AI in Sustainable Energy and Environment

Journal homepage: aisesjournal.com

Online

A Comprehensive Machine Learning Approach for Predictive Analysis of Energy Consumption in OPEC Countries

Halil GÖR

1.Hakkari University, Department of Electrical and Electronics Engineering, Hakkari, TÜRKİYE

Article Info

Received 6 January 2025

Received in Revised form 4 March 2025

Accepted 14 April 2025

Published online 30 April 2025

DOI:

Keywords

Energy policy,

OPEC,

Machine Learning,

Socio-economic factors

Abstract

The accurate energy consumption prediction for OPEC (Organization of the Petroleum Exporting Countries) member states is vital for strategic planning and policy-making in the global energy market. This study leverages advanced machine learning techniques to forecast energy consumption, utilizing historical data from the U.S. Energy Information Administration (EIA). We applied a variety of machine learning models, including Simple Linear Regression, Gaussian Processes, Multilayer Perceptron (MLP), SMOreg, IBK, Kstar, LWL, Random Subspace, Random Committee, and Random Forest, to the task of predicting energy consumption. The performance of these models was evaluated based on metrics such as R-squared (R^2), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Relative Absolute Error (RAE), and Root Relative Squared Error (RRSE). Our results demonstrated that the Random Committee model achieved the highest accuracy in both training (R^2 = 0.9999, MAE = 0.7411, RMSE = 1.0509, RAE = 1.2199%, RRSE = 1.2399%) and testing phases (R^2 = 0.9525, MAE = 11.4795, RMSE = 30.6585, RAE = 17.9586%, RRSE = 31.6700%), highlighting its robustness and predictive power. In contrast, the LWL model showed the poorest performance, with significant errors in both phases. The study also highlights the strengths and limitations of each model, with a focus on the applicability of these findings for policymakers and energy analysts. The insights gained from this research underscore the potential of machine learning to enhance energy consumption forecasting, providing a foundation for future studies to build upon. Directions for future research include incorporating additional socio-economic and environmental variables, real-time data, and more advanced machine learning techniques to improve prediction accuracy and reliability further.

1. Introduction:

The Organization of the Petroleum Exporting Countries (OPEC) has wielded considerable influence over global energy markets since its establishment in 1960. Comprising 13 member countries, including major oil-producing nations like Saudi Arabia, Iran, and Venezuela, OPEC's primary objective is coordinating petroleum policies to ensure market stability and fair prices for producers and consumers [1]. Predicting energy consumption for OPEC countries is paramount for several reasons [2]. Firstly, these nations are home to vast crude oil and natural gas reserves, which serve as critical economic assets. Accurate forecasts of energy consumption are essential for managing these resources sustainably and ensuring long-term economic viability [3]. Furthermore, OPEC's decisions regarding oil production quotas and pricing directly influence global oil markets. By forecasting energy consumption, OPEC can anticipate future demand trends, enabling informed decisions to stabilize markets and prevent supply-demand imbalances. Moreover, OPEC countries' energy consumption predictions inform policymaking [2]. These forecasts guide energy efficiency, infrastructure development, and environmental sustainability initiatives, facilitating effective policy formulation to meet future energy

Corresponding author: halilgor@hakkari.edu.tr (H. GÖR)

needs [4]. Additionally, accurate predictions of energy consumption support economic planning by governments and businesses in OPEC nations. By anticipating energy demands, stakeholders can plan investments, allocate resources, and devise strategies to foster economic growth and ensure energy security [5]. Predicting energy consumption for OPEC countries is crucial for maintaining energy market stability, informing policy decisions, and promoting sustainable economic development [6]. This paper explores the significance of energy consumption prediction and proposes a machine learning-based approach to address this critical need.

Machine learning, a field within artificial intelligence, has become increasingly indispensable in analyzing complex datasets and making accurate predictions across diverse domains [7]. In energy consumption analysis, machine learning offers unparalleled capabilities to extract valuable insights from vast data repositories and rapidly forecast future consumption patterns [8, 9]. Traditional methods of energy consumption analysis often struggle to contend with the intricacies inherent in energy systems [10]. Economic indicators, population demographics, climate variations, and technological advancements contribute to the complexity of energy consumption patterns. Machine learning algorithms excel in handling such multifaceted data, allowing for the identification of subtle patterns and nonlinear relationships that traditional techniques may overlook [11]. A myriad of interconnected factors influence energy consumption. Machine learning algorithms are adept at analyzing these complex interactions and identifying hidden patterns, providing a more comprehensive understanding of energy consumption dynamics [12]. The advent of sensor technologies and smart meters has led to an explosion of data in the energy sector [13]. Machine learning algorithms are inherently scalable, enabling real-time analysis of large-scale datasets [14]. Machine learning models can make accurate predictions about future energy consumption trends by leveraging advanced algorithms such as neural networks, decision trees, and support vector machines. These models capture the nonlinear relationships between predictors and energy consumption, leading to more reliable forecasts [15]. Energy systems are subject to constant change due to various factors. Machine learning models can adapt to these changes by continuously updating their parameters based on new data, ensuring that energy consumption predictions remain relevant and up-to-date [16, 17]. Machine learning empowers policymakers, energy analysts, and stakeholders by providing timely and accurate insights into energy consumption patterns. These insights inform critical decisions regarding resource allocation, infrastructure planning, and policy formulation. Machine learning is pivotal in energy consumption analysis, offering advanced analytical capabilities, scalability, prediction accuracy, adaptability, and decision support [8, 18]. Leveraging machine learning techniques has the potential to revolutionize our understanding and management of energy systems, paving the way for a more sustainable and efficient energy future.

