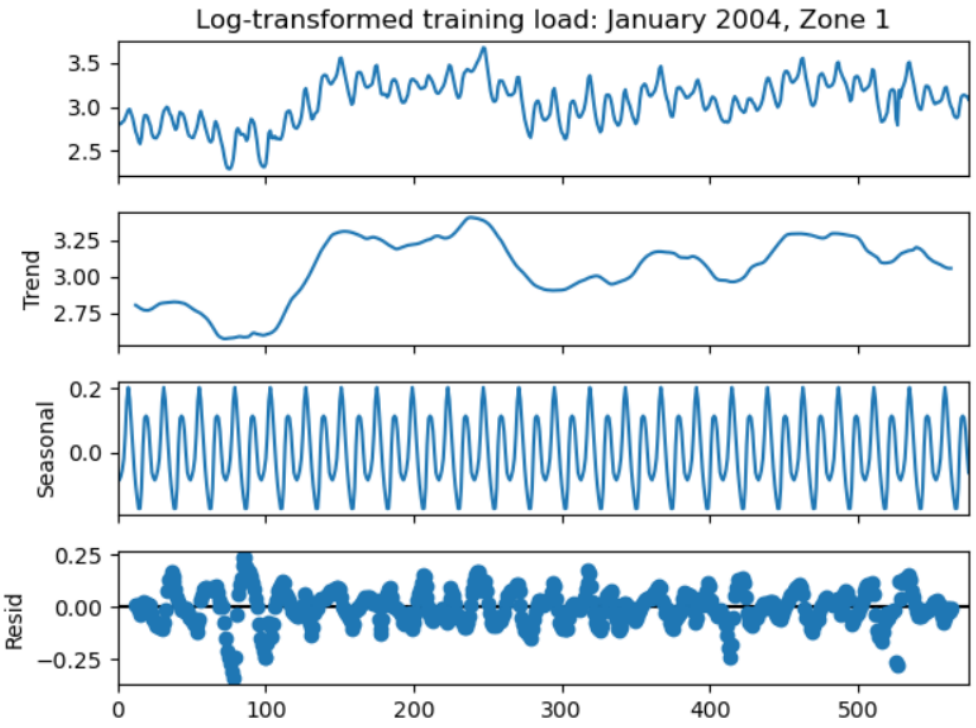


Figure 4.16: Seasonal Decomposition Plot for first training session: Jan 2004, Zone 1

The specified Holt Winter's method for Exponential Smoothing is performed on the first training session and the fit of the model on the training data is observed in addition to the generated forecast. In figure 4.17, it can be seen that the fitted values track the observed data very closely. For the forecasted portion, the seasonal pattern appears to be captured appropriately and there is no unreasonable explosion of the trend component too.
The findings from the "observed versus fitted values" plot for January 2004 are again found to be applicable to the remaining test sessions. The "observed versus fitted values" plot for the remaining test sessions can be seen in appendix A.11.

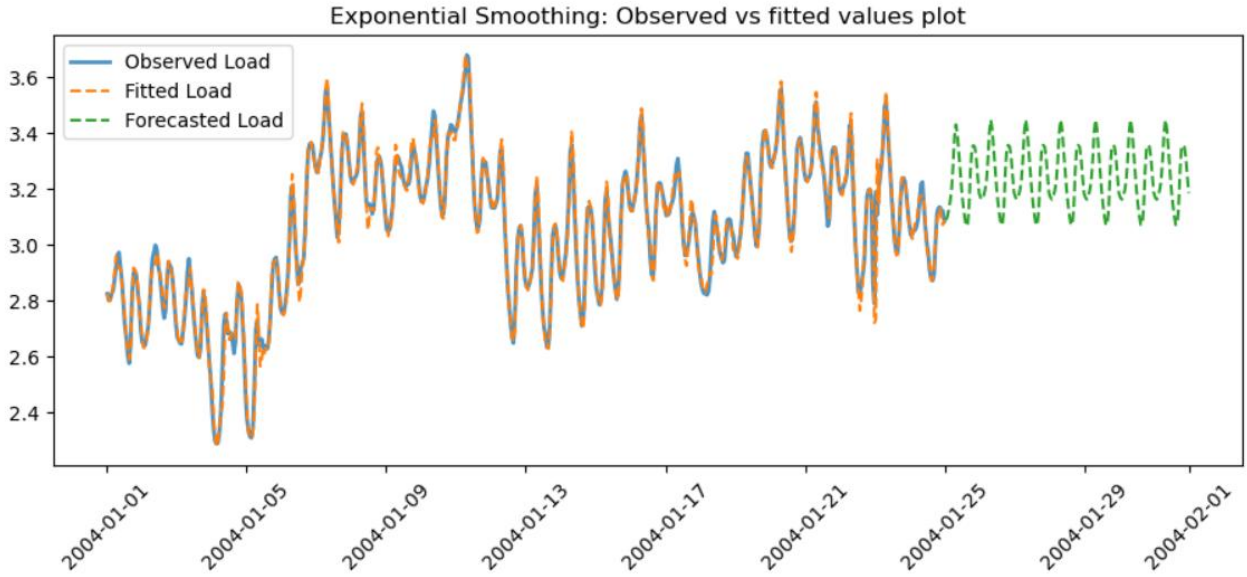


Figure 4.17: Observed versus fitted values plot for January 2004, Zone 1

The overall performance on test set for each month corresponding to training can be seen in table 4.3. Compared to the best naïve method, i.e. load averaged by hour from week prior to testing, Holt-Winter's Exponential Smoothing performs worse. The most notable areas are winter months such as December and January, where forecasting performance is particularly poor.

Testing Month (2004)	**R2 Score**	**RMSE**
January	0.114	0.164
April	0.594	0.117
July	0.843	0.111
October	0.421	0.127
December	-3.976	0.483
	Average Score: **- 0.4**	Average Score: **0.2**

Table 4.3: Exponential Smoothing performance on test set across 2004, by month

4.3.4 Time Series Regression Methods

The non-linear relationship between load and temperature was visually depicted in the exploratory data analysis section. This section begins with a simple load estimation technique, i.e. linear regression model with temperature as the explanatory variable. As the section progresses, more advanced modelling techniques are used to account for the observed non-linear relationship as well as the seasonal variations in load. Here again, the last week of the selected months of 2004 are used as test sets, while the remaining portion of the months are used as training sets.

4.3.4.1 Simple Linear Regression

Table 4.4 covers the performance for simple linear regression model for the test sessions across 2004. This method does perform better than Exponential Smoothing, but it still falls short when compared to the best naïve approach. Multiple "observed versus fitted values" plots, such as the one for April 2004 training set in figure 4.18, reveal that the model is struggling to capture the daily seasonal variation as well as the variation in average level of the series at certain portions of the year. This result is in line with what was found in the exploratory section of the paper, where the relationship between load and temperature was found to be ambiguous during periods of "shoulder months". The performance observed in some other months such as July, where load-temperature relationship was found to be clearer, is much better.

Testing Month (2004)	R2 Score	RMSE
January	0.302	0.145
April	-0.371	0.216
July	0.571	0.183
October	-0.008	0.168
December	0.727	0.113
	Average Score: **0.244**	Average Score: **0.165**

Table 4.4: Simple linear regression performance on test sets across 2004, by month

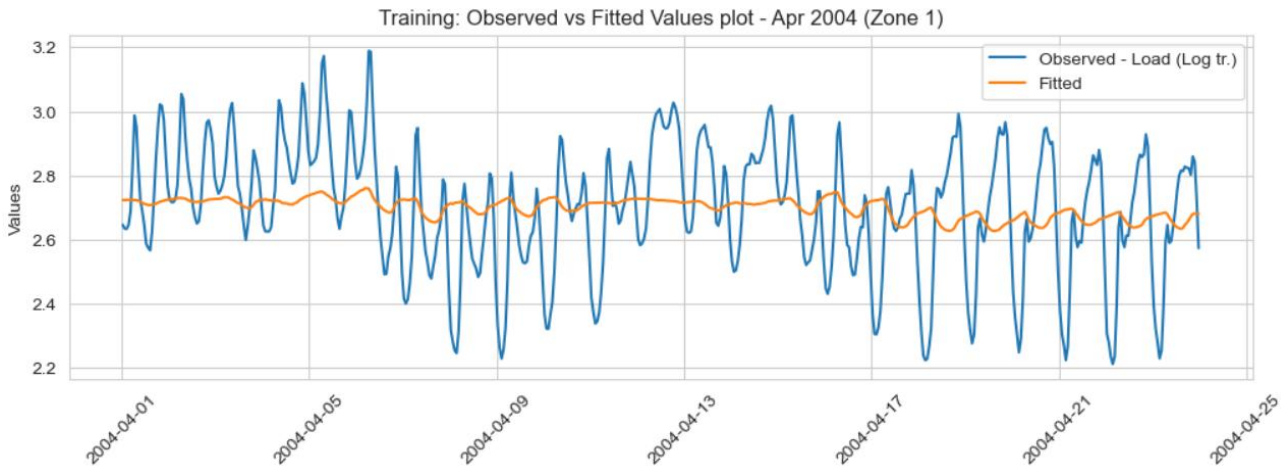


Figure 4.18: Observed vs fitted values plot for training set in April 2004, Simple Linear Regression

4.3.4.2 Harmonic Regression

Fourier series has been used to capture the hourly seasonality in load, without introducing any temperature information. Since daily seasonality is observed in load, the periodicity is specified to comprise 24-hour periods. It is possible that some other seasonal pattern spanning over a larger duration of time might also be present. However, since the training is done only on 3 weeks of data approximately with the goal of performing short term forecasting of load, these other forms of seasonality patterns are not accounted for in the harmonics generated. To accommodate changes in

the average level of the series, a linear trend term has also been included alongside the Fourier components. What is more essential to determine is the number of harmonics that can accurately capture the daily seasonal variation in load. Having more harmonics can capture local variations in load at a granular level, however, it can also lead to overfitting if the captured local patterns are not generalizable on a weekly or a monthly scale.

By comparing one harmonic only with cases comprising up to four harmonics on all individual training sets for specified months, the case of two harmonics was found to explain the variation in data sufficiently well. The comparison table of R2 scores used to reach this conclusion is included in appendix A.12

Table 4.5 covers the performance of harmonic regression with two harmonics on the test set, for specified months of 2004. Harmonic regression performs worse than simple linear regression with temperature, suggesting that temperature information is essential for improving load estimation. The fit of the model on one of the training sets can also be seen in figure 4.19 for a visual illustration. The two harmonics are able to replicate daily periodicity pattern in load but are still unable to match load movement precisely.

Test Set in Month	R2 score	RMSE
January, 2004	-4.182	0.397
April, 2004	-0.322	0.212
July, 2004	0.652	0.165
October, 2004	0.747	0.084
December, 2004	-1.300	0.328
	Average: **-0.881**	Average: **0.237**

Table 4.5: Harmonic regression performance on test sets across 2004, by month

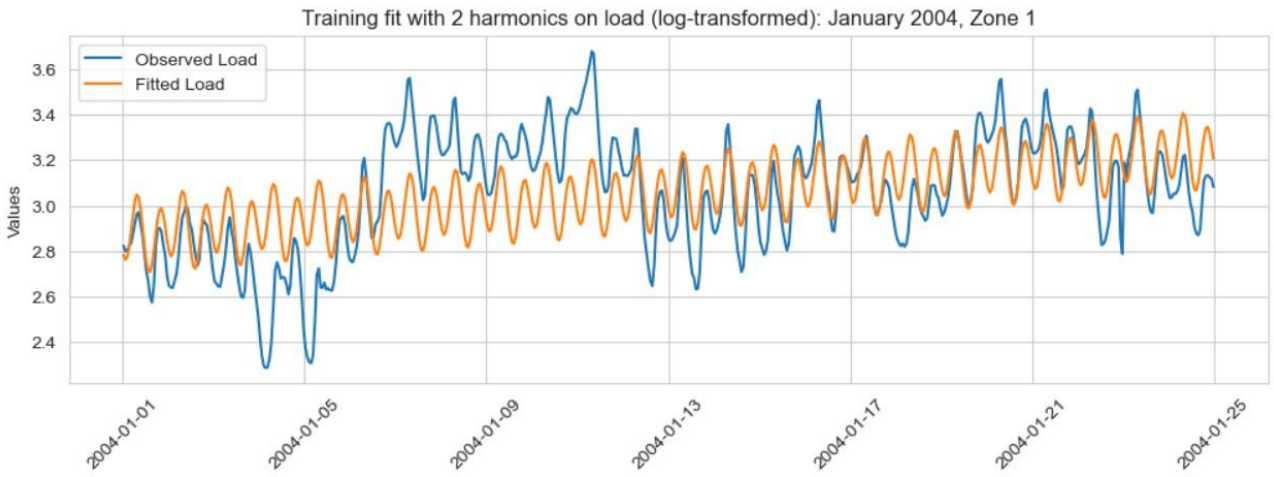


Figure 4.19: Observed vs fitted values plot for Jan training, Harmonic regression (2 harmonics)

4.3.4.3 Harmonic regression combined with Temperature

For matching load movement more precisely, temperature information is added as explanatory variable to the previously covered two harmonics approach. However, a linear load-temperature relationship is still assumed in this multiple regression model. Based on overall performance shown in table 4.6, the combined harmonic regression is the best time series regression method so far. Comparing the fit of the model on the same training set used in figure 4.19 for simple harmonic regression, combined harmonic regression now follows the load movement more precisely as shown in figure 4.20. However, for the month of April i.e. a shoulder month the performance on test set is still poor, likely due to the non-linear relationship between temperature and load that can't be properly captured by the existing model.

Test Set in Month	R2 score	RMSE
January, 2004	0.511	0.122
April, 2004	0.309	0.153
July, 2004	0.795	0.127
October, 2004	0.794	0.076
December, 2004	0.774	0.102
	Average: **0.636**	Average: **0.116**

Table 4.6: Combined harmonic regression performance on test sets across 2004, by month

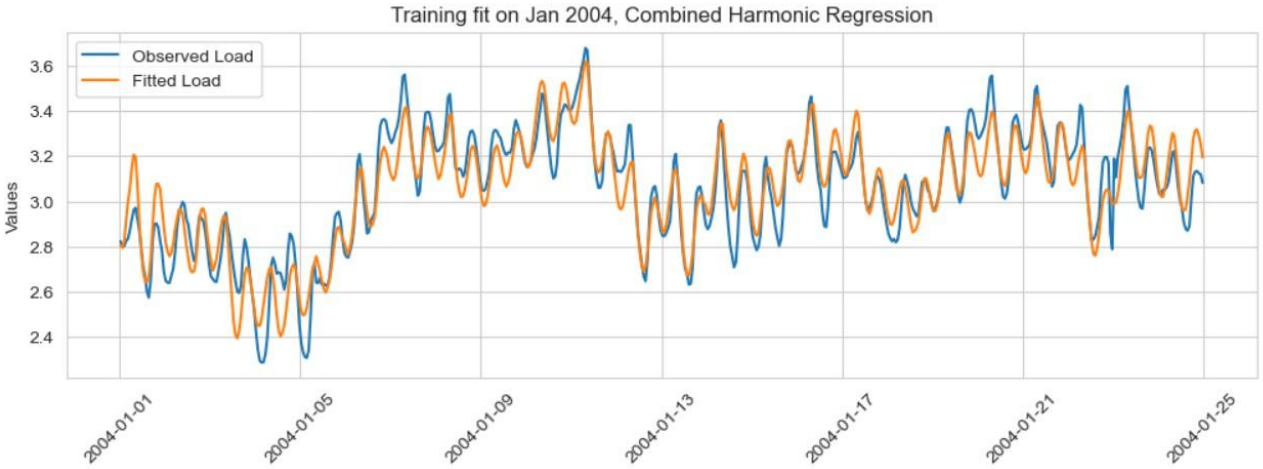


Figure 4.20: Observed vs fitted values plot for Jan 2004 training, Combined harmonic regression

4.3.4.4 Combining Piecewise Linear Regression with Harmonics

The combined harmonic regression mentioned previously performed well but did not account for the non-linear relationship between temperature and load. It is reasonable to assume that if the combined harmonic regression model is able to capture the non-linear relationship, improvement in prediction performance should be observed. Piecewise linear regression is a way to accommodate for this relationship by introducing "knots" or breakpoints on the temperature variable. These knots allow the variable to be broken down into linear segments so that a unique straight line (who's slope is estimated by the regression model) can be fit into each of the linear segment. However, what is essential is to know the number of knots to use and where to place them in the temperature variable so that the relationship with load is appropriately captured.

Exploratory data analysis section discussed that the relationship between load and temperature was positive in summer months, negative in non-summer months and ambiguous in shoulder months. Another look at this dynamic can be seen in figure 4.21 where a loess smoother (red curve) is used to reduce visual noise and make the pattern more clearly visible. The U-shaped pattern is most clear from the loess curve for shoulder months (i.e. April and October), with loess curves for other months appearing to be a straight line bent into two distinct linear segments. Load for temperatures in 55-65 °F range appear to be very close in value, roughly staying constant for most months. Below 55°F the load-temperature relationship can be well represented by a linear line for all months, the slope of which can be positive or negative depending on the season. The same can be said for the portion of the scatterplots where temperatures are higher than 65°F. Hence, two knots (one at 55°F and the other at 65°F) can be a good way to model load behavior for different temperature regions. They allow the creation of three segments that can capture the "U" shape with three distinct slopes along with the flexibility to adjust well to other shapes that can also be represented by two line segments.

A plausible explanation for the load-temperature pattern in multiple months could be as follows: In winter months, "heating" demand kicks-in below 55°F, where load increases rapidly with drop in temperatures to maintain heat. In summer months, "cooling" demand kicks-in above 65°F, where load increases rapidly with increase in temperatures to avoid over-heating. For the temperature range 55-65°F load movement is likely month specific.

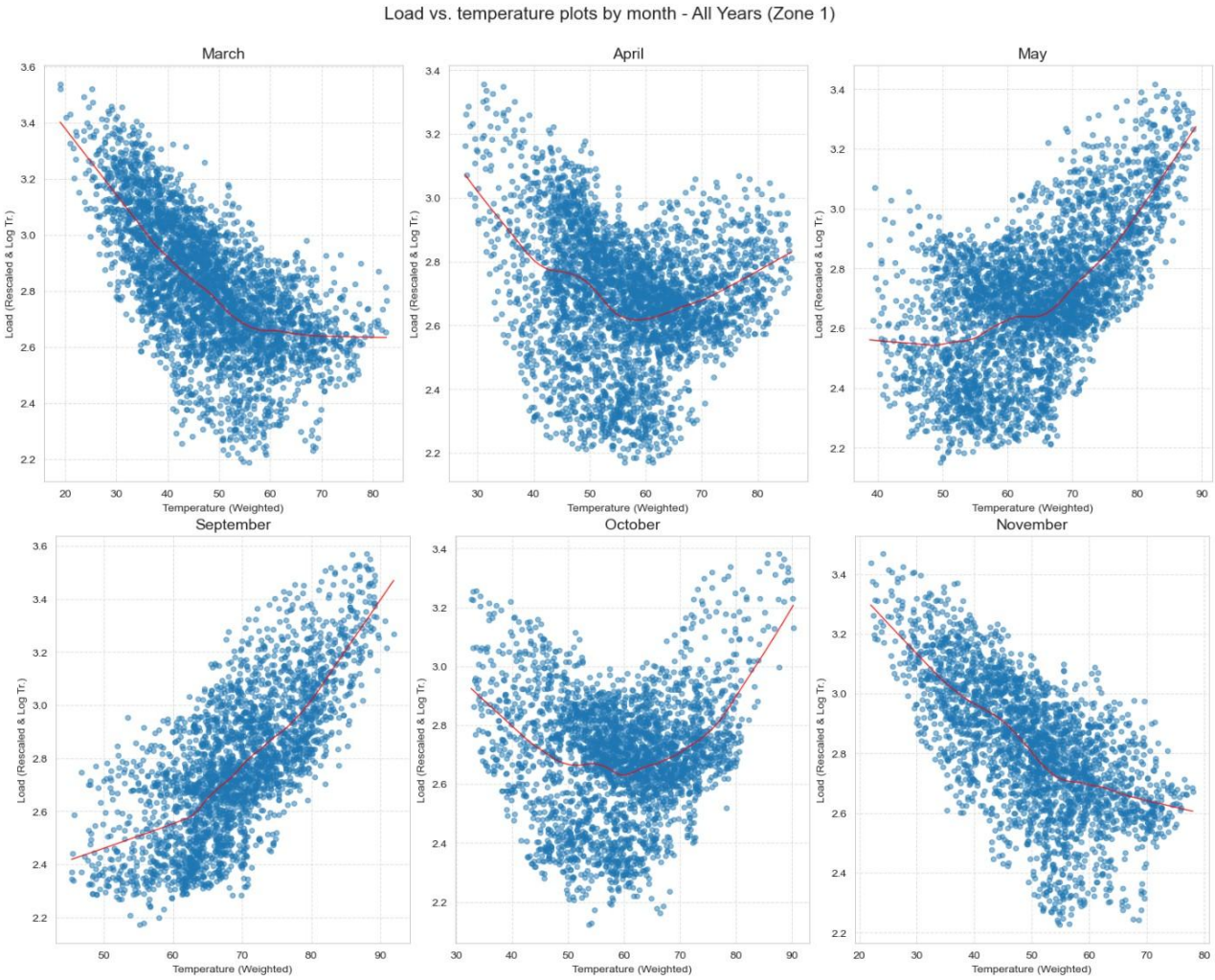


Figure 4.21: Load versus Temperature scatterplots with loess smoothing, All Years (Zone 1)

Allowing load-temperature relationship to be non-linear by breaking down temperature into three linear segments, the overall performance of this multiple regression model is shown in table 4.7. The piecewise linear regression model combined with harmonics is the best time series method in terms of overall performance. Performance on test set for April 2004 has also increased considerably, indicating that the non-linear pattern is now being captured more adequately.

Test Set in Month	R2 score	RMSE
January, 2004	0.519	0.121
April, 2004	0.654	0.108
July, 2004	0.797	0.126
October, 2004	0.723	0.088
December, 2004	0.802	0.096
	Average: **0.699**	Average: **0.107**

Table 4.7: Piecewise linear regression performance on test sets across 2004, by month

Explanatory Variable	VIF
"HDK" — First Knot (Below 55°F)	4.871807
"CDK" — Second Knot (Above 65°F)	1.055392
"trend" — Linear Trend term	4.744324
"sin(1,24)" — Harmonic 1 (24 hour periodicity)	1.096138
"cos(1,24)" — Harmonic 1 (24 hour periodicity)	1.070003
"sin(2,24)" — Harmonic 2 (24 hour periodicity)	1.026594
"cos(2,24)" — Harmonic 2 (24 hour periodicity)	1.007374

Table 4.8: VIF test for piecewise linear regression model

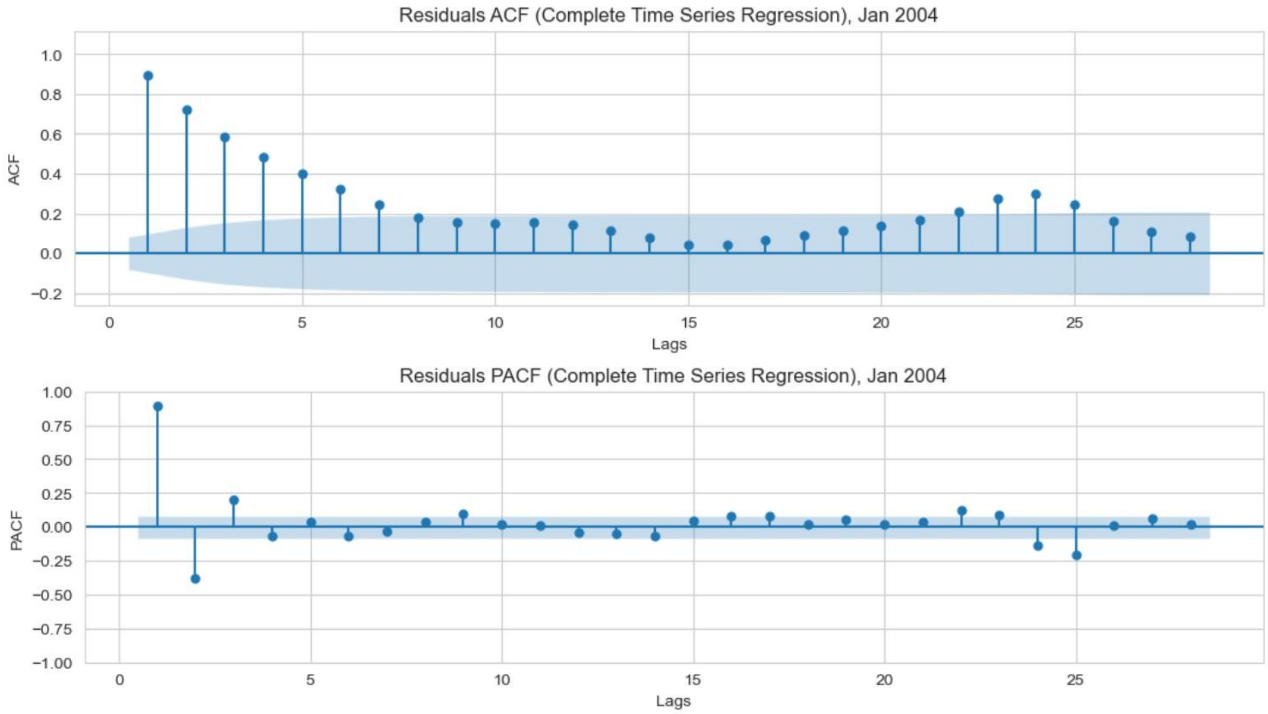


Figure 4.22: ACF and PACF of residuals, piecewise linear regression model (Jan 2004, training set)

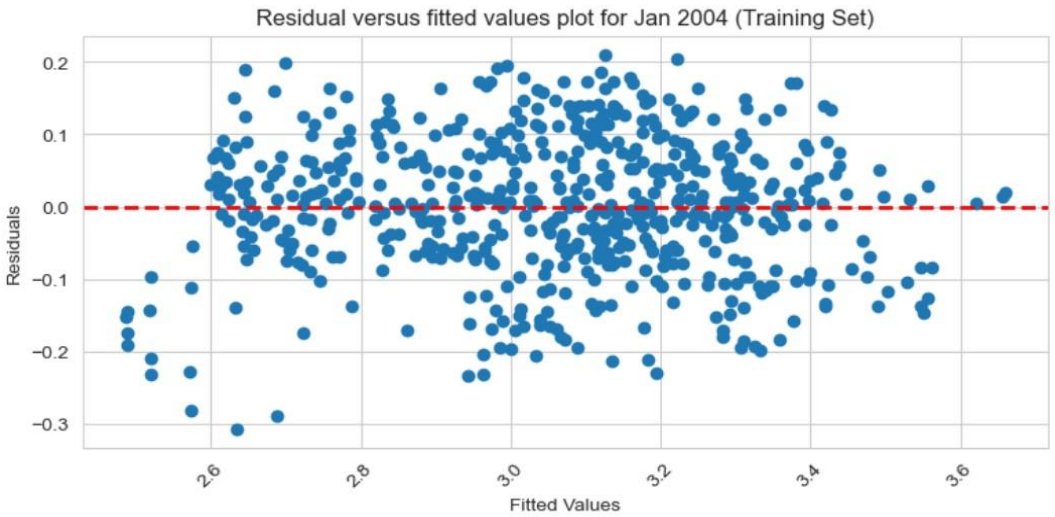


Figure 4.23: Residuals vs. fitted values, piecewise linear regression model (Jan 2004, training set)

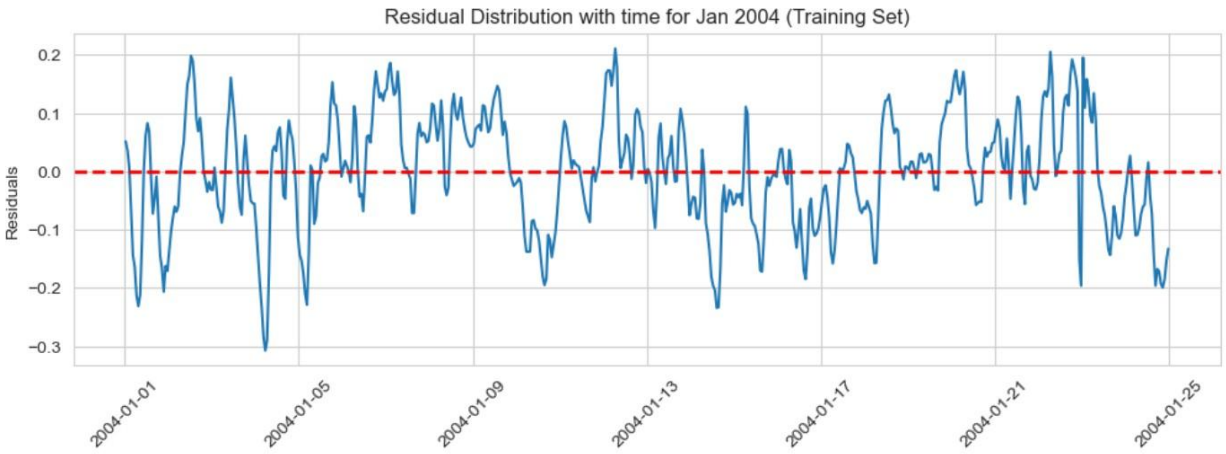


Figure 4.24: Residuals versus time plot, piecewise linear regression model (Jan 2004, training set)

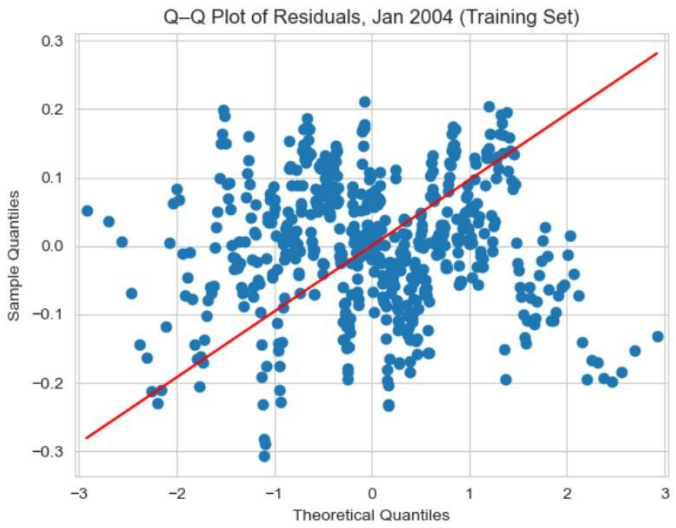


Figure 4.25: Residuals QQ plot, piecewise linear regression model (Jan 2004, training set)

The observed versus fitted values plots on training sets for piecewise linear regression methods, by individual months can be seen in appendix A.13. To check how meaningful the results are, month-specific diagnostics for piecewise linear regression models are performed on the training set. This section shows the diagnostics on training set for the month of January 2004. Diagnostic plots for the remaining testing months are not included in this section because they lead to the same conclusions. They can be seen in appendix A.14 – A.19.

