SAVONIA

University of Applied Sciences

TYPE OF REPORT – BACHELOR'S DEGREE PROGRAMME
TECHNOLOGY, COMMUNICATION AND TRANSPORT

COMPARATIVE ANALYSIS OF STATISTICAL AND MACHINE LEARNING MODELS FOR SOLAR ENERGY PRODUCTION FORECASTING AND MODEL SELECTION

AUTHOR/S : Sabuj Bhowmick

SAVONIA UNIVERSITY OF APPLIED SCIENCES

THESIS
Abstract

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology, Internet of Things			
Author(s) Sabuj Chandra Bhowmick			
Title of Thesis Comparative Analysis of Statistical and ML Models for Solar Energy Production Forecasting and Model Selection.			
Date	02/06/2024	Pages/Number of appendices	38
Client Organisation /Partners Savonia University of Applied Sciences			

Abstract (NOTE: write/insert all your text in the grey box below, also if you use copy + paste)6

Efficient energy forecasting methods have become the focal point to ensure sustainable energy production. The present study aimed to use cutting-edge machine learning models along with blended statistical models to precisely predict solar energy production. Three data-based Machine learning models-Random Forest, Support vector machine, and XGBoost-and three statistical models- Linear regression, ARMA, and ARIMA models were employed. Statistical indices such as Mean absolute error (MAE), Mean square error (MSE), and Root mean square error (RMSE) were used to determine the most precise predictive model. The results demonstrated that XGBoost was the most precise predictive model, with MAE value of 0.1618 MSE of 0.1542, and an RMSE of 0. 3927. It is recommended that the use of blended ML and statistical forecast model would be a valuable tool for policymakers, solar energy researchers, and solar farm developers.

Keywords
Artificial intelligence, Forecasting, Machine learning, Solar plant, RF, SVM, XGBoost, ARMA, ARIMA, LR, MAE, MSE, RMSE, Time series.

CONTENTS

LIST OF FIGURES

1 INTRODUCTION

In this modern era, energy is a fundamental requirement for the development of human civilization. Historically, energy production has relied heavily on burning coal and gas, both of which are based on fossil fuel and highly unsustainable. These traditional methods of energy production are the primary contributors to greenhouse gas emissions, which in turn lead to global warming and the El Niño effect. Globally, approximately 40% of greenhouse gas emissions stem from the energy sector (Bogdanov 2021). There is no alternative to reducing emissions other than diversifying from coal gas to renewable sources such as solar, hydro, wind, and geothermal for electricity production. Governments worldwide are taking numerous steps to facilitate the transition from dirty to clean energy production.

For this reason, energy researchers and research councils in many different countries are focusing on green energy production. One such initiative was taken by Savonia University of Applied Sciences research and development project on March 31,2023, when they installed the first solar plant on the campus premises.

The empirical energy production and forecasting model did not capture the entire picture. The rise of AI and its implications in the energy sector are the prime focus among the scientists and researchers in this field. Efficient energy forecasting methods become the focal point to ensure sustainable energy production.

This study focuses on utilizing a cutting-edge machine learning model along with a blended statistical model to precisely predict solar energy production. This aligns with the objective of the study which aims to maximize solar energy output as the most effective predictive model. Several Machine learning and statistical models were deployed to compare and determine which one produces the best predictive model.

The study has limitations regarding generalization across different settings and applying similar model fitting for all sources of data. However, this limitation could be mitigated by extending the study duration for longer periods. The availability of more data would enable a better understanding of insights drawn by the predictive models.

A famous Chinese proverb "*a journey of a thousand miles begins with a single step*" collected from Wikipedia, aptly illustrates Savonia's small step towards a larger transition in the energy sector.

1.1 Background of the study

Artificial intelligence (AI) plays a crucial role in advancing sustainable energy production (Ahmad 2021). This thesis is grounded in the Savonia solar energy production project, which was conducted during the summer of 2023 at Savonia University of Applied Sciences.

The outcome of this study can benefit various organizations, particularly those in the renewable energy sector, government agencies, and research bodies. Statistical and Machine learning models were implemented to get accurate and reliable results.

The primary focus of this study is to enhance and develop optimal models for forecasting solar energy output. Numerous machine learning and statistical models were examined and compared to obtain the most accurate predictions.

1.2 Research objectives.

The research objectives involve determining the maximum energy output potential of the Savonia solar production facility through the application of advanced analytical techniques. AI models will be employed to comprehensively analyze and interpret the extensive data set collected from the solar energy production facility. Additionally, the aim is to enhance future energy production by utilizing insights from the AI models to refine forecasting methods, thereby facilitating more accurate and efficient utilization of renewable energy resources.

1.3 Scope and limitations

The study explores the application of artificial intelligence (AI) in advancing sustainable energy production, with a specific focus on the Savonia solar energy production project undertaken at Savonia University of Applied Sciences during the summer of 2023. The goal is to enhance the accuracy and efficiency of solar energy production, benefiting renewable energy companies, solar power plant operators, energy grid management agencies, government bodies involved in energy policy, and researchers in the field of renewable energy and AI/ML learning. Extensive datasets collected from the Savonia solar production facility will undergo thorough analysis and interpretation using AI models to determine the maximum energy output potential. Furthermore, future energy production will be improved by refining forecasting methods through insights gained from AI models, aiming to promote more accurate and efficient utilization of renewable energy resources. Limitations of the study include potential constraints related to data availability, time, resources, generalizability, and ethical considerations regarding data privacy and confidentiality.

1.4 Literature review

While classic models for forecasting still dominate the scientific community and research enthusiast, machine learning and Artificial Intelligence models are gaining popularity for modelling

historic time series data due to their flexibility, optimization techniques, generalization, and software availability. A review paper on machine learning methods for solar radiation forecasting by Cyril (2017) demonstrates the increasing trend of utilizing ML methods in the scientific world. The paper reviewed dozens of published scientific papers on forecasting techniques based on the classic, AI, and ML models, revealing that ANN, ARIMA, naive method, SVM and k-means are the most frequently used methods, while boosting, regression tree and random forest are less commonly used.

In general AL and ML methods are gaining popularity among researchers and scientists in various fields alongside classic forecasting models. A comparative study on machine learning algorithm for forecasting solar energy production by Younes (2023) found that ANN algorithm is the most accurate model for energy production. AI and ML algorithms are predominantly used in studies for complex computational procedures and reliable output measures comparisons.

In many review papers show the comparative analysis of different statistical and machine learning model based on standard metrics of comparison (Vaia 2023). Data from different fields and applications were examined in that study. Some other studies mainly focus on one-to-one comparison of statistical models such as ARIMA and Machine learning model XGBoost in the field of epidemiology (Mirxat 2020).

The comparison-based studies were not only found in the medical field only but also found in energy sector, where ARIMA-XGBoost hybrid model is found to be more accurate in energy forecasting (Pin 2018). SVM, a popular machine learning model was found to be the best compared to the ARIMA model for forecasting daily solar energy generation (Atique 2020).

It is noticing that best performed model is study specific and based on the study objective. Statistical models were found to be best fitted in some studies and machine learning model found to be fitted accurately in another studies. It is difficult to choose only one universal machine learning model in every case or one universal statistical model in every case at a time. In many cases hybrid models perform better than machine learning and statistical models alone (Xu 2019).

This study focuses on selecting the best models for forecasting on premise solar energy production in the Savonia solar energy project. It will also assess whether Machine learning algorithms or classic time series and statistical models are the optimal options, considering time and resources constraints.

1.5 Savonia Solar power plant

Savonia University of Applied Sciences hosts a solar production facility on its campus, which serves as the focus of this research project. The main goal was to determine the maximum energy output potential since the facility was established. From the outset, we carefully recorded and securely

stored energy data on a server, which serves as the foundation for our analysis and modelling. Figure 1 shows the Savonia solar plant setup. Solar panels are installed in the roof of the main building and are restricted to the common visitors.

Figure 1.Solar plant system at Savonia UAS

1.6 Savonia Own Server: Data Record Server

All data are stored and maintained within Savonia own server infrastructure, ensuring a secure environment for the storage of valuable information. Regular backups are conducted to prevent data loss and ensure business continuity in the event of unexpected incidents. Additionally, continuous monitoring and auditing processes are employed to detect and mitigate any potential security threats or vulnerabilities.

Throughout the summer project in 2023, thorough data collection, modelling, and iterative adjustments using AI forecasting models were conducted. The collaborative efforts of project team members and the R&D specialist facilitated the gathering and analysis of relevant data, resulting in the successful achievement of all project objectives within the set timeframe.

This thesis aims to investigate the effectiveness of AI models in predicting solar energy production, exploring key questions regarding the suitability of various AI methods such as machine learning and time series analysis. Through careful examination and evaluation, valuable insights into the renewable energy sector are sought to be provided, promoting the sustainable and efficient use of solar energy resources.

2 THEORITICAL BACKGROUND

2.1 ML Models

In this thesis, machine learning models were employed to enhance the precision of solar energy production forecasting, thereby contributing to a more accurate assessment of energy resources. The subsequent sections of this thesis introduce various machine learning models.

2.1.1 Random Forest

Random forest is a widely used machine learning model in various real-life applications. The working principle of the model is illustrated in Figure 2. Several steps involved in this this model include the choice of decision tree, learning from the decision tree, bootstrap sampling, feature randomization, hyperparameters, and out of bag error estimation.

The simple algorithm is described in the following steps.

- Ensemble of decision tree: Decision trees are constructed in this case. Trees are independent from each other.
- Random feature selection: Random Forest employs random feature selection to ensure the uniqueness property. There are three different methods employed for feature selection such as Filter, Wrapper and Embedded methods. Main purpose is to reduce the dimensionality, speed up the learning process, accurate predictions and improve the results.
- Bootstrap aggregating or bagging: It is a sampling with replacement techniques by creating multiple bootstrap samples from the original dataset.
- Decision making and voting: It is a regression task that takes mean of individual tree predictions.

Algorithmically can be presented (Hastie 2009).

1. For $b=1$ to B:

(a) Draw a bootstrap sample Z^* of size N from the training data.

(b) Make a random forest tree T_b to bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.

   i. Select m variables at random from p variables.

   ii. Pick the best variable among the m.

   iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a new prediction for a new point x:

Regression: $\hat{f}_{\mathrm{rf}}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$

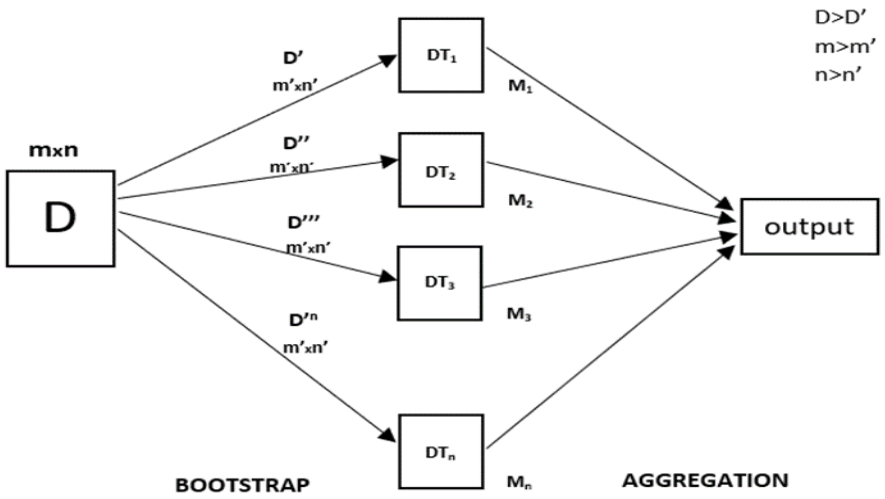

Figure 2. Random Forest regression model & how it is working (GeeksforGeeks 2024).