Previous research has explored the application of machine learning methods for predicting energy consumption across various contexts, including residential, commercial, industrial, and national levels. These studies have contributed valuable insights into the effectiveness of different machine learning techniques and the factors influencing energy consumption patterns. Several studies have focused on predicting residential energy consumption to inform energy efficiency initiatives and demand-side management strategies. For example, research by Burnett and Kiesling [19] applied machine learning algorithms such as decision trees and neural networks to household-level data to forecast electricity usage accurately. Their findings highlighted the importance of weather conditions, household demographics, and appliance usage patterns in predicting residential energy consumption. Other studies have explored energy consumption prediction in commercial and industrial settings to optimize energy usage and reduce operational costs. For instance, Liu, et al. [20] developed predictive models using support vector machines and time-series analysis techniques to forecast energy demand in manufacturing plants. Their research demonstrated the potential of machine learning for identifying energy-saving opportunities and improving production efficiency in industrial facilities. At the national level, researchers have utilized machine learning approaches to analyze and predict energy consumption trends for entire countries or regions. For example, Khan, et al. [2] conducted a comprehensive study on energy consumption forecasting for OPEC countries using a combination of regression analysis and neural networks. Their analysis revealed the significance of socio-economic indicators, energy prices, and policy interventions in shaping energy demand dynamics across OPEC nations. While machine

learning holds promise for energy consumption prediction, several challenges and limitations persist. These include data quality issues, model interpretability concerns, and the need for domain expertise in feature selection and model tuning. Addressing these challenges is crucial for ensuring the reliability and practical utility of machine learning-based energy forecasting models. Previous studies have demonstrated the effectiveness of machine learning methods in predicting energy consumption across various scales and contexts. By leveraging advanced algorithms and integrating diverse datasets, researchers have gained valuable insights into the drivers of energy demand and the potential for optimizing energy usage in residential, commercial, industrial, and national settings.

Previous studies on energy consumption prediction have employed a variety of methodologies, algorithms, and data sources to achieve accurate forecasts and gain insights into energy usage patterns. Many studies utilize time-series analysis techniques to model temporal patterns in energy consumption data [21, 22]. This involves analyzing historical consumption data to identify trends, seasonality, and other recurring patterns that can inform future predictions [23]. Regression models are commonly employed to capture the relationship between energy consumption and predictor variables such as temperature, population, economic indicators, and policy variables [24]. Linear regression, polynomial regression, and multiple regression are among the regression techniques used in energy consumption prediction studies [24]. Machine learning algorithms offer a powerful alternative to traditional statistical methods for energy consumption prediction [25]. Supervised learning algorithms such as decision trees, random forests, support vector machines, and neural networks have been widely used to build predictive models that capture complex patterns and nonlinear relationships in energy data. Decision tree algorithms are popular for their simplicity and interpretability. They partition the data into hierarchical decision nodes based on feature attributes, allowing an intuitive understanding of the decision-making process [26]. Neural network models, inspired by the human brain's structure, can learn intricate patterns from large datasets [27]. Multilayer perceptron (MLP), convolutional neural networks (CNN), and recurrent neural networks (RNN) are commonly used architectures for energy consumption prediction tasks. SVM algorithms are effective for both regression and classification tasks. They work by finding the optimal hyperplane that separates different classes or predicts continuous values with maximum margin from the data points. Longitudinal datasets containing historical energy consumption records are fundamental for training and evaluating predictive models [28]. These datasets may include information on electricity, natural gas, oil, and other energy sources, disaggregated by sector (residential, commercial, industrial) and geographical region. Weather variables such as temperature, humidity, solar radiation, and wind speed significantly impact energy demand. Meteorological data sources provide essential inputs for modeling weather-dependent variations in energy consumption patterns. Demographic data, economic indicators, and socio-economic variables such as population size, GDP, urbanization rate, and household income levels are often included as predictors in energy consumption prediction models. These indicators help capture the underlying drivers of energy demand at both individual and aggregate levels. Methodologies such as time-series analysis, regression analysis, and machine learning, coupled with algorithms like decision trees, neural networks, and SVMs, have been employed in similar studies to predict energy consumption [29]. These studies leverage diverse data sources, including historical energy consumption data, meteorological data, and socio-economic indicators, to build predictive models that can provide valuable insights for energy planning, policy-making, and resource allocation.

The primary aim of this study is to develop and evaluate various machine learning models for accurately predicting the energy consumption of OPEC member countries using historical data obtained from the U.S. Energy Information Administration (EIA). While numerous studies have examined energy forecasting, few have focused specifically on the collective group of OPEC nations, whose energy consumption trends are crucial due to their central role in the global oil and energy market. The novelty of this research lies in its comparative analysis of ten distinct machine learning algorithms, offering a robust and data-driven perspective on energy demand prediction. Furthermore, this work contributes by identifying the most accurate and efficient models tailored to OPEC's energy patterns, which can support policy-making, economic planning, and energy sustainability strategies. The outcomes of this research are expected to guide future efforts in data-driven energy modeling and encourage the

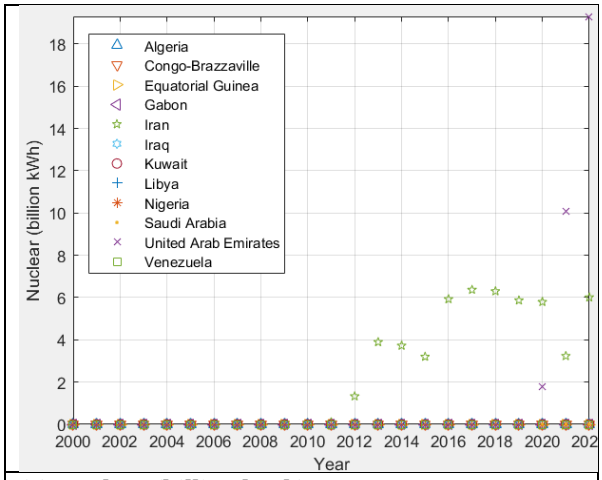
integration of machine learning approaches in international energy policy analysis.

