AI Based Techniques for Short-Term Load Forecasting

Meiirkhan Abdek, B.Eng. in Electrical and Computer Engineering

Submitted in fulfilment of the requirements

for the degree of Master of Science

in Electrical and Computer Engineering

School of Engineering and Digital Sciences

Department of Electrical and Computer Engineering

Nazarbayev Univesity

53 Kabanbay Batyr Avenue,

Nur-Sultan, Kazakhstan, 010000

Supervisors: Prashant Jamwal, Refik Kizilirmak

27.04.2025

DECLARATION

I hereby, declare that this manuscript, entitled “AI Based Techniques for Short-Term Load Forecasting”, is the result of my own work except for quotations and citations which have been duly acknowledged. I also declare that, to the best of my knowledge and belief, it has not been previously or concurrently submitted, in whole or in part, for any other degree or diploma at Nazarbayev University or any other national or international institution.

Name: Meiirkhan Abdek
Date: 27.04.2025

Abstract

This thesis investigates the application of artificial intelligence-based techniques for short-term load forecasting (STLF) using synthesized data representing Kazakhstan's electrical grid. The research compares traditional statistical approaches, machine learning algorithms, and deep learning methods to determine optimal forecasting solutions for the region's specific load patterns. Five models were implemented and evaluated: Seasonal Autoregressive Integrated Moving Average (SARIMA), gradient boosting methods (LightGBM and XGBoost), and recurrent neural network architectures (Long Short-Term Memory and Gated Recurrent Unit).

Following extensive exploratory data analysis and feature engineering, the study reveals that gradient boosting methods significantly outperform both statistical and deep learning approaches. XGBoost achieved the best performance with a Mean Absolute Percentage Error (MAPE) of 7.15%, closely followed by LightGBM at 7.42%. The deep learning models performed moderately well (LSTM: 11.73% MAPE, GRU: 12.21% MAPE), while SARIMA showed considerably poorer results (20.85% MAPE).

Analysis of temporal patterns revealed strong hourly dependencies but relatively weak weekly and monthly seasonality in Kazakhstan's load profiles. This characteristic partly explains the superior performance of gradient boosting methods, which excel at capturing complex feature interactions and non-linear relationships. The findings suggest that electricity consumption in Kazakhstan may be more influenced by consistent daily industrial patterns than by residential usage that typically varies between weekdays and weekends.

This research provides valuable insights for electric utility planning in Kazakhstan and demonstrates that appropriate model selection based on data characteristics is crucial for accurate load forecasting. The methodologies and findings contribute to the growing body of knowledge

on AI applications in power systems management, particularly in regions with unique consumption patterns.

Table of Contents

List of Abbreviations

ML	Machine Learning
RNN	Recurrent Neural Network
VSTLF	Very Short-Term Load Forecasting
STLF	Short-Term Load Forecasting
MTLF	Medium-Term Load Forecasting
LTLF	Long-Term Load Forecasting
ARMA	Auto-Regressive Moving Average
ARIMA	Auto-Regressive Integrated Moving Average
SARIMA	Seasonal Autoregressive Integrated Moving Average
EDA	Exploratory Data Analysis
ADF	Augmented Dickey-Fuller
GOSS	Gradient-based One-Side Sampling

List of Figures

Chapter I – Introduction

1.1 General Information

Electrical energy is a vital part of the modern world that plays a crucial role in society's technological and socioeconomic development. Due to its importance, society's prosperity is directly related to electricity infrastructure, grid, availability, and source of electricity. Forecasting the electricity load is a crucial factor that controls the proper development of infrastructure and availability. Especially in an era when technologies are directly dependent on electricity, the demand for which is only growing [1].

Electric load forecasting is needed because of the requirements of accurate planning and operation of power systems, revenue projection, rate designing, trading energy, etc. [2]. Despite many studies focusing on forecasting electricity demand to plan electricity generation and distribution, the power industry is still challenged by accurate load forecasting. Moreover, the modern trends tend to reshape it to a more sustainable form by increasing the proportion of renewable power sources. Complexity of planning stable power infrastructure is increased by non-stationarity of factors that affect renewable energy production [1] Typically, load forecasting is important because failure to accurately forecast the loads can lead to utility company bankruptcy and cause systemic blackouts, leaving several regions or an entire country without power. [3].

General load forecasting tasks can be subdivided into groups depending on the range of the forecasting horizon, however there is no united standard for the range of the classification for load forecasts. It can be groped as: very short-term load forecasting (VSTLF) for several hours ahead of the look-ahead window, short-term load forecasting (STLF), with a look-ahead window

of days to weeks ahead, medium-term load forecasting (MTLF), with a look-ahead window of up to three years and long-term load forecasting (LTLF), with a look-ahead window above three years [3].

The possibility of forecasting load for short horizons has improved with the development of computation technologies at the end of the last century. Since then, several approaches have been proposed and tested for load forecasting. Initially, STLF was based upon time series analysis and statistical approaches and significantly transitioned to machine learning-based (ML) techniques [3]. We generally grouped the methods currently used for STLF into classical ML-based and Recurrent Neural Network-based (RNN) approaches.

1.2 Problem Definition.

Applying ML-based and RNN-based techniques, we provide a tool for forecasting load. The main goal of this Master's thesis is to explicitly explore the Kazakhstani dataset that contains historical data for one year that contains meteorological data and electrical grid load with time step being one hour. Another task is to investigate AI-based tools for accurate forecasting the load on those data. The main challenge of this problem is the stochastic nature of the weather conditions and load demand itself. As mentioned before, another challenge arises with the global trend to shift from traditional production of electrical energy based on fossil fuels to sustainable wind source and solar source generation as they also have non-stationary nature.

1.3 Aims and Objectives.

Implementation of the most common ML-based and RNN-based approaches for STLF is the general goal of this Master's thesis. The choice of models and their effectiveness directly

depends on the data we use as most of them rely on specific mathematical and statistical assumptions about the data. For this reason, we provide detailed exploration analysis of the data we reveal the patterns and peculiarities hidden in the data.

1.4 Thesis Outline

This Master's thesis is organized as follows. Chapter II illustrates the state-of-the art models that are commonly used for Time-Series forecasting, especially we are interested in Short-Term Load Forecasting. Chapter III introduces the theory behind those ML methods. We explore Seasonal Auto-Regressive Integrated Moving Average (SARIMA), XGBoost, LightGBM. This section also overviews the currently popular Recurrent Neural Network (RNN) based studies, comparing the benefits of RNN-based methods with those of other approaches. Chapter IV describes the methodology used in this Thesis. Chapter V provides results of the explanatory data analysis of the generated data and simulations results. It also discusses the performance of our selected models and compares it with the literature review results.

Chapter II – Literature Review

2.1 Evolution of Load Forecasting

Load forecasting has undergone substantial advancements over the past decades, transitioning from statistical approaches to advanced machine learning approaches. Initially forecasting methods mostly relied on regression techniques and time series analysis [4]. Studies show that these conventional techniques were unable to capture highly nonlinear behaviors in the electricity consumption data. More recent approaches demonstrate ensemble learning, and deep learning methods might further improve the forecasting accuracy [5].

2.2 Statistical Methods for Load Forecasting

2.2.1 Seasonal Autoregressive Integrated Moving Average

Autoregressive Integrated Moving Average (ARIMA) models are classical predictive models for time series forecasting that were proposed in 1970. To produce forecasts, these models decompose time series into systematic patterns [6]. The AR model uses observation and some lagged observations as a dependent relationship. The Moving Average (MA) model utilizes past forecast errors in a regression framework, relating current observations to previous residuals. Integration is the process of stationary time series creation using differencing of raw observation [7],[8].