The conclusion drawn from diagnostics are as follows:

Firstly, no significant multicollinearity was found, with VIF values for all explanatory variables being less than 5 (Table 4.8). This suggests that no independent variable can be well-approximated as a linear combination of the others.
Secondly, ACF of model residuals (figure 4.22) shows both non-seasonal and seasonal autocorrelation. This suggests that the piecewise linear regression model still hasn't captured

temporal dependencies completely and residuals are not white noise. The ACF/PACF plots are considered reliable because the regression residuals are found to be stationary for each of the training sets considered, according to the ADF test. The ADF test results for each training set can be seen in appendix A.16.
Thirdly, the residuals versus fitted values plot (figure 4.23) reveals that the residuals are randomly scattered around the zero line without a clear pattern or shape, suggesting that the model is able to estimate the average movement in the data. The spread of residuals is also roughly the same, suggesting homoscedasticity in errors.
Forth, the residual versus time plot (figure 4.24) shows no trend and residual spread doesn't vary with time. However, residual at time "t" tend to be very close to residual at time "t-1" suggesting positive autocorrelation, as seen in the ACF plot for residuals.
And finally, it is seen that residual distribution considerably deviates from a Gaussian distribution (figure 4.25).

The competition dataset also includes days which were declared as public holidays. However, holiday information is not added as explanatory variable in the final time series model. This is because for every short-term forecasting done here (by month), the training period has been set to roughly 3 weeks. Since holidays occur very rarely in the training set (roughly 0 — 16 % of data in training set in most cases) and cluster together in time, the time series model would not have sufficient variation in the data to adequately determine the true coefficient for this variable.

4.3.5 Seasonal ARIMA

Regression methods typically rely on external factors to make predictions about the forecast variable. ARIMA, on the other hand, uses historical information that lies within the series structure to make those predictions. Studying the ACF and PACF plots is helpful in identifying the correct order of the AR and the MA components of a time series. However, it's also essential to check for stationarity before interpreting the plots. ACF plots were covered before as an exploratory tool, however, they couldn't be used reliably because the load series for each of the testing sets was found to be non-stationary. Seasonal differencing, by accommodating daily periodicity, successfully transformed the series into a stationary one. The ADF test results for each training set, before and after seasonal differencing, can be found in appendix A.20. Load distribution plots, before and after seasonal differencing, can also be found in appendix A.21 – A.25. They help depict how the series changes after performing seasonal differencing (using lag '24') on the load data.

The ACF and PACF plots for seasonally differenced (daily periodicity adjusted) data are shown in figure 4.26, for January 2004 training set. ACF and PACF plots for the remaining test sets are included in appendix A.26 – A.29 because they provide similar information about the underlying structure of the series.

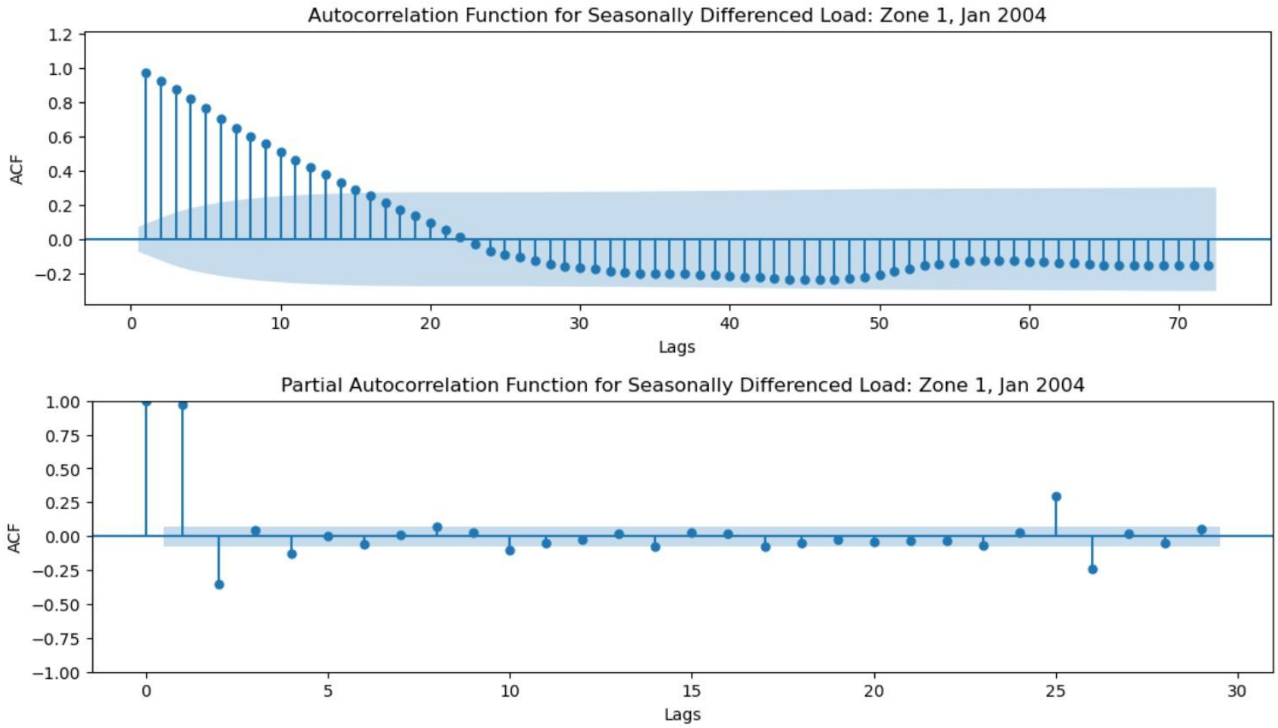


Figure 4.26: ACF and PACF plots for seasonally differenced load (Jan 2004, training set)

Looking at the first few lags, sharp, significant spike at *lag 1* and *lag 2* can be seen in the PACF plot that is followed by a drop-off. This is suggestive of an AR(2) process. The ACF shows a very slow decay rather than a sharp cutoff. This confirms that the series is dominated by AR behavior rather than MA behavior. Some months did show occasional spike in their respective PACF plots at *lag 3,* but the spikes were not frequent or significant enough to indicate the presence of an AR(3) process. In addition, significant spike around *lag 24/25* in PACF plot are observed suggesting that seasonal autoregressive component might also be present. From the existing PACF plots, it is difficult to tell whether spikes appear at further seasonal lags or not. In other words, the order of the seasonal autoregressive component is unclear.

For non-seasonal MA component to be present, sharp cut off in ACF plot and a gradual decay in PACF plot would be expected. For a seasonal MA component to be present, significant isolated spikes in ACF plot at seasonal lags (that directly proceeds in a cutoff) and gradual decay in PACF plot at seasonal lags would be expected. None of these signs are observed in the plots. However, adding lower order MA terms can adjust anomalies for the errors made by the low-order AR part of the model (Montgomery et al, 2015).

For determining the correct order of the seasonal autoregressive component as well as the MA terms (seasonal and non-seasonal), *Auto ARIMA* is used that makes use of Akaike Information Criterion (AIC) for parameter selection. Both seasonal and non-seasonal MA parameters were restricted to a maximum value of '1' so that the model remains relatively simple, reducing the chances of overfitting. The maximum value of the seasonal autoregressive component parameter was restricted to a maximum value of '4' since the series is more AR dominant and might contain memory that spans over a few days in the past.

For the training periods across 2004, the SARIMA models found to be the best quality according to Auto ARIMA are shown in table 4.9.

Training Set	Best (p,q)	Best (P,Q)	AIC Value
January, 2004	(1,1)	(3,0)	-1942.6
April, 2004	(2,1)	(0,1)	-2068.4
July, 2004	(1,1)	(3,0)	-2501.0
October, 2004	(2,1)	(2,0)	-2282.3
December, 2004	(2,0)	(2,1)	-2386.6

Table 4.9: Best SARIMA model according to Auto ARIMA, by different training sets

Considering strong evidence for an AR(2) process in ACF/PACF plots and the best model having *p=2* in majority of cases, *p=2* is considered the appropriate value. *q=1* is found to be the best value for *q* in almost all cases in table 4.9. The value for *P* varies between 0 and 3, so a moderate value of P=2 is considered an appropriate value. This value would allow the model to capture 'memory' effects from previous two days without making the model very complex. And lastly, *Q=0* is considered appropriate value for *Q* since it's found to be the best value in majority of cases, and the underlying process is found to be less MA dominant. Hence, the best model is found to be:

SARIMA(2,0,1)(2,1,0)[24].

This model is found to consistently hold an AIC value close to the lowest value in all training sets (not the lowest AIC value in every training set), compared to SARIMA models with different orders. For each training set, all SARIMA models tested by Auto ARIMA with their respective AIC values can be found in appendix A.30.

Figure 4.27 contains diagnostic plots for the SARIMA(2,0,1)(2,1,0)[24] model on January 2004 training set. Based on the first diagnostic plot i.e. the residual plot, the residuals appear to oscillate around a mean of zero with relatively constant variance for the majority of the time series. However, there is a significant spike toward the end of the training period (around Jan 21-23). It can be concluded that the model is generally stable, but it does struggle to predict sudden *shocks* in load, as seen towards the end of training period.

The Correlogram, or the ACF of residuals, shows that almost all autocorrelation spikes are well within the blue shaded confidence interval. A slight spike at *lag 8* is still observed though. This indicates that the non-seasonal and seasonal components of the chosen model have successfully extracted nearly all the information from the data and the residuals are close to White Noise.

For the Histogram plus Estimated Density plot, the orange KDE curve follows the general shape of the green *N(0,1)* normal distribution curve. However, the KDE is more "peaked" and has a slightly higher center than the theoretical normal line. It can be concluded that the residuals are approximately normal, but the peaked portion of KDE curve suggests that the model is slightly underestimating the probability of extreme values (i.e. outliers, as seen in Standardized Residuals plot).

For the Normal Q-Q Plot, the points follow the red linear reference line closely in the center however, they deviate at both the left and right tails. The chosen model is able to fit the data closely most of the time, except in the case of sudden jumps in load (outlier cases) that are preventing the errors from being perfectly normally distributed.

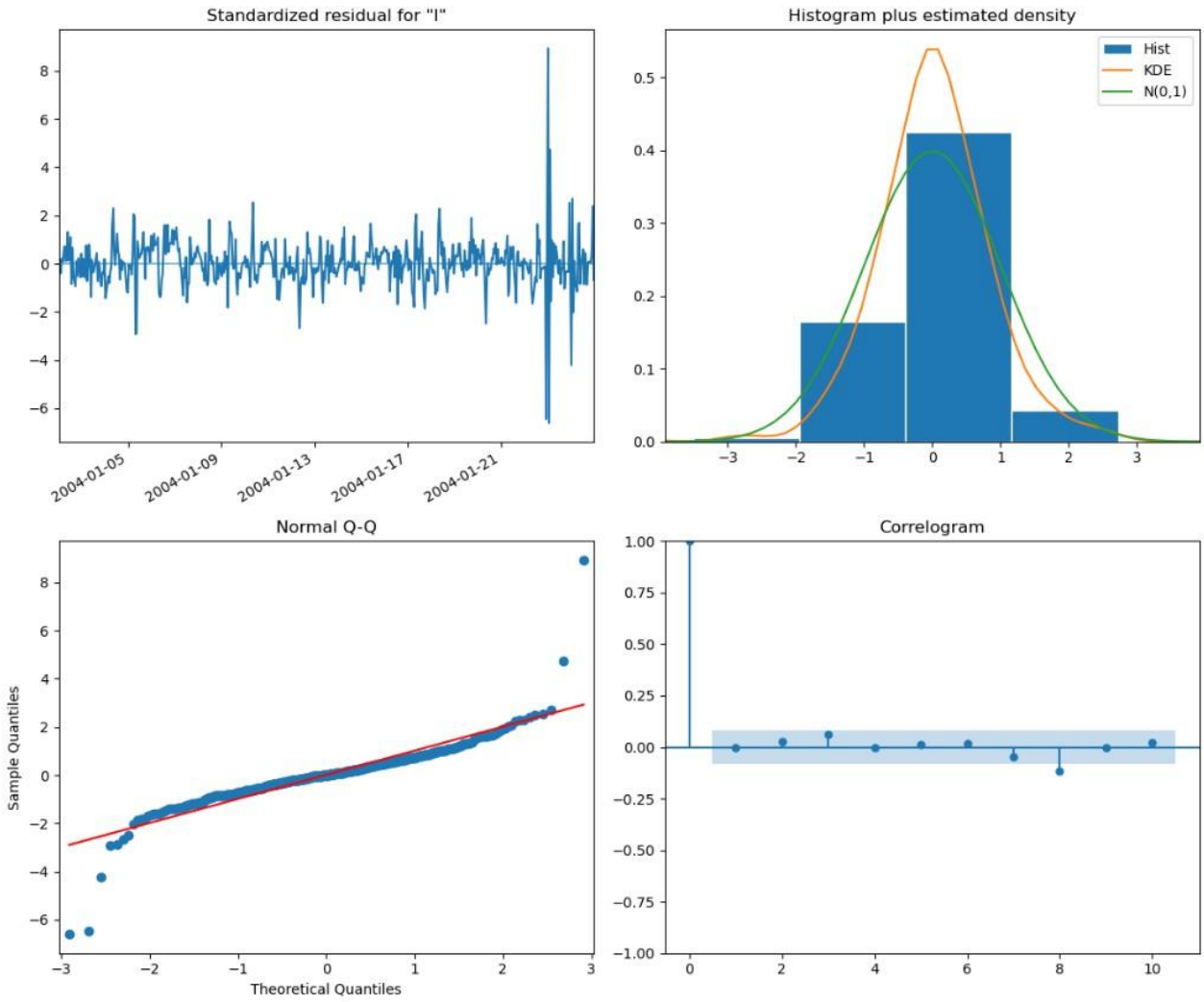


Figure 4.27: Diagnostic plots for SARIMA(2,0,1)(2,1,0)[24] model (Jan 2004, training set)

The diagnostics for the SARIMA(2,0,1)(2,1,0)[24] model for January 2004 training data indicate a strong goodness of fit. Same conclusions can be made for the remaining training sets from the diagnostics plots provided in appendix A.31 – A.34.

Table 4.10 summarizes the performance of the SARIMA(2,0,1)(2,1,0)[24] model on different test sets. The average performance is significantly poorer than piecewise linear regression method. The model performs even poorer than the best naïve method found earlier. It only performs slightly better than Holt Winter's Exponential Smoothing method. The gap in performance between piecewise linear regression method and SARIMA model highlights the importance of temperature information in producing accurate load forecasts.

Test Set in Month	R2 score	RMSE
January, 2004	-0.818	0.235
April, 2004	0.329	0.151
July, 2004	0.818	0.119
October, 2004	0.595	0.106
December, 2004	-0.290	0.246
	Average: **0.126**	Average: **0.171**

Table 4.10: SARIMA(2,0,1)(2,1,0)[24] performance on test sets across 2004, by month

4.3.6 Seasonal ARIMA with Exogenous Variables (SARIMAX)

Regression models are often extended with seasonal ARMA error structures to better explain variation in the data. These models are known as SARIMAX models. The exogenous regressors help capture the deterministic variation in the series from external factors, allowing the ARMA component to model the remaining serial dependence. All the regressors of the piecewise linear regression model— including the harmonics, trend, and intercept components — are included as exogenous variables in the SARIMAX model to determine the deterministic variation. The regression residuals of the piecewise linear regression model—representing the stochastic, serially dependent component of the model—were found to be stationary (see appendix A.16). Consequently, they are suitable for modeling by the ARMA component of the SARIMAX framework. The ACF and PACF plots for the piecewise linear regression model (see Figure 4.22 and appendix A.17 – A.18) show strong signs of an AR dominant model. The ACF shows an exponential decay pattern (rather than a sharp cut-off) that re-emerges at seasonal lag, suggesting that both non-seasonal and seasonal AR component are likely significant. The PACF shows significant spikes for the first 3 lags, which is immediately followed by a cut-off. A significant spike in PACF at the first seasonal lag ($s = 24$), consistent with a seasonal AR(1) component, is also observed. Overall, ACF and PACF behavior suggest an AR(3) structure with both non-seasonal and seasonal component.

To determine the appropriate order of the seasonal autoregressive component as well as the seasonal and non-seasonal moving average terms, the *Auto-ARIMA* procedure was employed using the Akaike Information Criterion (AIC) for model selection. Both seasonal and non-seasonal MA orders were constrained to a maximum value of '3' so that the effect of increasing the order of MA components on AIC values can be seen. The maximum orders of the seasonal and non-seasonal autoregressive components were set to '3', reflecting the AR-dominant nature of the series that might contain memory spanning over a few days in the past.

For the training periods across 2004, the SARIMAX models found to be MLE convergent and best in quality according to Auto ARIMA are shown in table 4.11.

Training Set	Best (p,q)	Best (P,Q)	AIC Value
January, 2004	(1,1)	(0,0)	-2094
April, 2004	(3,0)	(2,0)	-2205
July, 2004	(3,0)	(1,0)	-2636
October, 2004	(3,0)	(1,1)	-2452
December, 2004	(3,0)	(0,0)	-2282

Table 4.11: Best MLE-convergent SARIMAX model according to Auto ARIMA, by training set

Considering strong evidence for an AR(3) process in ACF/PACF plots and the best model having $p=3$ in majority of cases, it is considered the appropriate value for non-seasonal AR component. For non-seasonal MA component, $q=0$ is found to be the best value in almost all cases in table 4.11. The value for P varies between 0 and 2, however $P=1$ is considered appropriate to maintain model parsimony and simultaneously allow for some seasonal AR component since seasonal spike in ACF/PACF is significant. And lastly, $Q=0$ is considered appropriate value for Q since it's found to be the best value in majority of cases, and the underlying process is found to be less MA dominant. Hence, the best model is found to be:

$$\text{SARIMAX}(3,0,0)(1,0,0)[24].$$

where,
X: External Regressors from Piecewise Linear Regression

SARIMAX(3,0,0)(1,0,0)[24] consistently attains AIC values close to the minimum across all training sets, while also achieving successful MLE convergence. Although it did not always yield the lowest AIC in every training set, it performs competitively compared to alternative SARIMAX specifications with different orders. For each training set the full set of SARIMAX models evaluated by the Auto-ARIMA procedure, along with their corresponding AIC values, is reported in appendix A.35 – A.37.

Figure 4.28 contains diagnostic plots for the SARIMAX(3,0,0)(1,0,0)[24] model on April 2004 training set. Based on the first diagnostic plot i.e. the residual plot, the residuals appear to oscillate around a mean of zero with relatively constant variance for the majority of the time series. However, there is a significant spike toward the end of the training period (around April 19-20). It can be concluded that the model is generally stable, but it does struggle to predict sudden *shocks* in load, as seen towards the end of training period.

The Correlogram, or the ACF of residuals, shows that almost all autocorrelation spikes are within or at the boundary of the blue shaded confidence interval. This indicates that the non-seasonal and seasonal components of the chosen model are capturing serial dependence well and the residuals are good approximation of White Noise.

For the Histogram plus Estimated Density plot, the orange KDE curve approximately follows the general shape of the green *N(0,1)* normal distribution curve. It can be concluded that the residuals follow normal distribution well, but tails appear a bit heavier than *N(0,1)*. This suggests that prediction intervals are generally reliable but extreme quantiles may be less accurate.

For the Normal Q-Q Plot, the points follow the red linear reference line closely in the center however, they slightly deviate at both the left and right tails. The chosen model is able to fit the data closely most of the time, apart from the tail ends of the load distribution.

The diagnostics for the SARIMAX(3,0,0)(1,0,0)[24] model for April 2004 training data indicate a strong goodness of fit. Same conclusions can be made regarding the remaining training sets from the diagnostics plots provided in appendix A.38 - A.41.

Test Set in Month	R2 score	RMSE
January, 2004	-0.296	0.198
April, 2004	0.689	0.103
July, 2004	0.875	0.099
October, 2004	0.699	0.092
December, 2004	0.576	0.141
	Average: **0.508**	Average: **0.126**

Table 4.12: SARIMAX(3,0,0)(1,0,0)[24] performance on test sets across 2004, by month

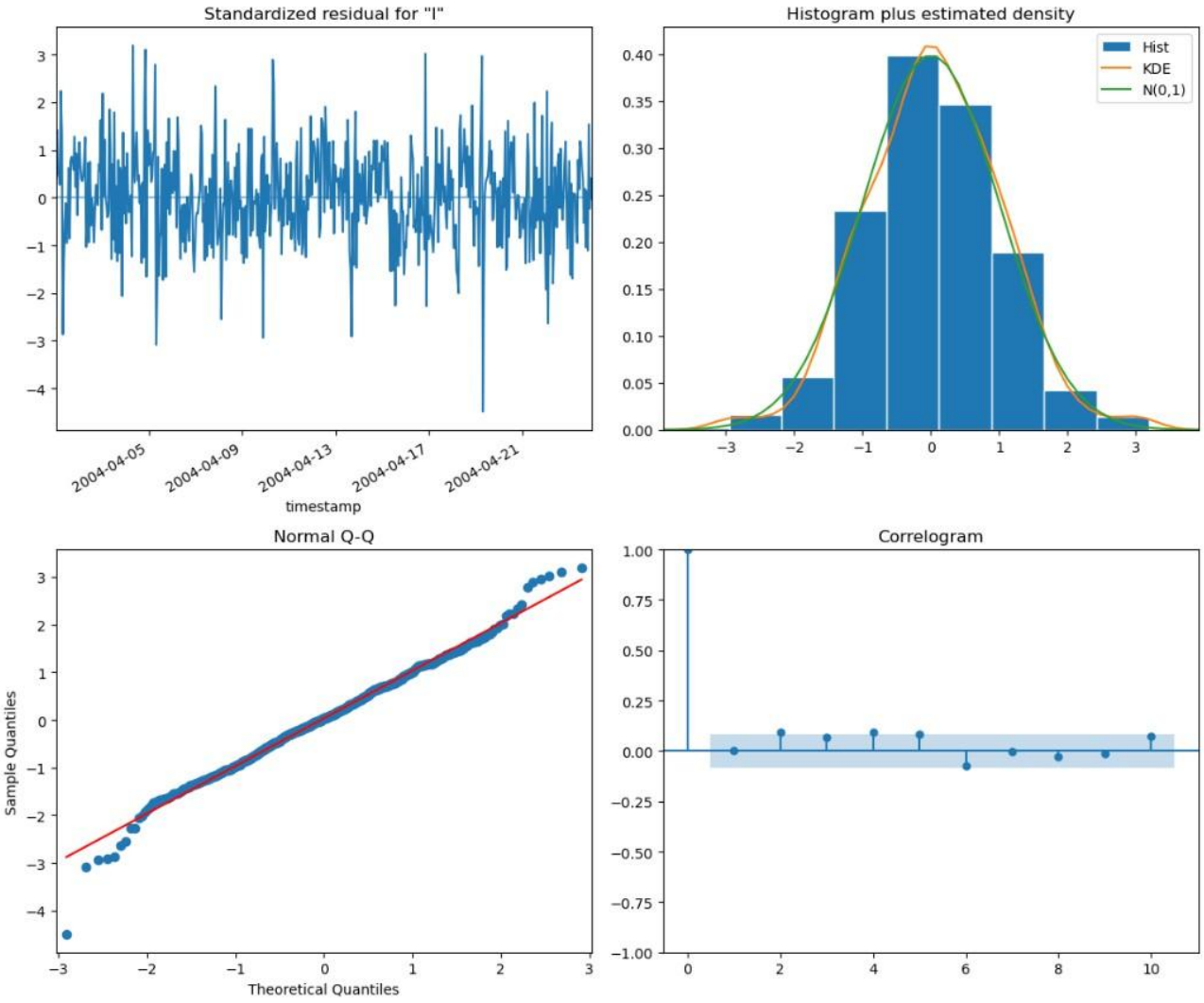


Figure 4.28: Diagnostic plots for SARIMAX(3,0,0)(1,0,0)[24] model (Apr 2004, training set)

Table 4.12 summarizes the performance of the SARIMAX(3,0,0)(1,0,0)[24] model on different test sets. Even with lower AIC value than a SARIMAX model with no AR and MA components (i.e. SARIMAX(0,0,0)(0,0,0)[0]), the average performance of the chosen model is significantly poorer than piecewise linear regression method with no SARIMA terms. This suggests that the chosen SARIMAX(3,0,0)(1,0,0)[24] model is resulting in overfitting and is modelling patterns that are not generalizable. Other, more parsimonious, SARIMAX models were also evaluated to assess whether simplifying the model could lead to better forecasting performance without overfitting. However, the same conclusion was drawn since none of the models performed better, on average, than piecewise linear regression method with no SARIMA terms. Average performance of more parsimonious forms of the SARIMAX model is listed in table 4.13.

Model	Average R2 score	Average RMSE
SARIMAX(3,0,0)(0,0,0)[24]	0.610	0.117
SARIMAX(1,0,1)(0,0,0)[24]	0.639	0.114
SARIMAX(1,0,0)(0,0,0)[24]	0.625	0.117
SARIMAX(2,0,0)(0,0,0)[24]	0.662	0.111

Table 4.13: Performance of other SARIMAX models, averaged on test sets across 2004

4.3.7 Load Forecasting combined with Backcasting

Backcasting is estimating past values using information from later points in time. Instead of forecasting forward, predictions are made backward in time (Hong, 2014). This paper attempted to find out if combining forecast predictions with backcast predictions on the same test set (the same 5 test sets across 2024) could improve the overall prediction performance and could help fill missing values better. To achieve this, the first training set was kept the same across all five periods as before. The second training set, being the same length as the first training set, was kept just ahead of the test set in question. Training was done after reversing the temporal order (i.e. temporal index) of the second training set. The motivation to do this is to allow the trained model to see the end of the testing period as the beginning of time, while making predictions. Time series regression does not model dependence between consecutive observations and is generally "unaware" of the temporal order. But time-indexed features (such as trend and harmonic components) still impose temporal directionality.

Backcasting builds on information near the end of the test window while forecasting builds on information near the beginning. To reduce boundary bias, the predicted value on the test set was considered to be the average of forecast-based predicted value and backcast-based predicted value. However, no improvement in average performance was seen as a result compared to regular forecasting approach that excludes backcasting (see Table 4.14).

Test Set in Month	R2 score	RMSE
January, 2004	0.421	0.132
April, 2004	0.500	0.130
July, 2004	0.899	0.088
October, 2004	0.795	0.075
December, 2004	0.594	0.137
	Average: **0.641**	Average: **0.112**

Table 4.14: Combining Forecasting with Backcasting — Performance on test set, by month

4.3.8 Filling-in Missing Load Values

Gefcom 2012 competition mentions eight periods of one week duration, with missing values for electric load in the entire power grid. However, temperature information has been provided for the missing periods. The same PCA technique is used to extract the temperature signal from eleven weather stations data. For the entire grid comprising 20 zones, piecewise linear regression in combination with two harmonics is used as the forecasting model. The length of each training set is 3 weeks of data prior to the period that contains missing values. The model's performance is compared to a naïve approach that simply repeats the load information from the week prior to the missing value week. This naïve method was found to be a strong contender among the four tested naïve approaches, in terms of a simple baseline. Gefcom 2012's benchmark values for missing data are also compared with this naïve approach to evaluate error improvement over baseline.

Table 4.15 contains RMSE scores for piecewise linear regression model, naïve — repeating last cycle — method, and the competition's benchmark values. The provided scores are averaged over all "20" zones. RMSE scores appear high because the load data was untransformed (from log-transformation) to original scale using the exponential function after retrieving predictions.

Missing Period	Piecewise Linear Regression	Naïve Last Cycle Repetition	Competition's Benchmark Value
6 Mar 2005 - 12 Mar 2005	6567.4	15805.3	5698.5
20 Jun 2005 - 26 Jun 2005	7605.1	22595.1	6212.3
10 Sep 2005 - 16 Sep 2005	8478.2	12729.1	6695.3
25 Dec 2005 - 31 Dec 2005	9710.3	21212.8	7866.0
13 Feb 2006 - 19 Feb 2006	8220.1	18991.1	6441.8
25 May 2006 - 31 May 2006	8896.1	20614.3	9385.0
02 Aug 2006 - 08 Aug 2006	10234.0	14533.3	6190.7
22 Nov 2006 - 28 Nov 2006	10834.1	13439.3	8393.0
	Avg: 8818.2	**Avg:** 17490	**Avg:** 7110.3

Table 4.15: Avg. RMSE, by missing period — Piecewise Linear Regression, Baseline Naïve & Competition Benchmark

For benchmarked values, average RMSE (across 20 zones, all missing periods) was reduced by 59.3% compared to the baseline naïve error of 17490. In comparison, the same average error for piecewise linear regression reduced by 50% relative to the baseline naïve error of 17490. Hence, the piecewise linear regression method has considerable value since it's reducing the error over the entire power grid by half. However, slight gap of almost 10% persists in terms of reaching the benchmark performance in forecast accuracy.

4.3.9 Temperature Forecasting

Gefcom 2012 dataset provided temperature information for 8 week periods with missing load values. For the forecasting part of the competition, that requires load forecast from 2008/7/1 to 2008/7/7, no temperature information is provided. Since the piecewise linear regression model uses temperature as an explanatory variable, it must rely in temperature forecast to generate load predictions for the mentioned forecast horizon.

For the forecast horizon period, temperature series plots from previous years (see Fig. 4.29) show considerable periodic pattern and daily seasonality in temperature. For each year, from 2004 to 2007, the temperature series for 1st week of July was found to be non-stationary according to the ADF test (see appendix A.42). Performing seasonal differencing and then performing non-seasonal differencing corrected non-stationarity (see Table 4.16).