2. Data Collection and Preprocessing:

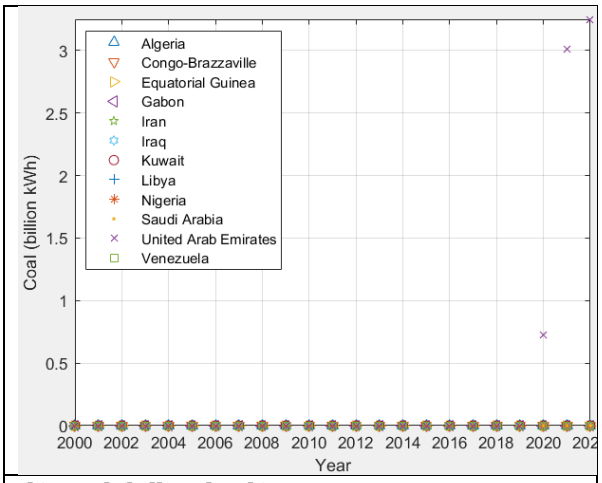
For this analysis, we will leverage multiple data sources to comprehensively understand energy consumption patterns in OPEC countries and develop predictive models for forecasting future consumption trends. Historical energy consumption data for OPEC countries will serve as the primary dataset for our analysis. This data, obtained from the U.S. Energy Information Administration (EIA) database, includes records of energy consumption over time, disaggregated by sector (residential, commercial, industrial) and energy source (electricity, petroleum, natural gas). By analyzing historical consumption trends, we aim to identify energy demand patterns and drivers in OPEC nations. Socio-economic indicators such as population demographics, GDP per capita, urbanization rate, and household income levels will be sourced from international databases such as the World Bank and the International Monetary Fund (IMF). These indicators provide insights into the socio-economic context and underlying factors influencing energy consumption patterns in OPEC countries.

Meteorological data will be sourced from meteorological agencies and global climate datasets, including temperature, humidity, solar radiation, and wind speed. Environmental factors play a significant role in shaping energy demand, particularly in sectors such as residential heating and cooling, transportation, and agriculture. By integrating meteorological data into our analysis, we can capture the impact of weather variations on energy consumption patterns. Information on energy prices, including the cost of electricity, petroleum products, and natural gas, will be obtained from national energy agencies and international organizations. Energy prices influence consumer behavior and investment decisions, affecting energy demand across various sectors. Incorporating price data into our analysis enables us to assess the relationship between energy prices and consumption levels in OPEC countries. By combining data from these diverse sources, we aim to develop comprehensive predictive models that account for the multifaceted nature of energy consumption dynamics in OPEC nations. Integrating historical consumption data, socio-economic indicators, environmental factors, and energy prices will facilitate a holistic understanding of energy demand drivers and enhance the accuracy of our predictions.

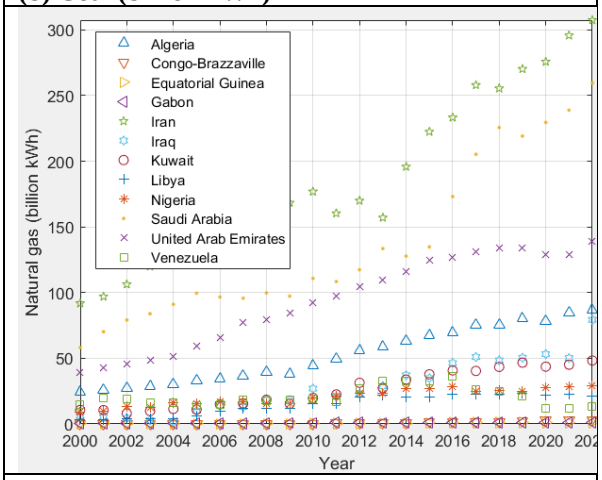
Before analysis, the collected data undergoes thorough preprocessing to ensure its quality and suitability for predictive modeling. Handling missing values is the priority, where techniques like mean, median, or mode substitution are employed for numerical features, and predictive models may be utilized to estimate missing values based on other relevant variables. Subsequently, normalization techniques such as Min-Max scaling or Z-score normalization are applied to rescale numerical features, ensuring that all features contribute equally to the analysis. Feature engineering plays a crucial role in enhancing the dataset's predictive power, involving creating new features or transforming existing ones to capture underlying patterns better. Time-related, lag, interaction, and polynomial features are generated to account for seasonality, temporal trends, autocorrelation, and nonlinear relationships in the data. Outliers are detected, removed, or transformed using robust methods to mitigate their impact on predictive models. Furthermore, categorical variables are encoded into numerical representations using one-hot or label encoding techniques to facilitate their incorporation into machine learning models. These preprocessing steps collectively ensure that the data is clean, normalized, and appropriately structured for analysis, laying the foundation for accurate forecasting of energy consumption patterns in OPEC countries.



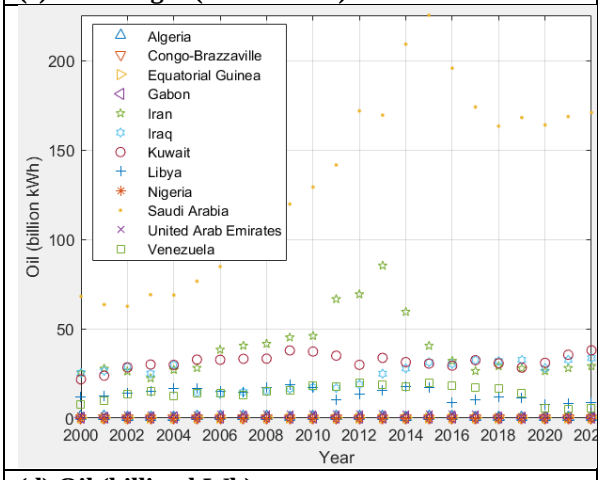
(a) Nuclear (billion kWh)



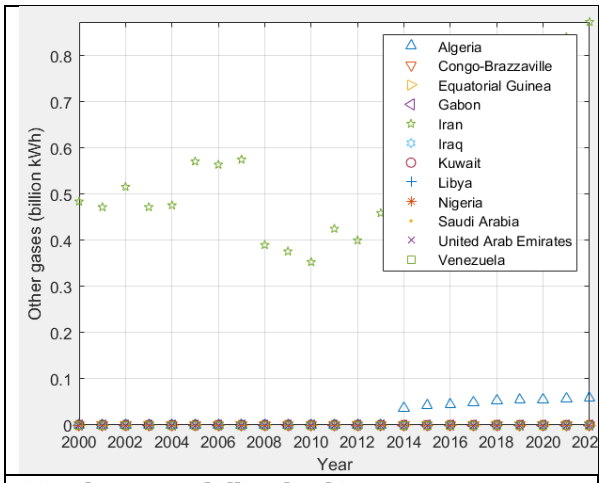
(b) Coal (billion kWh)