As mentioned, forecasting has become essential for optimizing and ensuring reliable energy planning and management operations. Numerous studies have proposed the ARIMA model and evaluated its effectiveness using sequential load data from different places. In another study, ARMA, ARIMA, and ARIMAX, the modified ARIMA model that includes exogenous variables, were compared. As a result, ARMA, ARIMA, and ARIMAX resulted in a MAPE of 17.7%, 4%, and 3.6%, respectively [9].

SARIMA, is another ARIMA based statistical model that considers seasonal effect, has shown efficiency in load forecasting applications as they usually contain seasonal patterns.

On the other hand, recent studies suggest that the ARIMA model is not reliable. In [10:8], the authors evaluate the model's robustness to handling noise within the data. The paper proposes feeding two data sets, one unmodified and the other modified, with different noise levels. As a result, noise loading affects the choice of the ARIMA model parameters, the decrease in performance, and the inclusion of time series noise filters is possible. ARIMA model also requires significant preprocessing to achieve optimal performance.

2.3 Machine Learning Approaches

2.3.1 Decision Tree-Based Ensemble Methods

Numerous studies investigated the application of gradient boosting decision tree methods like XGBoost and LightGBM. The main mechanism behind those methods is sequentially combining multiple weak learners.

XGBoost, introduced in 2016 by [11], has demonstrated exceptional performance in variety of load forecasting applications. It is a scalable tree-boosting system that benefits from gradient-boosting decision trees and speeds up optimization by utilizing second-order Taylor expansion. It achieves strong learners by combining several weak ones, specifically classification and regression trees [12]. In [12], a load forecasting method combining SVMD with XGBoost was introduced. To eliminate the need for manually specifying the number of modes in traditional VMD, an adaptive decomposition technique using VMD and Sample Entropy (SampEn) was proposed. The method separates the raw load signal into a trend sequence and a set of fluctuating components. The essential variables, such as calendar restrictions, environmental temperature, and industrial customers' operational patterns, are then examined. Furthermore, feature selection

tools optimize the supplied features. For the purpose of model simplification, the trend series is finally modeled using a linear regression (LR) approach, given that the trend series' direction changes quite obviously. Concurrently, the XGBoost regression model is presented for every fluctuation subseries; hyper-parameters of XGBoost are optimized via Bayesian optimization method (BOA). Model evaluation on the test set yields the following results: MSE 435.42 kW, MAE 12.01 kW, and MAPE 7.93% [12]. Another study shows that XGBoost load prediction resulted in a 2.80% MAPE, and 0.0288 RMSE [13:38].

LightGBM, was developed by Microsofr Research in 2017 [14], adopts a leaf-wise decision tree construction method in conjunction with GOSS to offer computational advantage over XGBoost. LightGBM was used for STLF and resulted in MAPE to be 2.80%, significantly outperforming classical statistical forecasting approaches. [13]

2.4 Deep Learning Approaches

2.4.1 Recurrent Neural Network (RNN)

A recurrent neural network (RNN) is a deep learning network capable of processing sequential data like text, speech, and temperature. Distinctive feature of RNN from other deep learning networks is the ability to process contextual information such as the daytime or type of weather. The RNNs gained popularity in recent years due to the increased efficiency of the parallel processing units of the Graphical Processing Units (GPU). GPU allows us to train and execute complex AI models at higher speeds. The ability of RNNs to work with sequential data opens a wide range of applications. This makes RNN one of the main tools for electrical load forecasting [15],[16]. In general, RNNs can deal with long input samples. However, the longer the input sequence, the lower the accuracy. The reason for what is so-called the problem of vanishing gradient. The problem of vanishing gradients occurs when deep neural networks must

deal with long data sequences, and as a result, while updating the gradients, they might become very small. The following RNN models were designed to solve this issue by implementing the memory cells [15],[16].

2.4.2 Long Short-Term Memory

In 2017, Hochreiter and Schmidhuber proposed an improved version of RNN as a tool to mitigate the vanishing gradient problem. It was called Long short-term memory (LSTM). An LSTM network is structured around a memory cell and three types of gates: input, output, and forget. LSTM is developed to store and retrieve information over a more extended data sequence. This allows the LSTM to “remember” the essential features, whereas the traditional RNN is most likely to “forget” previous information as new information comes in [15],[16].

2.4.3 Gated Recurrent Unit

A Gated Recurrent Unit (GRU) is another model that attempts to mitigate the problem of vanishing gradient. The system consists of four components: an update gate, reset gate, memory unit, and hidden state. The reset gate manages the degree to which past information is discarded. The update gate determines how much data to ‘remember’. Then, common backpropagation and gradient descent methods are used to train GRU. [17][15]

Chapter III – Theory

In this chapter of the Thesis, we explicitly explain the theory behind the methods we discuss in the literature review section. We also implement them on our dataset, demonstrate and discuss the results in the corresponding section of this Thesis.

3.1 Seasonal Autoregressive Integrated Moving Average

SARIMA is a statistical approach that is an extension to the ARIMA model. It is designed to handle data that preserves short-term seasonal patterns as well as long-term seasonal patterns. It relies on the stationarity of the time-series data which means constant statistical properties of the data over time such as mean and variance. Augmented Dickey-Fuller test (ADF) is a common tool to check stationarity.

The conventional notation of the SARIMA is SARIMA (p, d, q) (P, D, Q, s) where:

- p: AR term of order p which captures the autocorrelation in the data.
- q: MA term of order q tries to model relationship between the current data point and past prediction errors.
- d: Integrated term of order d which calculates the number of differences needed to preserve stationarity in the data.
- S: seasonal period which attributes to periodic patterns of the time-series.
- P: seasonal autoregressive term of order P.
- Q: seasonal moving average term of order Q.
- D: Seasonal integrated term of order D.

We can represent SARIMA mathematically as follows:

$$(1 - \varphi_1 B)(1 - \Phi_1 B)(1 - B)(1 - B^s)y_t = (1 + \theta_1 B)(1 + \Theta_1 B)\varepsilon_t$$

Where:

- y_t is the observed time series at time t.
- B is the backward shift operator, representing the lag operator.
- φ_1 is the non-seasonal autoregressive coefficient.
- Φ_1 is the seasonal autoregressive coefficient.
- θ_1 is the non-seasonal moving average coefficient.
- Θ_1 is the seasonal moving average coefficient.
- s is the seasonal period.
- ε_t is the white noise error term at time t.

3.2 LightGBM

LightGBM is a framework proposed by Microsoft in 2017 [14]. It utilizes gradient boosting method to construct strong models from sequence of weak decision tree models. An architecture of the boosting models might be seen on Figure 3.1. The initial model is trained in data, and the next model is trained on errors of the previous one to correct them. Accurate forecasting abilities of the final model is achieved due to the combination of the weak models. LightGBM is widely used for regression and classification tasks [18],[19].

Mathematically, we can define the model at step m as follows:

$$F_m(x) = F_{m-1}(x) + \eta h_m(x) \quad (1)$$

Where:

- $F_{m-1}(x)$ is a previous model,
- $h_m(x)$ is the new decision tree that fits residuals,
- η is the learning rate

Loss function $L(y, F(x))$minimization is the main goal of the model. It utilizes gradient descent.

In case of Mean Squared Error (MSE):

$$L\big(y, F(x)\big) = \frac{1}{2}(y_i - F(x_i))^2 \tag{2}$$

LightGBM uses histogram-based binning to optimize tree growth:

1. Continuous features are bucked into discrete bins.
2. Gain is calculated to find the best split:

$$Gain = \frac{1}{2}\left(\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right) - \gamma \tag{3}$$

Where:

- G_L, G_R are the sums of gradients for the left and right splits,
- H_L, H_R are the sums of Hessians,
- λ is the L2 regularization term,
- γ is the minimum split gain.

LightGBM uses Leaf-wise growth strategy (GOSS) instead of growing them level-wise. This may lead to improved performance of the load forecasting due to the more accurate gain estimation.