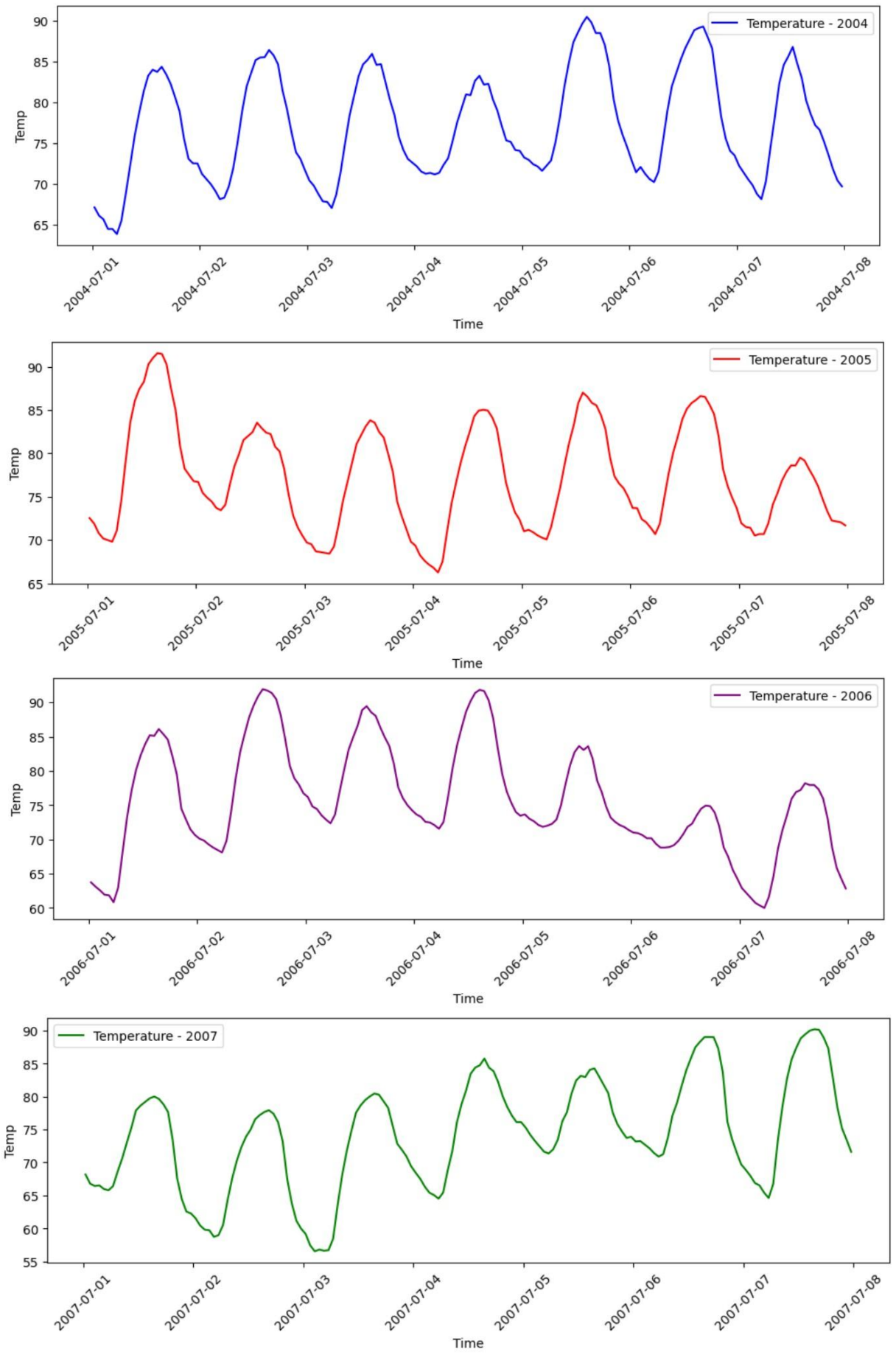


Fig 4.29: Temperature Series for first week of July, by year

	2004	2005	2006	2007
ADF statistic (after seasonal & non-seasonal differencing)	-6.612	-6.121	-3.894	-8.665
p-value of test statistic (after seasonal & non-seasonal differencing)	0.000	0.000	0.002	0.000
Conclusion: α = 0.05	Stationary	Stationary	Stationary	Stationary

Table 4.16: ADF test results for temperature series (after differencing) in 1st week of July, by year

For identifying a suitable forecasting approach to predict temperature, various traditional time series models are compared. Testing is spread across 4 periods, each period representing the 1st week of July from 2004 to 2007. For training, data 2 weeks prior to the test set is used. The length of the four training sets is deliberately kept short (i.e. two weeks) to align with the real constraint of limited data availability. A naïve approach of repeating last week's load (prior to test set) is kept as a baseline.

SARIMA or seasonal ARIMA is evaluated as one of the forecasting approaches. To determine the correct orders of the AR and MA components, Auto ARIMA method is used again. To maintain parsimony and preserve model's simplicity, (*p,q*) are both restricted to a maximum value of "3" and (*P,Q*) are both restricted to a maximum value of "2". For each training set, the best SARIMA model found can be seen in Table 4.17. Both seasonal (*D=1*) and non-seasonal differencing (*d=1*) is performed to maintain stationarity in series. The simplest SARIMA model i.e. SARIMA(1,1,0)(0,1,1)[24] is selected as the appropriate structure, since the AR and MA orders are low and are present in all the models found best for each training set.

Training Set	Best (p,q)	Best (P,Q)	AIC Value
17 to 30 June, 2004	(1,0)	(2,0)	726
17 to 30 June, 2005	(1,0)	(0,1)	724
17 to 30 June, 2006	(2,2)	(0,1)	747
17 to 30 June, 2007	(2,2)	(0,1)	754

Table 4.17: Best SARIMA model for temperature series according to Auto ARIMA, by training set

In addition to SARIMA, simple harmonic regression with a linear trend component is evaluated. With this approach, performance on test set did not increase after adding more than two harmonics. Holt Winters's additive method for exponential smoothing is also evaluated. Average performance on test set, for each of the three approaches, are compared with the naïve approach in table 4.18.

Method	Average RMSE
Naïve Aproach	6.132
Holt Winters's Exponential Smoothing	5.205
Simple Harmonic Regression (2 harmonics)	5.540
SARIMA	6.707
SARIMAX (with one harmonic and linear trend)	4.376

Table 4.18: Avg. performance of evaluated temperature forecasting approaches, across training sets

The best approach, however, was found to be SARIMAX approach that combines SARIMA with one harmonic. The correct AR/MA orders for SARIMAX was again identified with Auto ARIMA, using the same constrained defined for the simple SARIMA model. Adding more than one harmonic in SARIMAX didn't result in a performance increase on test set. The best SARIMAX model (with one harmonic) for each training set is shown in table 4.19. The simplest SARIMAX model, with non-seasonal and seasonal components as (2,0)(1,0) respectively, is selected as the appropriate structure since the AR and MA orders are low and are present in all the models found best for each training set.

Training Set	Best (p,q)	Best (P,Q)	AIC Value
17 to 30 June, 2004	(3,1)	(3,0)	745
17 to 30 June, 2005	(3,3)	(1,3)	772
17 to 30 June, 2006	(3,0)	(1,1)	802
17 to 30 June, 2007	(2,0)	(1,1)	819

Table 4.19: Best SARIMAX model for temperature according to Auto ARIMA, by training set

4.3.10 Total Grid Load Forecasting

Piecewise linear regression model in combination with harmonics uses the predicted temperature values to develop load forecast for the 1st week July 2008. The temperature values are predicted with the chosen SARIMAX model highlighted previously. As usual for temperature forecasting, two weeks of temperature data prior to the forecast horizon in 2008 is used for training.

For load forecasting, baseline naïve is kept the same i.e. repeating the last load cycle prior to the forecast horizon as predicted load. With temperature predictions in forecast horizon available, piecewise linear regression in combination with two harmonics is trained as usual to make load prediction in forecast horizon. Table 4.20 contains the prediction performance, averaged across 20 zones, for benchmark values provided by the author, naïve baseline and the piecewise linear regression model developed in this paper.

Method	Average RMSE
Naïve Aproach	17127.92
Benchmark Performance	15513.58
Piecewise Linear Regression (with 2 harmonics)	10804.30

Table 4.20: Performance comparison of piecewise linear regression with benchmark, averaged across 20 zones

It is found that competition benchmark relative to baseline naïve reduces average RMSE for forecast horizon by approximately 9.5%. With respect to the same naïve method, piecewise linear regression developed in this paper reduced average RMSE by 37%. If piecewise linear regression is directly compared with competition's benchmark, average RMSE gets reduced by approximately 30%. This result is in line with this paper's original goal of getting close to 30% error reduction, compared to benchmark load values. However, this error reduction only contains performance on forecasting horizon. If both missing values and forecast horizon are considered together, piecewise linear regression reduces average RMSE compared to benchmark load values by approximately 13%.

Chapter 5: Conclusion & Limitations

This thesis demonstrates that classical time series models, under realistic constraints of minimal training data and simplicity, can achieve competitive short-term load forecasting performance on the GEFCom2012 dataset. By focusing on zonal-level STLF without hierarchical reconciliation, the study meets its goal of approximating 30% error reduction relative to benchmarks, using only three-week training periods. Exploratory data analysis highlights periodic patterns and non-linear load-temperature dynamics, addressed through PCA-derived signals and model refinements.

The paper shows progressively improving regression methods: simple linear regression, harmonic regression, temperature-combined harmonic regression, and piecewise with harmonics best capturing non-linearities. SARIMA and SARIMAX are also used for performance comparison but are found to either be poor in performance or resulting in overfitting. Backcasting integration yields no improvement, suggesting limited boundary bias mitigation in short horizons. For missing values, piecewise regression reduces RMSE by 50% vs. naïve (8818 vs. 17490), nearing benchmarks (RMSE 7110). Temperature forecasting via SARIMAX(2,0,0)(1,0,0)[24] enables grid-level load predictions in forecasting horizon. The temperature predictions from this model allow piecewise linear regression model to make load predictions in forecasting horizon, achieving 37% error reduction relative to naïve approach and 30% error reduction relative to competition benchmark.

Overall, simpler statistical approaches match or exceed benchmarks, aligning with literature emphasizing efficiency in data-scarce environments. These findings contribute to practical energy analytics by validating lightweight models for utilities facing data limitations, such as in smart grids or transportation sectors dependent on reliable power. The 30% error target, achieved without complex DL or extensive tuning, underscores the value of harmonics and piecewise techniques for seasonality and non-linearity.

Limitations include dataset specificity: results are tailored to GEFCom2012's U.S. utility, potentially less generalizable to diverse climates or grids. Holidays were excluded due to rarity in short training sets, possibly underestimating their impact. No deep learning comparisons were made, focusing on classical methods; future work could benchmark against transformers or meta-learning under similar constraints. Temperature was the sole exogenous variable; incorporating humidity, wind, or socio-economic factors could enhance accuracy. Backcasting showed minimal gains, warranting exploration in longer horizons or with ensemble methods. The study ignores hierarchical aspects, assuming zonal averages suffice for grid totals—extending to full reconciliation could improve system-level insights.

References

- **Jolliffe, I. T.**
  Principal Component Analysis, 2nd ed., Springer, 2002.

- **Hastie, T., Tibshirani, R., Friedman, J.**
  The Elements of Statistical Learning, 2nd ed., Springer, 2009.

- **Montgomery, D. C., Jennings, C. L., & Kulahci, M.**
  Introduction to time series analysis and forecasting. John Wiley & Sons, 2015

- **Hyndman, R. J., & Athanasopoulos, G.**
  Forecasting: Principles and Practice, 3rd ed., OTexts, 2021

- **The MathWorks, Inc.**
  Unit Root Nonstationarity, MATLAB & Simulink, 2025
  www.mathworks.com/help/econ/unit-root-nonstationarity.html.

- **Smith A.**
  Pmdarima 2.1.1, The Python Package Index (PyPI), 2022
  https://pypi.org/user/arsmith/

- **Peixeiro, M.**
  Time series Forecasting in Python. Simon and Schuster, 2022

- **Hong, T., Pinson, P., Fan, S.**
  Global energy forecasting competition 2012. International Journal of Forecasting, vol.30, no.2, pp 357-363, 2014

- **National Academies of Sciences, Engineering, and Medicine**.
  Enhancing the Resilience of the Nation's Electricity System. Washington, DC: The National Academies Press, 2017. https://doi.org/10.17226/24836.

- **Mohamed, A. a. A.**
  On the Rising Interdependency between the Power Grid, *ICT Network, and E-Mobility: Modeling and Analysis*. Energies, *12*(10), 2019. https://doi.org/10.3390/en12101874.

- **Baran, E.**
  Barriers to Utility-Scale Electric Energy Storage. Western Interstate Energy Board, 2017

- **Karaduman, Ö.**
  Economics of Grid-Scale Energy Storage in Wholesale Electricity Markets. MIT CEEPR Working Paper 2021-005, 2021.

- **Tsoumplekas, G., Athanasiadis, C. L., Doukas, D. I., Chrysopoulos, A., Mitkas, P. A.**
  Few-Shot Load Forecasting Under Data Scarcity in Smart Grids: A Meta-Learning Approach. arXiv., 2024.

- **Nti, I. K., Teimeh, M., Nyarko-Boateng, O., & Adekoya, A. F.**

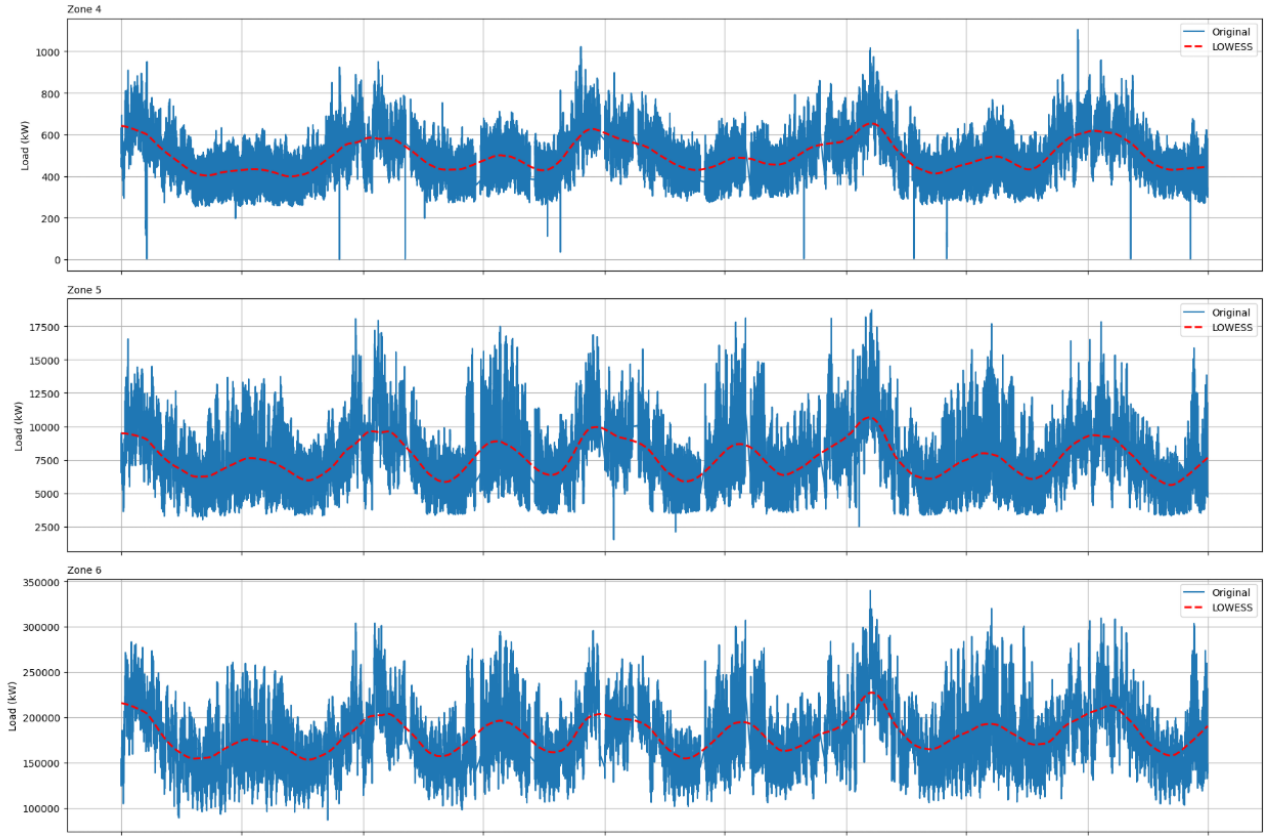
Electricity load forecasting: a systematic review. *Journal of Electrical Systems and Information Technology*, *7*(1)., 2020. https://doi.org/10.1186/s43067-020-00021-8

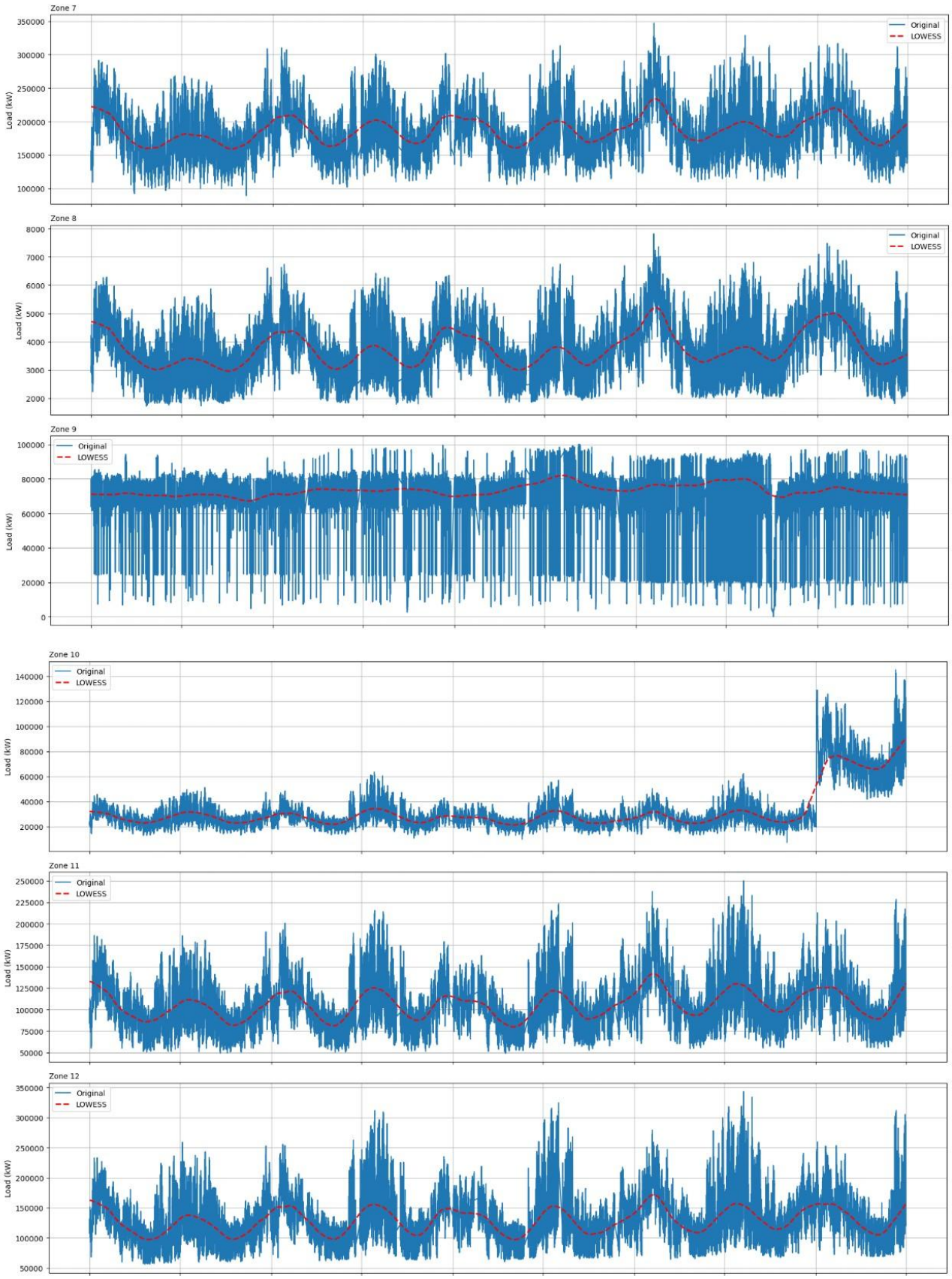
- **Roy, M., Pyltsov, V., & Hu, Y.**
  IISE PG&E Energy Analytics Challenge 2025: Hourly-Binned Regression Models Beat Transformers in Load Forecasting. arXiv (Cornell University)., 2025. https://doi.org/10.48550/arxiv.2505.11390

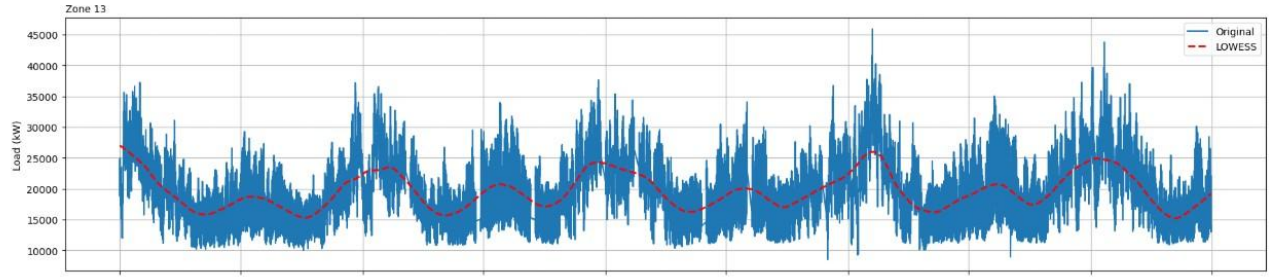
- **Wang, P., Liu, B., Hong, T.**
  Electric Load Forecasting with Recency Effect: a Big Data Approach. SAS - R&D, Cary, NC, USA, Energy Production and Infrastructure Center, University of North Carolina at Charlotte, USA, & Hugo Steinhaus Center, Wrocław University of Technology, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland., 2016.

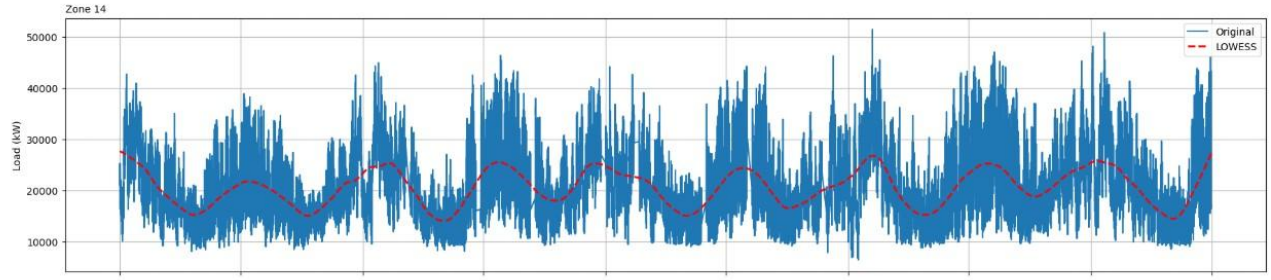
Appendix

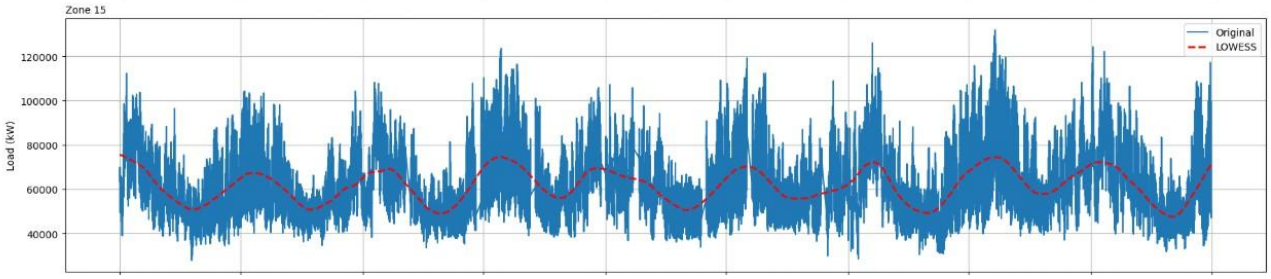
- **A.1** Hourly Load from 2004 — 2008 (in kW), by Zone

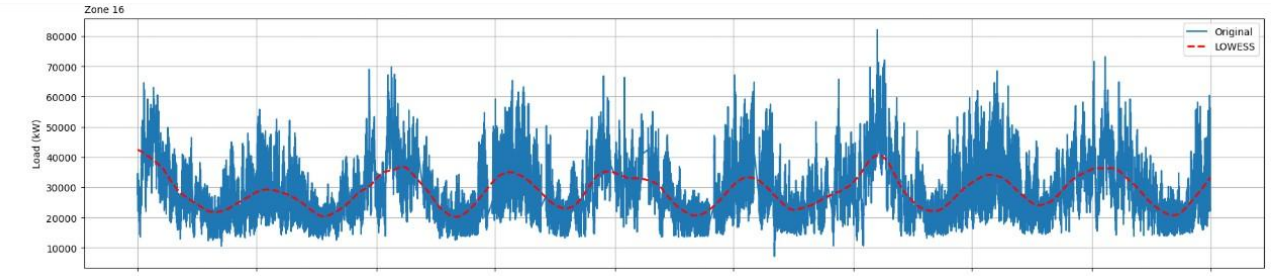


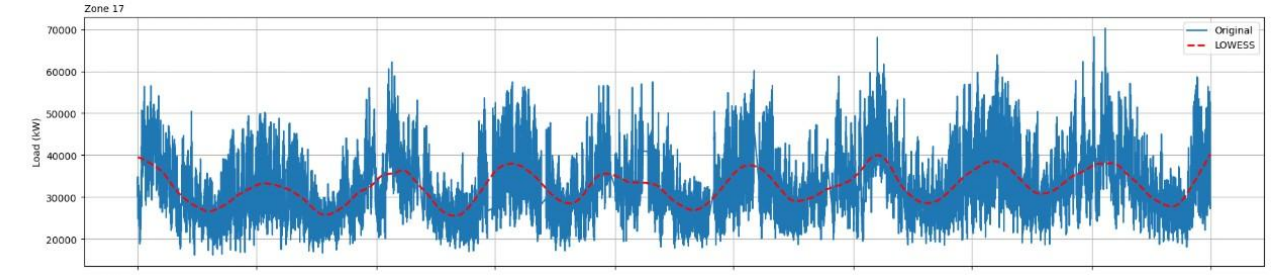

Zone 7
Zone 8
Zone 9
Zone 10
Zone 11
Zone 12
Original
LOWESS
Load (kW)

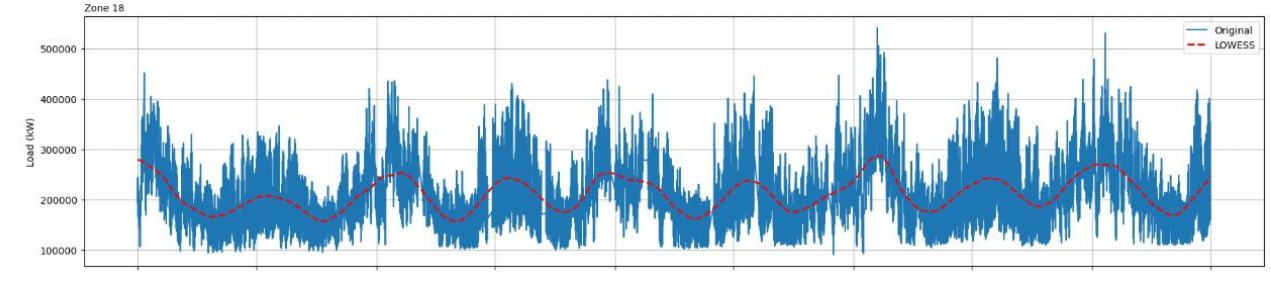

Zone 13
Load (kW)
45000
40000
35000
30000
25000
20000
15000
10000
Original
LOWESS