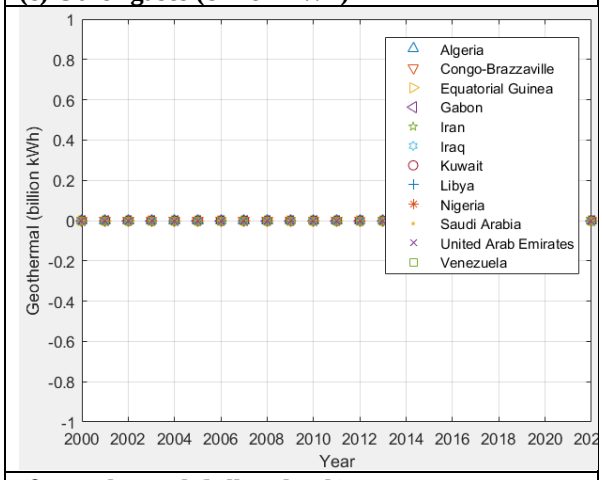
(c) Natural gas (billion kWh)



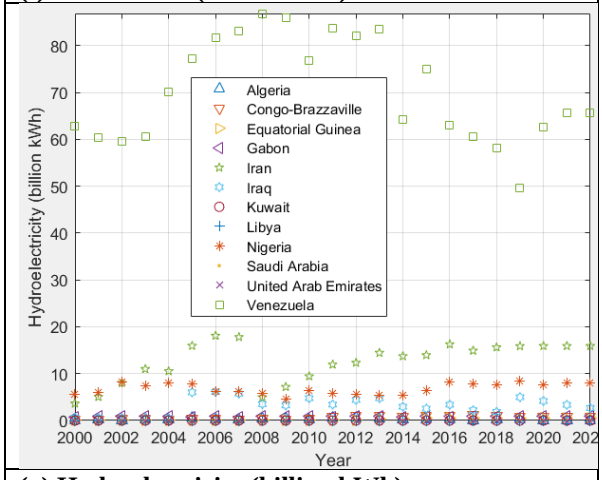
(d) Oil (billion kWh)



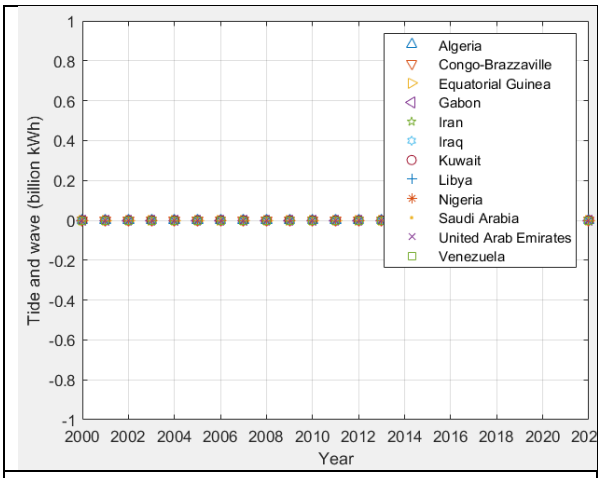
(e) Other gases (billion kWh)



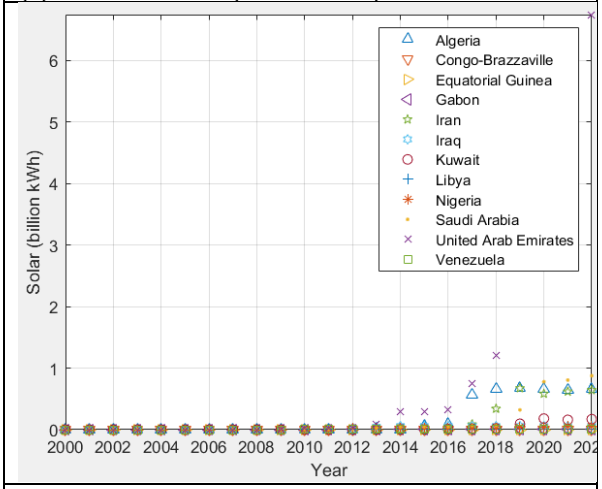
(f) Geothermal (billion kWh)



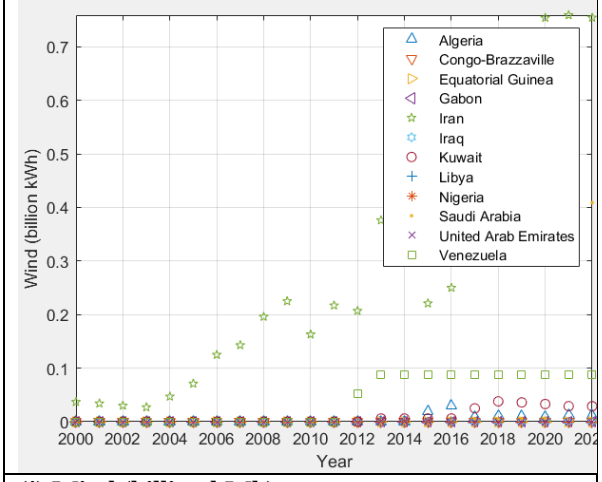
(g) Hydroelectricity (billion kWh)



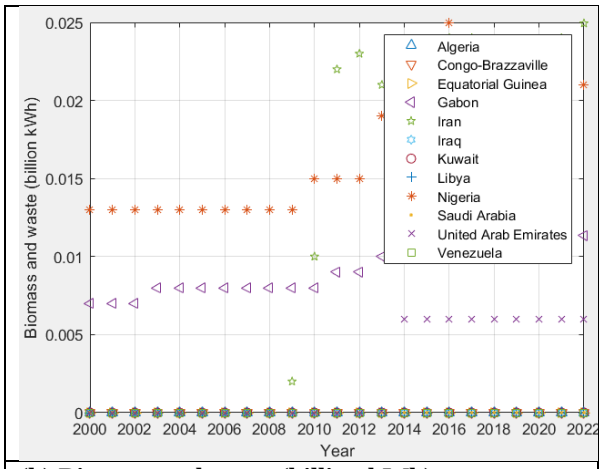
(h) Tide and wave (billion kWh)