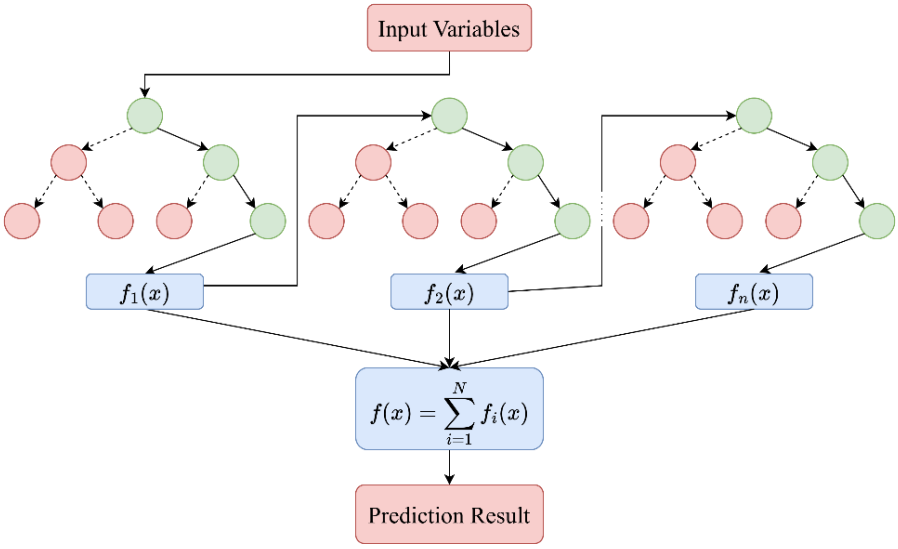

Figure 3.1 Boosting Model Architecture

3.3 XGBoost

XGBoost is a Gradient Boosting model and type of ensemble learning method. It also combines decision trees in sequential manner to improve performance of the model.

1. Ensemble Model:

The final prediction for a data point i is the average sum of predictions from all trees:

$$\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i) \quad (4)$$

Here, $\widehat{y_i}$ is the predicted value, K is the number of trees, $f_k(x_i)$ is the prediction of the k-th tree, and x_i is the input data point.

2. Objective Function:

The objective combines a loss function and a regularization term to balance accuracy and complexity:

$$obj(\Theta) = \sum_{i=1}^{n} l(y_i, \widehat{y_i}) + \sum_{k=1}^{K} \Omega(f_k) \quad (5)$$

- $l(y_i, \widehat{y_i})$ Measures the difference between true values y_i and predicted values $\widehat{y_i}$ (e.g., Mean Squared Error).
- $\Omega(f_k)$: Regularization term discouraging overly complex trees.

3. Iterative Optimization:

Starting with an initial prediction $\widehat{y_i^{(0)}} = 0$, the model adds trees sequentially to improve predictions:

$$\widehat{y_i^{(t)}} = \widehat{y_i^{(t-1)}} + f_t(x_i) \quad (6)$$

Here, $f_t(x_i)$ is the prediction of the new tree at iteration t.

4. Regularization:

To simplify trees, the regularization term penalizes the number of leaves and the squared weights of leaf values:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \quad (7)$$

- T: Number of leaves.
- γ: Regularization parameter for complexity.
- λ: Parameter penalizing large leaf weights.

5. Split Decision:

When building decision trees, XGBoost calculates the information gain for every possible split and chooses the one that maximizes gain:

$$\text{Gain} = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma \quad (8)$$

- G_L, G_R: Gradients for left and right child nodes.
- H_L, H_R: Hessians for left and right child nodes.

3.4 Long Short-Term Memory

An improved version of standard RNN called Long short-term memory (LSTM) was proposed to address the issue of the vanishing gradient. LSTM is designed to store and access information over a longer sequence of data. This allows the LSTM to "remember" the important features, whereas the traditional RNN is most likely to forget these features as the input sequence gets larger and gradients become smaller and smaller.

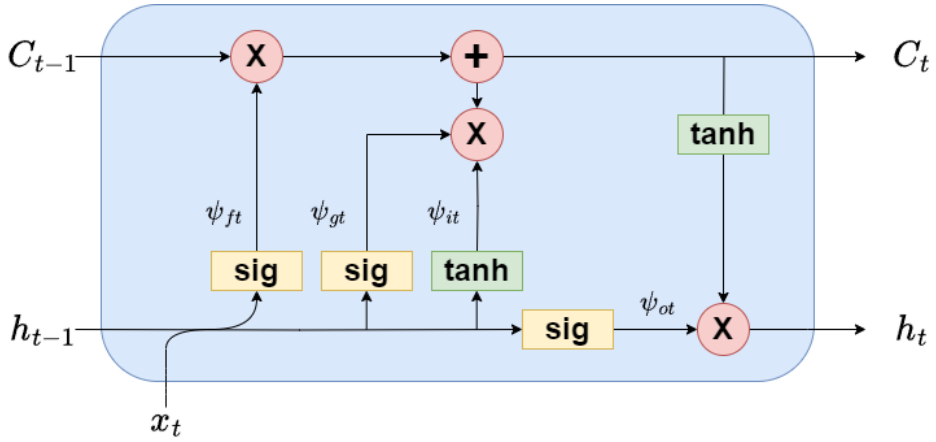

Figure 3.2. LSTM Cell Architecture

The conventional LSTM unit block is shown in Figure 3.2. The inward/outward variables are memory cell states (c_t), hidden states (h_t), and input (x_t). The intermediate variables inside the LSTM block are outputs of the forget gate (ψ_{ft}), input gate (ψ_{it}), input node (ψ_{gt}), and output gate (ψ_{ot}), respectively. The forget gate is responsible for how much information from the previous step will be used in the current one. The amount of stored information is determined by the output of the input gate. The output of the current step is calculated by the output gate.

The output of the gates is determined by the following formulas:

$$\psi_{ft} = sigmoid(w_{fx}x_t + w_{fh}h_{t-1} + b_f) \tag{10}$$

$$\psi_{it} = sigmoid(w_{ix}x_t + w_{ih}h_{t-1} + b_i) \tag{11}$$

$$\psi_{gt} = tanh(w_{gx}x_t + w_{gh}h_{t-1} + b_g) \tag{12}$$

$$\psi_{ot} = sigmoid(w_{ox}x_t + w_{oh}h_{t-1} + b_o) \tag{13}$$

The states are determined by these formulas:

$$c_t = \psi_{gt} * \psi_{it} + c_{t-1} * \psi_{ft} \tag{14}$$

$$h_t = tanh\,(c_t)\psi_{ot} \tag{15}$$

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{16}$$

$$tanh\,(x) = \frac{e^{2x}-1}{e^{2x}+1} \tag{17}$$

3.5 Gated Recurrent Unit

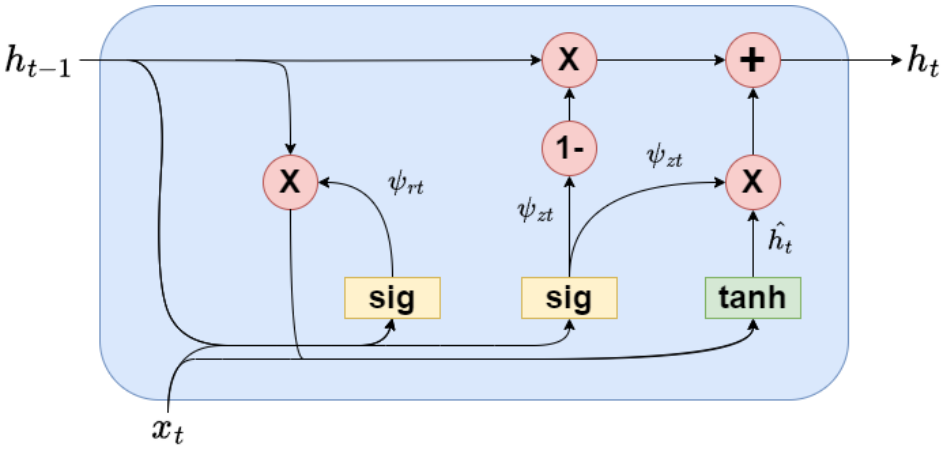

Figure 3.3. GRU Cell Architecture

Update-gate ψ_{zt} is updated as follows:

$$\psi_{zt} = sigmoid(W_z \cdot [h_{t-1}, x_t]) \tag{18}$$

Reset gate $\psi_{rt}$is updated as follows:

$$\psi_{rt} = sigmoid(W_r \cdot [h_{t-1}, x_t]) \tag{19}$$

In Equation (18) and (19) x_t represents the input series, while h_{t-1} is the previous hidden state, $sigmoid$ serves as non-linear function, and W is the input variable parameters.