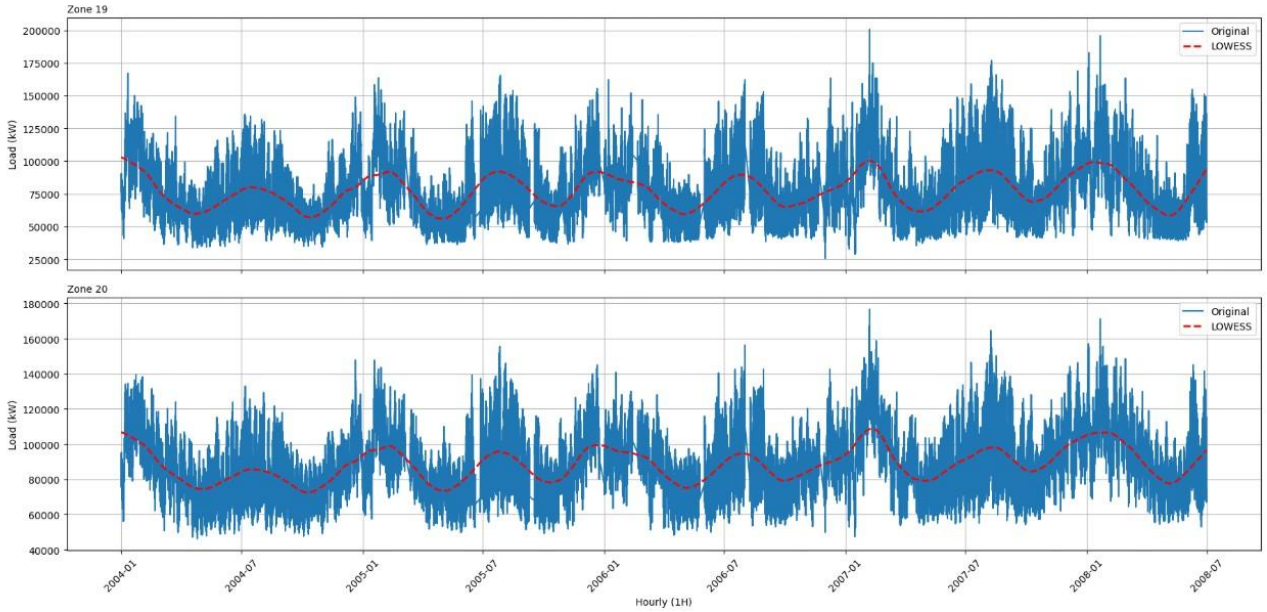

Zone 14
Load (kW)
50000
40000
30000
20000
10000
Original
LOWESS


Zone 15
Load (kW)
120000
100000
80000
60000
40000
Original
LOWESS

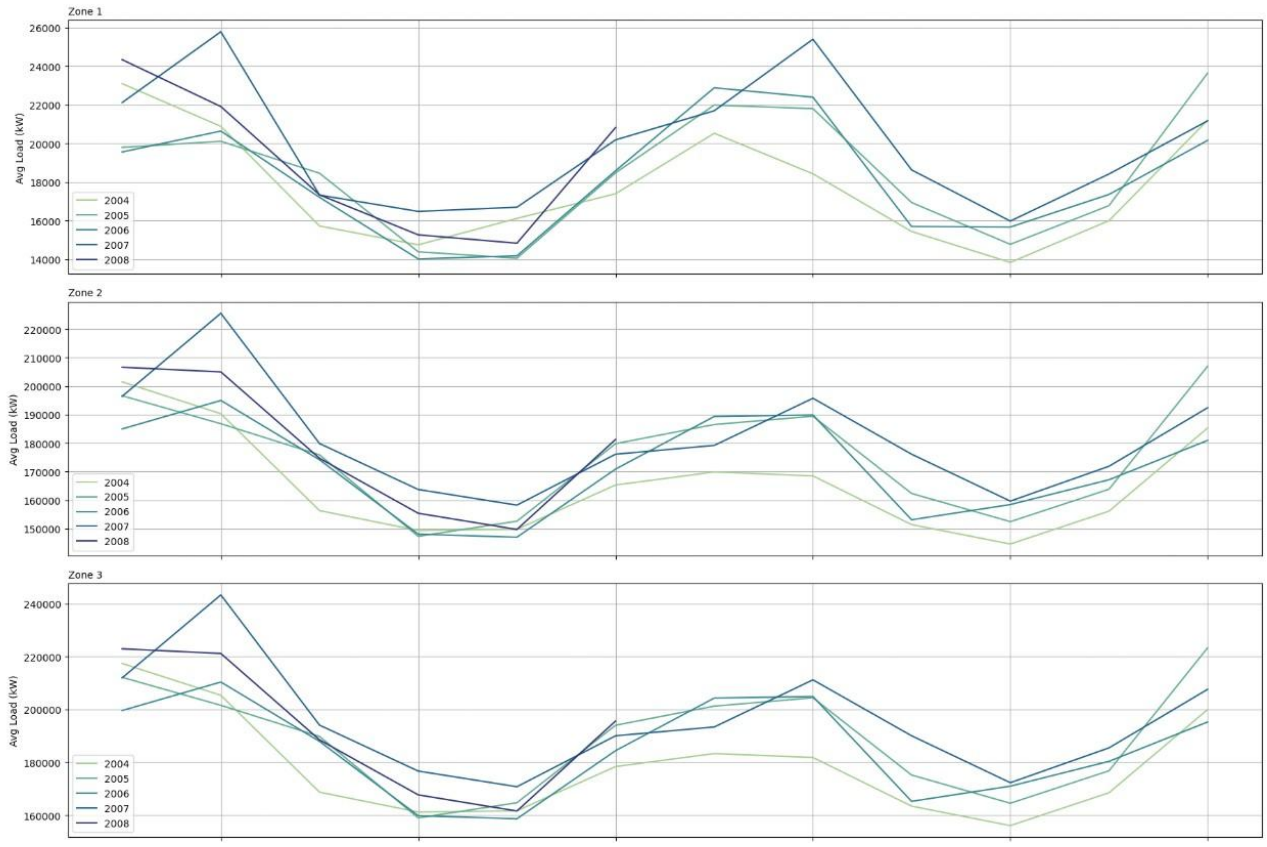

Zone 16
Load (kW)
80000
70000
60000
50000
40000
30000
20000
10000
Original
LOWESS

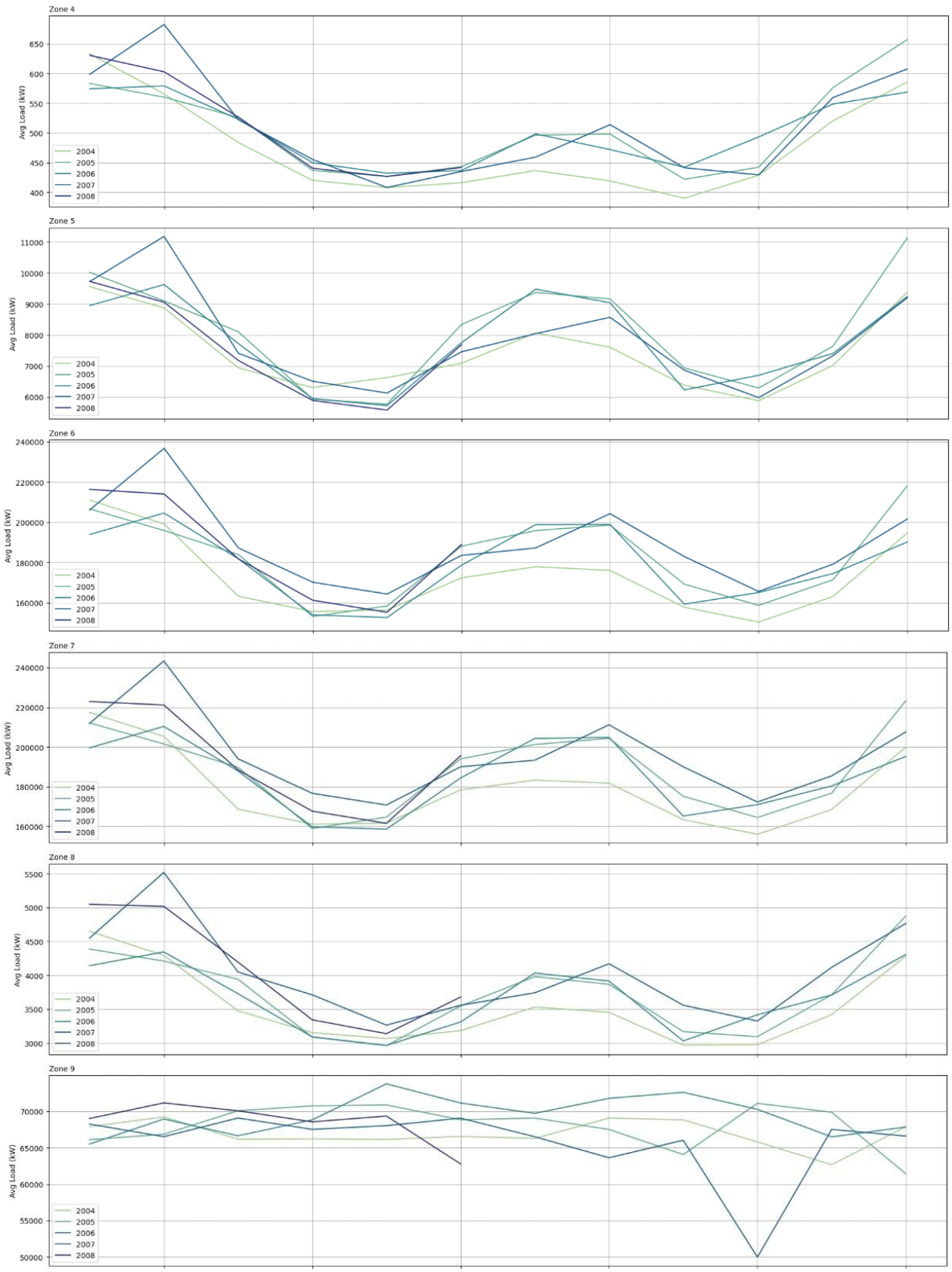

Zone 17
Load (kW)
70000
60000
50000
40000
30000
20000
Original
LOWESS

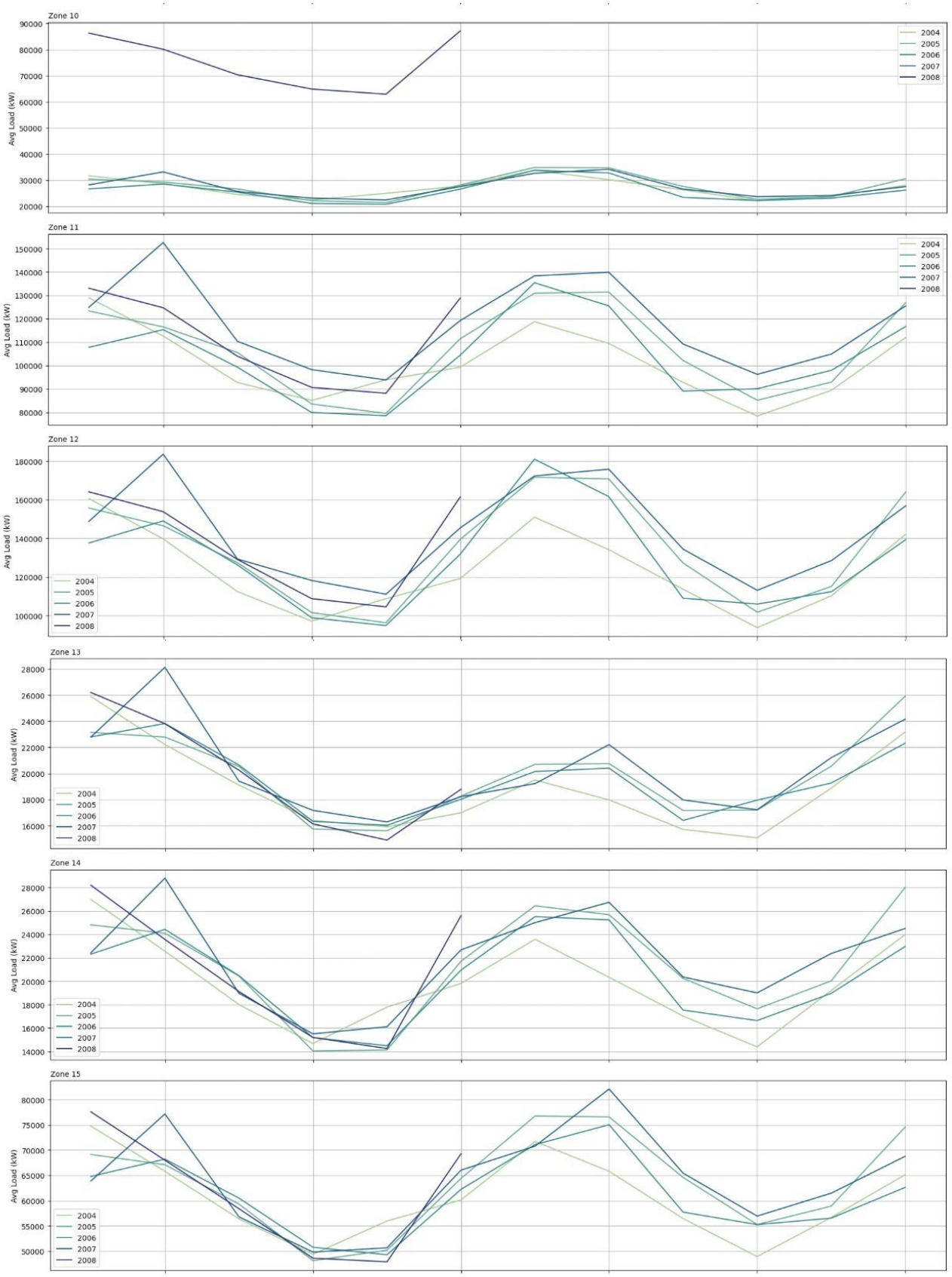

Zone 18
Load (kW)
500000
400000
300000
200000
100000
Original
LOWESS

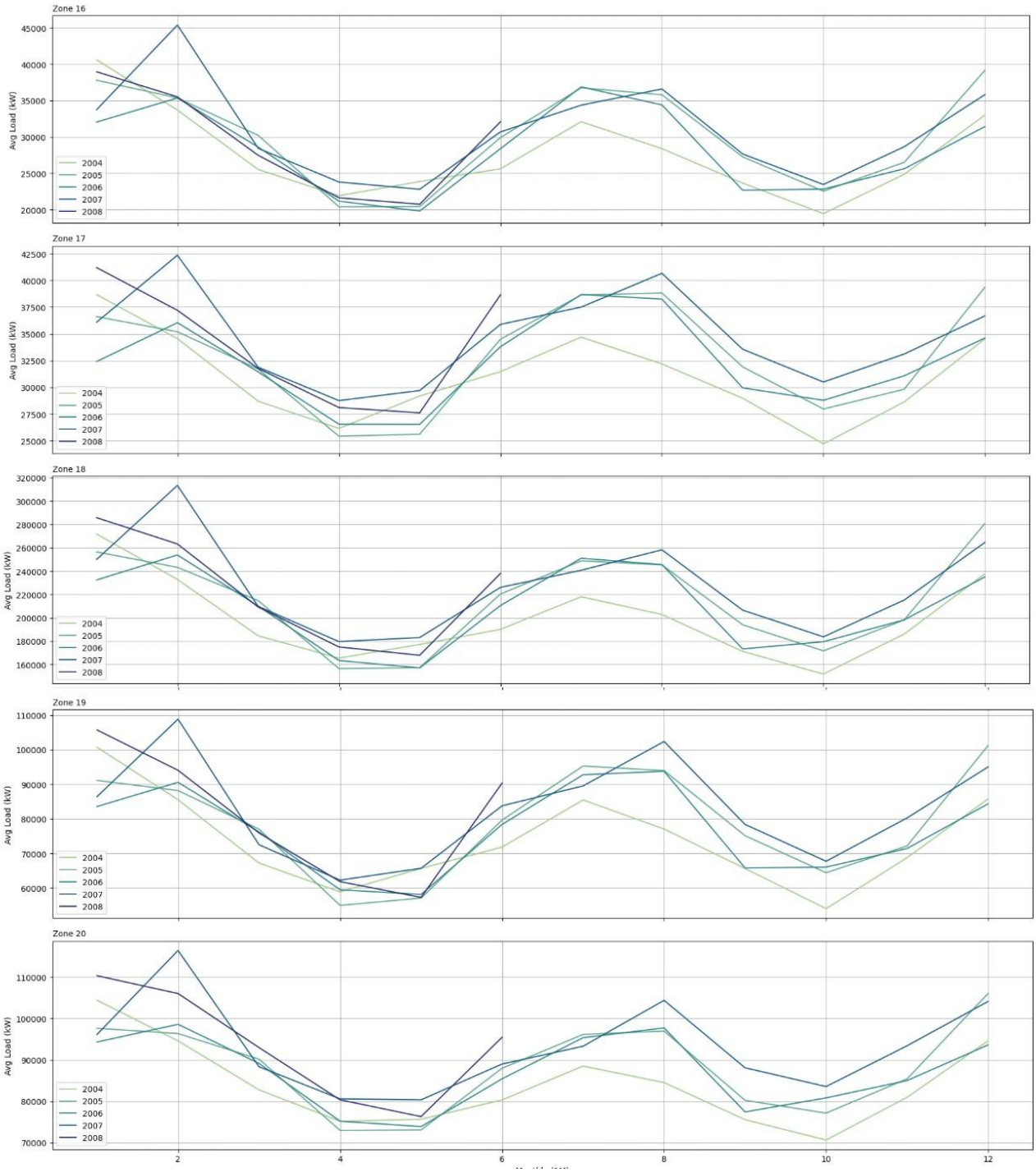


- **A.2** Average Monthly Load from 2004 — 2008 (in kW), by Zone

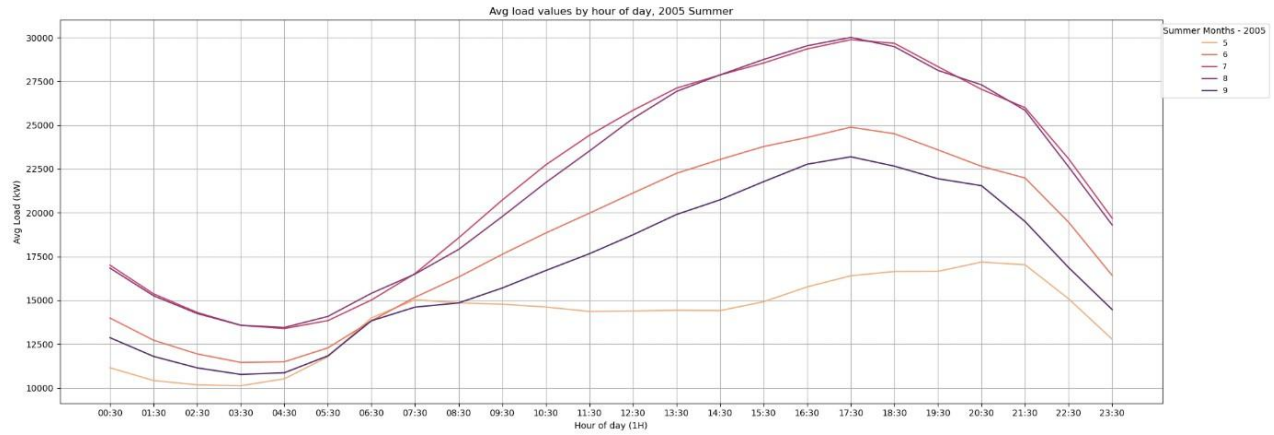



Zone 4
Avg Load (kW)
650
600
550
500
450
400
2004
2005
2006
2007
2008
Zone 5
Avg Load (kW)
11000
10000
9000
8000
7000
6000
2004
2005
2006
2007
2008
Zone 6
Avg Load (kW)
240000
220000
200000
180000
160000
2004
2005
2006
2007
2008
Zone 7
Avg Load (kW)
240000
220000
200000
180000
160000
2004
2005
2006
2007
2008
Zone 8
Avg Load (kW)
5500
5000
4500
4000
3500
3000
2004
2005
2006
2007
2008
Zone 9
Avg Load (kW)
70000
65000
60000
55000
50000
2004
2005
2006
2007
2008

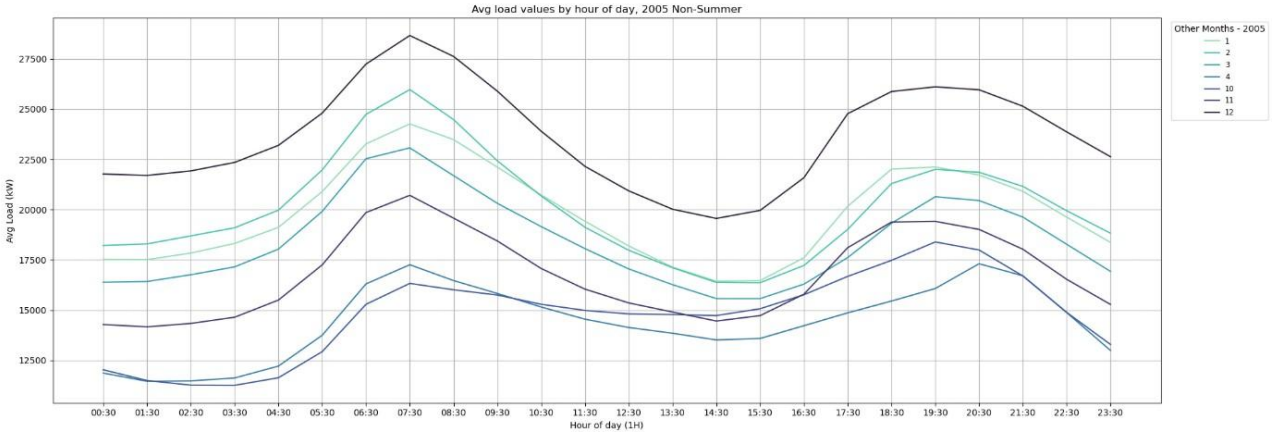

Zone 10
Avg Load (kW)
90000
80000
70000
60000
50000
40000
30000
20000
2004
2005
2006
2007
2008
Zone 11
Avg Load (kW)
150000
140000
130000
120000
110000
100000
90000
80000
2004
2005
2006
2007
2008
Zone 12
Avg Load (kW)
180000
160000
140000
120000
100000
2004
2005
2006
2007
2008
Zone 13
Avg Load (kW)
28000
26000
24000
22000
20000
18000
16000
2004
2005
2006
2007
2008
Zone 14
Avg Load (kW)
28000
26000
24000
22000
20000
18000
16000
14000
2004
2005
2006
2007
2008
Zone 15
Avg Load (kW)
80000
75000
70000
65000
60000
55000
50000
2004
2005
2006
2007
2008


Zone 16
Avg Load (kW)
45000
40000
35000
30000
25000
20000
2004
2005
2006
2007
2008
Zone 17
Avg Load (kW)
42500
40000
37500
35000
32500
30000
27500
25000
2004
2005
2006
2007
2008
Zone 18
Avg Load (kW)
320000
300000
280000
260000
240000
220000
200000
180000
160000
2004
2005
2006
2007
2008
Zone 19
Avg Load (kW)
110000
100000
90000
80000
70000
60000
2004
2005
2006
2007
2008
Zone 20
Avg Load (kW)
110000
100000
90000
80000
70000
2004
2005
2006
2007
2008
2
4
6
8
10
12

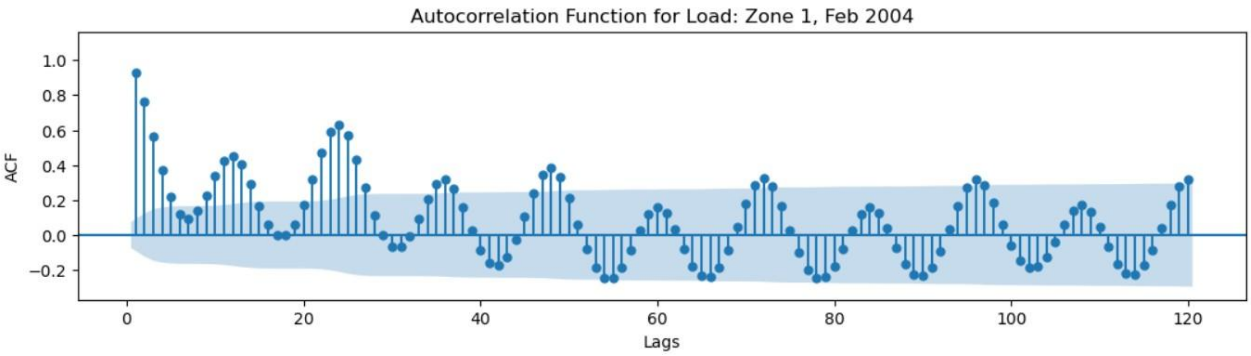
- **A.3** Monthly Average Hourly Load Distribution, Summer (Zone 1 - 2005)

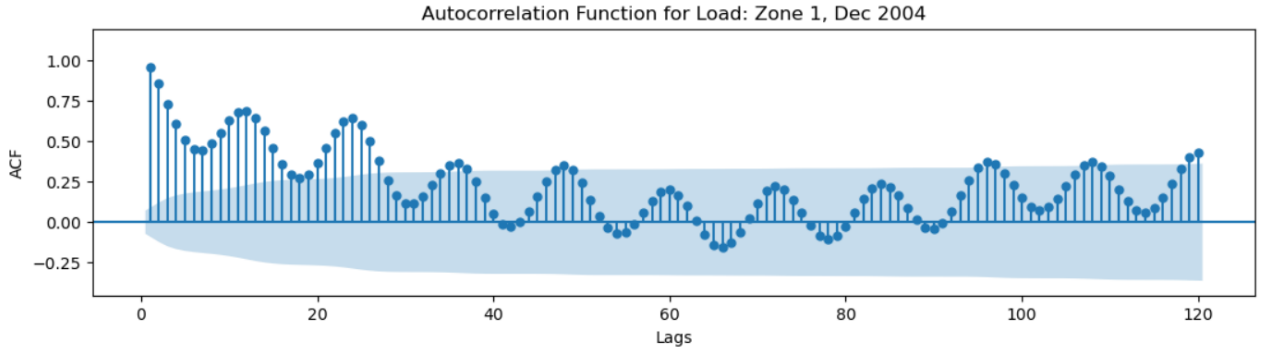


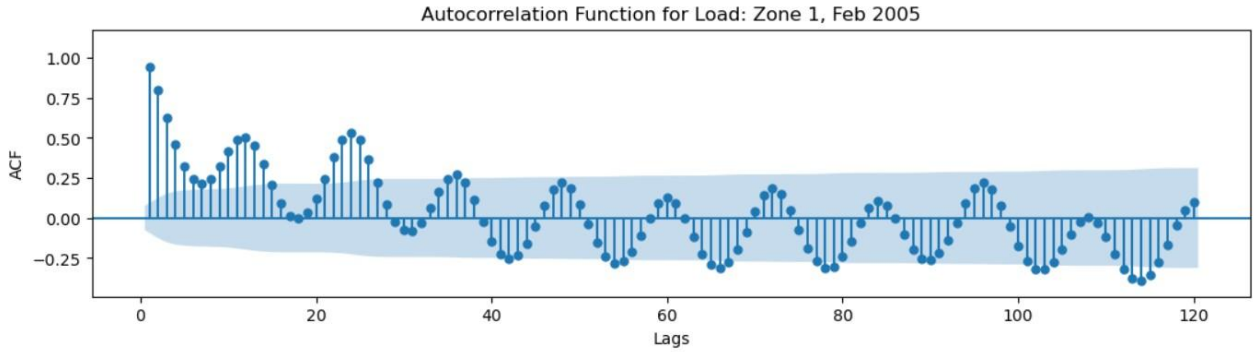
- **A.4** Monthly Average Hourly Load Distribution, Non Summer (Zone 1 - 2005)

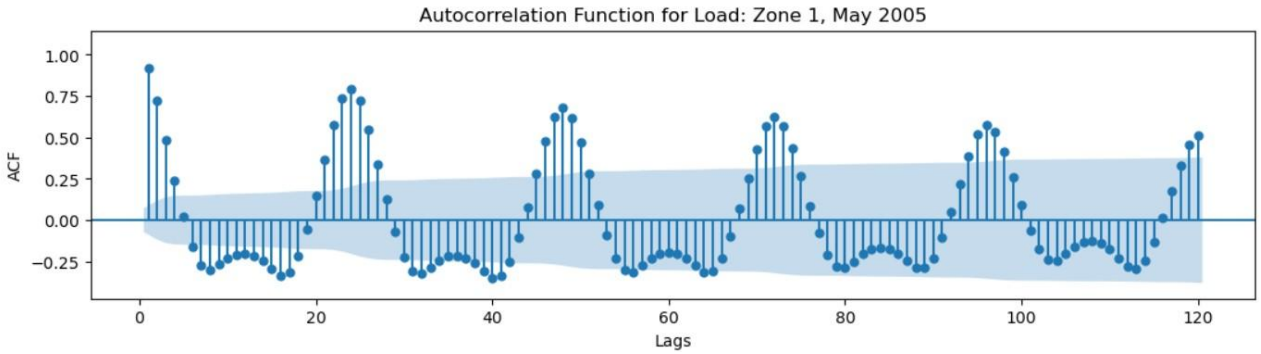


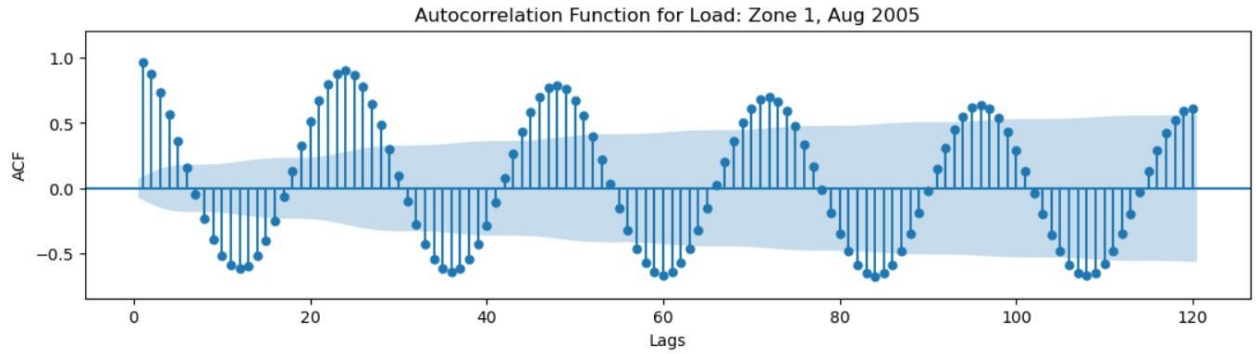
- **A.5** Monthly Autocorrelation Function for Zone 1, by Year