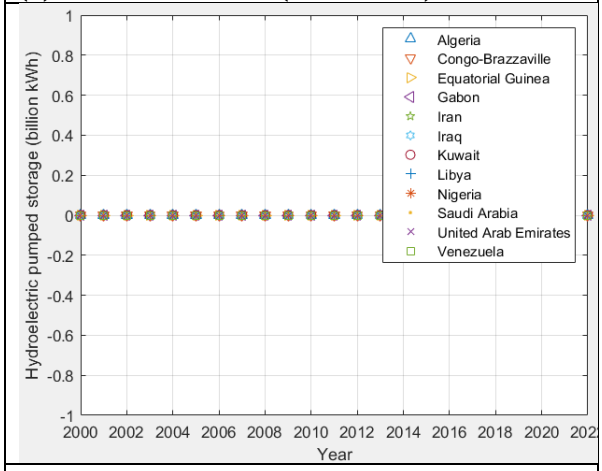
(i) Solar (billion kWh)



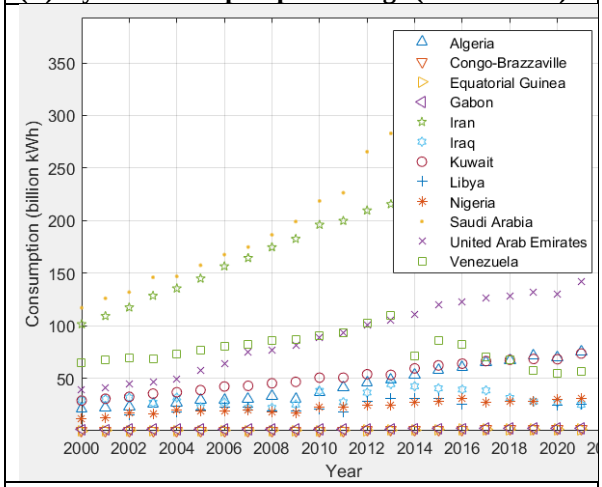
(j) Wind (billion kWh)



(k) Biomass and waste (billion kWh)



(L) Hydroelectric pumped storage (billion kWh)



(m) Consumption (billion kWh)

Figure 1: Parameters of dataset number

3. Machine Learning:

Machine learning, a transformative field within artificial intelligence, focuses on developing algorithms and models that enable computer systems to learn from data and improve their performance without explicit programming [30]. The essence of machine learning lies in the ability of systems to identify patterns, make predictions,

and derive insights from vast and complex datasets. This multidisciplinary field integrates principles from computer science, statistics, and mathematics to create algorithms capable of autonomously learning and adapting to new information. Supervised learning involves training models on labeled datasets, while unsupervised learning discovers patterns in unlabeled data. Reinforcement learning emphasizes decision-making in dynamic environments through trial-and-error learning. Machine learning applications span diverse domains, including natural language processing, image and speech recognition, medical diagnosis, finance, and predictive analytics. As technology advances, the pervasive influence of machine learning continues to reshape industries, offering solutions to intricate problems and unlocking unprecedented possibilities in data-driven decision-making.

3.1 Simple Linear Regression:

It is one of the oldest regression techniques, dating back to the early 19th century. It assumes a linear relationship between the dependent and independent variables and seeks to find the best-fit line [31]. A linear regression model that establishes a relationship between a dependent variable and a single independent variable. It assumes a linear relationship and aims to find the best-fitting line through the data points. It is suitable when there is a clear linear correlation between input and output variables. Modeling the link between a single independent and dependent variable is done statistically. The objective is to create a linear connection between the two variables to make predictions or understand the link between them. The approach is predicated on the idea that a straight line may adequately describe the connection between the variables. A basic linear regression model's equation is as follows:

$$Y = \beta 0 + \beta 1 \cdot X + \varepsilon \quad (1)$$

where:

- Y is the dependent variable (the variable you are trying to predict).
- X is the independent variable (the variable used for prediction).
- $\beta 0$ is the y-intercept (the value of Y when X is 0).
- $\beta 1$ is the slope of the line (representing the change in Y for a unit change in X).
- ε is the error term (represents the unobserved factors affecting Y that X does not explain).

3.2 Gaussian Processes:

Developed in the middle of the 20th century, Gaussian Processes (GPs) is a non-parametric approach that may depict complex interactions without requiring a specific functional form [32]. They come in handy for regression and classification. A non-parametric, probabilistic model that defines a distribution over functions. For regression tasks, uncertainty estimates are provided in addition to predictions. Helpful in effectively communicating prediction uncertainty and managing nonlinear interactions. Considering everything, GPs are a powerful and flexible framework used in machine learning for modeling and predicting situations where the relationships between variables are hazy or unclear. GPs are non-parametric models that may find complex, nonlinear patterns in data. They are frequently used in probabilistic classification problems and regression. It is a powerful and flexible machine-learning method primarily used for regression and probabilistic classification tasks. Gaussian Processes offer a powerful, probabilistic approach to regression that is particularly useful when it is important to quantify uncertainty in predictions. However, their computational complexity can be a limiting factor for very large datasets.

At the core of Gaussian Processes is defining a distribution over functions. Unlike traditional machine learning models that provide point estimates, GPs offer a probabilistic approach, predicting a distribution of possible functions that fit the data. GPs assume that any finite set of function values follows a multivariate Gaussian distribution. This distribution is defined by a mean function and a covariance function (kernel). The mean function ($m(x)$) represents the average of the function values at each point (x). It is often assumed to be zero for simplicity ($m(x) = 0$). The covariance function $k(x, x')$ defines the covariance between pairs of points x and x'. It encodes assumptions about the function's properties, such as smoothness and periodicity. Common kernels include the Radial Basis Function (RBF), the Gaussian kernel, and the Matérn kernel. Before observing any data, GPs define a prior distribution over functions based on the mean and covariance functions. These prior captures our initial beliefs about the function's behavior. When data points (X, y) are observed, where X is the set of input features, and y is the set of target values,

GPs update the prior distribution to a posterior distribution using Bayes' theorem. The posterior distribution combines the prior and the observed data, resulting in a new distribution that is more concentrated around the observed data points. The updated mean and covariance functions are computed as follows:

The new mean function $m_*(x)$ is influenced by the observed data and indicates the expected function value at each point x. The new covariance function $k_*(x, x')$ reflects the updated uncertainty about the function values after observing the data. Given a new input x_*, the GP model provides a Gaussian distribution for the corresponding output y_*. The mean of this distribution gives the predicted value, and the variance gives the uncertainty of the prediction.