The current time step $\hat{h}_t$ is calculated as follows:

$$\hat{h}_t = tanh(W \cdot [\psi_{rt} * h_{t-1}, x_t]) \tag{20}$$

The current hidden state h_t is calculated as follows:

$$h_t = (1 - \psi_{zt}) * h_{t-1} + \psi_{zt} * \hat{h}_t \tag{21}$$

Chapter IV – Methodology

4.1 Data Generation

For this thesis we use synthetic data that were generated using HOMER Software. Beside the electrical load we are interested in the meteorological data such as solar radiation and wind speed in the Kazakhstan region over the year. Homer software utilizes NASA's database which provides average data for the selected location. To reflect realistic fluctuations in electrical load it introduces randomness.

After the data generation we have 8672 observations with the next list of the 36 features:

Time

Global Solar

Fronius Primo 8.2-1 with Generic PV Solar Altitude

Fronius Primo 8.2-1 with Generic PV Solar Azimuth

Fronius Primo 8.2-1 with Generic PV Angle of Incidence

Fronius Primo 8.2-1 with Generic PV Incident Solar

Fronius Primo 8.2-1 with Generic PV Power Output

Fron8.2 Dedicated Converter Power Input

Fronius Primo 8.2-1 with Generic PV Cell Temperature

Wind Speed

Generic 3 kW Power Output

Generic 3 kW Operating Status

Ambient Temperature AC Primary Load

AC Primary Load Served

Total Electrical Load Served Renewable Penetration

Excess Electrical Production Unmet Electrical Load

Total Renewable Power Output

Inverter Power Input

Inverter Power OutputRectifier

Power Input Rectifier Power Output

Generic 1kWh Lead Acid Maximum Charge Power

Generic 1kWh Lead Acid Maximum Discharge Power

Generic 1kWh Lead Acid Charge Power

Generic 1kWh Lead Acid Discharge Power

Generic 1kWh Lead Acid Input Power

Generic 1kWh Lead Acid Energy Content

Generic 1kWh Lead Acid State of Charge

Generic 1kWh Lead Acid Energy Cost

AC Required Operating Capacity

DC Required Operating Capacity

AC Operating Capacity

DC Operating Capacity

4.2 Exploratory Data Analysis (EDA)

First, we explore each column of the synthesized data to decide which of them are going to be used based on our domain knowledge. Next, analysis of the preliminary selected features requires us to visualize the data. Graphical representation of the data enables us to reveal the features, including patterns, anomalies, trends, seasonality, and connection among variables. For instance, we plot time graphs, box plots, correlation matrixes etc. We describe graphs in the

results sections explicitly. Another important aspect of EDA is handling missing values, outliers, and permutation of the timestamps.

4.3 Feature Selection and Feature Engineering

We will meticulously identify the most influential features significantly impacting short-term load forecasting. This involves statistical analysis, correlation studies, and domain knowledge to select most valuable features for the forecasting models. For example, state-of-the-art data demonstrates that information about which days are weekends, and which working days greatly influence the forecasting results. Not all articles use data that includes such information. Thus, feature selection is an essential part of our methodology. This will help use data that directly affects forecasting and exclude data that has little or no effect on the results. This will simplify the model, speed up learning, and make it more relevant and accurate.

In this part we also perform feature engineering. We introduce new features from available ones. Generally, this is an essential part of improving the forecasting abilities of the models that are used for STLF. Time features are split into hour, day of week, and month components to capture seasonality and periodicity. Lag features and rolling means of the load are also introduced as input features. Another important transformation of the features is scaling as it ensures that numerical features are on a similar scale. It prevents any single feature from disproportionately influencing the model and ensures smoother training of the models that utilize gradient-based optimization like LSTM and GRU.

4.4 Model Selection

Initially, model selection is based on the most common and well studies approaches from literature review. However, the solution normally depends on the dataset, in other words the solution is for each unique dataset. This means that good results for datasets from other countries

do not guarantee good results for the Kazakhstani dataset. However, it is an excellent point to start with.

In this thesis, we provide a comparative analysis of SARIMA, LightGBM, XGBoost, LSTM, and GRU models.

4.5 Model Training and Validation

After the feature engineering part, we split the data into 2 parts, including the train set and test set in proportion 80% to 20%. We further split the test set into validation and test sets in proportion 10% and 10%. All the chosen models are trained on the training set. After that, the validation set is used to find the best parameters of the models. This means that preliminary results for the models' accuracy are estimated at this point but not finalized, as we adjust the hyperparameters of our models without introducing the bias into the obtained results. We adjust hyperparameters as follows. For SARIMA model we auto_arima function to find optimal order of its components. For LightGBM and XGBoost we conduct grid searches, optimize learning rate, tree depth, regularization parameters and use Bayesian optimization for more efficient tuning. For LSTM and GRU we try different network architectures.

4.6 Evaluation Metrics

To estimate the forecasting capabilities of the models, final versions of the models predict future values of the load on unseen data. After that we use common for STLF metrics, such as Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Square and Error (RMSE). Calculation of the performance metrics might be seen below:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{22}$$

$$RMSE = \sqrt{\sum_{i=1}^{N} \frac{(y_i - \hat{y}_i)^2}{N}} \tag{23}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (24)$$

These metrics provide comprehensive insights into the forecasting models' accuracy and reliability. After that, performance of the models is summarized in the results section of this thesis. Mean Absolute Error (MAE) measures the average magnitude of errors in the predictions, without considering their direction. It’s simple to understand, making it ideal for assessing the absolute accuracy of the models. Root Mean Squared Error (RMSE) gives greater weight to larger errors due to squaring the differences before averaging. This makes it valuable for situations where larger errors are particularly undesirable, as it penalizes them more heavily than MAE. Mean Absolute Percentage Error expresses errors as percentages relative to the actual values. This normalization makes it useful for comparing models across different datasets with varying scales, or when relative error is more significant than absolute error.

Chapter V – Results and Discussion

5.1. Results

5.1.1 Explanatory Data Analysis Results

We calculate descriptive statistics of the selected features, and present results in Table 5.1. These statistics give a general idea of the average tendency, variability, and range of meteorological data and electricity consumption.

Table 5.1: Descriptive statistics of the selected features

	Ambient Temperature [C°]	**Total Renewable Power Output [kW]**	**Wind Speed [m/s]**	**Global Solar [kW/m2]**	**AC Primary Load [kW]**
count	8760	8760	8760	8760	8760
mean	3.912	1.979	6.494167	0.153	0.470
min	-15.0756	0	0.052545	0	0.023113
25%	-6.24895	0.188	3.910024	0	0.271655
50%	6.122019	1.306236	6.065309	0.008849	0.439
75%	19.59632	3.036701	8.629875	0.233549	0.605013
max	21.6077	9.866294	23.35846	1.020618	2.235969
std	13.311	2.071	3.434	0.232	0.298

We also provide analysis of the outliers of the selected features. Results are summarized in Table 5.2

Table 5.2: Outlier Analysis

Feature	**Number of Outliers**	**Percentage of Outliers**
Ambient Temperature	0	0.00
Total Renewable Power Output	191	2.18
Wind Speed	104	1.19
Global Solar	699	7.98
AC Primary Load	348	3.97

Next, we demonstrate the distribution of the features in Figure 5.1. In general, we can see that peaks for ambient temperature distribution at -15 C° and 20 C° which means that those temperatures occur most frequently. Distribution of the wind speed is right-skewed, while global solar distribution and total renewable power output are heavily skewed toward 0 demonstrating power low distribution. Log transformation is used here. We also plot distribution of the AC Primarily Load in Figure 5.2. The histogram is right-skewed, this suggests that smaller loads occur more frequently in the demand side.