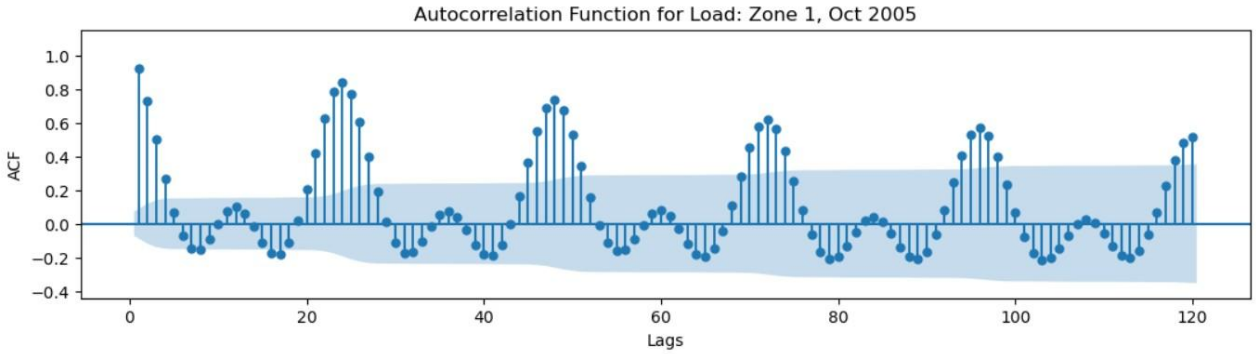


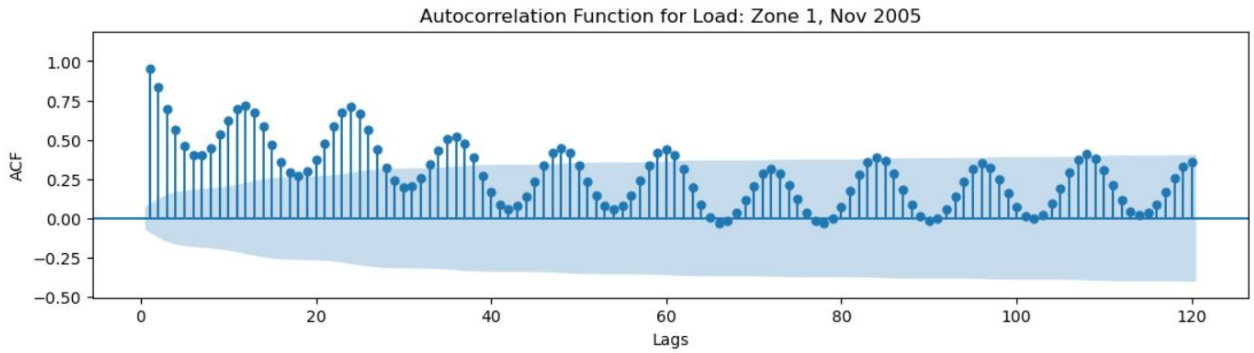

Autocorrelation Function for Load: Zone 1, Dec 2004
ACF
Lags


Autocorrelation Function for Load: Zone 1, Feb 2005
ACF
Lags

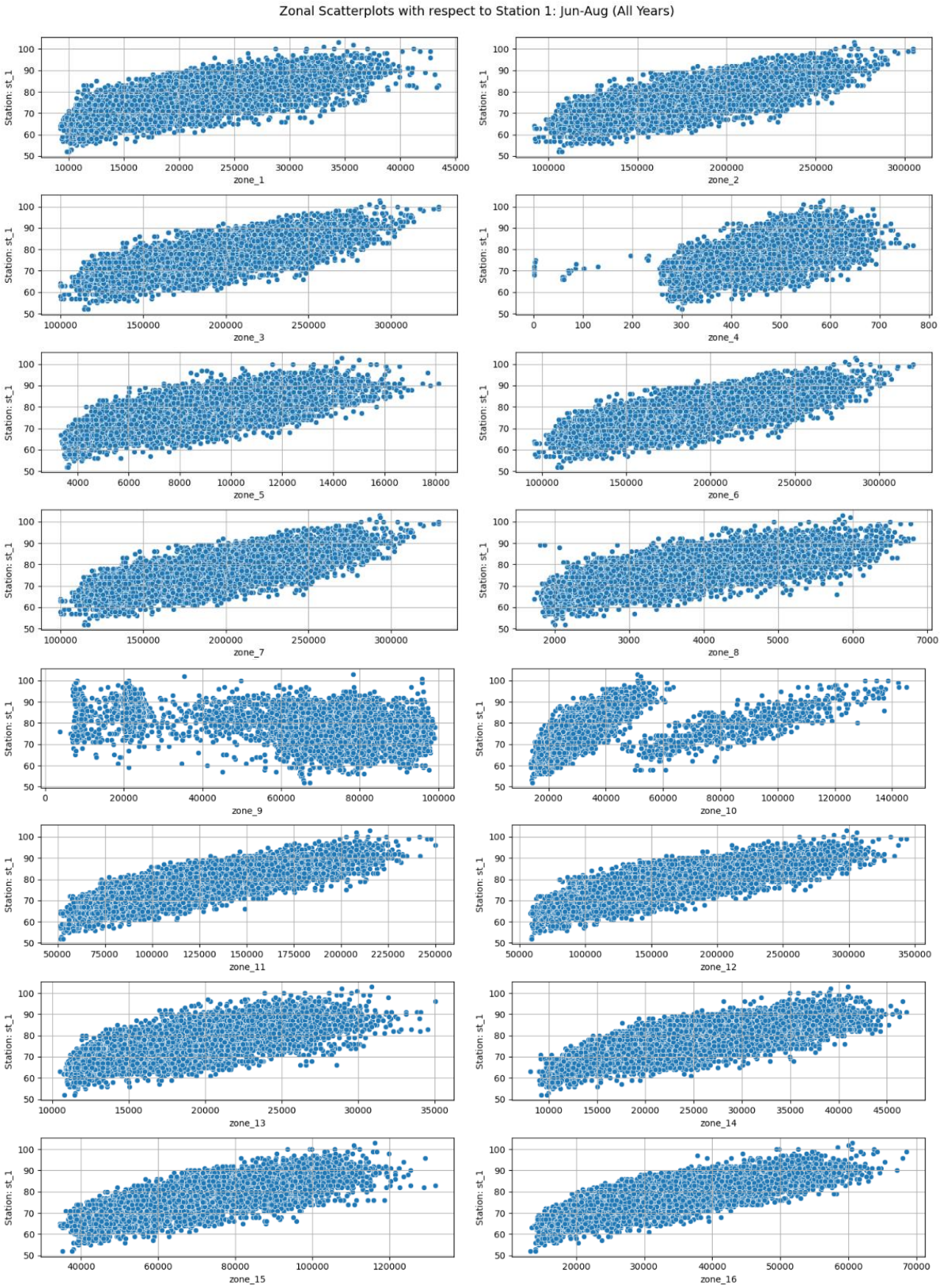

Autocorrelation Function for Load: Zone 1, May 2005
ACF
Lags

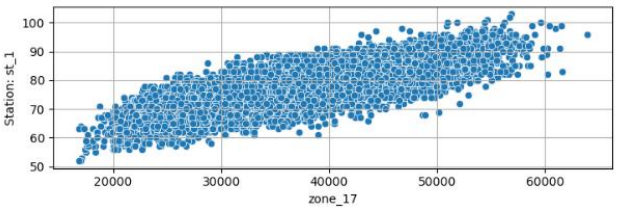

Autocorrelation Function for Load: Zone 1, Aug 2005
ACF
Lags

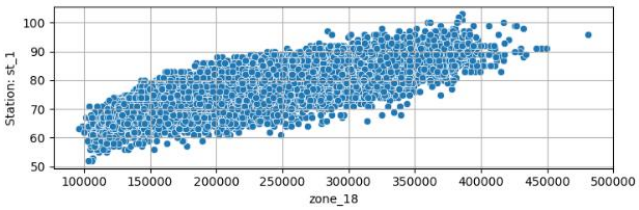


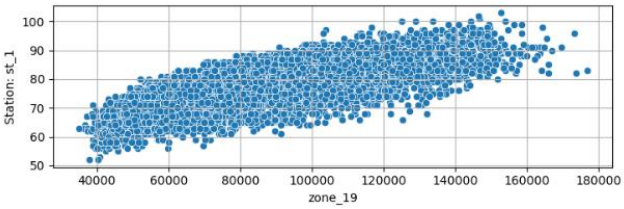


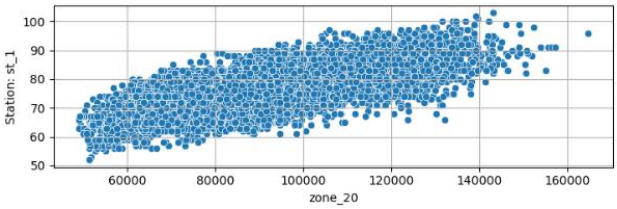
- **A.6** Zonal Scatterplots for all years with respect to Temperature Station 1, by month

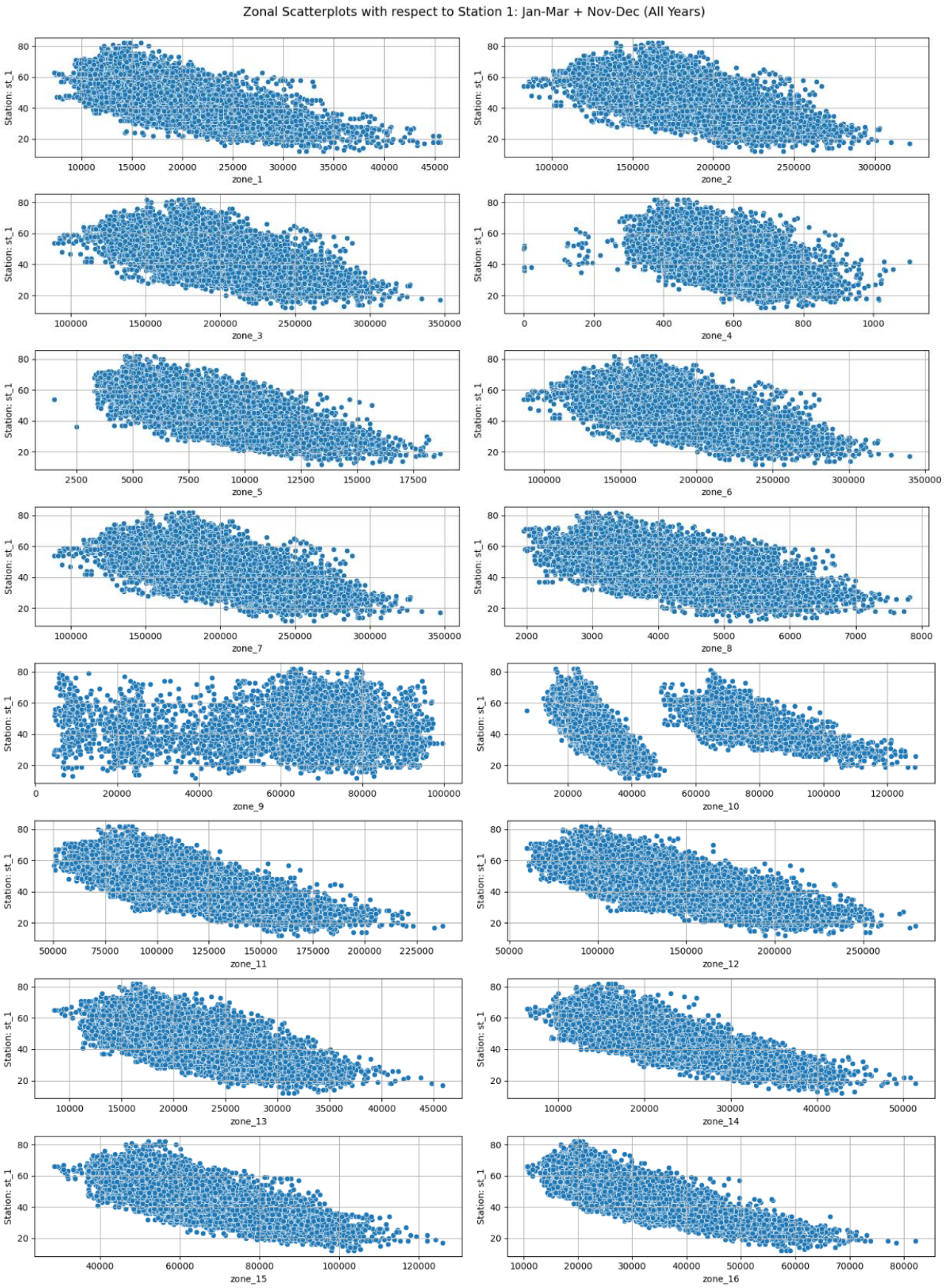

Zonal Scatterplots with respect to Station 1: Jun-Aug (All Years)
Station: st_1
zone_1
zone_2
zone_3
zone_4
zone_5
zone_6
zone_7
zone_8
zone_9
zone_10
zone_11
zone_12
zone_13
zone_14
zone_15
zone_16

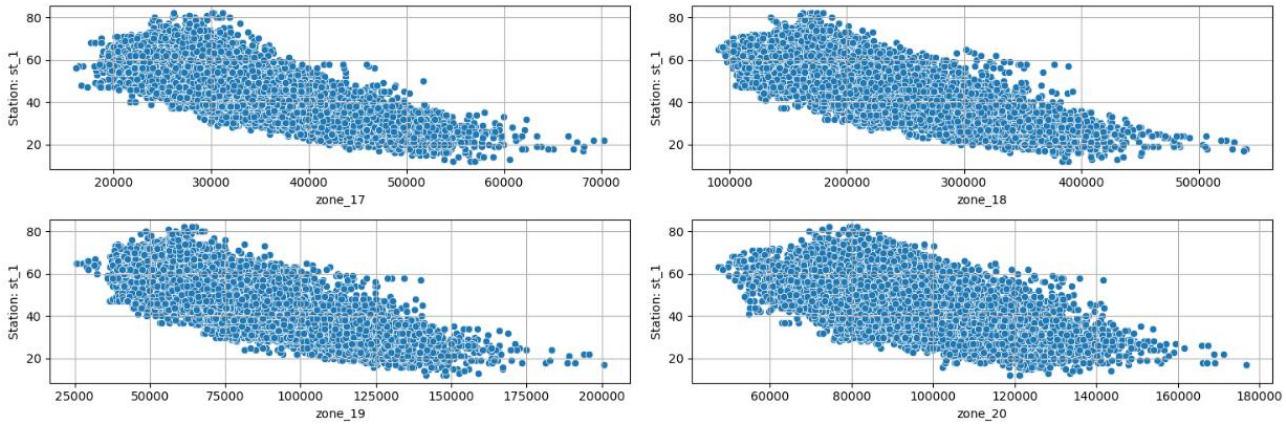

Station: st_1
zone_17


Station: st_1
zone_18


Station: st_1
zone_19


Station: st_1
zone_20


Zonal Scatterplots with respect to Station 1: Jan-Mar + Nov-Dec (All Years)
Station: st_1
zone_1
zone_2
zone_3
zone_4
zone_5
zone_6
zone_7
zone_8
zone_9
zone_10
zone_11
zone_12
zone_13
zone_14
zone_15
zone_16



- **A.7** Original weights assigned to each temperature station, according to 1st principal component

Station	Original Weight from PC1
1	0.28488634
2	0.29923833
3	0.30211105
4	0.29923997
5	0.30080121
6	0.29506554
7	0.31103738
8	0.30592595
9	0.30954738
10	0.30116938
11	0.30672135
SUM	3.315743891550456

- **A.8** Rescaled weights assigned to each temperature station, based on 1st principal component

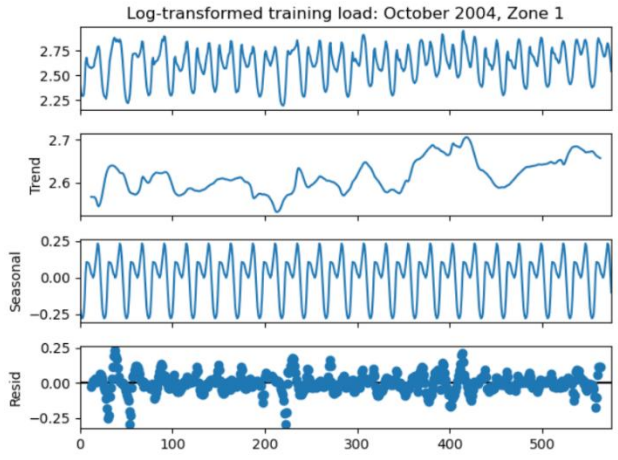
Station	Re-scaled Weight from PC1
1	0.08591928
2	0.09024772
3	0.09111411
4	0.09024822
5	0.09071907
6	0.08898924
7	0.09380621
8	0.09226465
9	0.09335684
10	0.09083011
11	0.09250454
SUM	1.0000000000000002

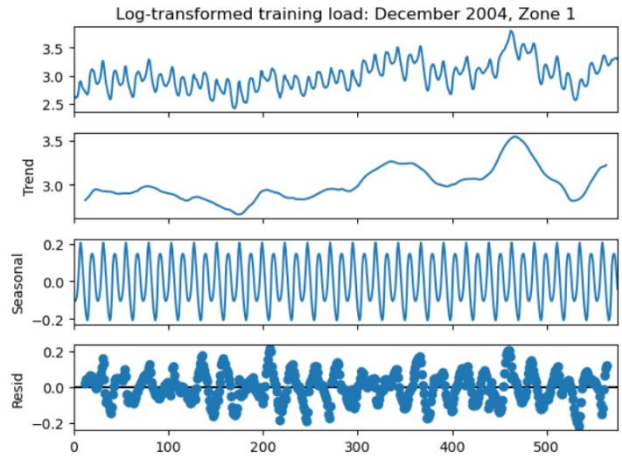
- **A.9** Naïve methods' performance scores for each test set, by individual month

**Testing in January 2004*				
Performance Metric	**Repeating Last Cycle**	**Seasonal Naïve**	**Overall Average Load (Week Prior)**	**Hourly Average Load (Week Prior)**
RMSE	0.25	0.277	0.215	0.204
R2 Score	-1.056	-1.515	-0.516	-0.374
**Testing in April 2004*				
Performance Metric	**Repeating Last Cycle**	**Seasonal Naïve**	**Overall Average Load (Week Prior)**	**Hourly Average Load (Week Prior)**
RMSE	0.162	0.155	0.192	0.151
R2 Score	0.225	0.295	-0.080	0.326
**Testing in July 2004*				
Performance Metric	**Repeating Last Cycle**	**Seasonal Naïve**	**Overall Average Load (Week Prior)**	**Hourly Average Load (Week Prior)**
RMSE	0.106	0.122	0.281	0.109

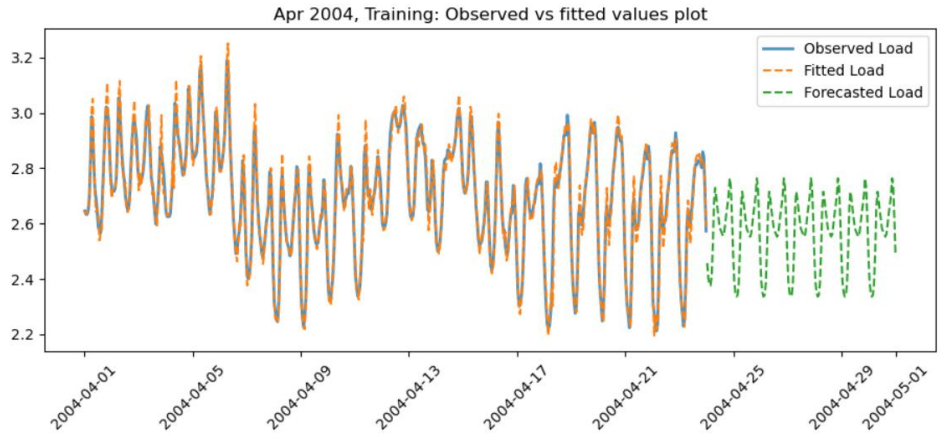
R2 Score	0.856	0.808	-0.001	0.847
**Testing in October 2004*				
Performance Metric	**Repeating Last Cycle**	**Seasonal Naïve**	**Overall Average Load (Week Prior)**	**Hourly Average Load (Week Prior)**
RMSE	0.091	0.119	0.173	0.090
R2 Score	0.705	0.497	-0.068	0.711
**Testing in December 2004*				
Performance Metric	**Repeating Last Cycle**	**Seasonal Naïve**	**Overall Average Load (Week Prior)**	**Hourly Average Load (Week Prior)**
RMSE	0.235	0.251	0.224	0.191
R2 Score	-0.183	-0.352	-0.075	0.220

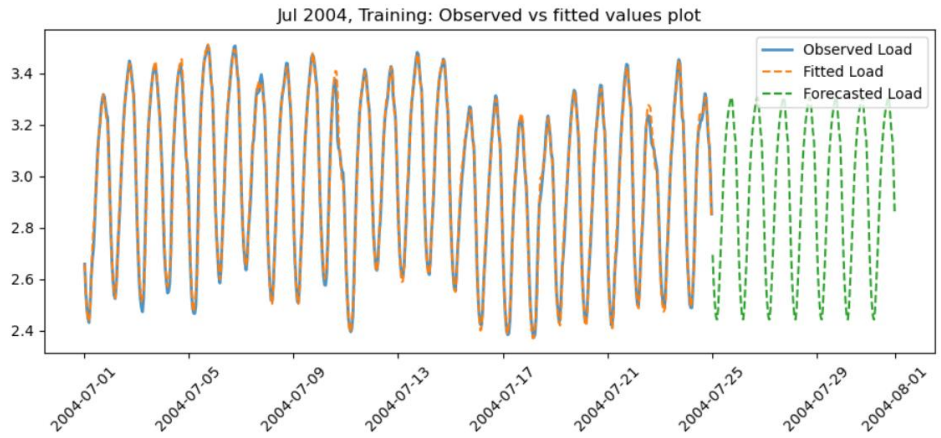
- **A.10** Monthly Seasonal Decomposition Plots for training sessions (Zone 1)

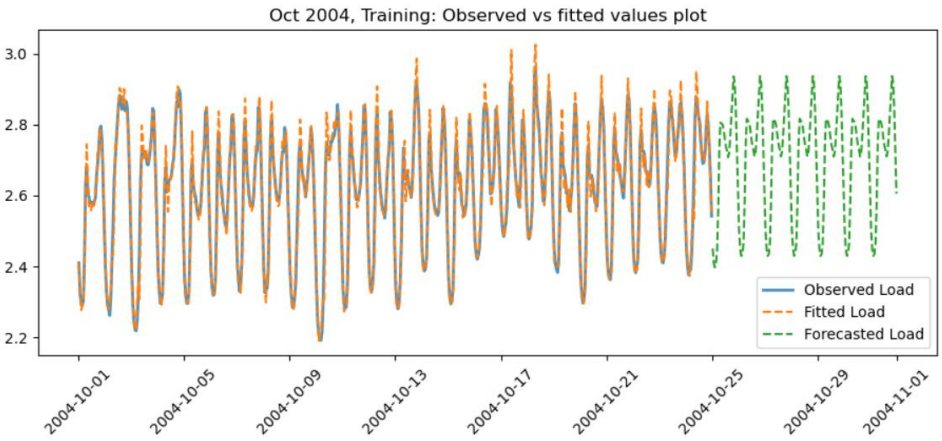


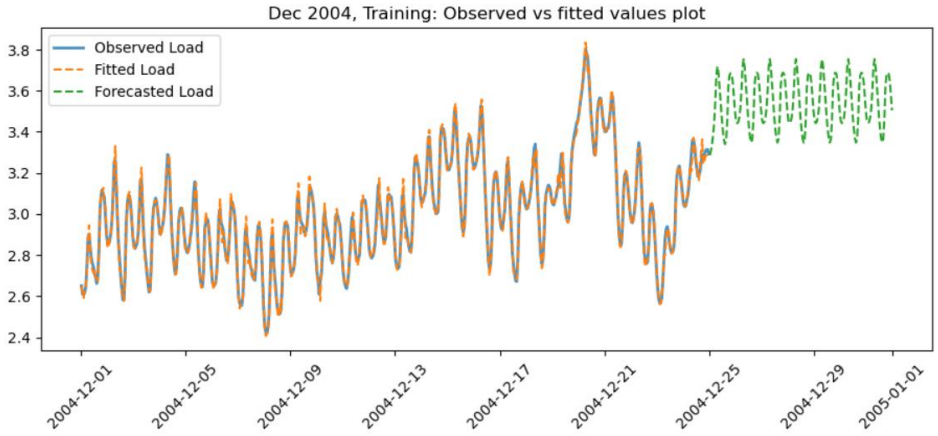


- **A.11** Observed versus fitted values plot for Zone 1, by month





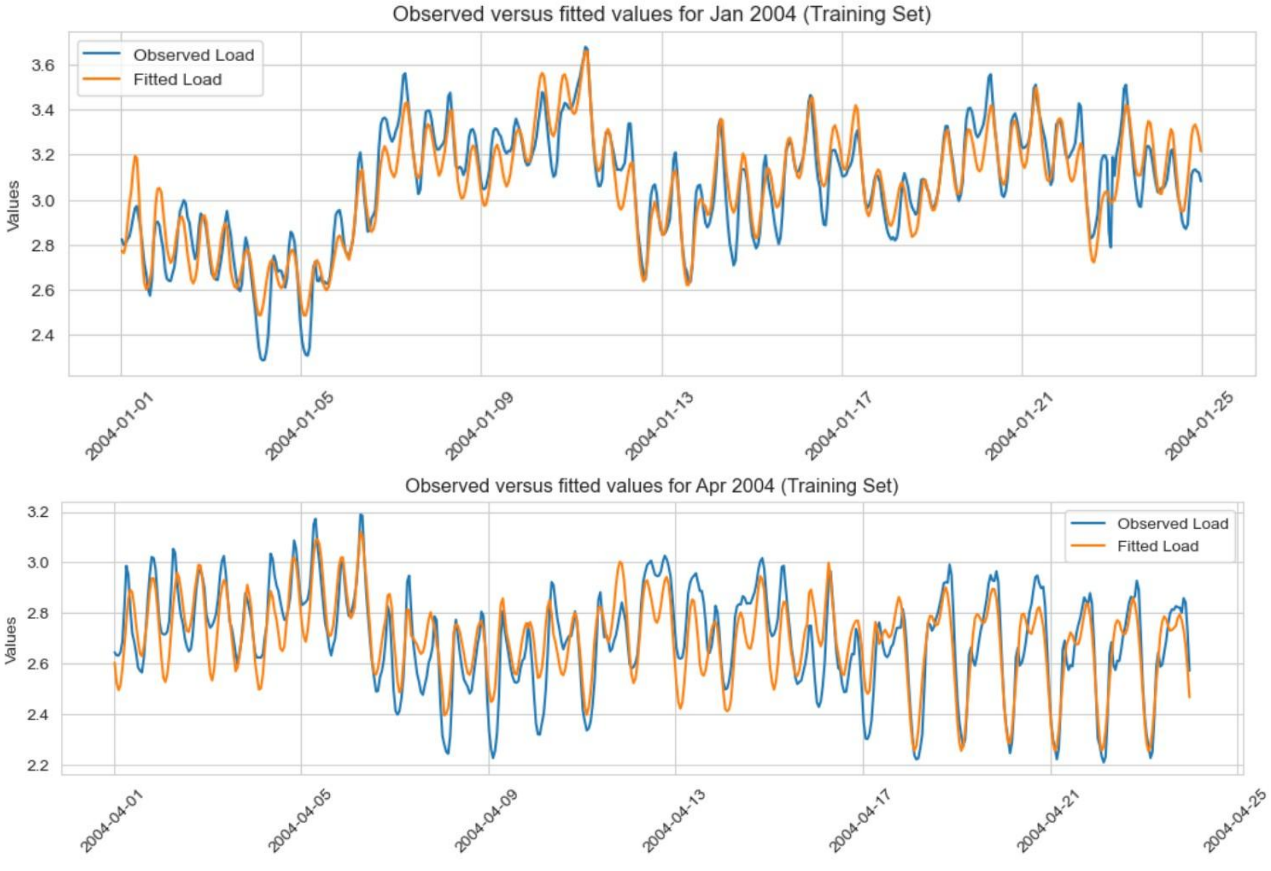


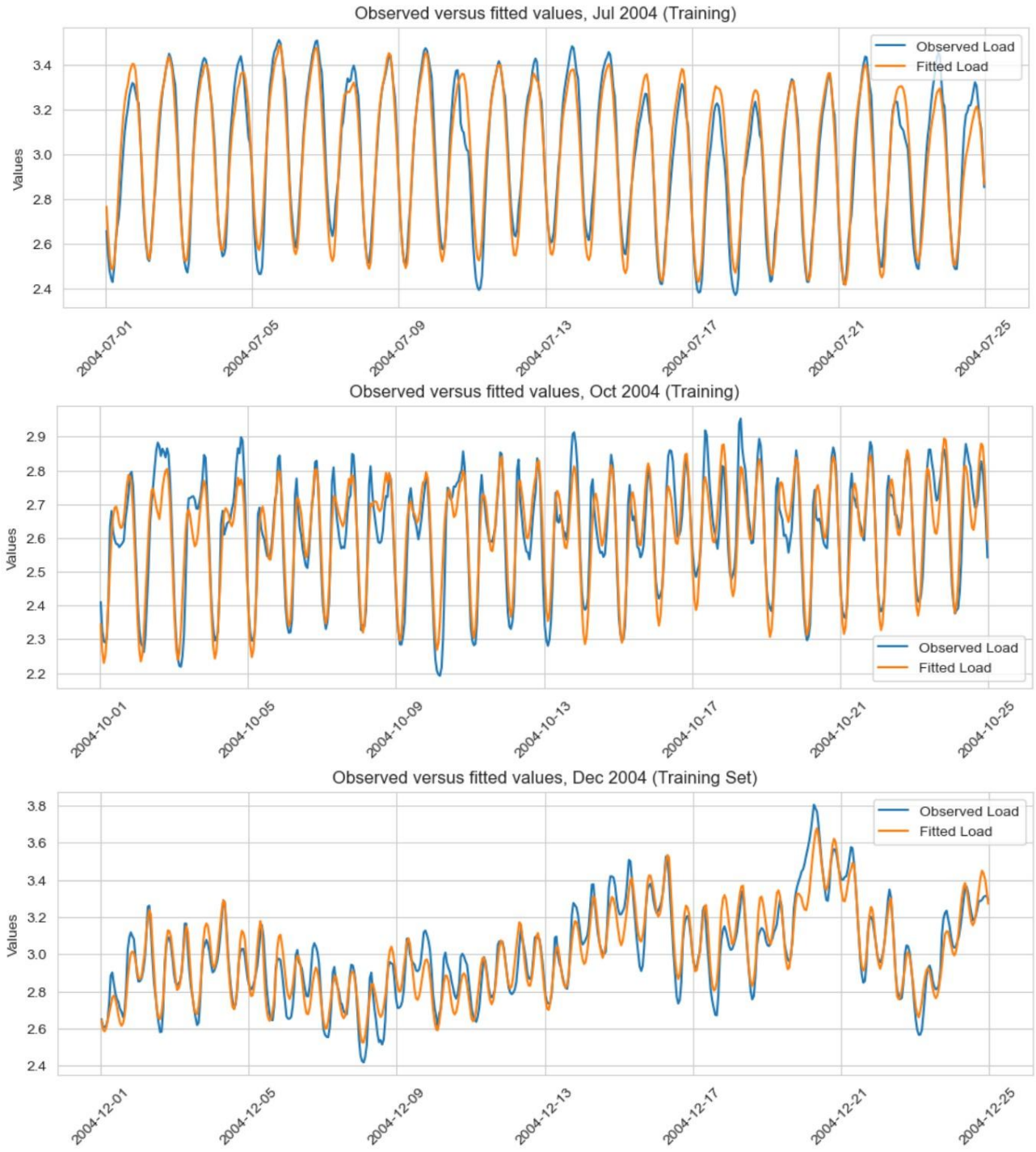


- **A.12** Training R2 scores for number of harmonics considered in training, by month