For a set of training points (X, y), where $X = [x_1, x_2, \ldots, x_n]$ and $y = [y_1, y_2, \ldots, y_n]$, the joint distribution of the training outputs y and the test outputs y_* for test inputs x_* is given by:

$$\begin{pmatrix} y \\ y_* \end{pmatrix} \sim N\left(0, \begin{pmatrix} K(X,X) + \sigma_n^2 I & K(X, x\_*) \\ K(x_*, X) & K(x_*, x_*) \end{pmatrix}\right) \quad (2)$$

Here, $K(X, X)$ is the covariance matrix for the training points, $K(X, x_*)$ is the covariance between training points and test points, and σ_n^2 is the noise variance.

The conditional distribution of y_* given X and y is:

$$y_* \mid X, y, x_* \sim N\left(m_*(x_*), k_*(x_*, x_*)\right) \quad (3)$$

where:

$$m_*(x_*) = K(x_*, X)[K(X,X) + \sigma_n^2 I]^{-1} y \quad (4)$$

$$k_*(x_*, x_*) = K(x_*, x_*) - K(x_*, X)[K(X,X) + \sigma_n^2 I]^{-1} K(X, x_*) \quad (5)$$

GPs are flexible and do not assume a fixed form for the modeled function. GPs provide not just predictions but also a measure of uncertainty for those predictions. The choice of kernel allows GPs to model various types of data patterns. GPs require the inversion of an $n \times n$ matrix, which is computationally expensive for large datasets (where n is the number of training points). The performance of GPs depends on the choice of kernel and its hyperparameters, which may require optimization.

3.3 Multilayer Perceptron (MLP):

MLP traces its roots back to the 1940s as a fundamental neural network architecture. It gained popularity in the 1980s with the development of backpropagation algorithms [33]. MLPs are versatile and widely used for various machine-learning tasks—an artificial neural network with nonlinear activation functions and several layers of nodes or neurons. MLPs are versatile and can learn complex patterns. It is well-suited for capturing intricate data relationships, especially in nonlinear cases. Multilayer Perceptrons are powerful tools for various machine learning tasks, leveraging layers of neurons and nonlinear activation functions to learn from data. Their effectiveness depends on careful design, training, and evaluation to balance complexity and generalization.

A Multilayer Perceptron (MLP) is an artificial neural network commonly used for classification and regression tasks. It is composed of multiple layers of nodes (neurons) that are fully connected, where each layer transforms the input data using weights that are adjusted during training. The input layer consists of nodes representing the input data's features. The number of nodes in this layer equals the number of input features—one or more layers between the input and output layers where computation is performed. The nodes in these layers apply activation functions to introduce non-linearity and enable the network to learn complex patterns. The final layer produces the output of the network. The number of nodes in this layer depends on the task (e.g., one node for regression multiple nodes for classification). Each connection between nodes has an associated weight, and each node (except those in the input layer) has an associated bias. These parameters are adjusted during training to minimize network error. Functions are applied to each node's output in the hidden and output layers. Common activation functions include Sigmoid, Tanh, and ReLU (Rectified Linear Unit).

Input data is fed into the input layer and passed through the network layer by layer. A weighted sum of inputs is calculated for each node in a hidden or output layer, including a bias term.

$$z = \sum_i (w_i . x_i) + b \quad (6)$$

where z is the weighted sum, w_i are the weights, x_i are the inputs, and b is the bias. The weighted sum

is passed through an activation function to introduce non-linearity.

$$a = \sigma(z) \quad (7)$$

where σ is the activation function, and a is the node's output. The network's output is compared to the target values using a loss function, such as Mean Squared Error (MSE) for regression or Cross-Entropy Loss for classification. The loss quantifies the difference between the predicted and actual values. The error from the output layer is propagated backward through the network to update the weights and biases. The gradient of the loss function concerning each weight and bias is calculated using the chain rule. Weights and biases are updated using an optimization algorithm such as Stochastic Gradient Descent (SGD) or Adam. The update rule for a weight w is:

$$w_{new} = w_{old} - \eta . \frac{\partial L}{\partial w} \quad (8)$$

where η is the learning rate and $\frac{\partial L}{\partial w}$ is the gradient of the loss function concerning the weight. The process of forward propagation, loss calculation, and backpropagation is repeated for many epochs (iterations) until the network's performance converges to an acceptable level.

MLPs can model complex, nonlinear relationships due to their multiple layers and nonlinear activation functions. Properly trained MLPs can generalize to unseen data if sufficient training data and regularization techniques are used. Training MLPs, especially with many layers and neurons, can be computationally intensive and require significant computational resources. MLPs are prone to overfitting, especially with limited training data. Techniques such as dropout, regularization, and cross-validation are often used to mitigate this.

Specify the number of layers, the number of neurons in each layer, and the activation functions. Initialize weights and biases, typically using small random values. A large dataset is used to train the network, adjusting the weights and biases through backpropagation and an optimization algorithm. Assess the network's performance on a separate validation set to ensure it generalizes well to new data. Adjust hyperparameters to improve performance, such as learning rate, number of layers, and regularization terms.

3.4 Sequential Minimal Optimization for Regression (SMOreg):

Developed by John Platt in the late 1990s, SMOreg is a support vector machine (SVM) algorithm tailored for regression tasks [34]. It efficiently sequentially optimizes the SVM objective function—a regression version of the Sequential Minimal Optimization (SMO) algorithm. SMOreg is mainly designed to solve regression problems using support vector machines. Effective in regression tasks where support vector machines are suitable. SVMs are supervised learning models commonly used for classification and regression tasks. In the context of regression, SVMs are called Support Vector Regression (SVR). The purpose of the SMO method is to effectively resolve the quadratic programming (QP) issue that comes up when training support vector machines (SVMs). SMO is an efficient algorithm for solving the quadratic programming problem associated with SVR. It iteratively optimizes pairs of Lagrange multipliers, making the optimization process more tractable than solving the entire problem simultaneously.