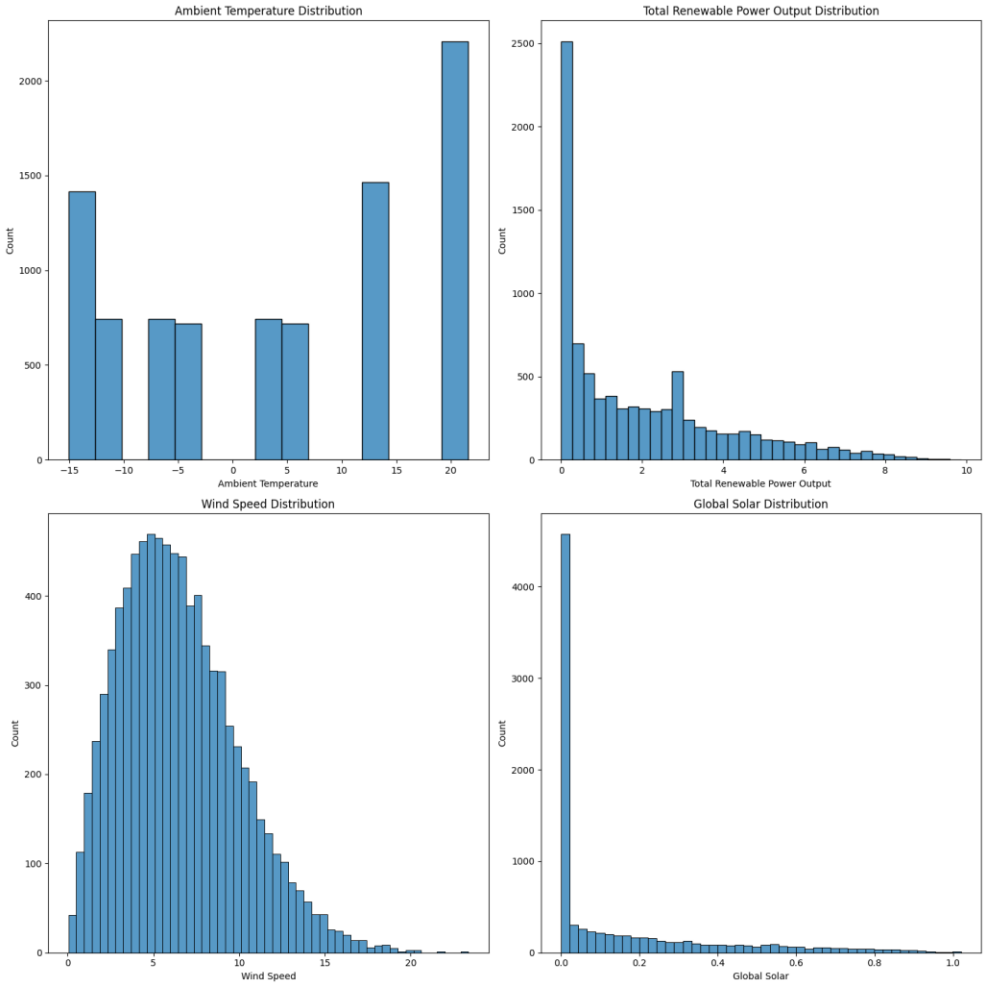

Figure 5.1: Distribution of Predictive Features

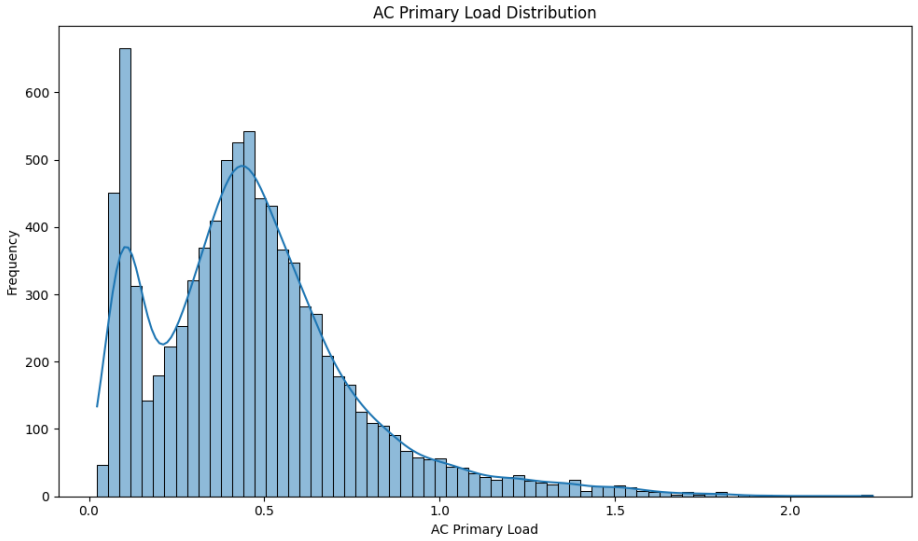

Figure 5.2: Distribution of AC Primary Load

Next, we plot correlation matrix of the features to investigate the relationship between the features and target features. We can see a strong positive correlation of Global Solar, and Wind Speed with Total Renewable Output.