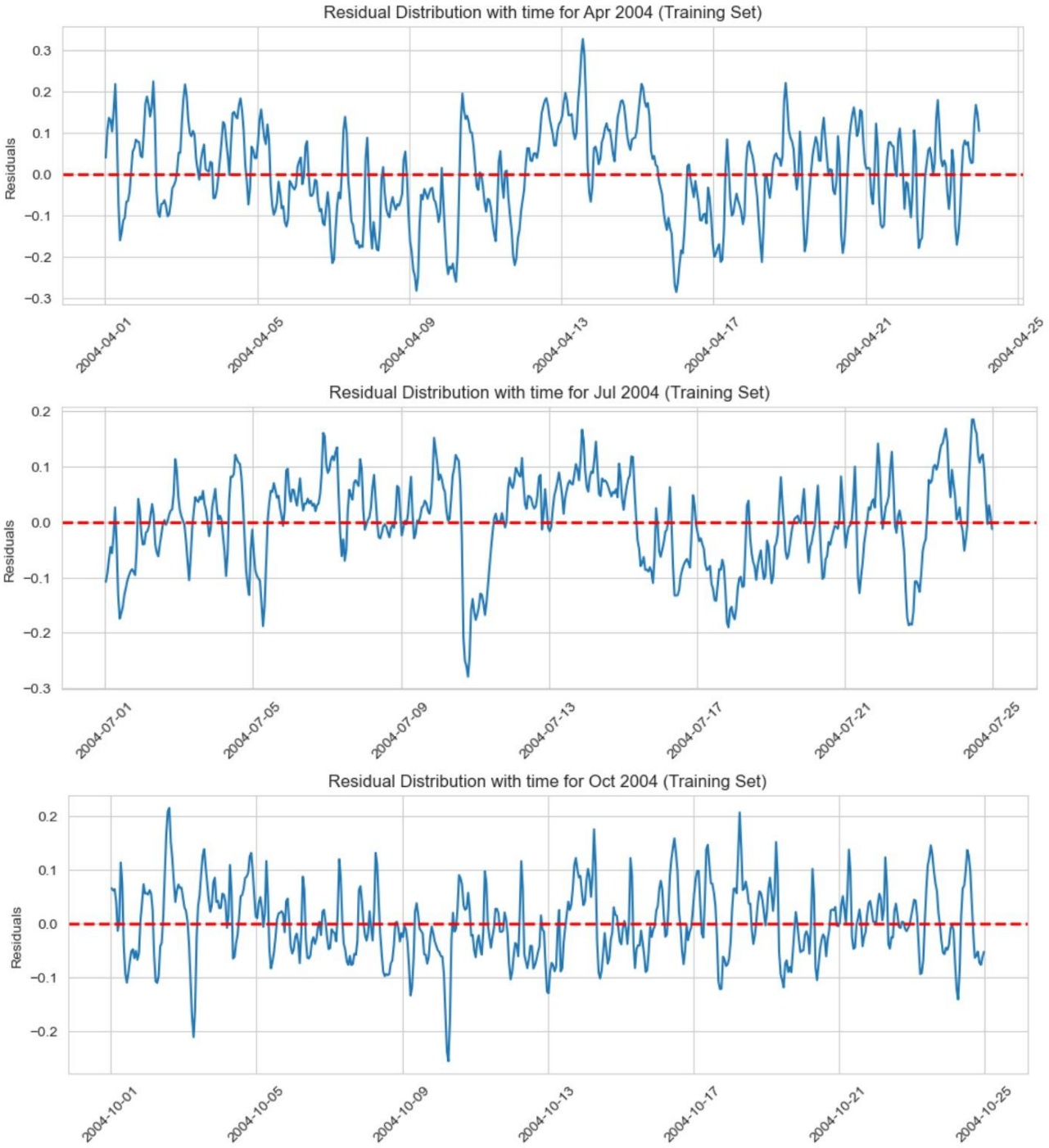
Training Set in Month	1 Harmonic	2 Harmonics	3 Harmonics	4 Harmonics
January 2004	0.191	0.338	0.339	0.344
April 2004	0.166	0.427	0.442	0.446
July 2004	0.893	0.920	0.920	0.921
October 2004	0.390	0.795	0.798	0.806
December 2004	0.240	0.464	0.465	0. 468

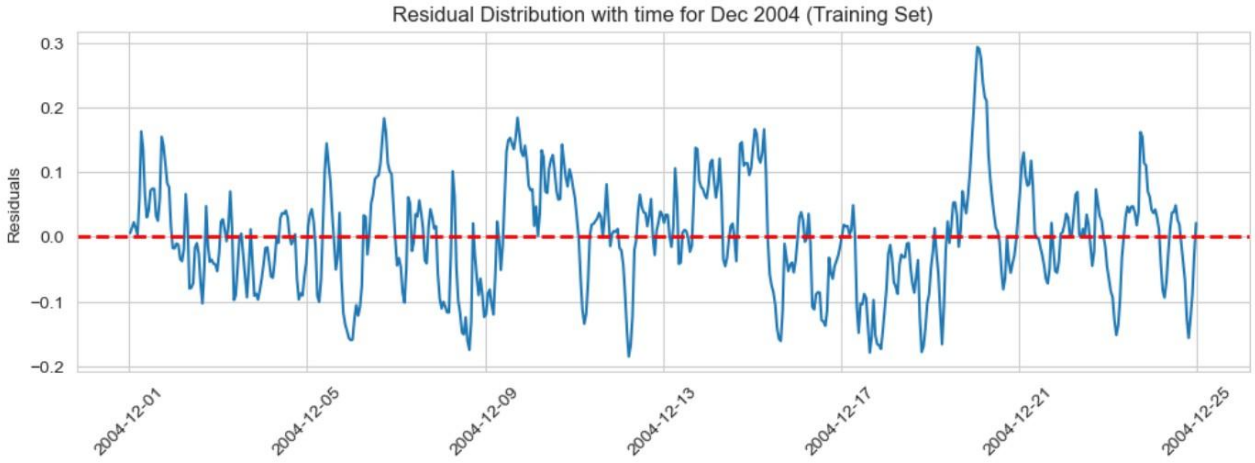
- **A.13** Piecewise Linear Regression, observed versus fitted values plot on training set (by month)



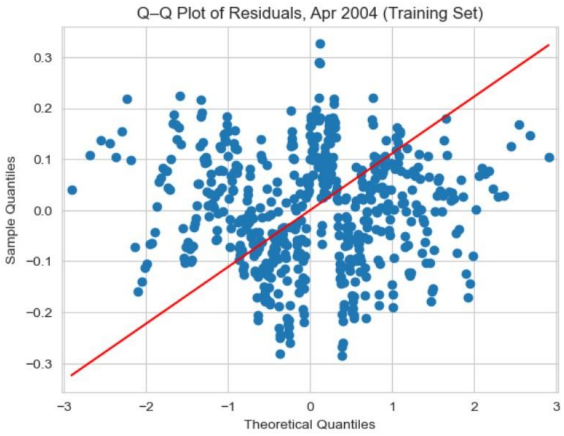

Observed versus fitted values, Jul 2004 (Training)
Observed Load
Fitted Load
Values
3.4
3.2
3.0
2.8
2.6
2.4
2004-07-01
2004-07-05
2004-07-09
2004-07-13
2004-07-17
2004-07-21
2004-07-25
Observed versus fitted values, Oct 2004 (Training)
Values
2.9
2.8
2.7
2.6
2.5
2.4
2.3
2.2
Observed Load
Fitted Load
2004-10-01
2004-10-05
2004-10-09
2004-10-13
2004-10-17
2004-10-21
2004-10-25
Observed versus fitted values, Dec 2004 (Training Set)
Values
3.8
3.6
3.4
3.2
3.0
2.8
2.6
2.4
Observed Load
Fitted Load
2004-12-01
2004-12-05
2004-12-09
2004-12-13
2004-12-17
2004-12-21
2004-12-25

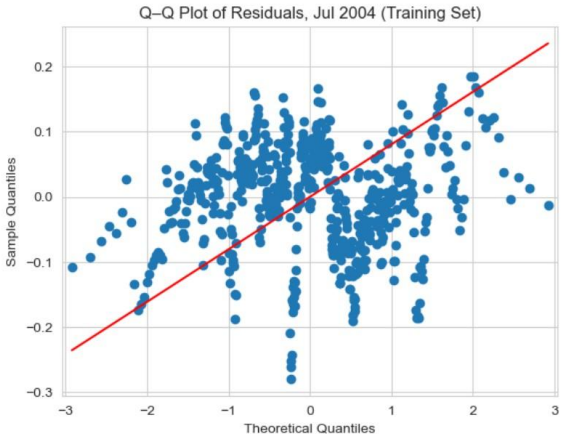
- **A.14** Piecewise Linear Regression, residuals versus time plots on training set (by month)

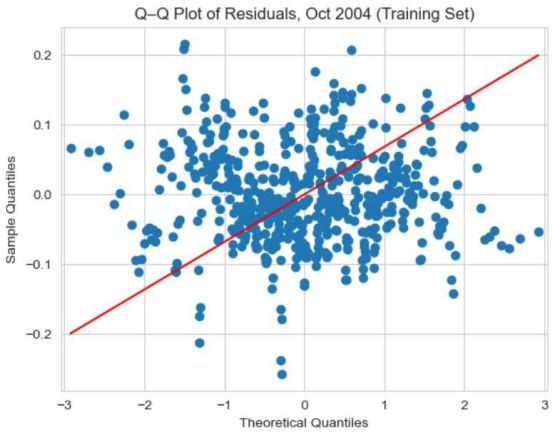


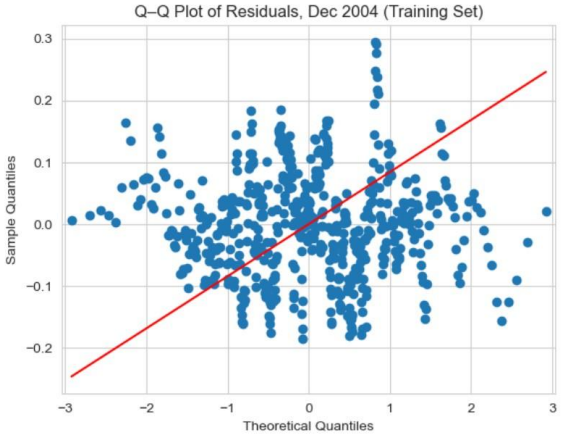


- **A.15** Piecewise Linear Regression, QQ plots for residuals on training set (by month)





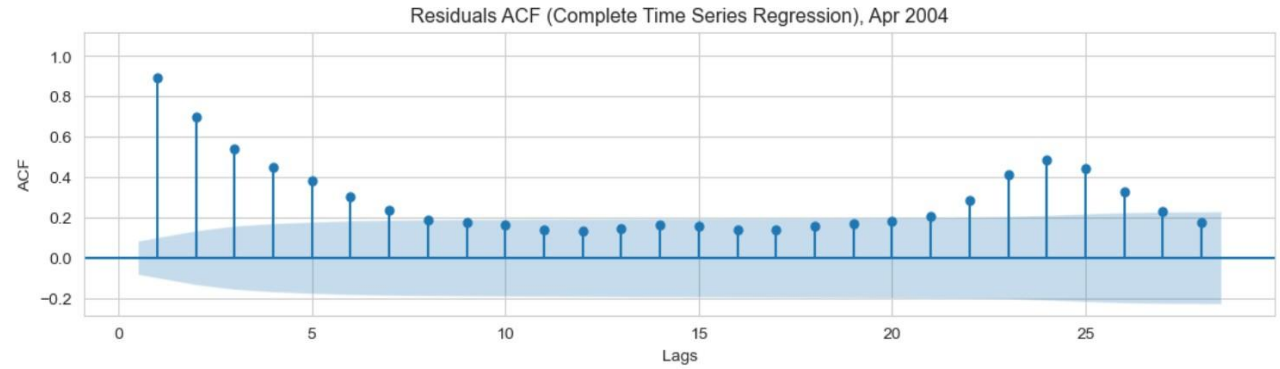


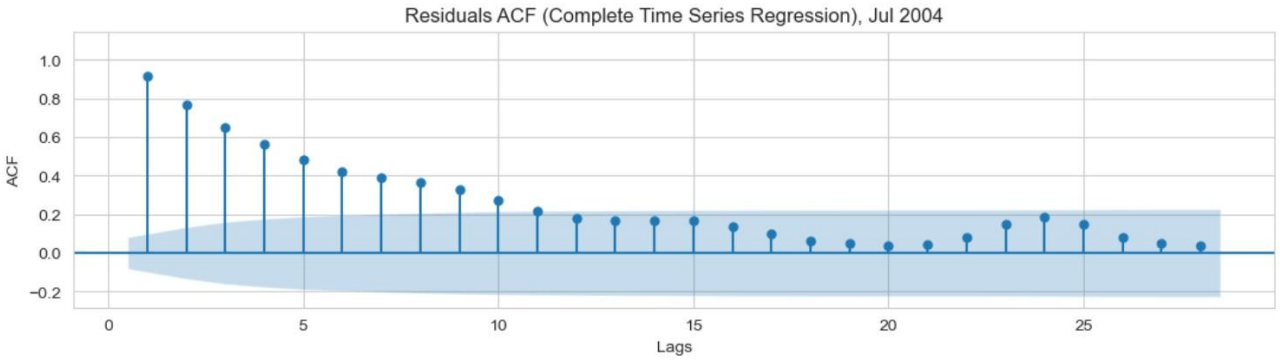


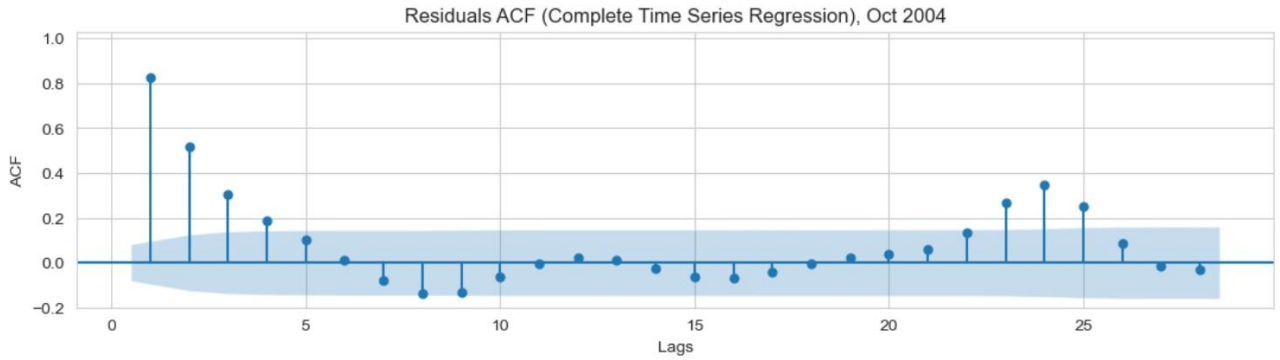
- **A.16** Piecewise Linear Regression, ADF test results for residuals on training set (by month)

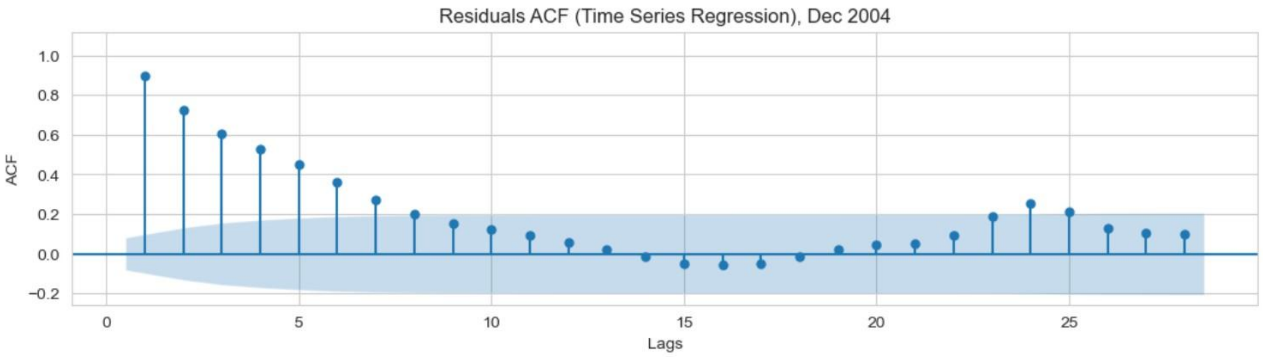
Training Set	**ADF Statistic**	**P-Value**	**Stationary if: p-value < 0.05**
January 2004	-6.201492952767946	5.7925847580140435e-08	Stationary
April 2004	-4.0394675264219915	0.0012173790858884307	Stationary
July 2004	-4.476584315085787	0.00021671089701973028	Stationary
October 2004	-6.867219304421103	1.5480492867748033e-09	Stationary
December 2004	-4.133425376646943	0.0008523864475285511	Stationary

- **A.17** Piecewise Linear Regression, ACF plots for residuals on training set (by month)

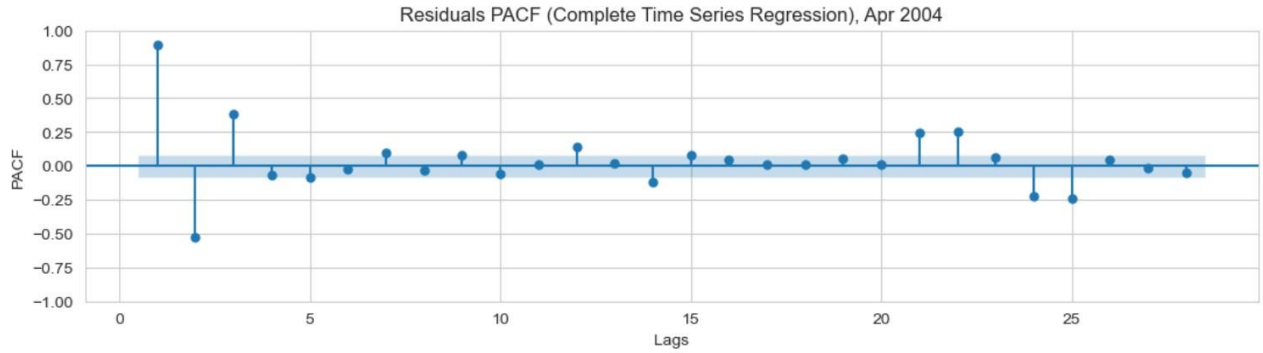


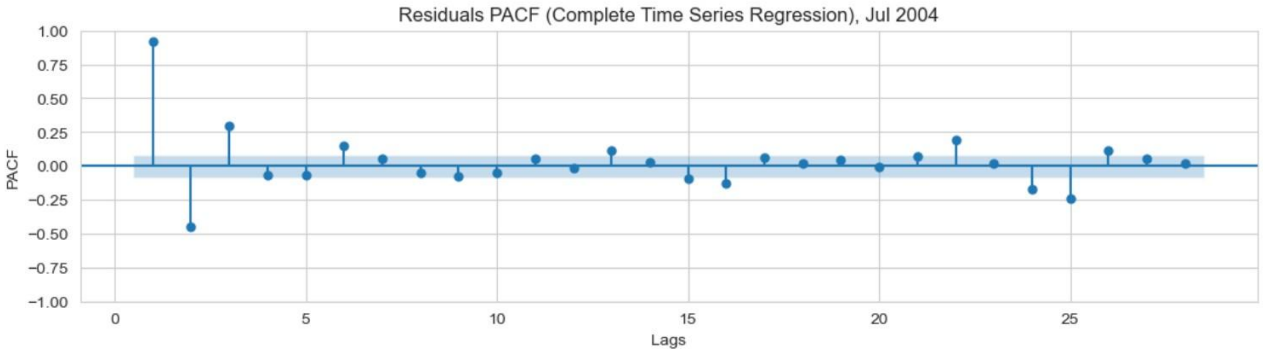


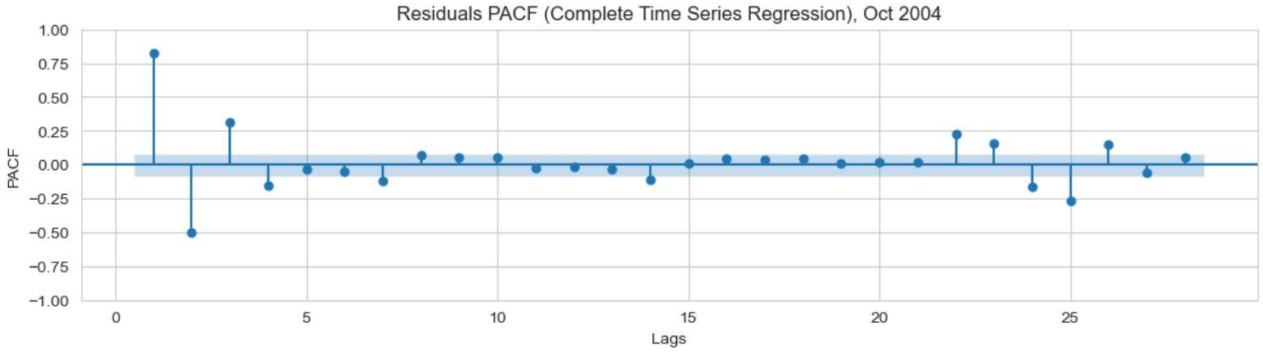


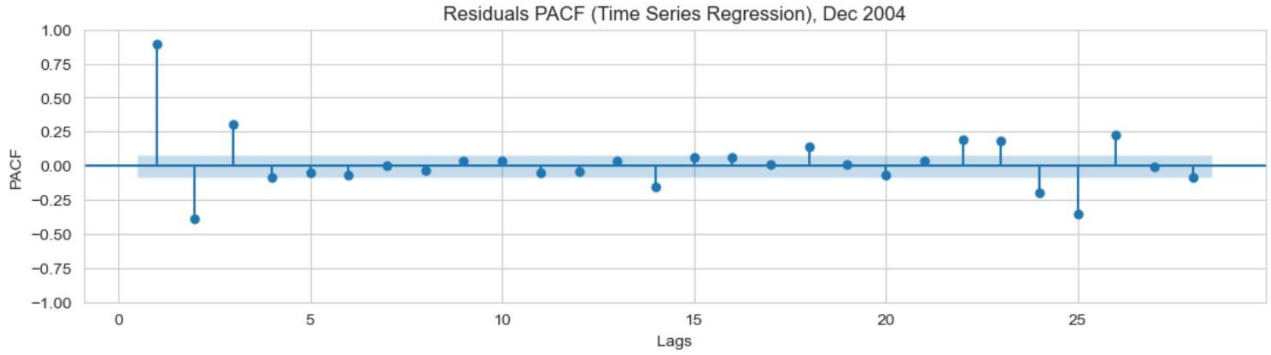


- **A.18** Piecewise Linear Regression, PACF plots for residuals on training set (by month)

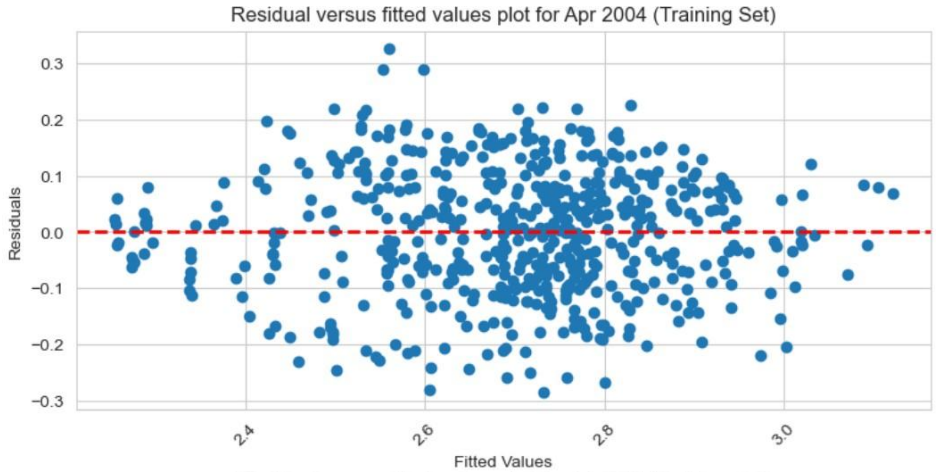


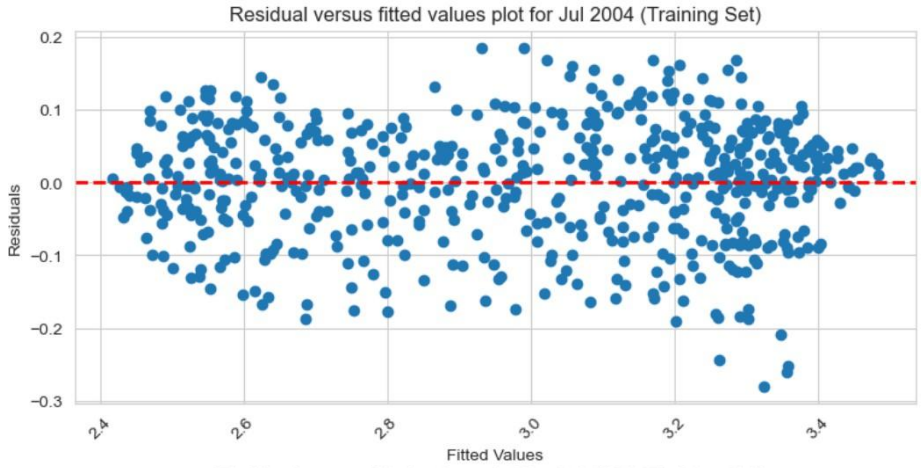


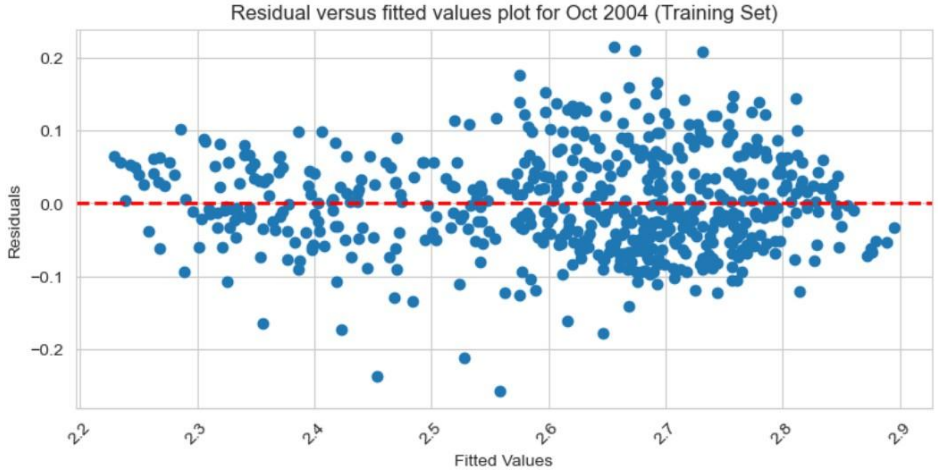


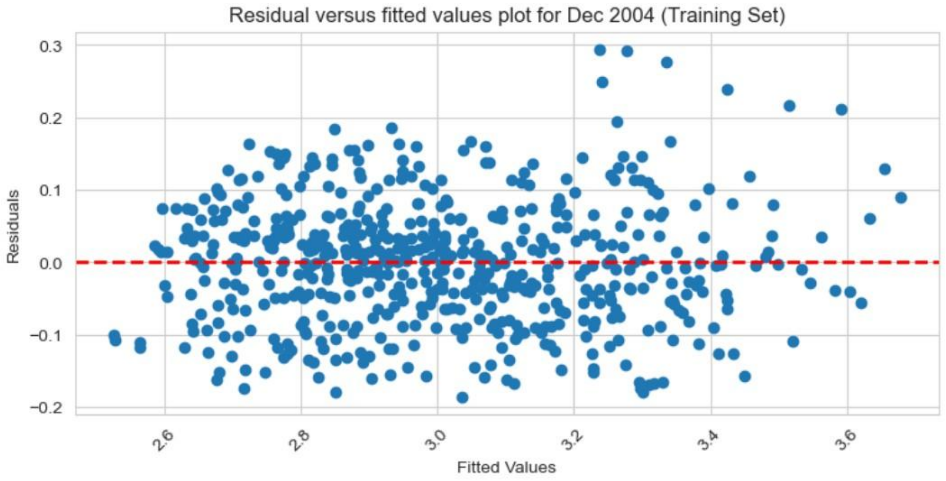


- **A.19** Piecewise Linear Regression, residuals vs fitted values plot on training set (by month)





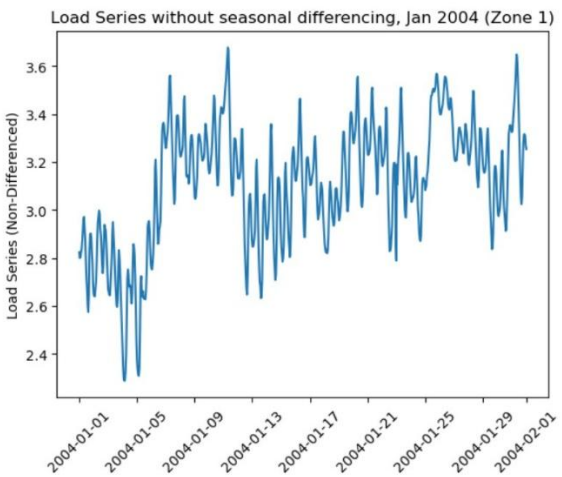


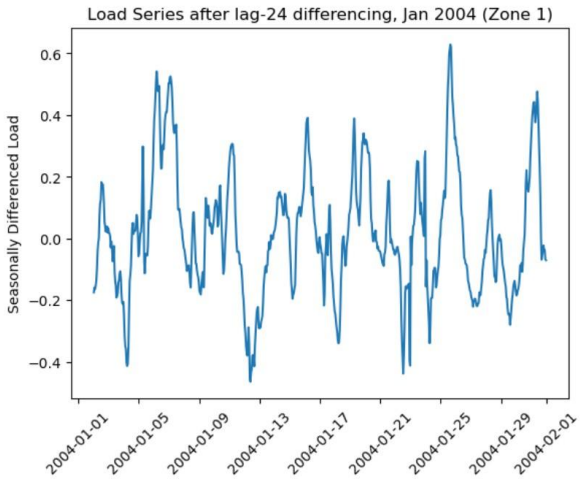


- **A.20** ADF test results, before and after *lag-24* seasonal differencing (adjusted for linear trend)