The Random Forest (RF) algorithm has numerous real-life applications, such as financial forecasting, demand forecasting, energy demand forecasting, weather forecasting, healthcare forecasting, sales forecasting, and many more. Random Forest offers advantages such as handling large datasets, various input features, managing non-linear relationships, and providing easily interpretable results. In a short-term load forecasting study Random Forest (RF) model outperformed other ML models due to its high dimensionality (Fan 2022). Another study based on a random forest regression model found that RF had better accuracy than SVM and IARMA model (Shijun 2020).

2.1.2 Support Vector Machine (SVM)

Support vector machine is another machine learning algorithm employed in the solar energy forecasting process. Support vector regression is a type of support vector machine that is used for regression tasks. This supervised learning algorithm operates in the principle of minimizing the error between the predicted values and the actual values. Support vector machine (SVM) is renowned for its ability to capture complex non-linear relationships in the data.

- Hyperplane: It is the decision boundary separating the data points in different classes in a feature space.
- Support vectors: It is the closest data points to the hyperplane which plays crucial role in deciding the hyperplane and margin.
- Margin: It is the distance between the support vector and hyperplane. Main objective is to maximize the margin.
- Kernel: It is a mathematical function to map the original input data points into high-dimensional feature space.
- Hard margin: It separates the data points without any misclassifications.

- Soft margin: It is employed when there exists outlier in the data points by compromising between increasing the margin and minimizing violations.
- C: Parameter C regularization is employed by a strict penalty imposed with a greater value of C resulting in fewer misclassifications.
- Hinge loss: It is a loss function used in SVM; it is used as a combination with the regular term.
- Dual problem: It enables the use of kernel tricks and more effective computing.

Support vector machine algorithm can be explained in the following step-by-step algorithm.

1. Define parameters:
    - C: regularization parameter controlling margin width
    - Kerner: kernel function(linear)
2. Initialize:
    - Alpha(α): vector of Lagrange multiplier (all zeros)
    - b: bias term (zero)
3. Iteration of training data:
    - For each data point (X_i, Y_i):

I. Calculate pair-wise inner products between X_i and all points in data.

II. Calculate violation of optimality condition (KKT conditions) for current α_i.

- Choose a violating pair of data points (x_i, x_j).
- Update α_i and α_j using chosen kernel function and violation values.
- Update bias term b based on new α_i and α_j.

4. Repeat step 3 until convergence or maximum iterations are reached.

Support Vector Machine (SVM) plays a crucial role in energy forecasting, with applications spanning solar and wind power forecasting, load forecasting, price forecasting, demand response forecasting, energy consumption forecasting, and renewable energy integration. In solar power forecasting, Support Vector Machine (SVM) accurately predicts solar generation by analyzing solar radiation, panel characteristics, and weather data. Similarly, in wind power forecasting, it predicts generation based on wind speed and turbine features. Support Vector Machine (SVM) also aids in predicting electricity demand, prices in markets, effectiveness of demand response, and energy consumption patterns across various sectors. Its versatility and accuracy contribute significantly to the efficient management and utilization of energy resources, ensuring the reliability and sustainability of energy infrastructure. A review paper focused on the application of Support Vector Machine (SVM) in solar and wind energy found that 50% of the studies utilized SVM model for solar radiation data (Zendehboudi 2018). This highlights SVM's is a popularity as a model of choice for forecasting and achieving accurate predictions.

2.1.3 XGBoost

XGBoost is another machine learning algorithm for building supervised regression models. The generic XGBoost algorithm is explained in the Figure 3.

- Initialization: Start with an initial prediction.
- Gradient Calculation: Compute the gradient of the loss function with respect to predicted values.
- Tree construction: Build decision tree for prediction of gradient.
- Update: Updating the gradient with newly constructed tree.
- Repeat: Repeating the process until the convergence.

Input: training set $\{(x_i, y_i)\}_{i=1}^{N}$, a differentiable loss function $L(y, F(x))$, a number of weak learners M and a learning rate α.

Algorithm:

1. Initialize model with a constant value:

$$\hat{f}_{(0)}(x) = \arg\min_{\theta} \sum_{i=1}^{N} L(y_i, \theta).$$

2. For m = 1 to M:
    1. Compute the 'gradients' and 'hessians':

$$\hat{g}_m(x_i) = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=\hat{f}_{(m-1)}(x)}.$$

$$\hat{h}_m(x_i) = \left[\frac{\partial^2 L(y_i, f(x_i))}{\partial f(x_i)^2}\right]_{f(x)=\hat{f}_{(m-1)}(x)}.$$

    2. Fit a base learner (or weak learner, e.g. tree) using the training set $\left\{x_i, -\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)}\right\}_{i=1}^{N}$ by solving the optimization problem below:

$$\hat{\phi}_m = \arg\min_{\phi \in \Phi} \sum_{i=1}^{N} \frac{1}{2}\hat{h}_m(x_i)\left[\phi(x_i) - \frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)}\right]^2.$$

$$\hat{f}_m(x) = \alpha\hat{\phi}_m(x).$$

    3. Update the model:

$$\hat{f}_{(m)}(x) = \hat{f}_{(m-1)}(x) + \hat{f}_m(x).$$

3. Output $\hat{f}(x) = \hat{f}_{(M)}(x) = \sum_{m=0}^{M} \hat{f}_m(x)$.

Figure 3. Generic unregularized XGBoost algorithm (Wikipedia. 2024).

XGBoost, a highly effective machine learning algorithm, finds widespread use in forecasting across diverse fields. Its ability to analyze historical data and predict future trends makes it valuable in forecasting stock prices, currency exchange rates, consumer demand for products, energy consumption patterns, and weather conditions (Shin 2022).

In finance, XGBoost assists in predicting market trends and stock prices by analyzing historical data and market indicators. Similarly, in demand forecasting, it accurately predicts consumer demand based on sales history and external factors (Nti 2019).

For energy forecasting, XGBoost predicts demand and supply dynamics using historical consumption patterns, weather data, and economic indicators. Moreover, in weather forecasting, XGBoost efficiently predicts meteorological variables like temperature and precipitation by analyzing past weather data. In a probabilistic solar irradiance forecasting study based on XGBoost model, accurately predicted the real dataset compared to other algorithms (Li 2022).

2.2 Statistical models

Statistical models are extensively utilized in forecasting and model building across various fields. In this thesis three such models are employed for model fitting and forecasting solar energy production.

2.2.1 Linear Regression

Classic linear regression is a commonly used statistical model in time series data analysis. While linear regression is often associated with predicting continuous outcomes in cross-sectional data, it can also be adapted for time series forecasting by incorporating time-related features.

A simple linear regression model can be represented in the equation form.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon_i \quad (1)$$

Where, β represents the parameter and ε is the error term.

We can represent the linear regression model in terms of lagged value for time series data.

$$Y_t = \beta_0 + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \cdots + \beta_p X_{t-p} + \epsilon_t \quad (2)$$

Where:

Y_t is the dependent variable at time t.

X_{t-i} are lagged values of the independent variables.

$\beta_0, \beta_1, \beta_2, \ldots\ldots\ldots\ldots, \beta_p$ are the coefficients to be estimated.

The estimated coefficients $\hat{\beta}$ can be written as matrix form.

$$\hat{\beta} = (X^T X)^{-1} X^T \mathrm{y} \quad (3)$$

Where:

- X is the design matrix of independent variables, where row represents a data point and column represents a feature,

- y is the vector of observed values of the dependent variable,
- X^T denotes the transpose of X,
- $(X^T X)^{-1}$ is the inverse of the matrix $X^T X$
- $\hat{\beta}$ is the vector of estimated coefficients.

Linear regression is a versatile and widely used statistical method with applications in time series forecasting. Although time series data often exhibit complex patterns and dependencies, linear regression can still be valuable in certain contexts. It can be applied for trend estimation, seasonal adjustment, and as a simple baseline model for forecasting. For instance, a linear regression model for global solar radiation in Nigeria demonstrated that temperature based linear regression model could be provide a better fit for estimating global solar radiation data compared to an existing sunshine-based model (Okundamiya 2013).

2.2.2 ARIMA

ARIMA (autoregressive integrated moving average) is a commonly used time series model for forecasting. The ARIMA model extends the ARMA model by including the Integration component (I). Integration component involves differencing the time series data to make it stationary. If the data shows nonstationary behavior, such as varying mean and variance, in such case ARIMA model with appropriate differencing is chosen (Vaia 2023). The parameters *p, d*, and *q* represent ARIMA model, denoted as (*p, d, q*), where *p, d,* and *q* are the orders of autoregressive AR part, differencing, and moving average (MA) part, respectively. The Autoregressive and the moving average process can be represented in the form of mathematical equations (Alsharif 2019).

AR Model: AR model can represent as follows.

$$y_t = \delta + \emptyset_1 y_{t-1} + \emptyset_2 y_{t-2} + \ldots\ldots + \emptyset_p y_{t-p} + \epsilon_t \qquad (4)$$

The term has a similar form as linear regression. δ is the intercept term, y_{t-i} is the regressor, $\emptyset_{t-i}$ is the parameters and ε is the error term. The only special thing is the regressors are the dependent variable's own lagged terms. If lag up to p is included in the model like above, the AR process is said to be of order *p.*

Moving Average Model: MA model has the following form.

$$y_t = \mu + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots\ldots + \theta_p \epsilon_{t-p} + \epsilon_t \qquad (5)$$

The Autoregressive Integrated Moving Average (ARIMA) model is a popular and effective time series forecasting method that can be applied to solar energy estimation. Solar energy forecasting is essential for optimizing the integration of solar power into the energy grid, managing energy demand, and ensuring grid stability. The ARIMA model captures the temporal dependencies and patterns in solar energy production by modelling the series as a combination of autoregressive (AR), differencing (I), and moving average (MA) components (Rigby 2024).

A comprehensive study on forecasting solar radiation based on 37 years data in Seoul, South Korea, modeled and fitted by the ARIMA model, produced an excellent weekly and monthly solar forecasting with a 68% of R-squared value (Alsharif 2019). The ARIMA model is recognized as being the best for forecasting energy consumption in many settings (Ozturk 2018).

2.2.3 ARMA

In the observed time series, sometimes high order autoregressive (AR) or moving average (MA) model is needed to model the underlying process. The ARMA model is particularly for stationary time series data. In this case autoregressive moving average process (ARMA) model would be better choice.

An ARMA model typically expressed as the conditional mean of y_t as a function of past observations, $y_{t-1}, y_{t-2}, \ldots\ldots, y_{t-p}$ and the past errors, $\epsilon_{t-1}, \epsilon_{t-2}, \ldots\ldots, \epsilon_{t-p}$. The number of past observations y_t that depends on *p,* is the AR degree the number of errors y_t that depends on *q*, is the MA degree. The general ARMA model representation is denoted by ARMA (*p, q*) (Rojas 2008).