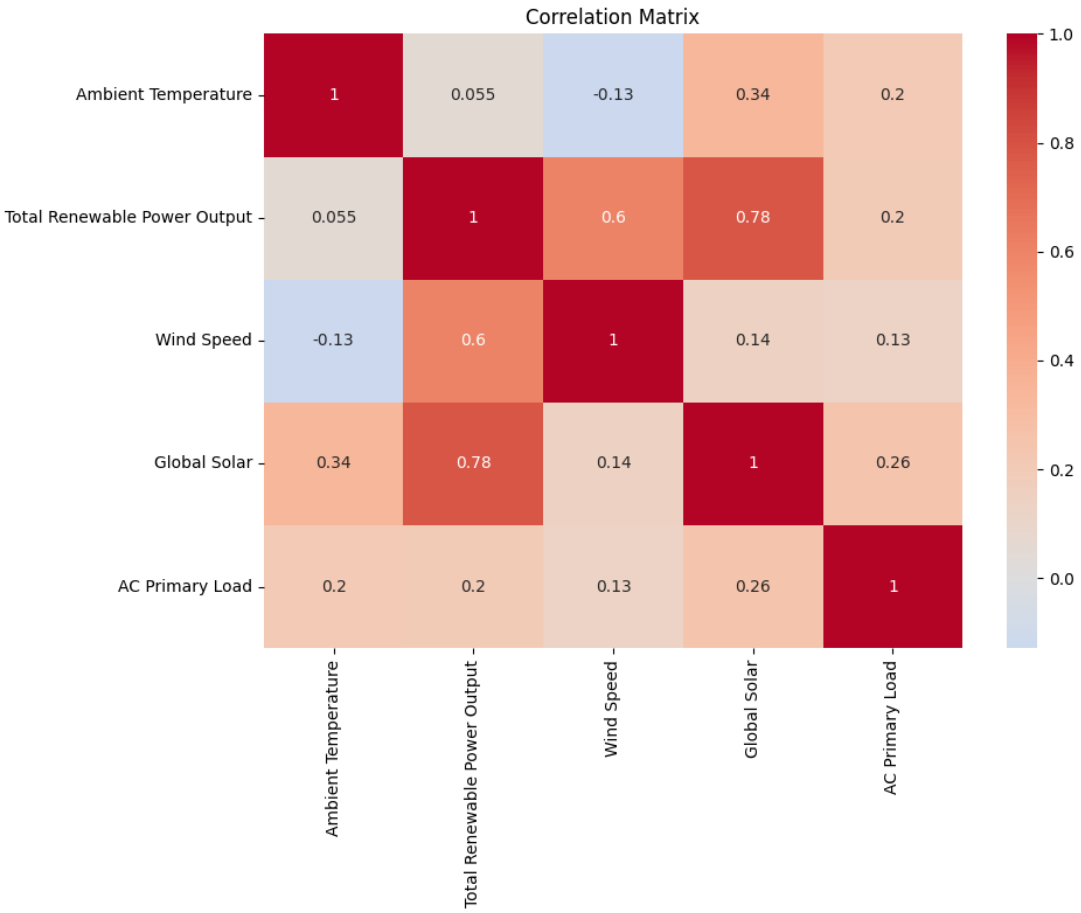

Figure 5.3: Correlation Matrix of Features

Then, we plot of the AC Primary Load against the time demonstrates that the data is permutated. Results of the sorted data might be seen in Figure 5.4. This required as time-series models rely on temporal dependencies, shuffled data breaks seasonal patterns, trends, and influence of past values on future ones.

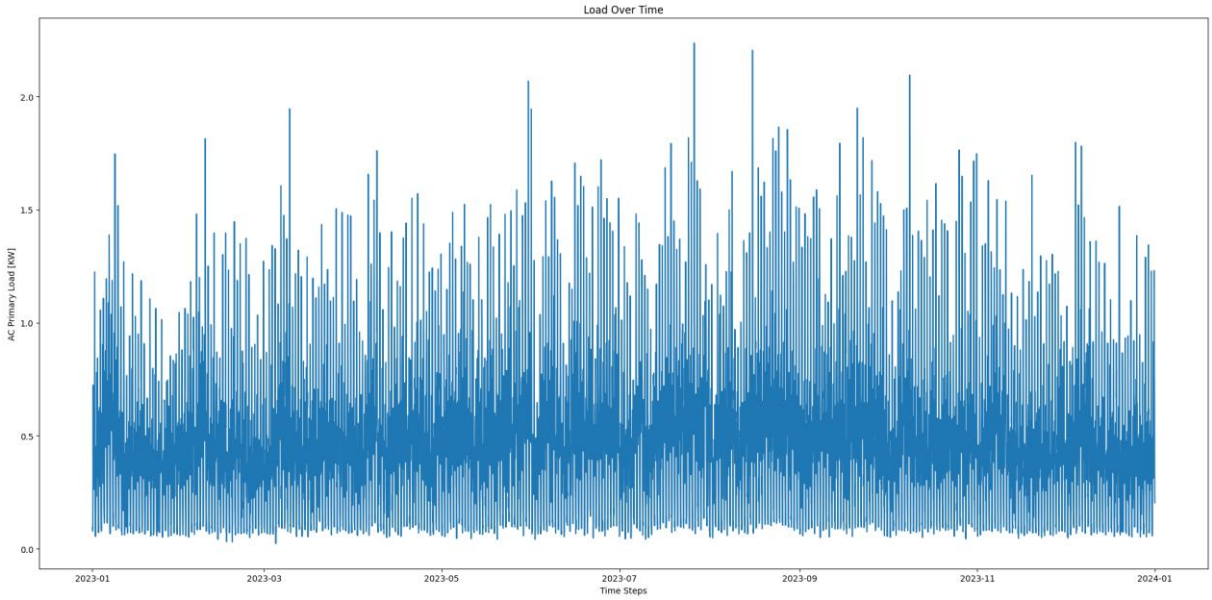

Figure 5.4: AC Primary Load Against Time

Next, we plot boxplots of the load by hour, week and month. We observe that electricity consumption exhibits a steady trend throughout the week and month. This suggests the absence of the significant effect of the weekends and year seasonality. However, we can see that the hour of the day significantly affects the load. This information is used while models' selection and training.