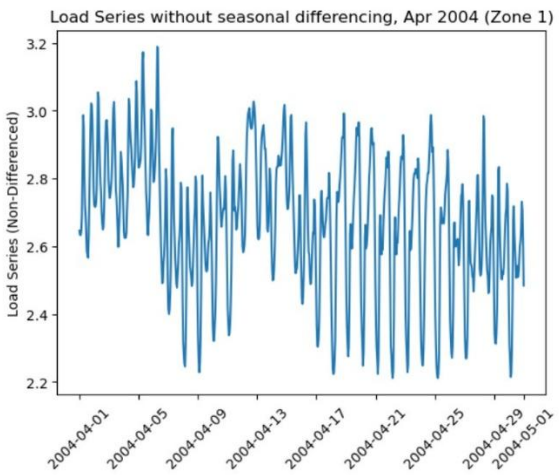
	Jan, 2004	**Apr, 2004**	**Jul, 2004**	**Oct, 2004**	**Dec, 2004**
ADF statistic (before differencing)	-2.355	-1.551	-1.800	-3.251	-1.790
p-value of test statistic (before differencing)	0.154	0.508	0.380	0.017	0.385
Conclusion: α = 0.05 (before differencing)	Non-Stationary	Non-Stationary	Non-Stationary	Stationary	Non-Stationary
ADF statistic (after differencing)	-4.695	-6.376	-7.065	-8.642	-4.581
p-value of test statistic (after differencing)	0.0000	0.0000	0.0000	0.0000	0.0001
Conclusion: α = 0.05 (after differencing)	Stationary	Stationary	Stationary	Stationary	Stationary

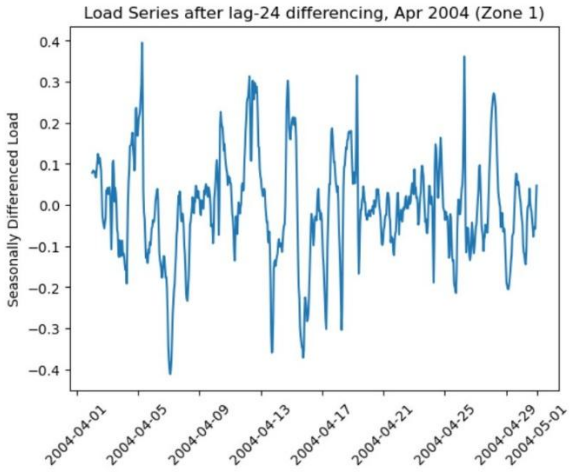
- **A.21** Load Distribution, before and after seasonal differencing (Jan 2004 – Zone 1)



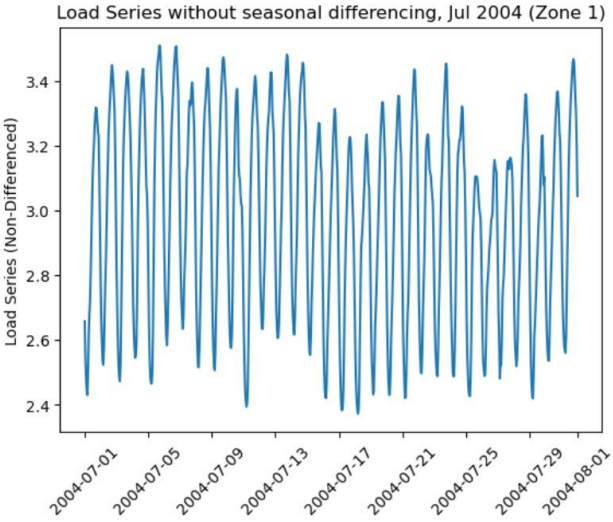


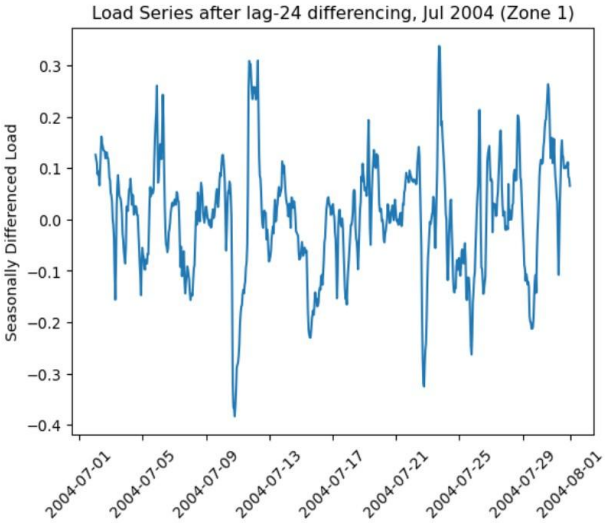
- **A.22** Load Distribution, before and after seasonal differencing (Apr 2004 – Zone 1)



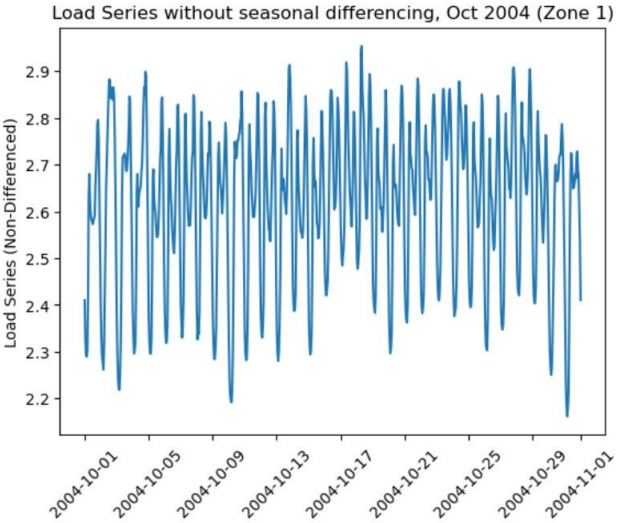


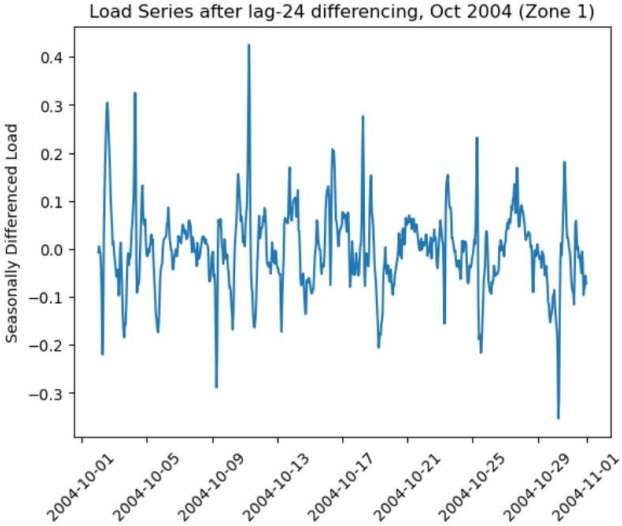
- **A.23** Load Distribution, before and after seasonal differencing (Jul 2004 – Zone 1)



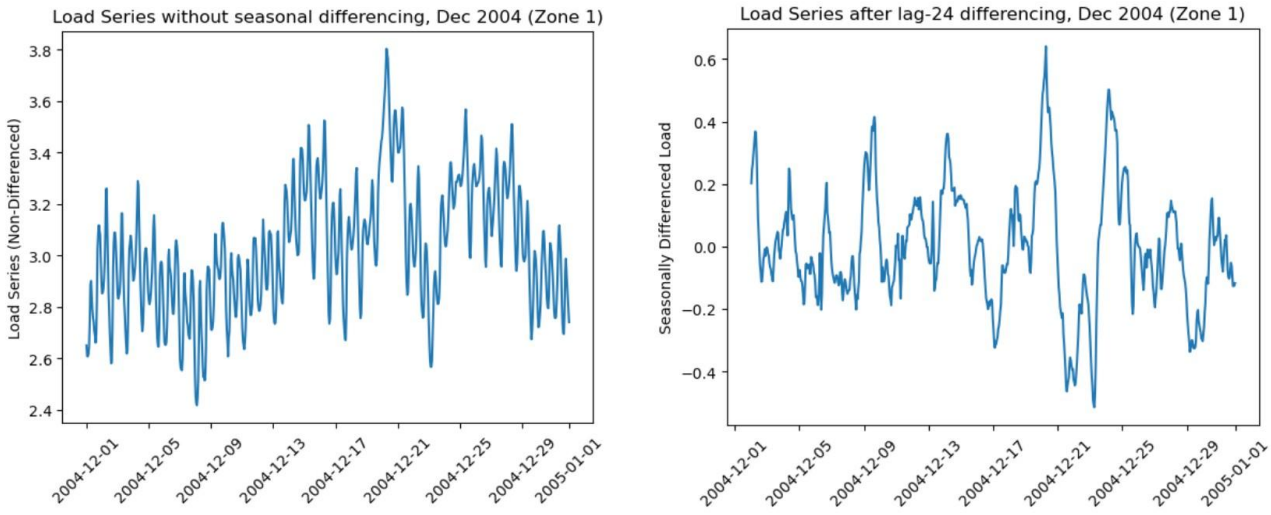


- **A.24** Load Distribution, before and after seasonal differencing (Oct 2004 – Zone 1)

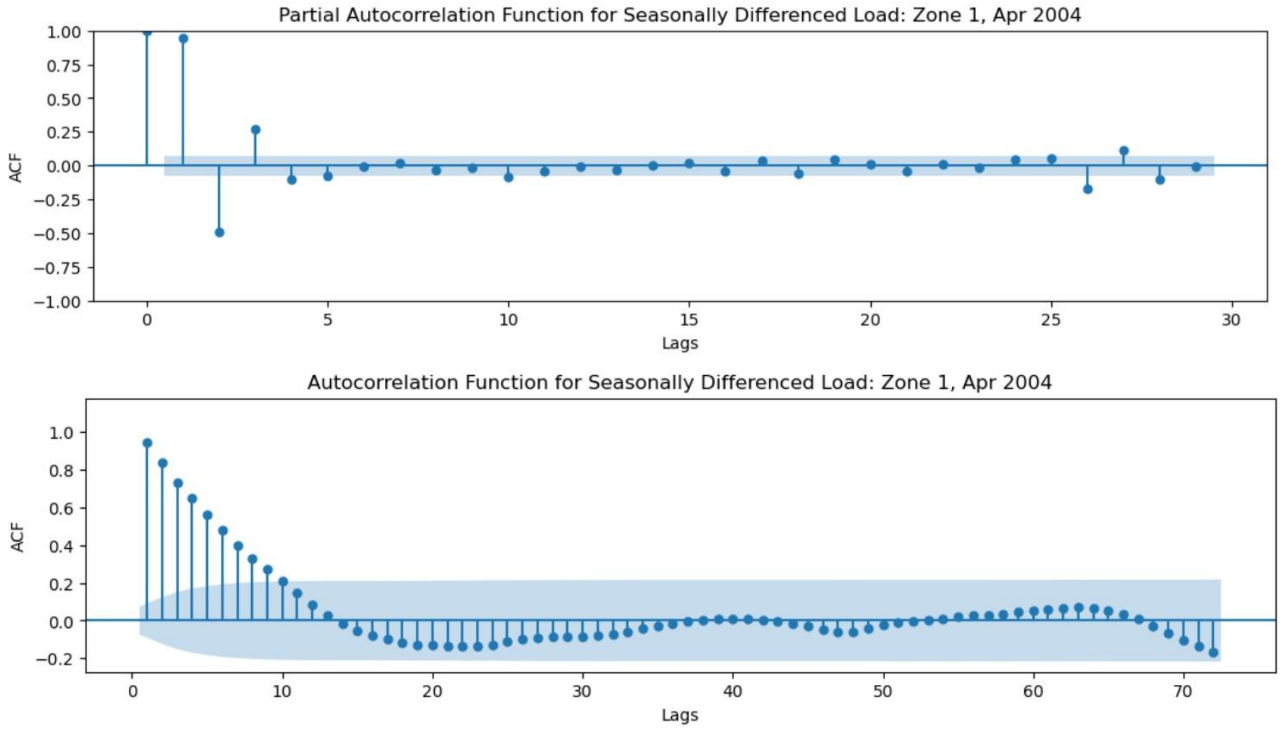




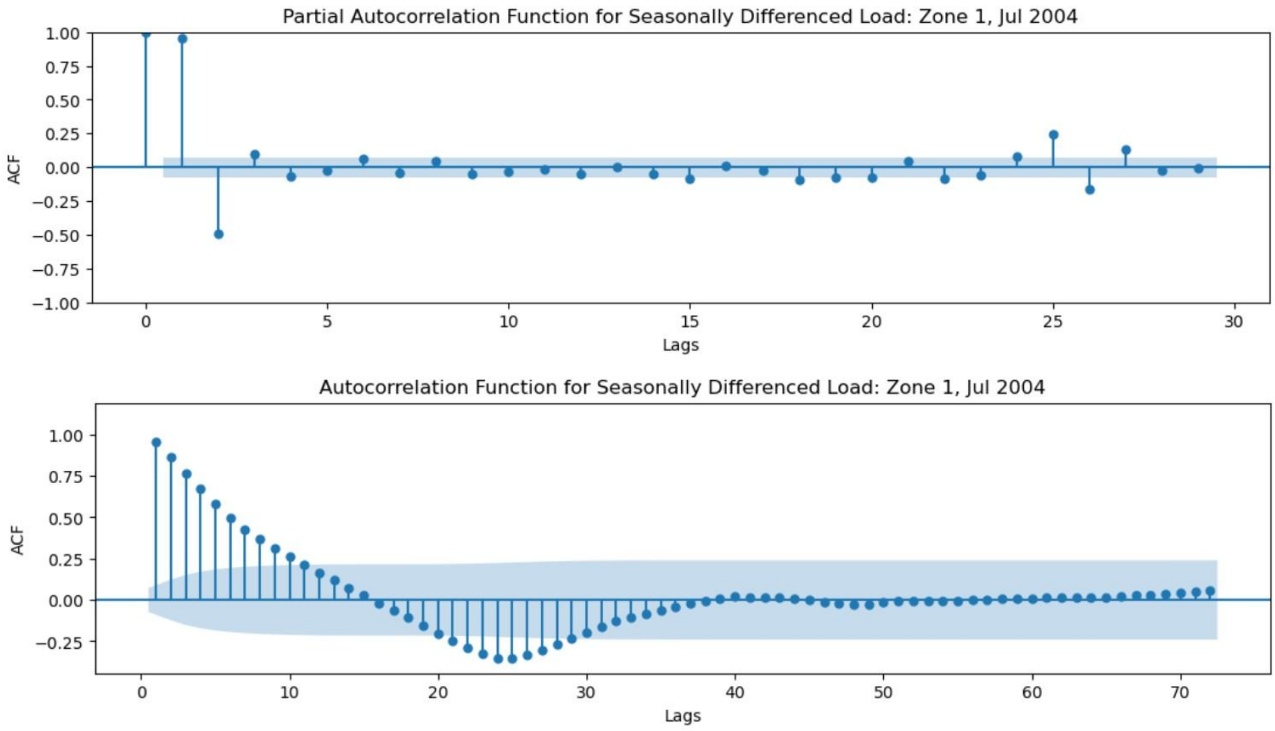
- **A.25** Load Distribution, before and after seasonal differencing (Dec 2004 – Zone 1)



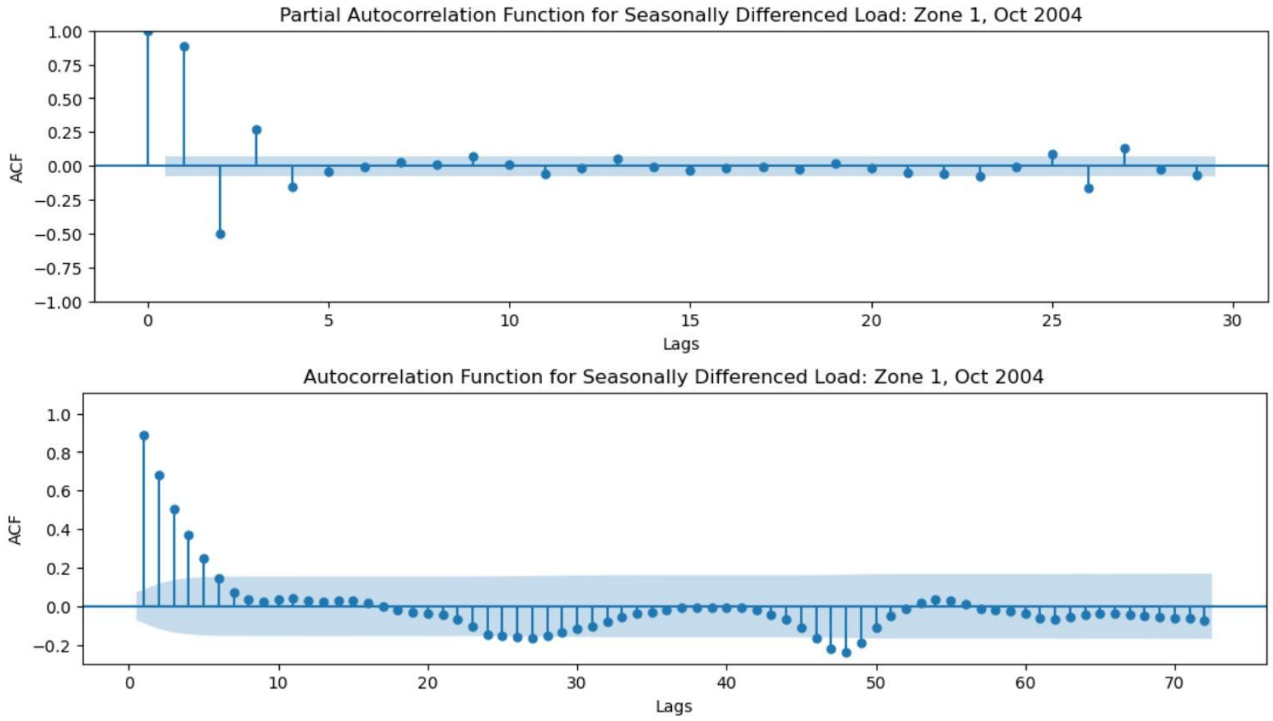
- **A.26** ACF & PACF plots for seasonally differenced data (Apr 2004 – Zone 1)



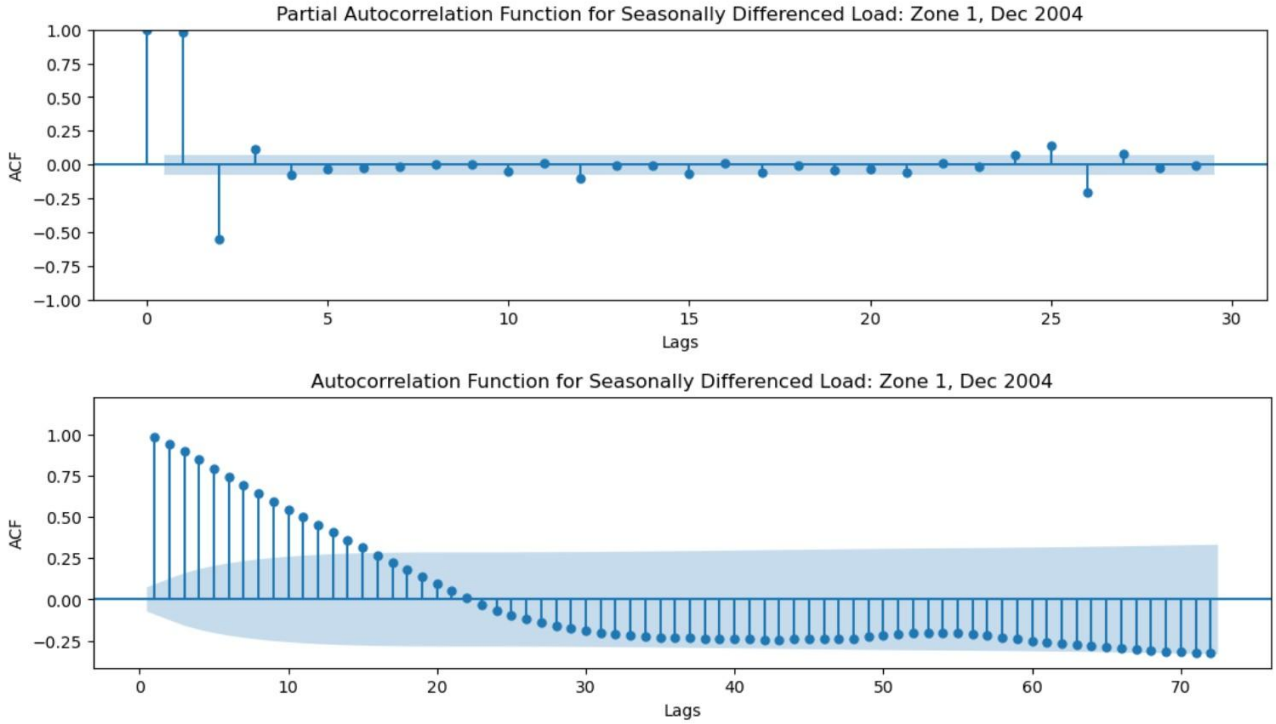
- **A.27** ACF & PACF plots for seasonally differenced data (Jul 2004 – Zone 1)



- **A.28** ACF & PACF plots for seasonally differenced data (Oct 2004 – Zone 1)



- **A.29** ACF & PACF plots for seasonally differenced data (Dec 2004 – Zone 1)

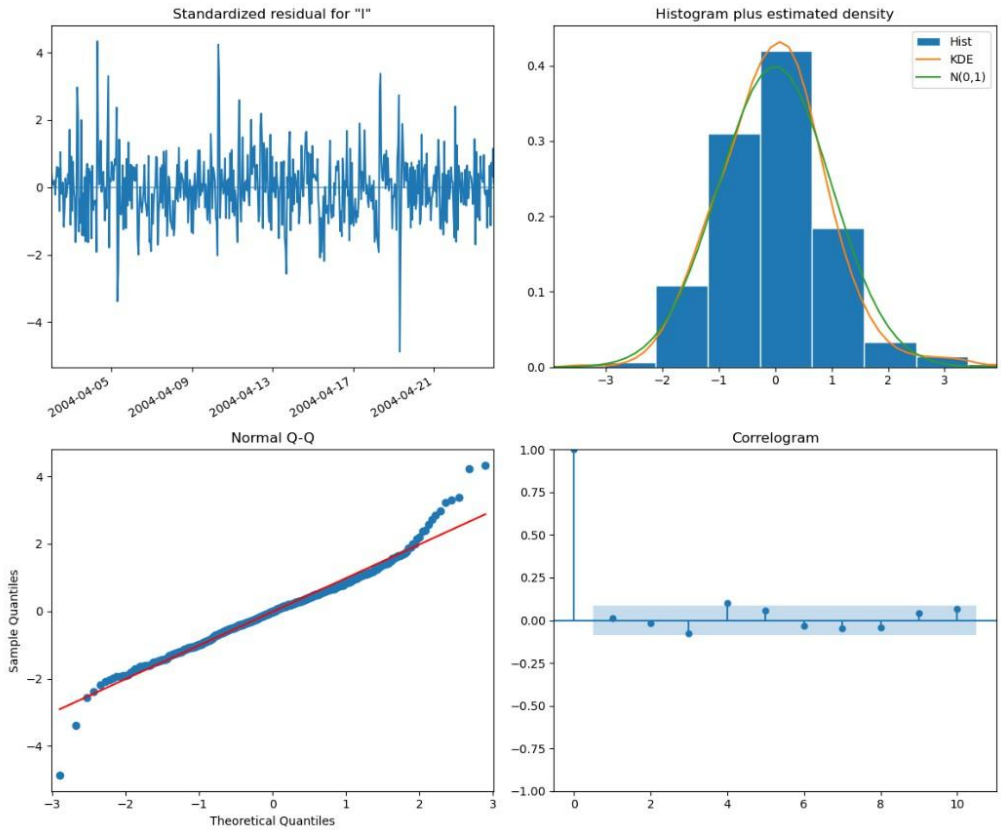


- **A.30** AIC values of fitted SARIMA models according to Auto ARIMA, by training set

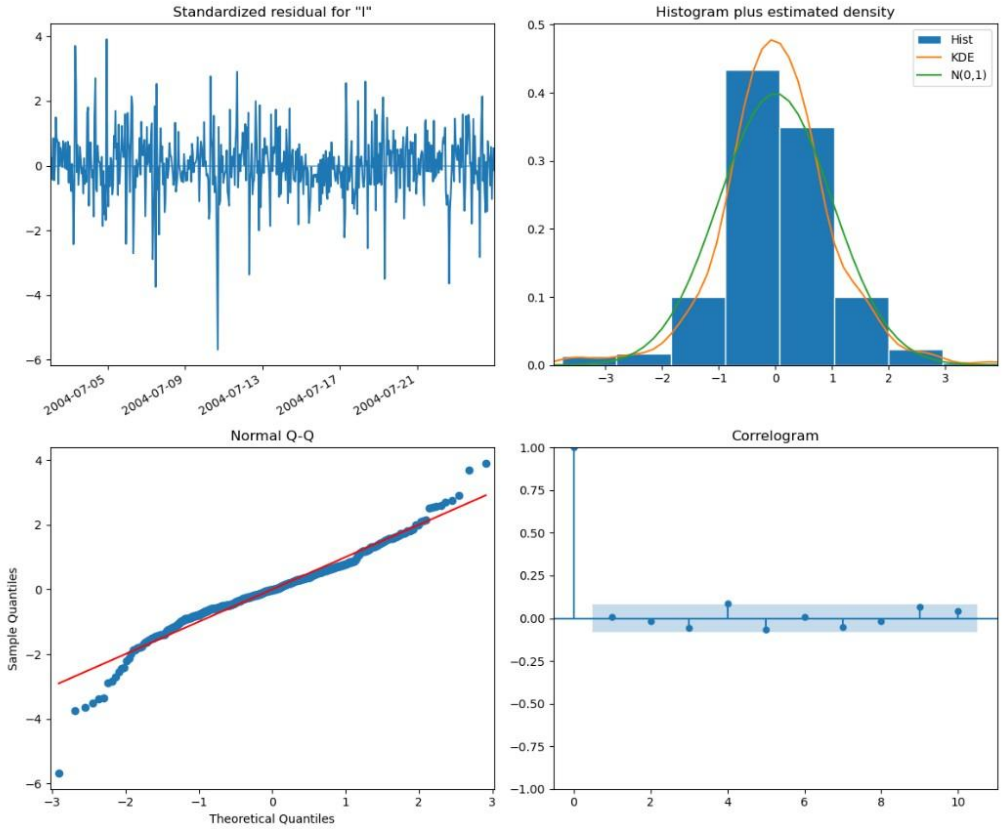
	Jan 2004	**Apr 2004**	**Jul 2004**	**Oct 2004**	**Dec 2004**
ARIMA(0,0,0)(0,1,0)[24]	-222.247	-606.059	-848.997	-1166.12	-176.323
ARIMA(0,0,0)(0,1,1)[24]	-220.894	-614.875	inf	-1231.46	-189.388
ARIMA(0,0,0)(1,1,0)[24]	-220.504	-612.706	-960.982	-1186.15	-217.53
ARIMA(0,0,0)(1,1,1)[24]	-227.674	inf	inf	inf	-193.632
ARIMA(0,0,0)(2,1,0)[24]	-268.392	-614.2	-969.472	-1217.56	inf
ARIMA(0,0,0)(2,1,1)[24]	inf	inf	inf	inf	-226.77
ARIMA(0,0,0)(3,1,0)[24]	-276.659	-626.182	-968.56	-1232.23	inf
ARIMA(0,0,0)(3,1,1)[24]	inf	inf	inf	inf	inf
ARIMA(0,0,0)(4,1,0)[24]	-275.183	-634.164	-992.561	-1271.78	-400.032
ARIMA(0,0,0)(4,1,1)[24]	inf	-684.892	inf	-1333.2	inf
ARIMA(0,0,1)(0,1,0)[24]	-882.468	-1232.67	-1490.68	-1762	-406.676
ARIMA(0,0,1)(0,1,1)[24]	-881.915	-1247.33	inf	-1823.09	-897.109
ARIMA(0,0,1)(1,1,0)[24]	-881.175	-1244.08	-1599.46	-1781.99	-921.883
ARIMA(0,0,1)(1,1,1)[24]	inf	inf	inf	inf	-900.829
ARIMA(0,0,1)(2,1,0)[24]	-915.94	-1245.09	-1610.07	-1818.17	inf
ARIMA(0,0,1)(2,1,1)[24]	inf	inf	inf	inf	-930.194
ARIMA(0,0,1)(3,1,0)[24]	-928.354	-1257.05	-1611.78	-1830.98	inf
ARIMA(0,0,1)(3,1,1)[24]	inf	inf	inf	inf	-1078.39
ARIMA(0,0,1)(4,1,0)[24]	-927.099	-1266.72	-1634.84	-1861.75	-1093.11
ARIMA(1,0,0)(0,1,0)[24]	-1718.89	-1838.45	-2190.47	-1992.86	-1099.32
ARIMA(1,0,0)(0,1,1)[24]	inf	-1855.79	inf	inf	-2047.93
ARIMA(1,0,0)(1,1,0)[24]	-1792.45	-1847.47	-2262.49	-1999.64	-2105.19
ARIMA(1,0,0)(1,1,1)[24]	inf	inf	inf	inf	-2071.61
ARIMA(1,0,0)(2,1,0)[24]	-1838.34	-1864.54	-2286.82	-2063.2	inf