The ARMA (*p, q*) model representation is denoted by:

$$y_t = c + \emptyset_1 y_{t-1} + \emptyset_2 y_{t-2} + \ldots\ldots + \emptyset_p y_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots\ldots + \theta_p \epsilon_{t-p} \quad (6)$$

where $\epsilon_t$is an uncorrelated error process with mean zero. Which is the stationarity of the data.

In the lag operator polynomial notations, $L^i y_t = y_{t-i}$ defines the degree AR lag operator polynomial.

$$\emptyset(L) = (1 - \emptyset_1 L - \ldots\ldots - \emptyset_p L^p) \quad (7)$$

MA with q degree can be expressed as a lag operator.

$$\theta(L) = 1 + \theta_1 L + \ldots\ldots + \theta_q L^q \quad (8)$$

In lagged operation ARMA model can be represented as:

$$\emptyset(L) y_t = c + \theta(L) \epsilon_t \quad (9)$$

From this model the following expression can be achieved.

$$y_t = \mu + \frac{\theta(L)}{\emptyset(L)} \epsilon_t = \mu + \varphi(L) \epsilon_t \quad (10)$$

Where, μ is the unconditional mean of the process, and $\varphi(L)$ is infinite degree lagged operator.

The ARMA model assumes a normal distribution of the error, but considering this forecast did not fit well with a Gaussian law of normality. Therefore, a recursive ARMA-GARCH model is adopted to compensate for better prediction (David 2016). In a laboratory level micro grid application of solar energy generation and prediction processes, the ARMA model excels at short- and medium-term solar forecasting (Huang, 2012). In this study the normality assumption for ARMA and

ARIMA model did not fulfill (ARMA residuals D = 0.24746, p-value < 2.2e-16 and ARIMA residuals D = 0.24746, p-value < 2.2e-16). Therefore, simple ARIMA and ARMA model were chosen for analysis.

3 METHODOLOGY

This methodology section is divided into three sections. In the first section the data acquisition procedure is described. Data preparation is described in the second section and in the final section selection of models and their implementation is broadly described.

3.1 Data Acquisition

Savonia solar production facility was installed on March 31, 2023. The data was recorded on an hourly basis, with the final record of the energy production completed on August 28, 2023. Table 1 illustrates the energy production pattern for the past five months (March 31st to August 31st). Data is stored in the database with three variables timestamp, energy record and total energy record. Energy is recorded hourly, and the total energy represents the cumulative energy production of the energy record. A sample dataset is provided below. The data was recorded in the database with a digit decimal as a csv format.

Table 1. Sample dataset representation of energy data.

	time	energy	total_energy
1	2023-03-31 12:30:49	0.0	16.6
2	2023-03-31 12:31:49	0.1	16.7
3	2023-03-31 12:32:49	0.0	16.7
4	2023-03-31 12:33:49	0.0	16.7
5	2023-03-31 12:34:49	0.0	16.7
6	2023-03-31 12:35:49	0.1	16.8
7	2023-03-31 14:00:00	1.2	18.5
8	2023-03-31 15:00:00	0.3	18.8
9	2023-03-31 16:00:00	0.0	18.8
10	2023-03-31 17:00:00	0.0	18.8

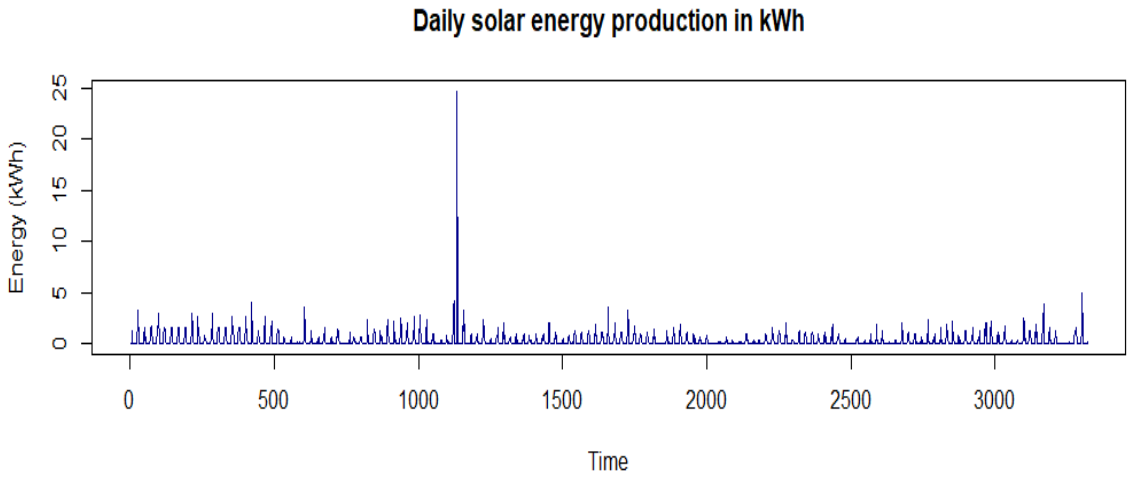

Figure 4. Solar energy production in kWh.

3.2 Data Preparation

Actual data is collected and prepared for analysis using RStudio software (Setiawan 2020). RStudio is an integrated development environment for R programming language. R programming language is widely used for data analysis, data manipulation, statistical modeling, machine learning modelling for prediction and other complex analyses.

Solar energy data was recorded in a csv file format, read.csv("S:\\Soldata\\solardata-full.csv") reads the file in table format and creates a data frame from it. For forecasting statistical models and testing forecast library was used. Libraries such as ggplot2, randomForest, e1071, xgboost, and MASS were used in the data analysis (The R project for Statistical Computing 2024). Forecast is a powerful library for time series data analysis. It has built-in functions used to analyze patterns, fitting time-series models, and forecast the time series data. For visualization, ggplot2 was used to create elegant data representations.

The e1071 library is used for functions such as latent class analysis, short-time Fourier transform, fuzzy clustering, support vector machines, shortest path computation, bagged clustering, naïve Bayes classifier and generalized k-nearest neighbor (The R project for Statistical Computing 2024).

The energy data frame was loaded, split into 80% for training and 20% for testing the model. Differencing and box-cox transformation was adopted to transform the actual data. A simple R code is provided for reference purposes.

```
# Load required libraries
library(ggplot2)
library(forecast)
library(randomForest)
```

```
library(e1071)
library(xgboost)
library(caret)
library(diagram)
# Load and preprocess data

Soldatafull <- read.csv("S:\\Soldata\\solardata-full.csv")
FullTime <- (Soldatafull$time)[-1]
Full_Total_energy <- diff(Soldatafull$energy)
Time_Solfull <- as.POSIXct(FullTime, format = "%Y-%m-%d %H:%M:%S")
FullDF <- data.frame(Time_Solfull, Full_Total_energy)
# Train-test split

train_size <- 0.8
train_rows <- round(nrow(FullDF) * train_size)
train_data <- FullDF[1:train_rows, ]
test_data <- FullDF[(train_rows + 1):nrow(FullDF), ]

# Box-Cox transformation
lambda <- BoxCox.lambda(train_data$Full_Total_energy)
train_data$Transformed_Energy <- BoxCox(train_data$Full_Total_energy, lambda)
test_data$Transformed_Energy <- BoxCox(test_data$Full_Total_energy, lambda)

# Fit ARIMA model
arima_model <- auto.arima(train_data$Transformed_Energy)
arima_residuals <- residuals(arima_model)
# Fit ARMA model (using auto.arima with stationary=TRUE)
arma_model <- auto.arima(train_data$Transformed_Energy, stationary=TRUE)
arma_residuals <- residuals(arma_model)
# Histogram with normal curve
hist(arima_residuals, main = "Histogram of ARIMA Residuals", xlab = "Residuals", breaks = 20,
probability = TRUE)
lines(density(arima_residuals), col = "blue")
curve(dnorm(x, mean=mean(arima_residuals), sd=sd(arima_residuals)), add=TRUE, col="red",
lwd=2)
# Q-Q plot
qqnorm(arima_residuals, main = "Q-Q Plot of ARIMA Residuals")
qqline(arima_residuals, col = "red")

# Shapiro-Wilk test for ARIMA residuals
shapiro_test_arima <- shapiro.test(arima_residuals)
print(shapiro_test_arima)
# Kolmogorov-Smirnov test for ARIMA residuals
ks_test_arima <- ks.test(arima_residuals, "pnorm", mean=mean(arima_residuals),
sd=sd(arima_residuals))
print(ks_test_arima)

# Visual inspection: ARMA residuals

# Histogram with normal curve
hist(arma_residuals, main = "Histogram of ARMA Residuals", xlab = "Residuals", breaks = 20,
probability = TRUE)
lines(density(arma_residuals), col = "blue")
curve(dnorm(x, mean=mean(arma_residuals), sd=sd(arma_residuals)), add=TRUE, col="red",
lwd=2)

# Q-Q plot
```

```
qqnorm(arma_residuals, main = "Q-Q Plot of ARMA Residuals")
qqline(arma_residuals, col = "red")
# Shapiro-Wilk test for ARMA residuals
shapiro_test_arma <- shapiro.test(arma_residuals)
print(shapiro_test_arma)
# Kolmogorov-Smirnov test for ARMA residuals
ks_test_arma <- ks.test(arma_residuals, "pnorm", mean=mean(arma_residuals), sd=sd(arma_re-
siduals))
print(ks_test_arma)
par(mfrow=c(1, 1))
```

In another way R-studio is a popular data science development environment.