The objective of SVR involves minimizing a quadratic function subject to linear constraints. The formulation of the Lagrange dual problem results in a QP problem that involves maximizing a quadratic objective function while adhering to equality and inequality constraints. Initialize the Lagrange multipliers. α_i for each training example. These multipliers represent the importance of each data point in the SVR model. In each iteration, SMO selects a pair of Lagrange multipliers (α_i, α_j) for optimization. The selection criteria can be based on heuristics or optimization strategies to efficiently identify pairs that violate the KKT (Karush-Kuhn-Tucker) conditions. Optimize the chosen pair $((\alpha_i, \alpha_j))$ by holding all other Lagrange multipliers fixed. The optimization involves updating the chosen multipliers subject to the equality constraints derived from the KKT conditions. Compute the bias terms (b) for the updated Lagrange multipliers. This involves considering the support vectors (data points with non-zero Lagrange multipliers) and their contributions to the bias. Check whether the chosen pair α_i, α_j satisfies the KKT conditions. The KKT conditions include equality and inequality constraints and complementary slackness conditions. Update the Lagrange multipliers for the remaining data points to maintain the equality constraints. This step ensures that the solution remains feasible. Check for

convergence by assessing whether the Lagrange multipliers and bias have changed significantly. If the changes are below a certain threshold, the algorithm has converged. Repeat steps 3 to 8 until convergence is achieved. In each iteration, SMO selects different pairs of Lagrange multipliers for optimization. The optimized Lagrange multipliers and bias determine the final SVR model.

3.5 Instance-Based Learning with k-Nearest Neighbors (lazy. IBK):

A sort of machine learning in which the model is not explicitly trained is termed instance-based learning, also known as lazy learning or instance-based approaches. Instead, predictions are made based on the similarity between new instances and instances from the training dataset. One popular instance-based learning algorithm is k-Nearest Neighbors (k-NN). K-Nearest Neighbors (k-NN) has been used since the 1950s—the lazy.IBK variant involves minimal training and defers most computations until predictions are needed [35]. It's effective for both regression and classification. A lazy learning algorithm that stores instances of the training data and classifies new instances based on their similarity to the nearest neighbors. It is suitable for cases where instances are locally clustered, and local information is essential for prediction.

In the training phase of k-NN, the algorithm stores the entire training dataset. There is no explicit training involved; the dataset is the model. Choose a distance metric to measure the similarity between instances. The Manhattan distance, the Euclidean distance, and other similarity metrics are examples of standard distance metrics. Represent each instance in the dataset as a point in a multidimensional space, where each feature corresponds to a dimension. This representation allows the algorithm to measure distances between instances. When a new instance is presented for prediction or classification, k-NN identifies the k-nearest neighbors of the recent example in the feature space based on the chosen distance metric. "k" is a hyperparameter that determines the number of neighbors to consider. In classification tasks, the technique uses the majority class among its k-nearest neighbors to predict the class label of the new instance. The technique forecasts the average of the goal values of the k-nearest neighbors in regression problems. If there is a tie in the case of classification or if you want to give more importance to closer neighbors, you can use weighted voting. Assign weights to each neighbor based on distance and consider the weighted votes for the final prediction. Tune the hyperparameter "k" based on the dataset's characteristics and the desired trade-off between bias and variance. A smaller k makes the model more sensitive to noise, while a larger k may smooth out the decision boundaries. Calculating the distances between each new instance and every instance in the training set is the prediction phase in k-NN. For large datasets, this can be computationally expensive. Efficient data structures like KD trees or ball trees can be used to speed up this process.

3.6 k*-Nearest Neighbors (lazy.Kstar):

Modifying k-NN and k*-NN improves efficiency by using an adaptive approach to select neighbors based on distances [36]. It maintains simplicity while addressing some limitations of traditional k-NN. A variant of the k-Nearest Neighbors (k-NN) algorithm that dynamically adapts the distance metric based on local instance density. Effective in scenarios where the density of instances varies across the feature space.

The k*-Nearest Neighbors (k*-NN) algorithm is also known as lazy.Kstar is an instance-based learning or lazy learning algorithm used for classification and regression tasks. Unlike eager learning algorithms, which build a model from the training data before making predictions, lazy learning algorithms delay the learning process until a prediction is requested. k*-NN stores all available cases and only computes the results for a new query when required, using the stored instances to make predictions. Instead of using a fixed number of nearest neighbors (k), k*-NN uses a probability distribution to weigh the influence of all training instances based on their similarity to the query instance. The algorithm employs a similarity measure that can handle various data types and adapt to the specific data context, making it more flexible than traditional k-NN.

In the prediction phase, when a new query instance needs to be classified, the algorithm computes the distance or similarity between the query instance and all stored instances. The distance metric can vary depending on the data type and context. k*-NN uses a probability distribution to determine the influence of each training instance based on its distance from the query instance, using a smooth weighting function that assigns higher weights to closer instances and lower weights to farther instances. For classification tasks, the weighted votes of the neighbors are aggregated to determine the class of the query instance. The weighted

average of the neighbors' values is computed for regression tasks. The key steps in k*-NN include defining the similarity measure, selecting an appropriate distance metric (e.g., Euclidean, Manhattan) or a more complex contextual similarity measure, and computing the weight for each training instance based on its distance to the query instance. A common approach is to use a kernel function that assigns weights based on the distance, such as a Gaussian kernel. For each new query instance, the algorithm calculates the distance or similarity to all stored instances, computes the weights for each instance based on the chosen similarity measure, and aggregates the results to make a prediction. The advantages of k*-NN include its flexibility in handling different types of data and adapting the similarity measure to the specific context, improving prediction accuracy, and its simplicity in implementing without requiring a complex training phase. However, the algorithm can be slow for large datasets due to the computational complexity of calculating the distance to all training instances for each query, and storing all training instances can be memory-intensive. Overall, k*-NN is a flexible and powerful instance-based learning method suitable for various classification and regression tasks.