ARIMA(1,0,0)(2,1,1)[24]	inf	inf	inf	inf	-2094.69
ARIMA(1,0,0)(3,1,0)[24]	-1843.24	-1877.76	-2305.97	-2069.2	inf
ARIMA(1,0,0)(3,1,1)[24]	inf	inf	inf	inf	-2121.58
ARIMA(1,0,0)(4,1,0)[24]	-1877.09	-1893.19	-2355.52	-2087.88	inf
ARIMA(1,0,1)(0,1,0)[24]	-1783.82	-1999.77	-2356.14	-2198.29	-2147.03
ARIMA(1,0,1)(0,1,1)[24]	inf	-2050.91	inf	-2277.24	-2230.53
ARIMA(1,0,1)(1,1,0)[24]	-1891.22	-2024.3	-2446.88	-2218.48	-2342.58
ARIMA(1,0,1)(1,1,1)[24]	inf	inf	inf	inf	-2275.84
ARIMA(1,0,1)(2,1,0)[24]	-1939.17	-2043.41	-2471.69	-2271.14	inf
ARIMA(1,0,1)(2,1,1)[24]	inf	inf	inf	inf	-2300.11
ARIMA(1,0,1)(3,1,0)[24]	-1942.63	-2055.99	-2501.08	-2279.33	inf
ARIMA(2,0,0)(0,1,0)[24]	-1775.52	-1982.88	-2362.35	-2157.73	-2326.16
ARIMA(2,0,0)(0,1,1)[24]	inf	-2025.86	inf	inf	-2245.73
ARIMA(2,0,0)(1,1,0)[24]	-1882.51	-2002.84	-2443.88	-2176.8	inf
ARIMA(2,0,0)(1,1,1)[24]	inf	inf	inf	inf	-2281.02
ARIMA(2,0,0)(2,1,0)[24]	-1926.56	-2024.97	-2465.01	-2229.88	inf
ARIMA(2,0,0)(2,1,1)[24]	inf	inf	inf	inf	-2323.14
ARIMA(2,0,0)(3,1,0)[24]	-1930.49	-2036.45	-2487.49	-2240.64	-2386.68
ARIMA(2,0,1)(0,1,0)[24]	-1782.54	-2008.5	-2373.42	-2207.46	-2341.12
ARIMA(2,0,1)(0,1,1)[24]	inf	-2068.48	inf	inf	-2253.26
ARIMA(2,0,1)(1,1,0)[24]	-1889.23	-2035.22	-2467.21	-2232.98	inf
ARIMA(2,0,1)(1,1,1)[24]	inf	inf	inf	inf	-2299.1
ARIMA(2,0,1)(2,1,0)[24]	-1936.6	-2057.27	-2490.51	-2282.37	-2370.59

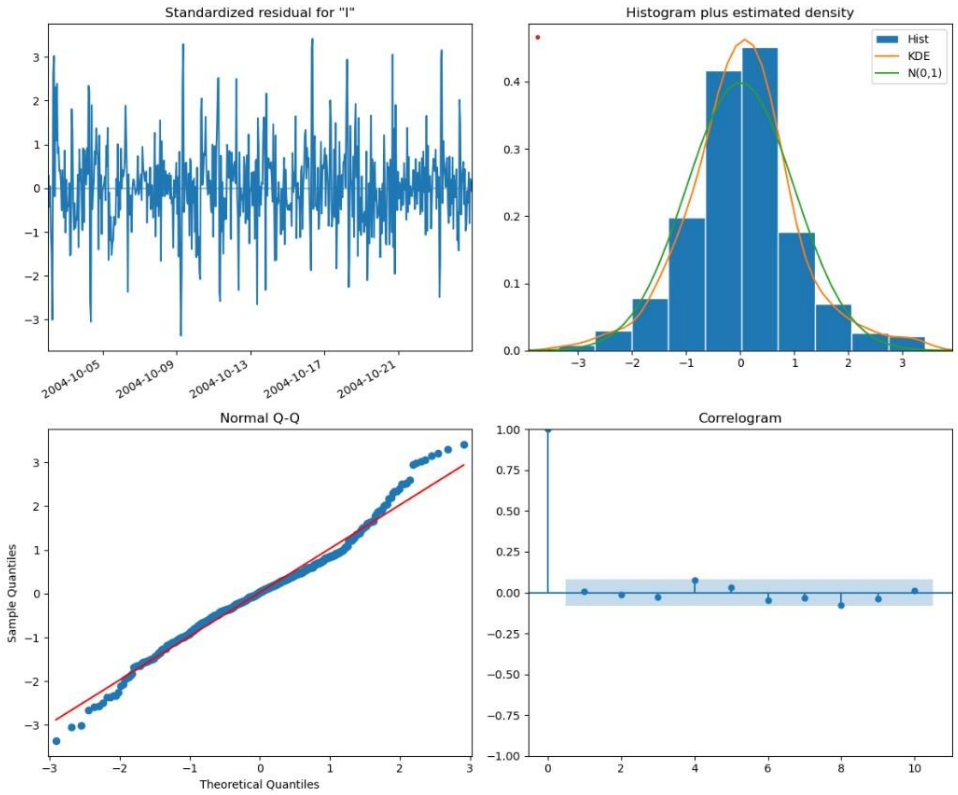
- **A.31** Diagnostic plots for SARIMA(2,0,1)(2,1,0)[24] model (Apr 2004, training set)



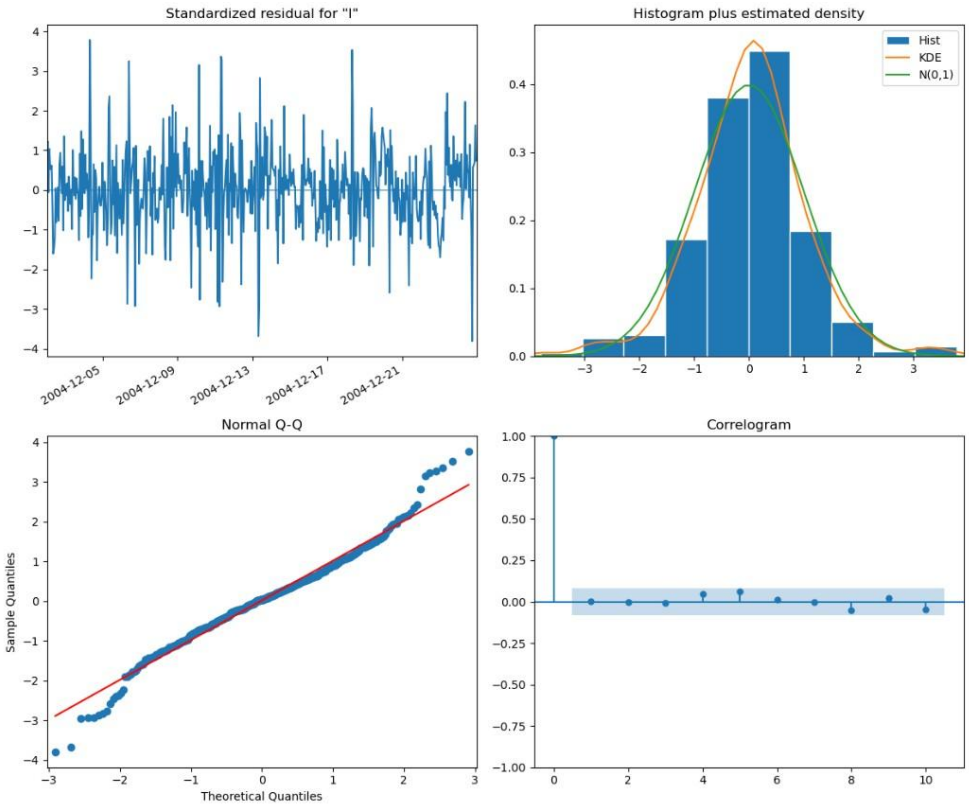
- **A.32** Diagnostic plots for SARIMA(2,0,1)(2,1,0)[24] model (Jul 2004, training set)



- **A.33** Diagnostic plots for SARIMA(2,0,1)(2,1,0)[24] model (Oct 2004, training set)



- **A.34** Diagnostic plots for SARIMA(2,0,1)(2,1,0)[24] model (Dec 2004, training set)



- **A.35** AIC of fitted SARIMAX models — stepwise Auto ARIMA, Jan & Apr 2004 training sets

SARIMAX Model	Jan 2004
ARIMA(0,0,0)(0,0,0)[24]	-1042.544
ARIMA(1,0,0)(1,0,0)[24]	-2070.885
ARIMA(0,0,1)(0,0,1)[24]	-1312.018
ARIMA(1,0,0)(0,0,0)[24]	-1982.262
ARIMA(1,0,0)(2,0,0)[24]	-2058.418
ARIMA(1,0,0)(1,0,1)[24]	-2067.866
ARIMA(1,0,0)(0,0,1)[24]	-2061.1
ARIMA(1,0,0)(2,0,1)[24]	-2055.651
ARIMA(0,0,0)(1,0,0)[24]	-1107.438
ARIMA(2,0,0)(1,0,0)[24]	-2129.48
ARIMA(2,0,0)(0,0,0)[24]	-2067.475
ARIMA(2,0,0)(2,0,0)[24]	-2087.827
ARIMA(2,0,0)(1,0,1)[24]	-2123.322
ARIMA(2,0,0)(0,0,1)[24]	-2120.717
ARIMA(2,0,0)(2,0,1)[24]	-2126.323
ARIMA(3,0,0)(1,0,0)[24]	-2141.954
ARIMA(3,0,0)(0,0,0)[24]	-2087.896
ARIMA(3,0,0)(2,0,0)[24]	-2110.897
ARIMA(3,0,0)(1,0,1)[24]	-2143.587

ARIMA(3,0,0)(0,0,1)[24]	-2128.727
ARIMA(3,0,0)(2,0,1)[24]	-2140.179
ARIMA(3,0,0)(1,0,2)[24]	-2127.93
ARIMA(3,0,0)(0,0,2)[24]	-2104.375
ARIMA(3,0,0)(2,0,2)[24]	-2137.869
ARIMA(3,0,1)(1,0,1)[24]	-2138.873
ARIMA(2,0,1)(1,0,1)[24]	-2148.542
ARIMA(2,0,1)(0,0,1)[24]	-2129.683
ARIMA(2,0,1)(1,0,0)[24]	-2152.107
ARIMA(2,0,1)(0,0,0)[24]	-2091.992
ARIMA(2,0,1)(2,0,0)[24]	-2109.198
ARIMA(2,0,1)(2,0,1)[24]	-2146.659
ARIMA(1,0,1)(1,0,0)[24]	-2152.67
ARIMA(1,0,1)(0,0,0)[24]	-2094.019
ARIMA(1,0,1)(2,0,0)[24]	-2111.195
ARIMA(1,0,1)(1,0,1)[24]	-2142.714
ARIMA(1,0,1)(0,0,1)[24]	-2140.303
ARIMA(1,0,1)(2,0,1)[24]	-2141.218
ARIMA(0,0,1)(1,0,0)[24]	-1091.589
ARIMA(1,0,2)(1,0,0)[24]	-2150.507
ARIMA(0,0,2)(1,0,0)[24]	-1787.186
ARIMA(2,0,2)(1,0,0)[24]	-1810.495

SARIMAX Model	Apr 2004
ARIMA(0,0,0)(0,0,0)[24]	-840.022
ARIMA(1,0,0)(1,0,0)[24]	-2023.894
ARIMA(0,0,1)(0,0,1)[24]	-930.77
ARIMA(1,0,0)(0,0,0)[24]	-1750.59
ARIMA(1,0,0)(2,0,0)[24]	-2031.211
ARIMA(1,0,0)(3,0,0)[24]	-1997.781
ARIMA(1,0,0)(2,0,1)[24]	-2027.121
ARIMA(1,0,0)(1,0,1)[24]	-2030.317
ARIMA(1,0,0)(3,0,1)[24]	-2021.377
ARIMA(0,0,0)(2,0,0)[24]	-988.67
ARIMA(2,0,0)(2,0,0)[24]	-2161.768
ARIMA(2,0,0)(1,0,0)[24]	-2162.254
ARIMA(2,0,0)(0,0,0)[24]	-1927.434
ARIMA(2,0,0)(1,0,1)[24]	-2168.901
ARIMA(2,0,0)(0,0,1)[24]	-2086.688
ARIMA(2,0,0)(2,0,1)[24]	-2160.849
ARIMA(2,0,0)(1,0,2)[24]	-2160.467
ARIMA(2,0,0)(0,0,2)[24]	-2138.308
ARIMA(2,0,0)(2,0,2)[24]	-2159.032
ARIMA(3,0,0)(1,0,1)[24]	-2187.635
ARIMA(3,0,0)(0,0,1)[24]	-2153.655
ARIMA(3,0,0)(1,0,0)[24]	-2201.26
ARIMA(3,0,0)(0,0,0)[24]	-2026.405
ARIMA(3,0,0)(2,0,0)[24]	-2205.004
ARIMA(3,0,0)(3,0,0)[24]	-2192.558
ARIMA(3,0,0)(2,0,1)[24]	-2209.387
ARIMA(3,0,0)(3,0,1)[24]	-2194.796
ARIMA(3,0,0)(2,0,2)[24]	-2206.399
ARIMA(3,0,0)(1,0,2)[24]	-2207.612
ARIMA(3,0,0)(3,0,2)[24]	-2202.5
ARIMA(3,0,1)(2,0,1)[24]	-2186.073
ARIMA(2,0,1)(2,0,1)[24]	-2177.202

- **A.36** AIC of fitted SARIMAX models — stepwise Auto ARIMA, Jul & Oct 2004 training sets

SARIMAX Model	Jul 2004
ARIMA(0,0,0)(0,0,0)[24]	-1247.431
ARIMA(1,0,0)(1,0,0)[24]	-2488.429
ARIMA(0,0,1)(0,0,1)[24]	-1232.904
ARIMA(1,0,0)(0,0,0)[24]	-2395.82
ARIMA(1,0,0)(2,0,0)[24]	-2460.964
ARIMA(1,0,0)(1,0,1)[24]	-2485.745
ARIMA(1,0,0)(0,0,1)[24]	-2460.07
ARIMA(1,0,0)(2,0,1)[24]	-2486.532
ARIMA(0,0,0)(1,0,0)[24]	-1267.245
ARIMA(2,0,0)(1,0,0)[24]	-2623.872
ARIMA(2,0,0)(0,0,0)[24]	-2467.319

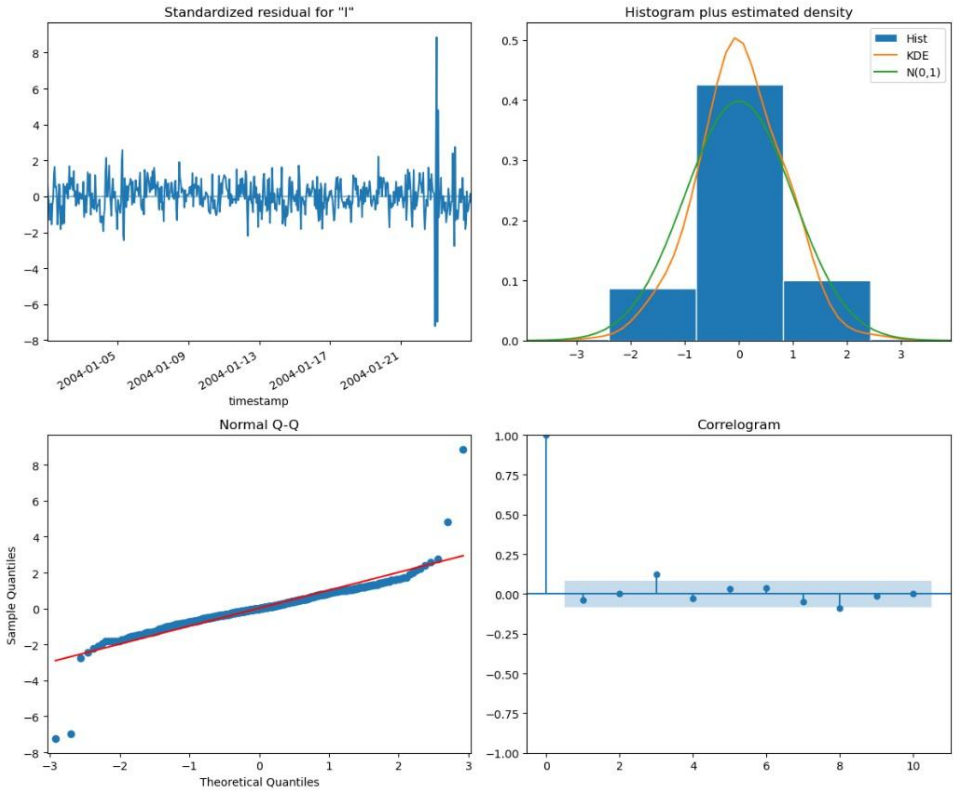
ARIMA(2,0,0)(2,0,0)[24]	-2629.556
ARIMA(2,0,0)(3,0,0)[24]	-2613.006
ARIMA(2,0,0)(2,0,1)[24]	-2629.979
ARIMA(2,0,0)(1,0,1)[24]	-2608.729
ARIMA(2,0,0)(3,0,1)[24]	-2608.739
ARIMA(2,0,0)(2,0,2)[24]	-2626.981
ARIMA(2,0,0)(1,0,2)[24]	-2624.258
ARIMA(2,0,0)(3,0,2)[24]	-2436.615
ARIMA(3,0,0)(2,0,1)[24]	-2665.592
ARIMA(3,0,0)(1,0,1)[24]	-2654.174
ARIMA(3,0,0)(2,0,0)[24]	-2667.284
ARIMA(3,0,0)(1,0,0)[24]	-2636.607
ARIMA(3,0,0)(3,0,0)[24]	-2660.647
ARIMA(3,0,0)(3,0,1)[24]	-2653.944
ARIMA(3,0,1)(2,0,0)[24]	-2647.027
ARIMA(2,0,1)(2,0,0)[24]	-2643.875

SARIMAX Model	**Oct 2004**
ARIMA(0,0,0)(0,0,0)[24]	-1442.88
ARIMA(1,0,0)(1,0,0)[24]	-2250.61
ARIMA(0,0,1)(0,0,1)[24]	-1460.15
ARIMA(1,0,0)(0,0,0)[24]	-2103.3
ARIMA(1,0,0)(2,0,0)[24]	-2259.45
ARIMA(1,0,0)(3,0,0)[24]	-2255.76
ARIMA(1,0,0)(2,0,1)[24]	-2250.04
ARIMA(1,0,0)(1,0,1)[24]	-2252.12
ARIMA(1,0,0)(3,0,1)[24]	-2253.92
ARIMA(0,0,0)(2,0,0)[24]	-1524.49
ARIMA(2,0,0)(2,0,0)[24]	-2395.23
ARIMA(2,0,0)(1,0,0)[24]	-2400.23
ARIMA(2,0,0)(0,0,0)[24]	-2260.72
ARIMA(2,0,0)(1,0,1)[24]	-2403.98
ARIMA(2,0,0)(0,0,1)[24]	-2394.2
ARIMA(2,0,0)(2,0,1)[24]	-2391.95
ARIMA(2,0,0)(1,0,2)[24]	-2395.36
ARIMA(2,0,0)(0,0,2)[24]	-2394.58
ARIMA(2,0,0)(2,0,2)[24]	-2386.5
ARIMA(3,0,0)(1,0,1)[24]	-2452.58
ARIMA(3,0,0)(0,0,1)[24]	-2445.63
ARIMA(3,0,0)(1,0,0)[24]	-2449.58
ARIMA(3,0,0)(2,0,1)[24]	-2443.07
ARIMA(3,0,0)(1,0,2)[24]	-2445.15
ARIMA(3,0,0)(0,0,0)[24]	-2321.86
ARIMA(3,0,0)(0,0,2)[24]	-2444.2
ARIMA(3,0,0)(2,0,0)[24]	-2446
ARIMA(3,0,0)(2,0,2)[24]	-2437.58
ARIMA(3,0,1)(1,0,1)[24]	-2441.04
ARIMA(2,0,1)(1,0,1)[24]	-2451.39

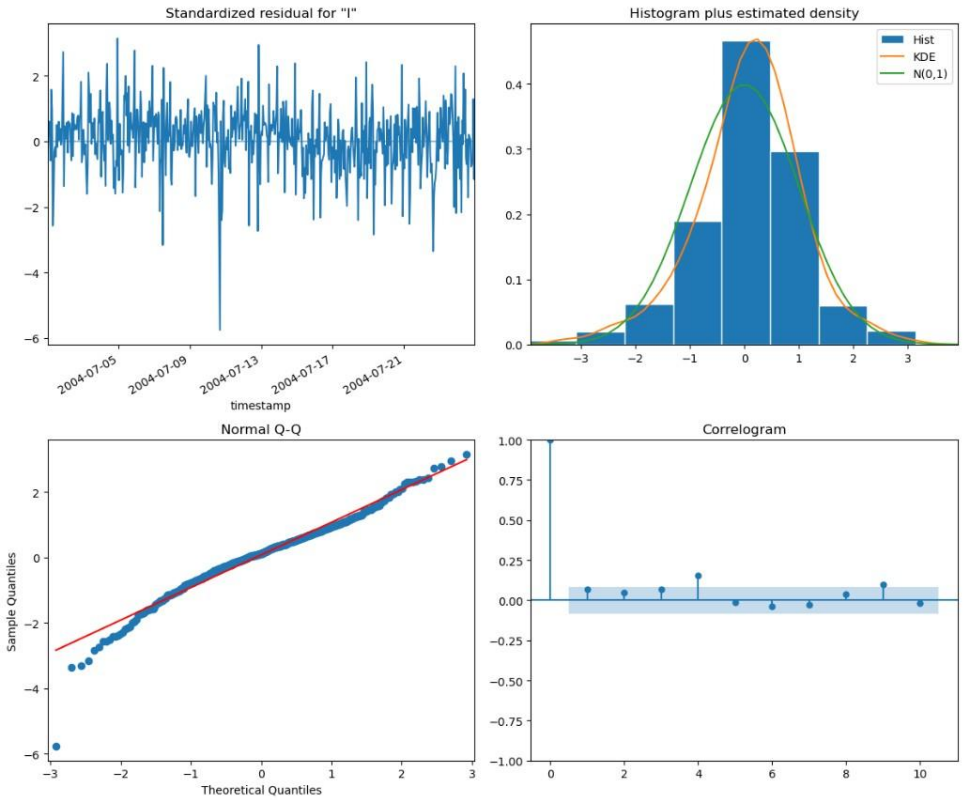
- **A.37** AIC of fitted SARIMAX models — stepwise Auto ARIMA, Dec 2004 training set

SARIMAX Model	Dec 2004
ARIMA(0,0,0)(0,0,0)[24]	-1198.08
ARIMA(1,0,0)(1,0,0)[24]	-2354.33
ARIMA(0,0,1)(0,0,1)[24]	-1204.87
ARIMA(1,0,0)(0,0,0)[24]	-2126.68
ARIMA(1,0,0)(2,0,0)[24]	-2349.44
ARIMA(1,0,0)(1,0,1)[24]	-2233.69
ARIMA(1,0,0)(0,0,1)[24]	-2309.69
ARIMA(1,0,0)(2,0,1)[24]	-2188.57
ARIMA(0,0,0)(1,0,0)[24]	-1236.22
ARIMA(2,0,0)(1,0,0)[24]	-2393.62
ARIMA(2,0,0)(0,0,0)[24]	-2218.8
ARIMA(2,0,0)(2,0,0)[24]	-2440.09
ARIMA(2,0,0)(3,0,0)[24]	-2445.36
ARIMA(2,0,0)(3,0,1)[24]	-2455.25
ARIMA(2,0,0)(2,0,1)[24]	-2438.74
ARIMA(2,0,0)(3,0,2)[24]	-2199.66
ARIMA(2,0,0)(2,0,2)[24]	-2170.13
ARIMA(1,0,0)(3,0,1)[24]	-2348.54
ARIMA(3,0,0)(3,0,1)[24]	-2471.59
ARIMA(3,0,0)(2,0,1)[24]	-2466.41
ARIMA(3,0,0)(3,0,0)[24]	-2430.5
ARIMA(3,0,0)(3,0,2)[24]	-2435.93
ARIMA(3,0,0)(2,0,0)[24]	-2470.57
ARIMA(3,0,0)(2,0,2)[24]	-2467.17
ARIMA(3,0,1)(3,0,1)[24]	-2430.87
ARIMA(2,0,1)(3,0,1)[24]	-2299.2

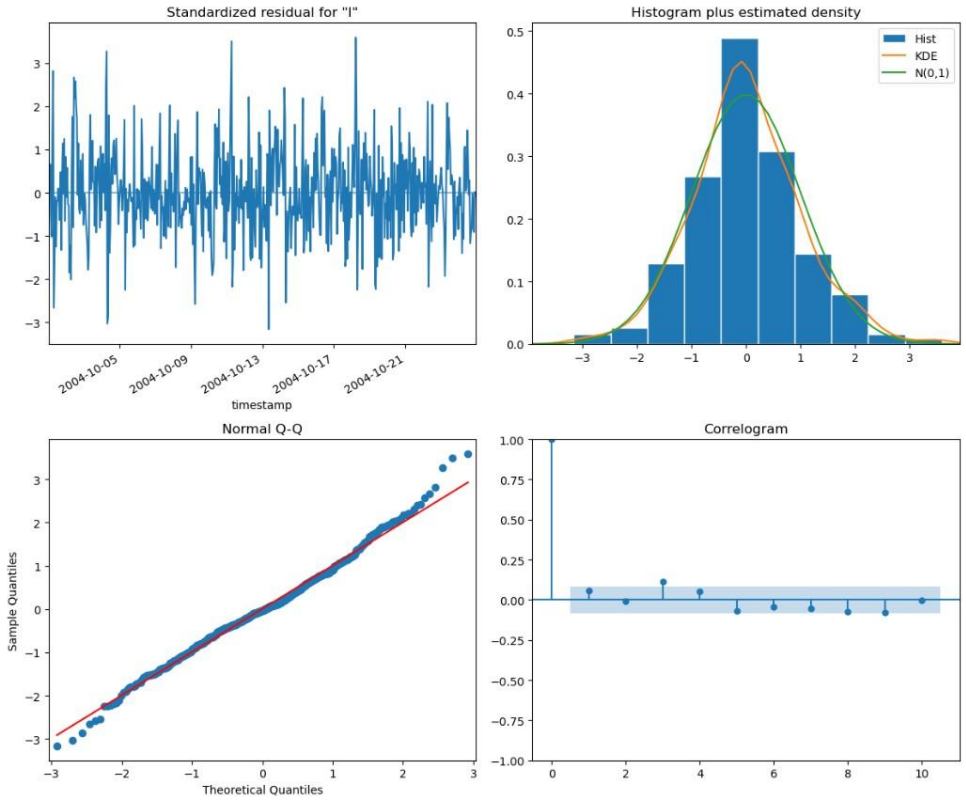
- **A.38** Diagnostic plots for SARIMAX(3,0,0)(1,0,0)[24] model (Jan 2004, training set)



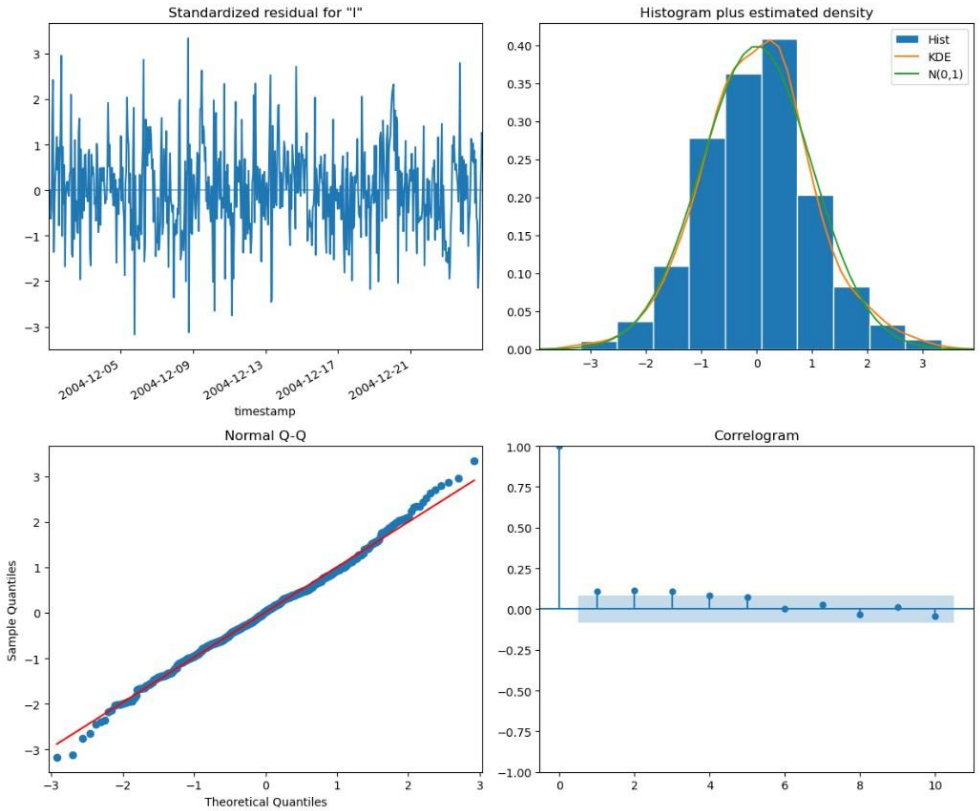
- **A.39** Diagnostic plots for SARIMAX(3,0,0)(1,0,0)[24] model (Jul 2004, training set)



- **A.40** Diagnostic plots for SARIMAX(3,0,0)(1,0,0)[24] model (Oct 2004, training set)



- **A.41** Diagnostic plots for SARIMAX(3,0,0)(1,0,0)[24] model (Dec 2004, training set)



- **A.42** ADF test results for temperature series in 1st week of July, by year

	2004	**2005**	**2006**	**2007**
ADF statistic	-2.926	-1.883	-1.215	-0.919
p-value of test statistic	0.042	0.339	0.666	0.781
Conclusion: α = 0.05	Stationary	Non-Stationary	Non-Stationary	Non-Stationary

Declaration of independent authorship

I, hereby, confirm that I have completed the submitted work without any inadmissible help from third parties or the use of any aids beyond those specified.
Furthermore, I confirm that arguments, text passages, program codes, pictorial and other representations (figures, diagrams, overviews, tables, etc.) as well as other content-based results created by generative Artificial Intelligence (AI) or with the help of programs based on generative AI have been labeled as such by me without exception, and the sources (including program, date of access, input data) have been indicated.
I confirm that I adhere to the Guidelines for Good Scientific Practice at TU Dresden. I also certify that the paper version I have submitted is identical to the digital version and that this academic work has not been submitted to any other examining authority in Germany nor abroad in the same or a similar form, nor has it already been published.

Dresden, 08 January 2026

Bhavay Singhal