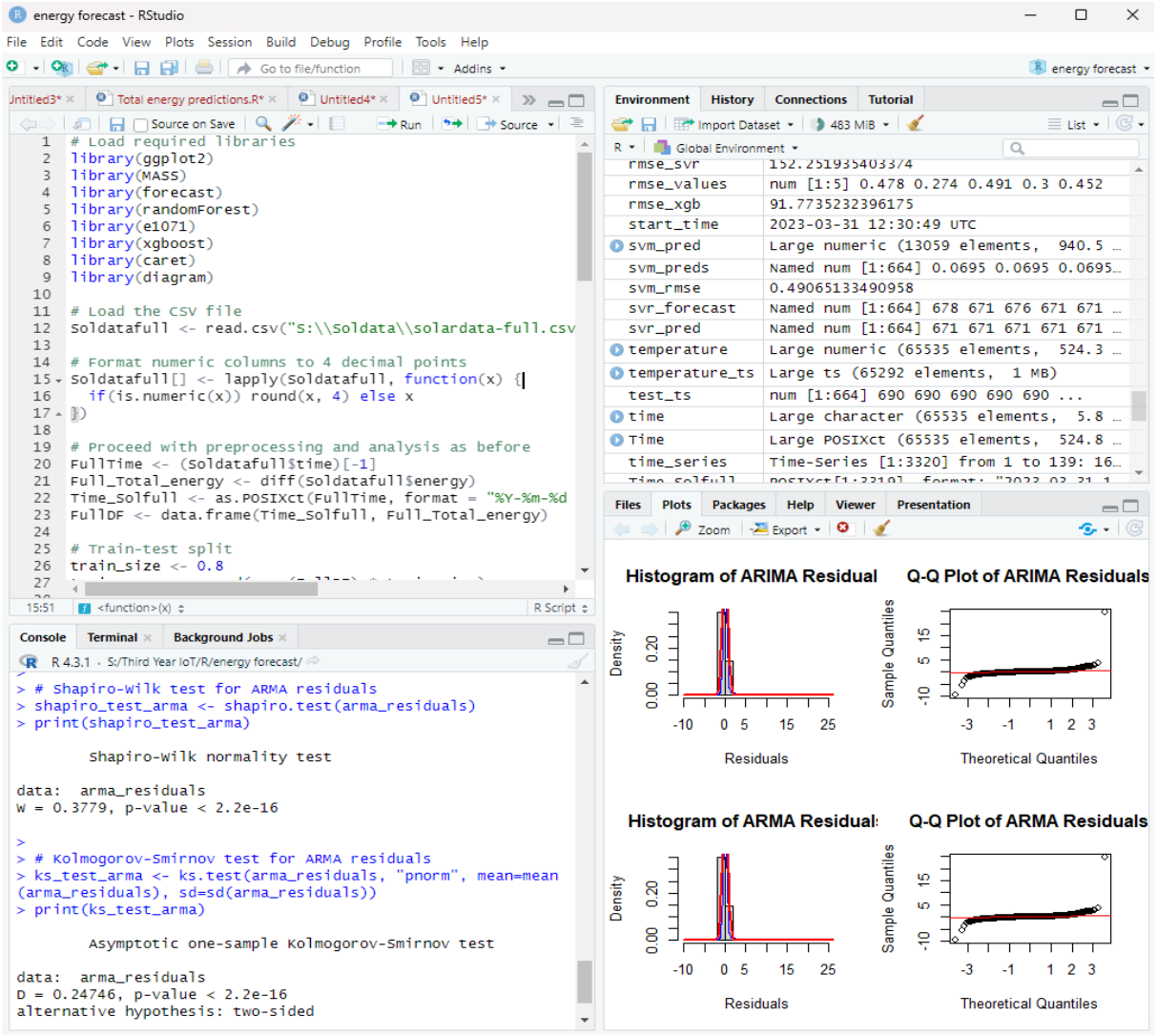

Figure 5. R-studio development environment.

Figure 5 displays the R-studio development environment. This is an integrated environment that enables users to access four windows at a time.

3.3 Model Selection

Model performance indices were considered to evaluate the best model. The main performance indices include Mean absolute error (MAE), Mean square error (MSE), Root mean squared error (RMSE), and R-squared (R^2) are. These metrics are standard for assessing model performance. The following Table 2 summarizes the model performance indices:

Table 2. Table of model performance index.

Performance index	Range	Model selection criteria
Mean Absolute error (MAE)	[0, ∞)	Choose the model with the lowest MAE value. Lower MAE indicates better predictive accuracy.
Mean Squared Error (MSE)	[0, ∞)	Choose the model with the lowest MSE value. Lower MSE indicates better predictive accuracy.
Root Mean Squared Error (RMSE)	[0, ∞)	Choose the model with the lowest RMSE value. Lower RMSE indicates better predictive accuracy.
R-squared $\boldsymbol{R^2}$	[0, 1]	Choose the model with the highest R-squared value. Higher R-squared values indicate better explanatory power of the model.

3.3.1 Mean absolute error (MAE)

Mean absolute error (MAE) is utilized to assess the accuracy of the forecasting model. It quantifies the absolute difference between predicted and the target value. Mathematically, MAE can be represented in the following formula.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}|}{n} \qquad (11)$$

Where, n is the number of the observation in the dataset.

y_i is the target value and $\hat{y}$ is the predicted value.

3.3.2 Mean square error (MSE) & Root mean square error (RMSE)

Mean square error (MSE) is the average of the squared difference between the actual and the predicted value. It measures the variance of the dataset. Root mean squared error is a standard

metric used as a performance measure for algorithms involving prediction or forecasting. It is calculated by taking the square root of the average of the squared difference between the prediction and the actual value. RMSE represents the sample standard deviation of the differences between predicted values and observed values (also known as residuals). The advantage of the RMSE is that it is less sensitive to the outliers compared to mean square error (MSE). MSE is always positive and a value closer to 0 or lower indicates better performance.

$$Mean\ square\ error(MSE) = \frac{\sum_{i=1}^{n}(\hat{Y}-Y_i)^2}{n} \qquad (12)$$

Were, $\hat{Y}$ is the predicted valuer and Y_i is the observed or actual value.

$$Root\ mean\ square(RMSE) = \sqrt{MSE} \qquad (13)$$

RMSE is always between 0 and 1 lower the score better the performance. In the ideal condition score is zero which is never achieved in the real-life scenario.

3.3.3 R-Squared

R-squared is used to evaluate the effectiveness of fit of a regression model. In the optimal condition R^2 value is 1, the value closer to 1 means model fitted better. R^2 b be represented as follows.

$$R^2 = 1 - \frac{SS_{residual}}{SS_{Total}} \qquad (14)$$

The sum of squares of the residuals, also called the residual sum of squares:

$$SS_{residual} = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \qquad (15)$$

Total sum of squares proportional to the variance of the data.

$$SS_{total} = \sum_{i=1}^{n}(Y_i - \bar{Y})^2 \qquad (16)$$

In the best fitted model $SS_{residual} = 0$ and $R^2 = 1$.

4 RESULTS

The comparison of statistical and ML models are shown in Figure 6 and Figure 7, stacked together in one plot, and individually presented in Figure 8 through Figure 19. Plots are paired by actual vs predicted for observed and transformed datasets. The dataset was transformed with the difference technique, a common technique used in time series analysis to transform the data. Another technique such as box-cox transformation is also used in the time series data analysis. The

first three sections present the Statistical models, followed by the ML models. ARIMA, ARMA, and Linear regression models are presented in Figure 8 through Figure 13.

The visual representation of the observed vs predicted plots in the statistical models does not fit well. This trend was also observed in the first two ML models, Random Forest and

SVM models. However, only one ML model fits well with the observed dataset in both the actual dataset and the transformed dataset. XGBoost, based on gradient boosting, demonstrates a good fit with the observed dataset.

Visualization of the predicted data is presented in Figure 18 and Figure 19.

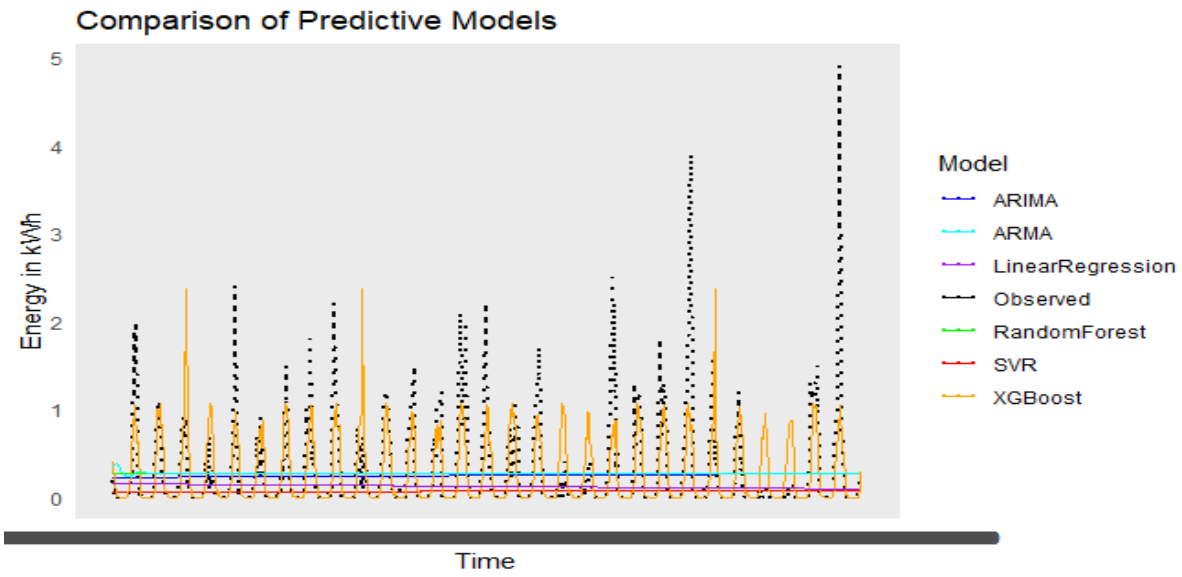

Figure 6. Comparison of Predictive Models for actual vs predicted values.

The Figure 6 depicts the comparison of fitted vs actual values in the actual dataset for six models. It clearly demonstrates that XGBoost performs better than other models.

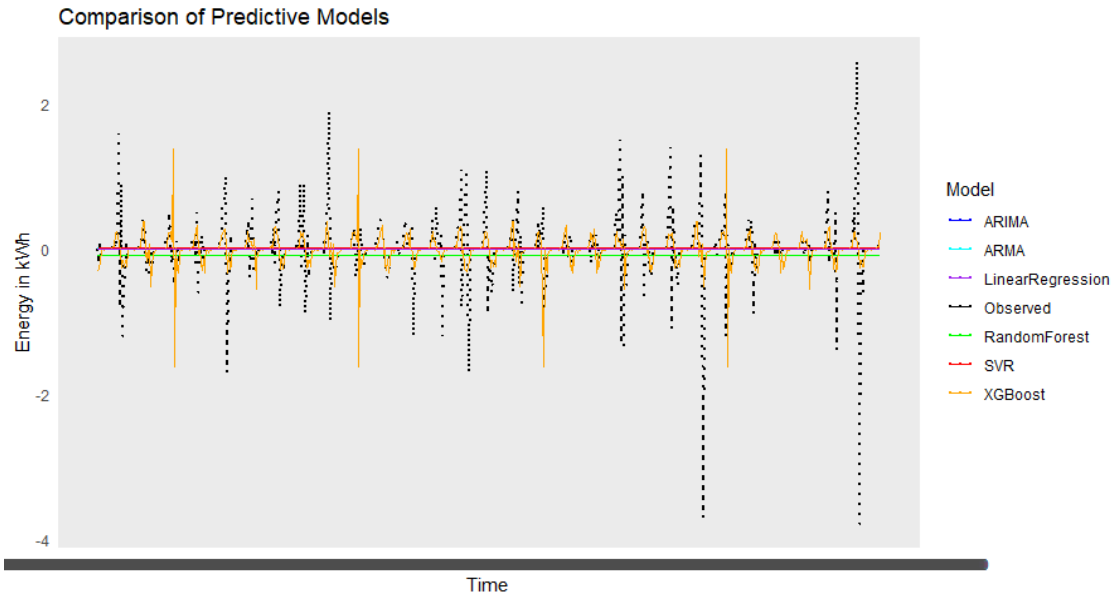

Figure 7. Comparison of Predictive Models for transformed dataset (Differencing).

In the transformed dataset XGBoost closely follows most of the observed fitting, whereas other models fail to adequately fit the observed datapoints.