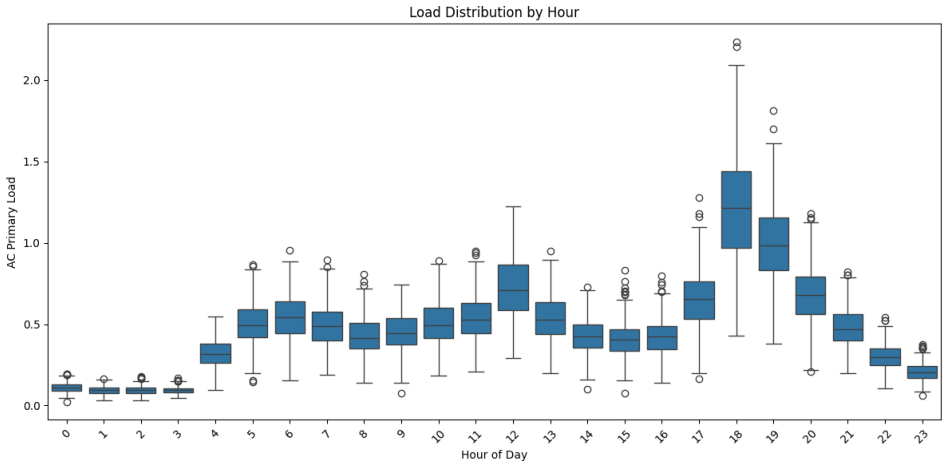

Figure 5.5: Boxplots of Load Distribution by Hour

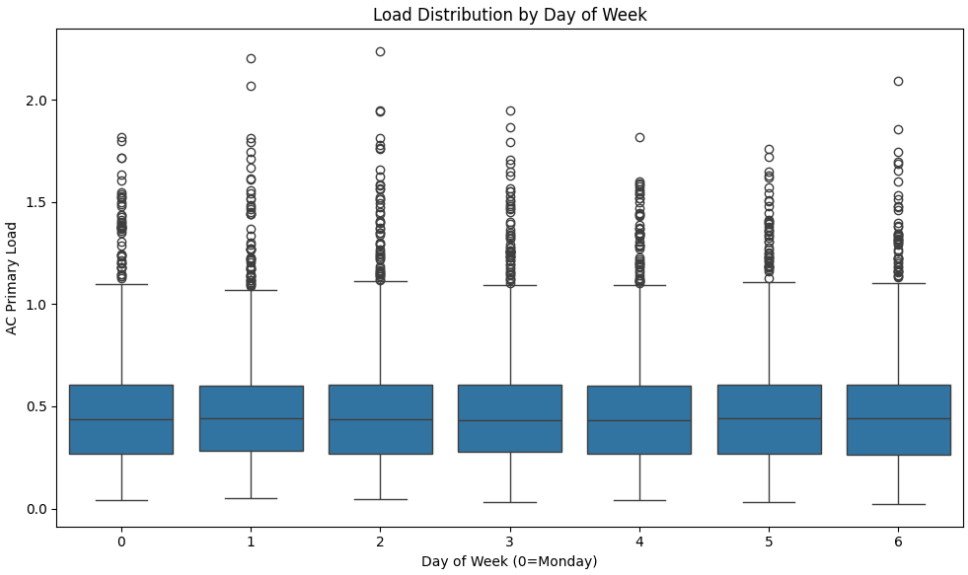

Figure 5.6: Boxplots of Load Distribution by Week

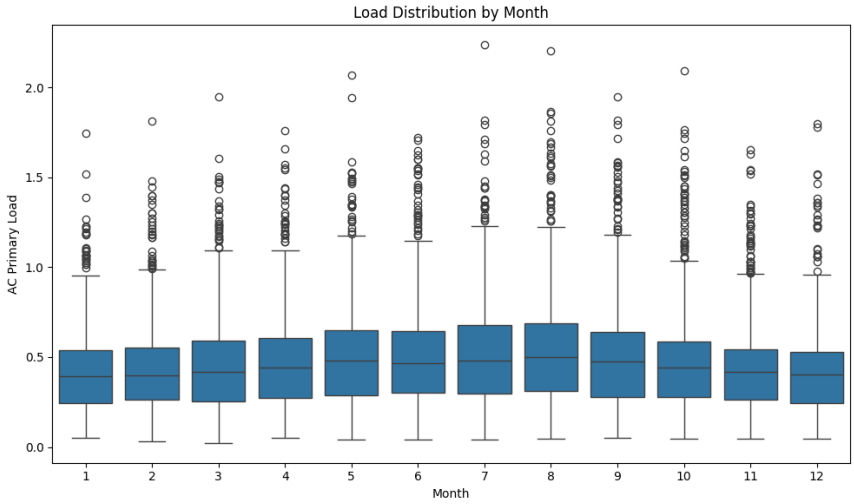

Figure 5.7: Boxplots of Load Distribution by Month

5.1.2 Results of Feature selection and Feature Engineering

In section 4.1.1 we discussed the features that are preliminary selected for load forecasting and its transformations. We also apply Standard scaler to make sure that features are on the same scale. The minor part of the outliers and non-Gaussian distribution of the features support this choice. Another important part of feature engineering is the introduction of new

predictive variables. For this Thesis we obtain lag features – previous load values for the same hour previous day, and the same hour previous week. We also introduce rolling mean variables: average of the last 3 observations and average over the last 24 observations. By this we allow models to detect very short-term trends and daily cycles and long-term trends. Another advantage of moving average features is noise reduction as models learn patterns from smoothed data and variance stabilization.

5.1.3 Trian Test Split

Next step is the train test split. Unlike standard machine learning problems, time-series data cannot be split in the random way with shuffle. An important aspect of the train-test split for the sequential data is preservation of the temporal order by chronological split. As mentioned in the methodology chapter of this Thesis we leave 80% of the first observations for the training and last 20% for the validation and test. The result of the train-test split might be seen on Figure 5.8 below.

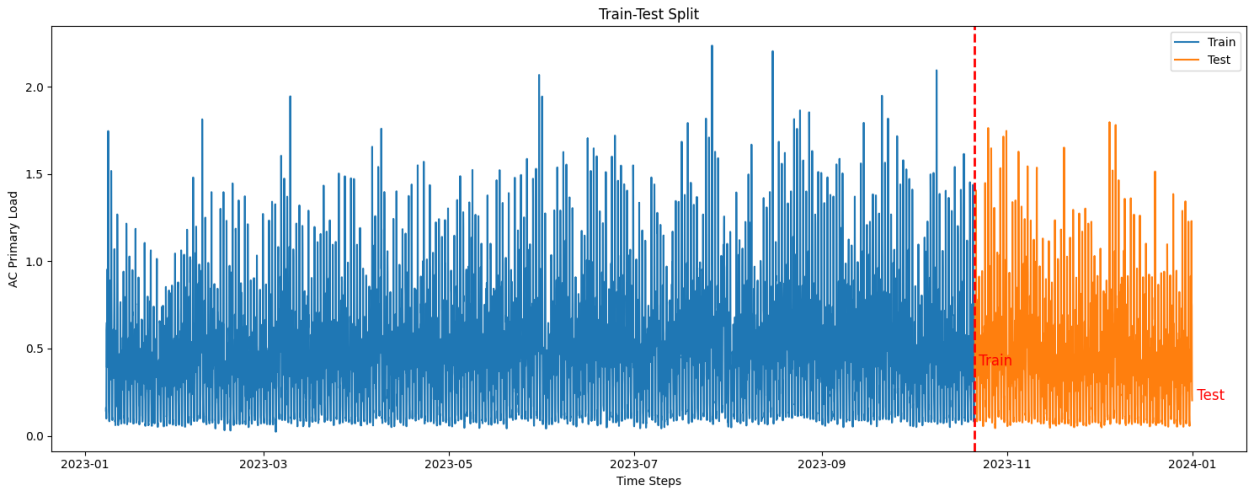

Figure 5.8: Train-Test Split

5.1.4 Stationarity Check

Augmented Dickey-Fuller (ADF) Test is used to check presence of stationery in the given rime-series. For models like ARIMA and SARIMA stationarity is crucial as they assume constant mean, variance, and autocorrelation over time. ADF test suggests the presence of stationarity, and results are presented in Table 5.3

Table 5.3: Results of the ADF Test

ADF Statistic		-9.607981368139978
p-value		1.8440851655209428e-16
Critical values	**1%**	-3.4313063970455224
	5%	-2.861962649102583
	10%	-2.566994972320543

5.2 Results of the Simulations

In this section we provide the results of the simulations. We trained several models discussed before, validated them to obtain optima parameters and tested them on the relevant set to estimate performance. Table 5.4 below summarizes the test results.

Table 5.4: Evaluation of Trained and Validated Models on Test Set

	SARIMA	**LightGBM**	**XGBoost**	**LSTM**	**GRU**
$RMSE(kW)$	0.1102	0.0412	0.0398	0.0622	0.0637
$MAE(kW)$	0.0783	0.0287	0.0276	0.0434	0.0451
$MAPE(\%)$	20.85	7.42	7.15	11.73	12.21

5.3 Discussion

The explicit analysis of various models for STLF in Kazakhstan reveals several significant insights regarding model performance, feature importance, and methodological considerations.

5.3.1 Model Performance Comparison

The results demonstrate a clear hierarchy in model performance, with gradient boosting methods (LightGBM and XGBoost) substantially outperforming both statistical and deep learning approaches. LightGBM achieved the best overall performance with the lowest error metrics (RMSE of 0.0412, MAE of 0.0287), closely followed by XGBoost (RMSE of 0.0398, MAE of 0.0276). This superior performance aligns with findings in recent literature [9] that highlight the effectiveness of boosting algorithms in handling complex, non-linear relationships in load forecasting.

The deep learning models (LSTM and GRU) performed moderately well but did not match the accuracy of the boosting methods, achieving MAPE values of 11.73% and 12.21% respectively. This contradicts some recent literature that positions RNN-based approaches as superior for time series forecasting [15],[16]. This discrepancy may be attributed to several factors:

1. The limited seasonality observed in our dataset, particularly the absence of strong weekly or monthly patterns, may have reduced the advantage of RNNs in capturing long-term dependencies.
2. The relatively small dataset size (8,760 hourly observations) might have limited the learning capacity of deep neural networks, which typically excel with larger datasets.

3. The gradient boosting algorithms' ability to handle feature interactions and non-linearities might be particularly well-suited to the specific characteristics of Kazakhstan's load patterns.

SARIMA performed notably worse than all other models (RMSE of 0.1102, MAE of 0.0783), despite the dataset passing stationarity tests. This underperformance might be attributed to the complex, non-linear relationships between load and other external variables, which linear statistical models struggle to capture effectively.

5.3.2 Feature Importance and Temporal Patterns

Our exploratory data analysis revealed several key insights that influenced model performance:

1. The strong hourly pattern observed in load distribution (Figure 5.5) proved to be highly informative for all models, with the engineered hour-of-day feature contributing significantly to prediction accuracy.
2. The relative absence of weekly and monthly seasonality (Figures 5.6 and 5.7) explains why traditional time series models like SARIMA, which rely heavily on capturing seasonal patterns, underperformed in our dataset.
3. The lag features and rolling means introduced during feature engineering were particularly valuable for the boosting models, allowing them to capture both short-term and longer-term patterns without requiring RNNs memory mechanism.
4. The correlation between meteorological variables and load was not as strong as anticipated, which explains why models that can adaptively determine feature importance (LightGBM and XGBoost) performed better than those with fixed architectures.