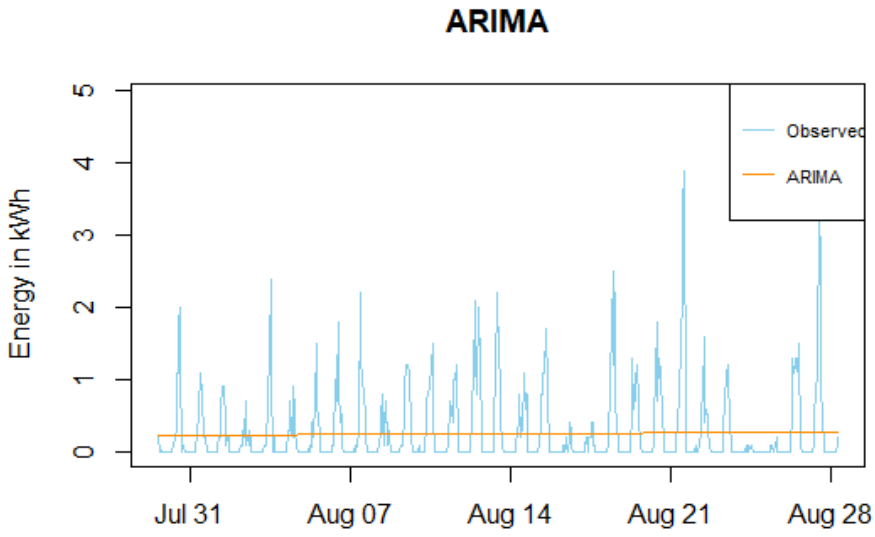

Figure 8. ARIMA model for observed and predicted values actual dataset.

The ARIMA model is based on the autoregressive, differencing and moving average components to capture the characteristics of the time-series data. Figure 8 illustrates the representation of

observed and fitted ARIMA model for the actual dataset. The figure clearly shows a discrepancy between observed and the actual datasets.

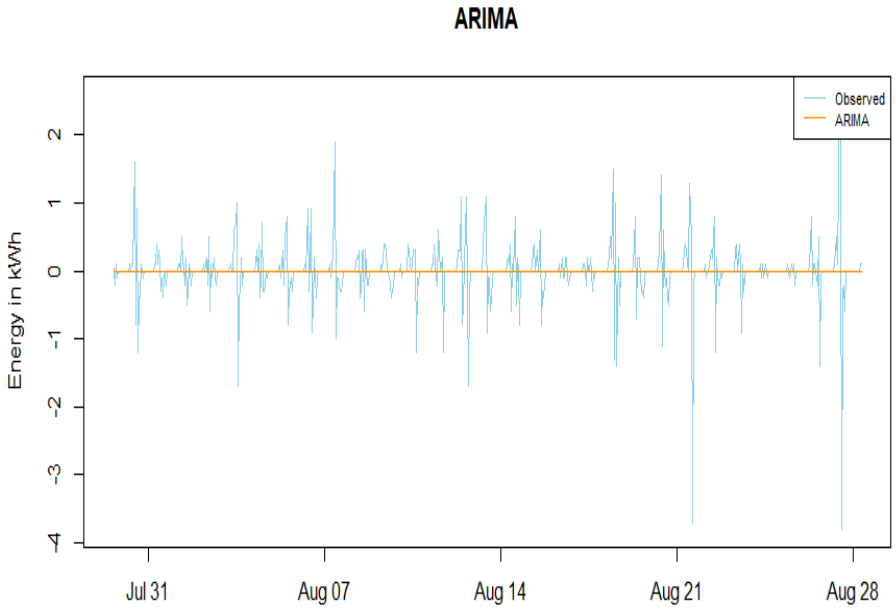

Figure 9. ARIMA model for observed and predicted values transformed dataset (Differencing). Even in the transformed dataset, the ARIMA model fails to adequately fit the observed points.

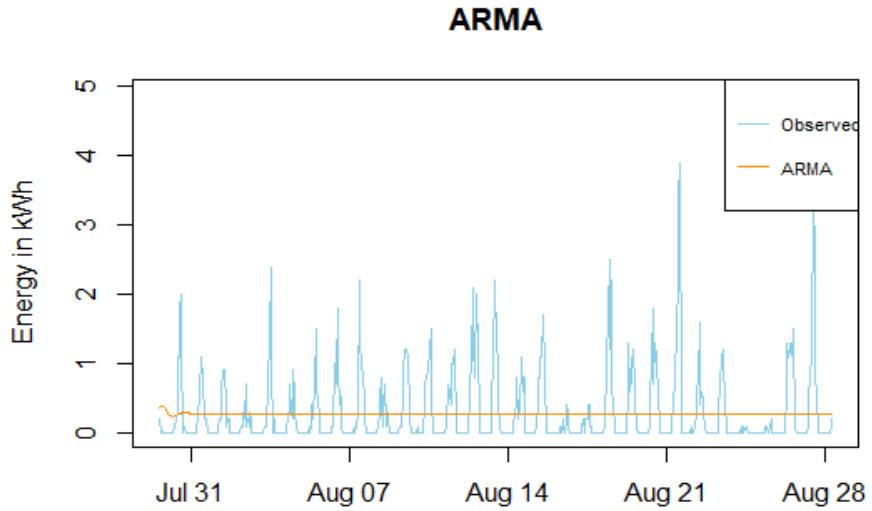

Figure 10. ARMA model for observed and predicted values actual dataset.

The ARMA model, another time-series model for forecasting and predicting actual datasets, relies on two main components: autoregressive and moving average. However, ARMA model does not adequately fit the observed dataset.

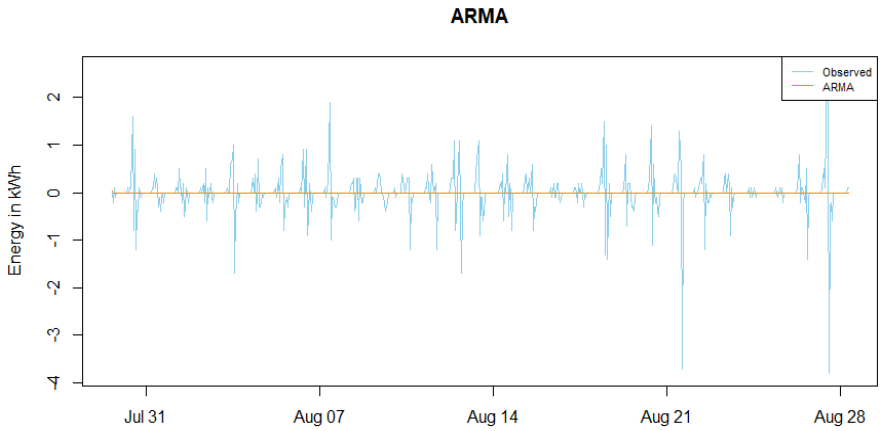

Figure 11. ARMA model for observed and predicted values transformed dataset (Differencing).

A similar trend is observed in the transformed dataset, where the ARMA model fails to capture most of the information in the observed dataset.

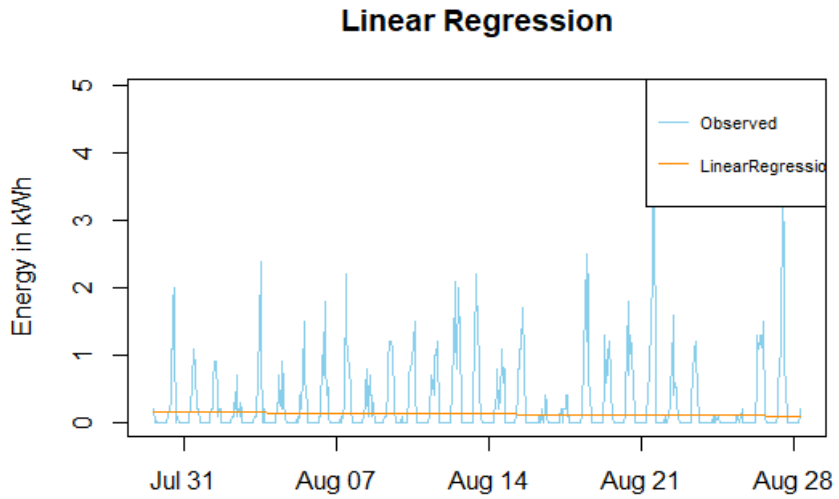

Figure 12. Linear Regression model for observed and predicted values actual dataset.

While linear regression is a commonly used model in the prediction and forecasting process, it does not produce the best results in this case.

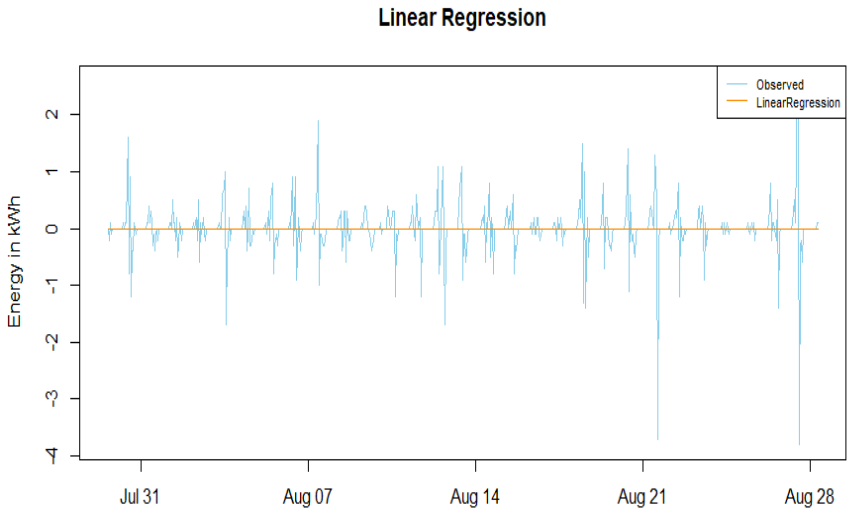

Figure 13. Linear Regression model for observed and predicted values transformed dataset (Differencing).

In the transformed dataset, the trend is linear and cannot capture the nonlinear aspects of the observed dataset.

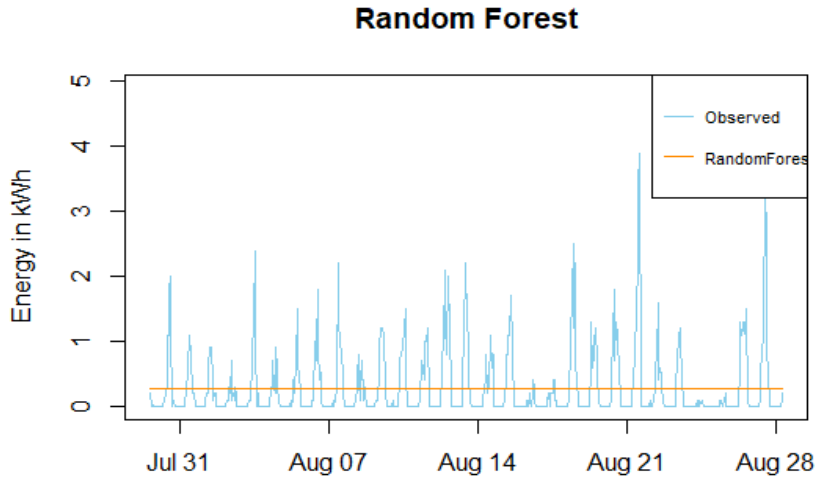

Figure 14. Random Forest model for observed and predicted values actual dataset.

Random forest, as the first ML model fitted for forecasting and prediction in this case, did not perform well in capturing the nonlinearity of the observed dataset.

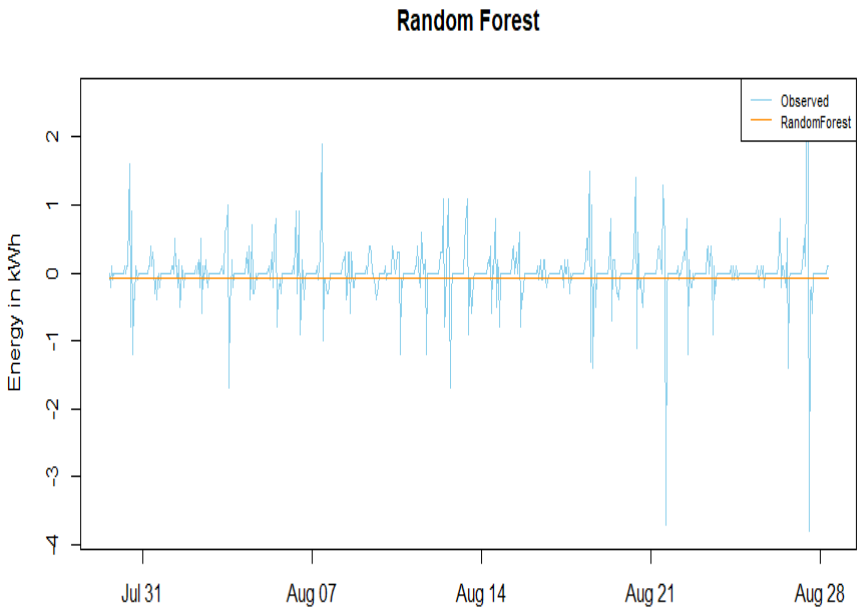

Figure 15. Random Forest model for observed and predicted values transformed dataset (Differencing).

The nature of the Random Forest model is to capture most of the observed dataset and the non-linearity of the dataset. However, its feature importance and complex memory management, it does not adequately fit the predicted data points to the observed dataset.

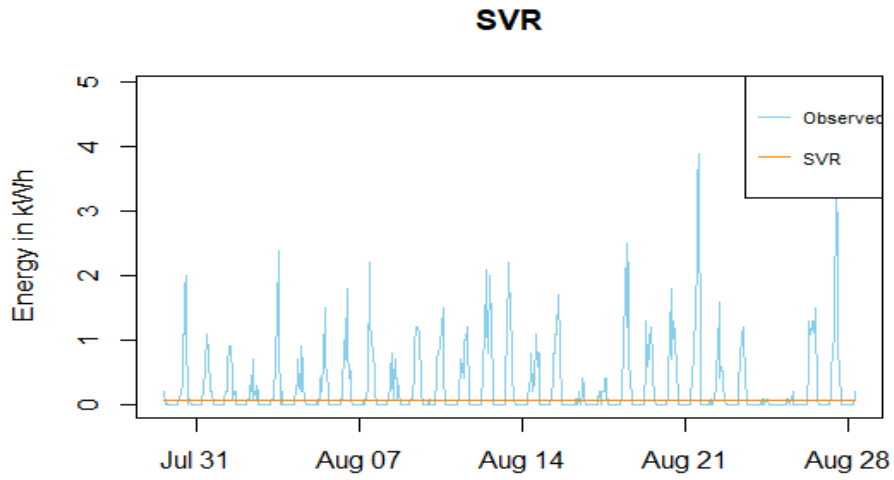

Figure 16. SVM model for observed and predicted values actual dataset.

Support vector machine is best suited for its high dimensionality and robustness. However, SVM did not fit well in this scenario.

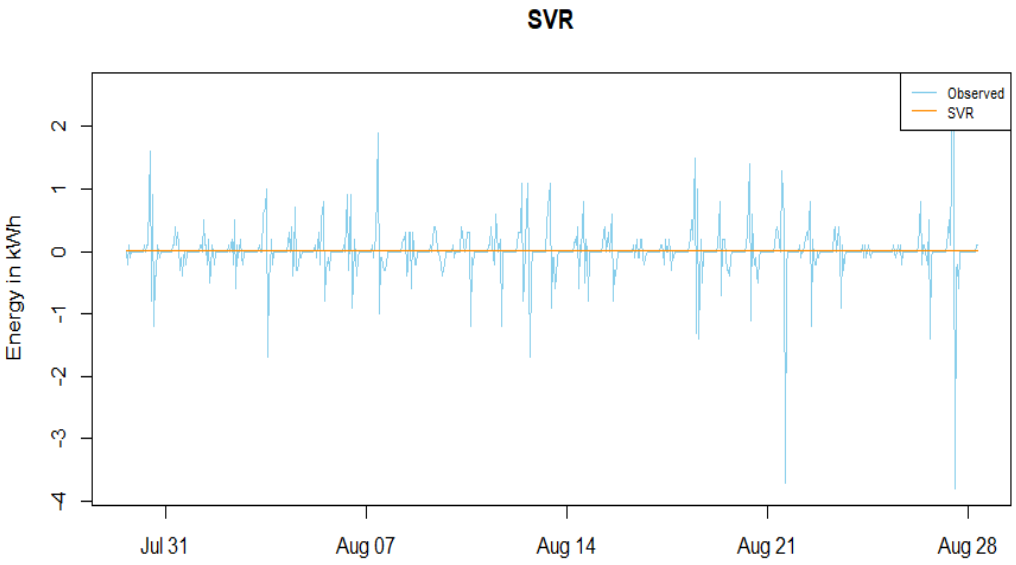

Figure 17. SVM model for observed and predicted values transformed dataset (Differencing).

In the transformed dataset SVM model remained unchanged and did not capture the non-linearity of the dataset.

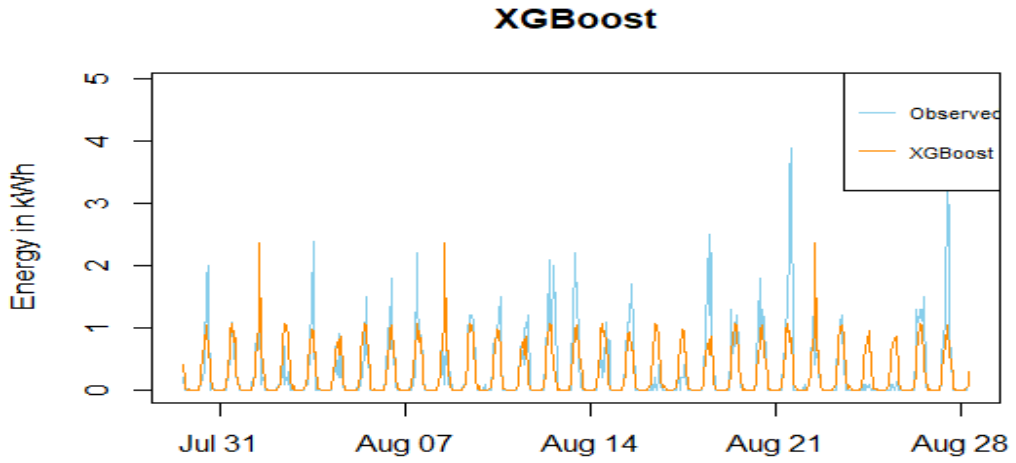

Figure 18. XGBoost model for observed and predicted values for actual dataset.

XGBoost is renowned for its feature importance, gradient boosting, ability to capture nonlinearity, and complex computation techniques. The model outperforms all other statistical and ML models in capturing the majority of the observed datapoints.

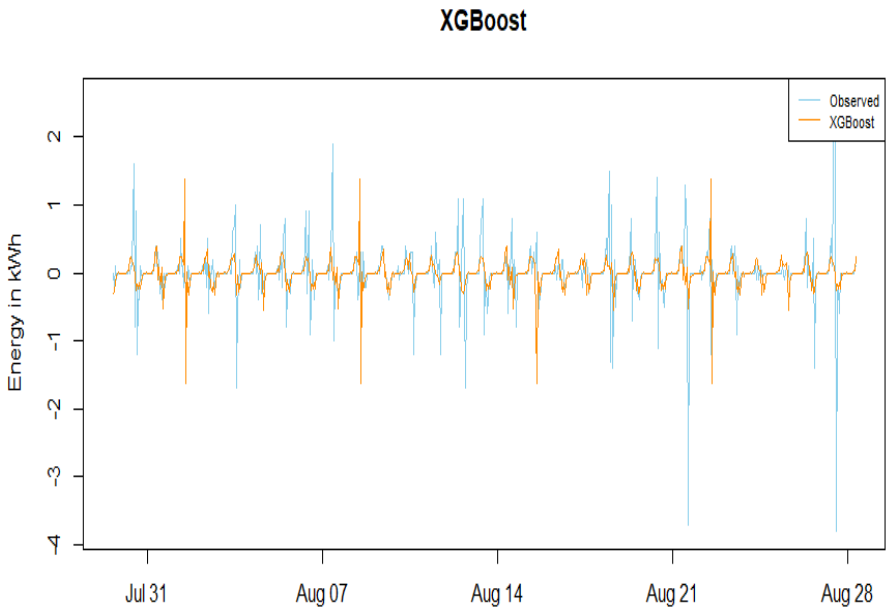

Figure 19. XGBoost model for observed and predicted values for transformed dataset (Differencing).

In the transformed dataset, XGBoost performs well and fits closely to the observed data points. Unlike other models that fail to adequately fit the observed data points, XGBoost provides better results for forecasting and prediction. Statistical models mainly based on the strong mathematical assumptions that needs to be fulfilled otherwise data need to be transformed and modified to meet the model assumptions. In this study the normality assumption for ARMA and ARIMA model did not fulfill (ARMA residuals D = 0.24746, p-value < 2.2e-16 and ARIMA residuals D = 0.24746, p-value < 2.2e-16). The normality test is based on the Asymptotic one-sample Kolmogorov-Smirnov test which is the standard test of normality.

The performance of the model fitting is primarily evaluated using performance metrics such as Mean square error (MSE), Mean absolute error (MAE) and Root mean square error (RMSE). Table 3 displays the model performance measurements for ARIMA, ARMA, Random Forest, SVM, XGBoost, and Linear Regression models for actual dataset and Table 4 display the transformed dataset.

In both cases, XGBoost exhibits the lowest measurements in MSE, MAE, and RMSE and indicates superior performance compared to other models.

Table 3. Performance Metrics for Different Models for actual dataset.

Model	MSE	MAE	RMSE
ARIMA	0.2856	0.3499	0.5344
ARMA	0.2858	0.3618	0.5346
RandomForest	0.2854	0.3631	0.5342
SVR	0.3286	0.2898	0.5732
XGBoost	0.1542	0.1618	0.3927
LinearRegression	0.3095	0.3022	0.5563

Table 3 shows the overall performance of six models, where XGBoost performs better than other models.

Table 4. Performance Metrics for Different Models Transformed dataset.