5.3.3 Implications for Load Forecasting in Kazakhstan

The superior performance of boosting methods on Kazakhstan's load data has several implications for the national power system:

1. The relatively low MAPE values achieved by LightGBM and XGBoost (7.42% and 7.15% respectively) indicate that these models could provide sufficiently accurate forecasts for operational planning and resource allocation in Kazakhstan's power grid.
2. The absence of strong weekly or seasonal patterns differs from load profiles typically observed in many other countries, suggesting that Kazakhstan's electrical consumption may be more influenced by industrial operations or other factors with consistent daily patterns rather than residential usage that typically varies between weekdays and weekends.
3. The moderate correlation between meteorological variables and load suggests that while temperature and renewable resource availability do influence electricity demand, other factors (possibly economic activity or industrial operations) may play equally important roles in determining Kazakhstan's electricity consumption patterns.

5.3.4 Methodological Considerations and Limitations

Several methodological aspects of our study warrant discussion:

1. The synthetic nature of our dataset, while allowing for controlled experimentation, may not fully capture the patterns and anomalies that may present in actual Kazakhstan load data from Kazakhstan. The performance metrics should be interpreted with this limitation in mind.

2. The handling of outliers (as identified in Table 5.2) was appropriate given their relatively low frequency, but extreme weather events or unusual consumption patterns might require special consideration in operational forecasting systems.
3. Our choice of standard scaling for feature normalization proved effective, particularly for the neural network models that are sensitive to feature scales. However, alternative normalization techniques might yield different results, especially for highly skewed features like Global Solar.
4. The train-test chronological split respected the temporal nature of the data but might not have captured potential seasonal variations that would occur in a multi-year dataset.
5. We do not examine the time required to train, validate models.

5.3.5 Comparison with Previous Studies

Our findings both align with and diverge from previous research in interesting ways:

1. The superior performance of ensemble learning methods is consistent with recent studies like [9], which demonstrated the effectiveness of XGBoost for load forecasting when combined with feature decomposition techniques.
2. Unlike [6] and [7], which found SARIMA and ARIMA models to perform reasonably well with MAPEs of 4.332% and 4% respectively, our implementation of SARIMA yielded a substantially higher MAPE of 20.85%. This discrepancy highlights the dataset-specific nature of forecasting performance and highlights that model selection is based on data characteristics.
3. The moderate results for our application of LSTM and GRU models contrast with some studies [12,13] that position these architectures as state-of-the-art for load forecasting. However, it aligns with research suggesting that simpler models often outperform

complex deep learning approaches when the dataset size is limited or when the underlying patterns lack the complex seasonality that RNNs excel at capturing.

5.3.6 Future Research Directions

Based on our findings, several promising directions for future research emerge:

1. Hybrid models that combine the feature selection capabilities of boosting algorithms with the sequential learning strengths of RNNs could potentially outperform either approach individually.
2. Investigating the impact of different feature engineering techniques, particularly wavelet transforms or empirical mode decomposition, could improve forecast accuracy by better separating trend, seasonal, and residual components.
3. Extending the analysis to multi-year data would provide insights into longer-term seasonal patterns and improve the robustness of model comparisons.
4. Exploring the impact of renewable energy integration on load forecasting accuracy would be particularly relevant given Kazakhstan's growing investment in renewable resources.
5. Developing specialized models for different regions of Kazakhstan could account for geographical variations in climate and consumption patterns.

In conclusion, our comprehensive evaluation of forecasting models for short-term load prediction in Kazakhstan demonstrates that gradient boosting methods, particularly LightGBM, offer the most promising approach for practical implementation. The findings suggest that successful load forecasting in Kazakhstan requires models that can adaptively handle complex feature interactions and capture the strong daily patterns observed in electricity consumption,

while being less dependent on weekly or monthly seasonality that might be more prominent in other countries.

Bibliography/References

[1] K. Mustafa et al., "Toward Holistic Energy Management by Electricity Load and Price Forecasting: A Comprehensive Survey," *IEEE Access*, vol. 11, pp. 132604-132626, 2023.

[2] T. Hong and S. Fan, “Probabilistic Electric Load Forecasting: A tutorial review,” *International Journal of Forecasting*, vol. 32, no. 3, pp. 914–938, Jul. 2016.

[3] T. Hang and M. Shahidehpour, “Load forecasting case study,” *National Association of Regulatory Utility Commissioners*, https://pubs.naruc.org/pub.cfm?id=536E10A7 2354-D714-5191-A8AAFE45D626

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

[5] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *International Journal of Forecasting*, vol. 32, no. 3, pp. 914-938, 2016, doi:10.1016/j.ijforecast.2015.11.011.

[6] G. T. Tunnicliffe Wilson, "Time Series Analysis: Forecasting and Control, 5th Edition, by George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel and Greta M. Ljung, 2015," *Journal of Time Series Analysis*, vol. 37, no. 5, pp. 701–702, Sep. 2016, doi: 10.1111/jtsa.12194.

[7] R. Adhikari and R. K. Agrawal, “An Introductory Study on Time Series Modeling and Forecasting,” *LAP Lambert Academic Publishing*, 2013.

[8] R. H. Shumway and D. S. Stoffer, ‘‘Characteristics of Time Series,’’ Time Series Analysis and its Applications: With R Examples. New York, NY, USA: Springer, 2017, pp. 77-156.

[9] S. G. N and G. S. Sheshadri, "Electrical Load Forecasting Using Time Series Analysis," in 2020 IEEE Bangalore Humanitarian Tech. Conf., Vijiyapur, India, 2020, pp. 1-6.

[10] E. Chodakowska, J. Nazarko, and Ł. Nazarko, “ARIMA Models in Electrical Load Forecasting and Their Robustness to Noise,” *Energies*, vol. 14, no. 23, p. 7952, Nov. 2021.

[11] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785-794, doi: 10.1145/2939672.2939785.

[12] Y. Wang et al., “Short-term load forecasting of industrial customers based on SVMD and XGBoost,” International Journal of Electrical Power and Energy Systems, vol. 129, p. 106830, Jul. 2021.

[13] X. Yao, X. Fu and C. Zong, "Short-Term Load Forecasting Method Based on Feature Preference Strategy and LightGBM-XGboost," in *IEEE Access*, vol. 10, pp. 75257-75268, 2022, doi: 10.1109/ACCESS.2022.3192011.

[14] G. Ke and Q. Meng, "LightGBM: A highly efficient gradient boosting decision tree", *Proc. Adv. Neural Inf. Process. Syst.*, pp. 3147-3155, 2017.

[15] W. Guo, L. Che, M. Shahidehpour, and X. Wan, “Machine-Learning based Methods in Short-Term Load Forecasting,” *The Electricity Journal*, vol. 34, no. 1, p. 106884, Jan. 2021.

[16] Y. Miky, M. R. Kaloop, M. T. Elnabwy, A. Baik, and A. Alshouny, “A recurrent cascade-neural network- nonlinear autoregressive networks with exogenous inputs (NARX) approach for long-term time-series prediction of wave height based on wave characteristics measurements,” *Ocean Engineering*, vol. 240, p. 109958, Nov. 2021.

[17] H. Eskandari, M. Imani, and M. P. Moghaddam, “Convolutional and Recurrent Neural Network based Model for Short-Term Load Forecasting,” *Electric Power Systems Research*, vol. 195, p. 107173, Jun. 2021.

[18] R. Punmiya and S. Choe, "Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing", *IEEE Trans. Smart Grid.*, vol. 10, no. 2, pp. 2326-2329, Mar. 2019.

[19] X. Ma, J. Sha and D. Wang, "Study on a prediction of P2P network loan default based on the machine learning lightGBM and XGboost algorithms according to different high dimensional data cleaning", *Electron. Commer Res. Appl*, vol. 31, pp. 24-39, Aug. 2018.