Model	MSE	MAE	RMSE
ARIMA	0.1588	0.1691	0.3985
ARMA	0.1588	0.1691	0.3985
RandomForest	0.1647	0.2146	0.4059
SVR	0.1593	0.1803	0.3991
XGBoost	0.1697	0.1727	0.4119
LinearRegression	0.1588	0.1692	0.3985

Table 4 represents the performance metrics of the transformed dataset. The differences in the metrices are insignificant in many cases. In the transformed dataset XGBoost did not fit well whereas ARIMA and ARMA model perform better than other models in that case. After box-cox transformation for ARIMA residuals with X-squared = 2.7126, df = 1, p-value = 0.09956, and ARMA residuals with X-squared = 2.7126, df = 1, p-value = 0.09956 shows that there is no significant evidence to suggest that residuals are autocorrelated. Which fulfills the assumptions of independence and hence in this case ARIMA and ARMA models fit well. Differencing leads to loss of data, in general the more differences used, the more data loss occurs (Mirxat 2020). In this case, 665 data points were lost due to differencing processes.

The visual representation of the Table 3 and Table 4 presented in the following figures (Figure 20 to Figure 25).

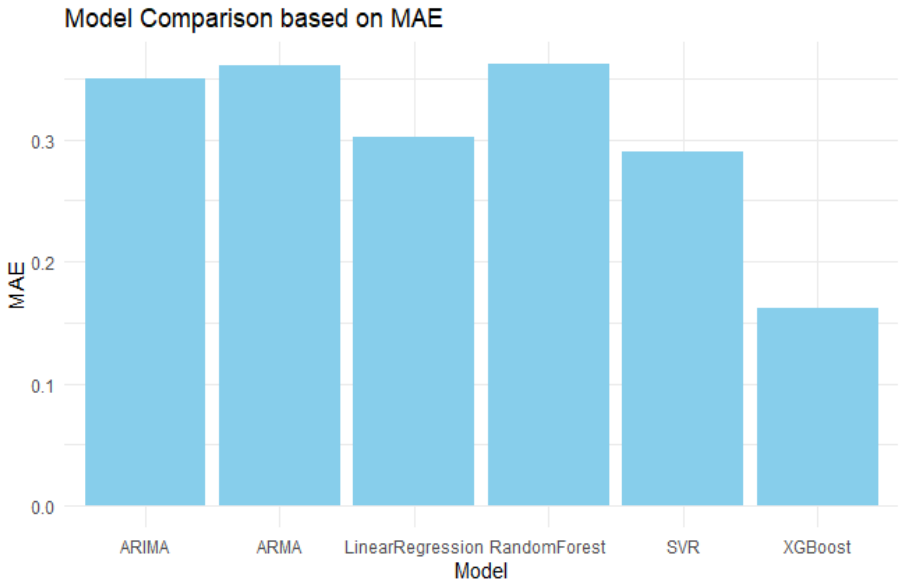

Figure 20. Model comparison based on MAE for actual dataset.

Figure 20 displays the comparison of the statistical and ML models Based on Mean Absolute Error (MAE). It is evident from the graph that XGBoost has the lowest MAE score compared to the other models.

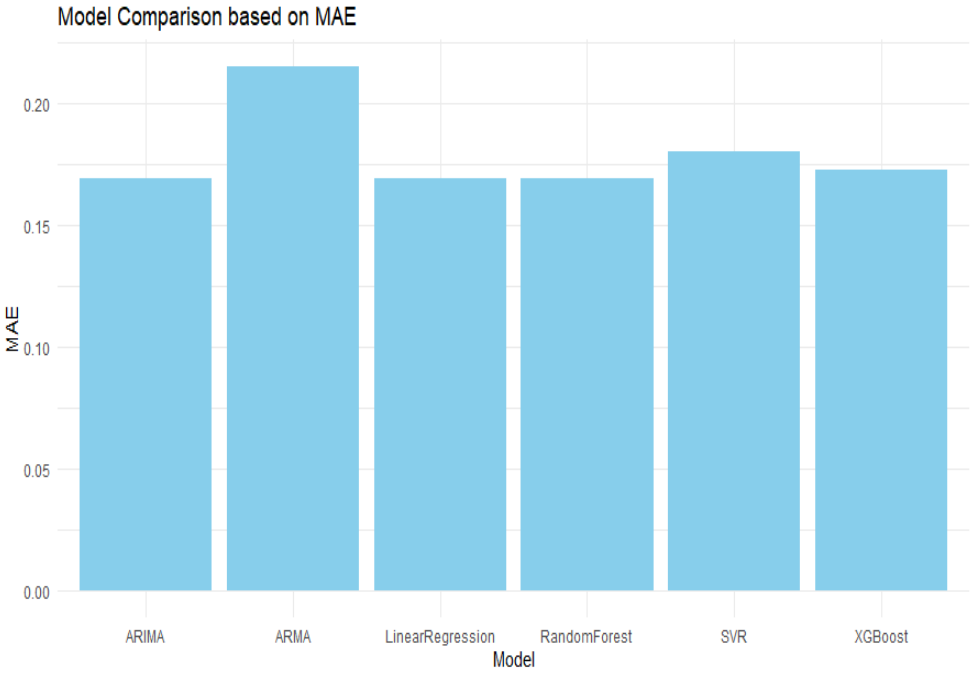

Figure 21. Model comparison based on MAE transformed dataset (Differencing).

Figure 21 illustrates the model comparison based on MAE for the transformed dataset. ARIMA emerges as the best performer based on the lowest MAE score.

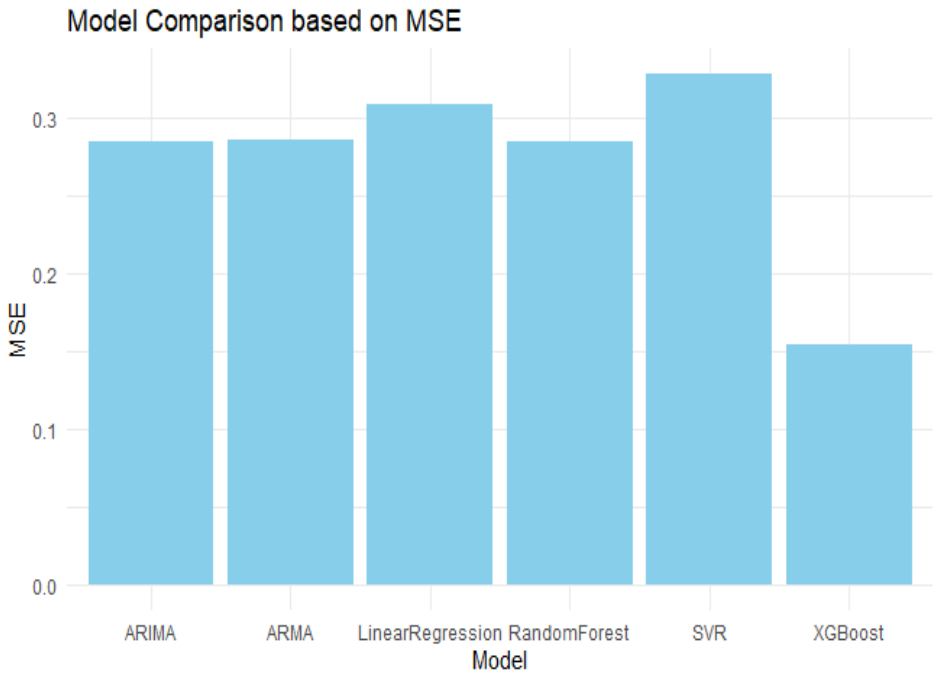

Figure 22. Model comparison based on MSE for actual dataset.

Mean Square Error (MSE) is another common metric used to assess model performance and predictive quality. Based on MSE, XGBoost outperforms all other models.

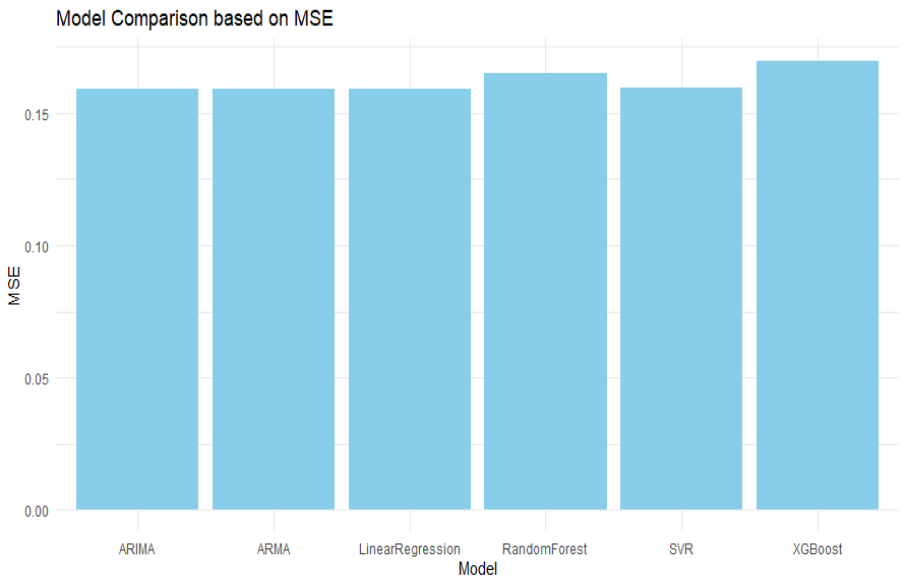

Figure 23. Model comparison based on MSE transformed dataset (Differencing).

However, in the transformed dataset, the differences in the MSE metric values are negligible, making it difficult to distinguish between the model performance.

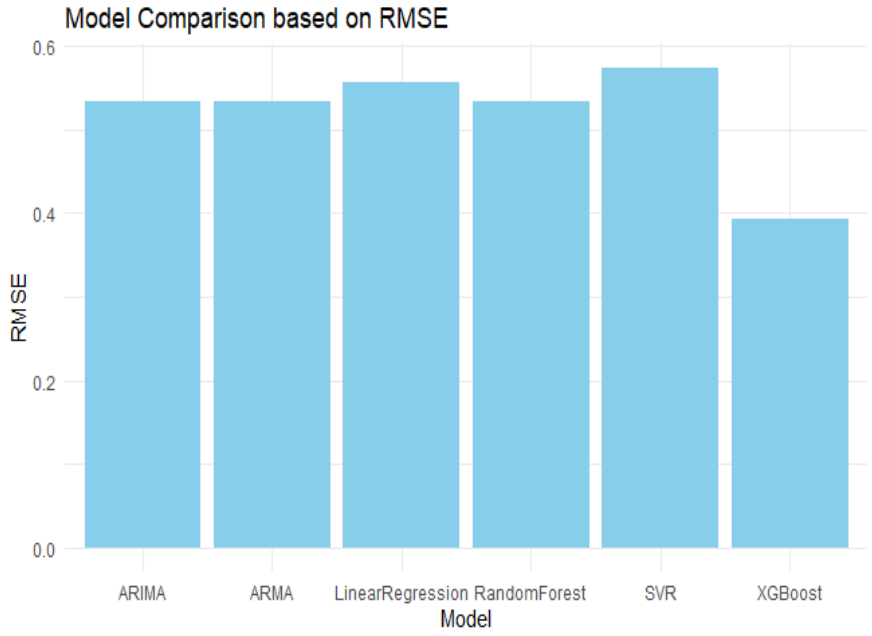

Figure 24. Model comparison based on RMSE for actual dataset.

MSE and RMSE are based on similar equations, with RMSE being the square root of MSE. Therefore, it follows that XGBoost has the lowest value for both metrics.

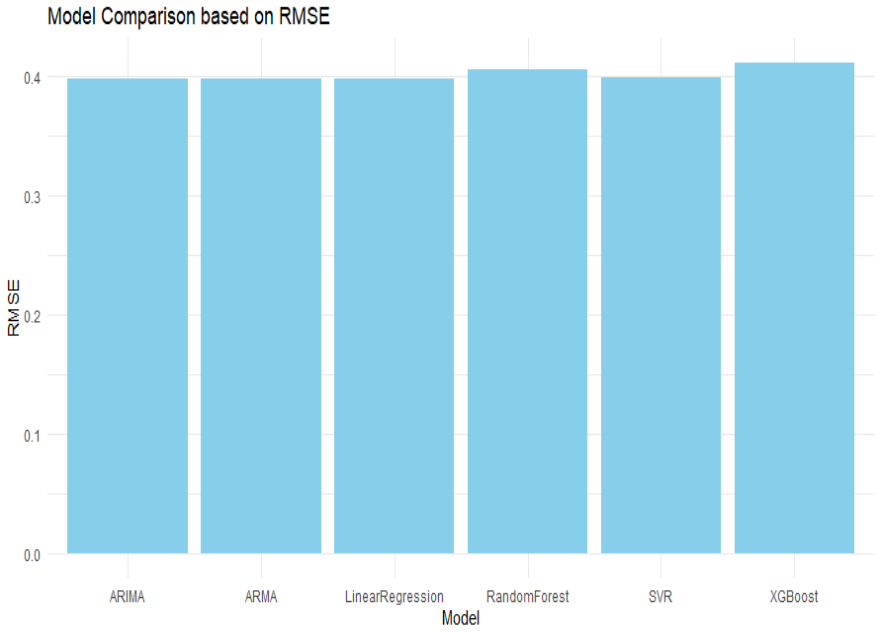

Figure 25. Model comparison based on RMSE for transformed dataset (Differencing).

One noticeable observation in the transformed dataset is that most of the statistical models have a similar score, whereas ML models differ in an insignificant manner.

5 SUMMARY AND DISCUSSIONS

Artificial neural network (ANN), Random Forest (RF), Support vector machine (SVM), XGBoost, ARIMA, and Linear regression (LR) are primarily used for forecasting and prediction in energy production (Solano 2022). These are the popular ML models utilized for model fitting and forecasting. This study incorporates both Machine learning (ML) and statistical models commonly used in the energy sector.

To evaluate the model performance of the AI, ML, and statistical models many authors consider standard metrics MAE, MSE, RMSE and R-Squared value (Yousif 2019). This study mainly focuses on the standard metrics, which are widely followed by the scientific community for precise prediction. The choice of determining features is primarily based on the correlation, as found in most papers (Han 2022). In this study, the main feature is energy produced in an hour recorded at an hourly pace, which is ideal for time series data modeling.

Hybrid models have gained popularity in time-series modeling and predictions. In hybrid modeling, data is first fitted with a simple linear model, and then the residuals of the linear model are

used to fit the nonlinear model (Xu 2019). This approach produces the best fitted model that captures both the linear trend and the nonlinear pattern of the data.

This study combines statistical and ML models, with statistical models used to identify the linearity of the data and ML models used to capture the nonlinearity. Supervised learning techniques were employed, with an 80% training rate and 20% test rate. A comparative study of supervised and unsupervised machine learning found that supervised machine learning generally outperforms unsupervised machine learning (Randle 2013).

In summary, this study focuses on finding the best model for and interpretability. Based on performance metrics, XGBoost emerges as the best fitted model among all ML and statistical models. Table 3 and Table 4 provide detailed comparison of metrics for all models. While other models also perform relatively well in prediction, XGBoost consistently stands out in cross/checking with actual and transformed data. Machine learning and Deep learning techniques are particularly suited for the large datasets, while computational intelligence models are more suitable for the smaller datasets. The use of hybrid models is highly recommended for modeling and optimizing in wind and solar energy applications (Shamshirband 2019).

This study was carried out from March 31st to August 28th, with a total of 3020 observations. The limited number of data points may restrict generalization for further inferences. However, with more datapoints collected over time, alternative models for prediction may be considered in the future.

6 REFERENCES

ChatGPT. 2024. OpenAI. GPT-3.5. https://chatgpt.com/. Accessed for language check 30.05.2024.

Ahmad, T. Z. 2021. Artificial intelligence in sustainable energy industry: Status Quo, challenges and opportunities. Journal of Cleaner Production, 289, 125834. doi: 10.1016/j.jclepro.2021.125834. Accessed 10.01.2024.

Alsharif, M. H. 2019. Time Series ARIMA Model for Prediction of Daily and Monthly Average Global Solar Radiation: The Case Study of Seoul, South Korea. Symmetry, 11(2), 240. doi.org/10.3390/sym11020240. Accessed 10.01.2024.

Atique, S. S. 2020. Time series forecasting of total daily solar energy generation: A comparative analysis between ARIMA and machine learning techniques. IEEE Green Technologies Conference, 175-180. doi: 10.1109/GreenTech46478.2020.9289796. Accessed 11.01.2024.

Bogdanov, D. G. 2021. Full energy sector transition towards 100% renewable energy supply: Integrating power, heat, transport and industry sectors including desalination. apenergy, 283, 116273. doi: 10.1016/j.apenergy.2020.116273. Accessed 17.01.2024.

Cyril Voyant, G. N.-L. 2017. Machine learning methods for solar radiation forecasting: A review. renene. doi: 10.1016/j.renene.2016.12.095. Accessed 17.01.2024.

David, M. R. 2016. Probabilistic forecasting of the solar irradiance with recursive ARMA and GARCH models. Solar Energy, 55-72. doi: 10.1016/j.solener.2016.03.064. Accessed 17.01.2024.

Fan, G. F. 2022. Applications of random forest in multivariable response surface for short-term load forecasting. International Journal of Electrical Power & Energy Systems, 139, 108073. doi: 10.1016/j.ijepes.2022.108073. Accessed 28.01.2024.

GeeksforGeeks. 2024. Random Forest Regression in Python. https://www.geeksforgeeks.org/random-forest-regression-in-python/. Accessed 18.05.2024.

Han, J. P. 2022. Data mining: concepts and techniques. Morgan kaufmann. Kaufmann publication. Accessed 29.01.2024.

Hastie, T. T. 2009. Random forests. The elements of statistical learning. Data mining, inference, and prediction, 587-604. DOI: 10.1111/j.1751-5823.2009.00095_18. x. Accessed 23.01.2024.

Huang, R. H. 2012. Solar generation prediction using the ARMA model in a laboratory-level micro-grid. IEEE third international conference on smart grid communications, 528-533. doi: 10.1109/SmartGridComm.2012.6486039. Accessed 18.01.2024.

Li, X. M. 2022. Probabilistic solar irradiance forecasting based on XGBoost. Energy Reports, 1087-1095. doi: 10.1016/j.egyr.2022.02.251. Accessed 05.01.2024.

Mirxat Alim, G.-H. Y.-S.-S. 2020. Comparison of ARIMA model and XGBoost model for prediction of human brucellosis in mainland China: a time-series study. BMJ open, 10(12). doi.org/10.1136/bmjopen-2020-039676. Accessed 10.01.2024.

Nti, K. O. 2019. Random forest-based feature selection of macroeconomic variables for stock market prediction. American Journal of Applied Sciences, 16(7), 200-212. doi.org/10.3844/ajassp.2019.200.212. Accessed 14.02.2024.

Okundamiya, M. S. 2013. A Linear Regression Model for Global Solar Radiation on Horizontal. International Journal of Renewable Energy Development, 2(3), 121. doi.org/10.14710/ijred.2.3.121-126. Accessed 16.03.2024.

Ozturk, S. &. 2018. FORECASTING ENERGY CONSUMPTION OF TURKEY BY ARIMA. Journal of Asian Scientific Research, 52-60. doi.org/10.18488/journal.2.2018.82.52.60. Accessed 13.02.2024.

Pin Li, a.-S. Z. 2018. A New Hybrid Method for China's Energy Supply Security Forecasting Based on ARIMA and XGBoost. Energies, 11(7). doi.org/10.3390/en11071687. Accessed 12.02.2024.

Randle, O. A. 2013. A comparison of the performance of supervised and unsupervised machine learning techniques in evolving Awale/Mancala/Ayo game player. arXiv preprint, 1309.1543. doi.org/10.48550/arXiv.1309.1543. Accessed 20.02.2024.

Rigby, A. B. 2024. Generation and validation of comprehensive synthetic weather histories using auto-regressive moving-average models. Renewable Energy, 224, 120157. doi: 10.1016/j.renene.2024.120157. Accessed 21.03.2024.

Rojas, I. O. 2008. Soft-computing techniques and ARMA model for time series prediction. Neurocomputing, 71(4-6), 519-537. doi: 10.1016/j.neucom.2007.07.018. Accessed 15.05.2024.

S. Shamshirband, T. R.-W. 2019. A Survey of Deep Learning Techniques: Application in Wind and Solar Energy Resources. IEEE Access, vol. 7, pp. 164650-164666. doi: 10.1109/ACCESS.2019.2951750. Accessed 15.05.2024.

Setiawan, I. 2020. Time series air quality forecasting with R Language and R Studio. Journal of Physics, vol. 1450, no. 1, p. 012064. doi 10.1088/1742-6596/1450/1/012064. Accessed 01.03.2024.

Shijun, C. Q. 2020. Medium-and long-term runoff forecasting based on a random forest regression model. Water Supply, 3658-3664.doi: 10.2166/ws.2020.214. Accessed 19.04.2024.

Shin, S. Y. 2022. Energy Consumption Forecasting in Korea Using Machine Learning Algorithms. Energies, 15(13), 4880. doi.org/10.3390/en15134880. Accessed 13.03.2024.

Solano, E. S. 2022. Solar radiation forecasting using machine learning and ensemble feature selection. Energies, 15(19), 7049. doi.org/10.3390/en15197049. Accessed 10.05.2024.

Vaia I. Kontopoulou, A. D. 2023. A Review of ARIMA vs. Machine Learning Approaches for Time Series Forecasting in Data Driven Networks. Future Internet, 15(8), 255. doi.org/10.3390/fi15080255. Accessed 18.05.2024.

The R Project for Statistical Computing. 2024. e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. https://cran.rstudio.com/web/packages/e1071/index.html. Accessed 28.05.2024.

Wikipedia. 2024. XGBoost. https://en.wikipedia.org/wiki/XGBoost. Accessed 18.05.2024.

Xu, W. P. 2019. A hybrid modelling method for time series forecasting based on a linear regression model and deep learning. Springer Nature, 49, 3002-3015. doi: 10.1007/s10489-019-01426-3. Accessed 28.05.2024.

Younes Ledmaoui, A. E. 2023. Forecasting Solar Energy Production: A Comparative Study of Machine Learning Algorithms. Elsevier. doi.org/10.1016/j.egyr.2023.07.042. Accessed 18.05.2024.

Yousif, J. H. 2019. A comparison study based on artificial neural network for assessing PV/T solar energy production. Elsevier, 100407. doi.org/10.1016/j.csite.2019.100407. Accessed 18.05.2024.

Zendehboudi, A. B. 2018. Application of support vector machine models for forecasting solar and wind energy resources: A review. Journal of Cleaner Production, 272-285. doi: 10.1016/j.jclepro.2018.07.164. Accessed 18.05.2024.