3.7 Locally Weighted Learning (lazy.LWL):

Locally Weighted Learning, often called Locally Weighted Regression (LWR) or Locally Weighted Scatterplot Smoothing (LOWESS), is a non-parametric regression technique used for making predictions locally adaptive. Unlike global regression methods that assume a constant relationship across the entire dataset, LWR focuses on modeling the relationship within a local neighborhood of the query point. Overall, it was introduced by At&T Bell Labs in the 1990s, and it was *lazy.LWL* is another k-NN variant. It focuses on local regions of the input space and adapts weights based on the proximity of instances to the query point [37]. A lazy learning algorithm that assigns different weights to different instances during prediction, providing higher importance to locally relevant data. It is beneficial in situations where certain areas of the feature space have more influence on predictions.

LWR aims to predict the value of a target variable for a given input by considering a weighted combination of the observed instances in the dataset. The weight assigned to each instance is determined by its proximity to the query point. Define a local neighborhood around the query point. This is typically done using a kernel or weighting function that assigns higher weights to instances closer to the query point and lower weights to instances farther away. Choose a function (kernel) that assigns weights based on the distance between each data point and the query point. Common choices include the Gaussian kernel or the Epanechnikov kernel. Perform a regression within the local neighborhood by assigning weights to each data point based on the chosen kernel. The weighted regression may be linear or nonlinear depending on how the variables relate. Predict the target variable for the query point by combining the predictions from the locally weighted regression. LWR adapts to the local characteristics of the data, allowing for more flexibility in capturing complex and nonlinear relationships. It provides a smooth and adaptive way of modeling the underlying function.

3.8 Random Subspace:

Introduced in the early 2000s, Random Subspace is an ensemble method that combines multiple models trained on random subsets of features [38]. It enhances model diversity, reducing overfitting. A model ensemble technique that builds various models by training each on a random subset of features. The final prediction is typically an average or voting of individual model predictions. It is helpful when dealing with high-dimensional data and aiming to reduce the risk of overfitting. In ovreal, random Subspace is used in ensemble learning, specifically in methods like Random Forests. It involves creating multiple subsets or "subspaces" of the feature space and building individual models on each of these subsets. The idea is to introduce diversity among the base models, making the ensemble more robust and capable of capturing various aspects of the underlying data distribution. Even though random subspace is a valuable ensemble learning technique (e.g., it balances model diversity and generalization), it has some limitations, such as Computational Cost (Training multiple models with different feature subsets can be computationally expensive, especially with large datasets and a large number of features) and Interpretability (As the number of features and trees increases, the interpretability of the model may decrease.). The Random Subspace method enhances the performance and robustness of machine learning models by leveraging the power of ensemble learning. By training multiple

classifiers on different random subsets of features, the method effectively reduces overfitting, handles noisy features, and improves generalization, making it a valuable technique for various predictive modeling tasks.

The Random Subspace method is an ensemble learning technique primarily used to improve the performance and robustness of classifiers. It involves training multiple classifiers on different random subsets of the feature space and then combining their predictions. Ensemble Learning: A technique that combines the predictions of multiple models to improve the overall performance and reduce the risk of overfitting. Feature Subset Selection: Randomly selecting subsets of features from the original dataset to train different models, aggregating the predictions from multiple classifiers to make a final decision. Start with the original dataset, which consists of N instances and D features. Randomly select a subset of features from the original D features. The number of features selected can be a fixed number or a proportion of the total features. The selection process is typically done without replacement, meaning each feature can appear only once in a subset. Train each base classifier using the instances from the dataset but only with the selected subset of features. This results in multiple classifiers, each trained on a different "view" of the data. Different classifiers can be used as base learners, such as decision trees, support vector machines, or neural networks. When making predictions for a new instance, each classifier in the ensemble makes a prediction based on its subset of features. Combine the predictions from all classifiers to produce the final output. This can be done through various aggregation methods, such as majority voting for classification or averaging for regression.

By training classifiers on different subsets of features, the ensemble reduces the risk of overfitting to the training data. The method increases robustness against noisy features, as not all classifiers are exposed to every feature. The Random Subspace method can be used with various base classifiers and applies to both classification and regression tasks. Training and maintaining multiple classifiers require more computational resources and memory. Some classifiers may produce redundant or highly correlated predictions if the feature subsets are not sufficiently diverse.

Determine the number of base classifiers to include in the ensemble. Randomly choose subsets of features for each classifier. The size of each subset can be specified as a fixed number or a fraction of the total features. Train each classifier using the instances from the dataset with their respective feature subsets. When a new instance is to be predicted, collect the predictions from all classifiers and combine them using an appropriate aggregation method. Suppose you have a dataset with 100 features, and decide to use ten base classifiers in your ensemble. For each classifier, you randomly select 20 out of the 100 features. Each classifier is then trained on the instances using only these 20 features. When making a prediction, each classifier provides an output based on its trained model, and the final prediction is determined by combining these outputs, such as by majority vote for classification tasks.

3.9 Random Committee

The Random Committee method is an ensemble learning technique that improves the accuracy and robustness of predictive models by creating multiple instances of the same base classifier, each trained on the same dataset but with different random initializations [39]. These classifiers are then combined to produce a final output. The key concepts of Random Committee involve ensemble learning, random initialization, and classifier combination. In practice, the original dataset trains all classifiers in the ensemble, with each classifier initialized differently. During training, each classifier learns slightly different patterns from the data due to its unique initialization. In the prediction phase, each classifier provides its prediction for a new instance, and these predictions are aggregated to produce the final output, commonly using methods such as majority voting for classification or averaging for regression. The advantages of this method include improved accuracy, robustness to overfitting, and simplicity in implementation. However, it requires more computational resources and memory, and the diversity among classifiers is limited to their random initializations. The practical steps involve defining the ensemble size, initializing multiple instances of the base classifier, training each classifier on the entire dataset, and aggregating their predictions for new instances. For example, using Random Committee with ten instances of a decision tree classifier, each tree is trained on the same dataset but with different random seeds affecting splits or data processing order. The final prediction is determined by a majority vote or averaging, enhancing the model's overall performance and robustness.

3.10 Random Forest

Building on the concepts of ensemble learning, Random Committee involves training several models independently and combining their predictions [40]. Both generalization and robustness are improved. Model ensemble building is similar to Random Subspace, except it employs different subsets of the training data. Effective at merging many models' predictions to enhance generality. In 2001, Leo Breiman also introduced Random Forest. An ensemble learning method called Random Forest creates a lot of decision trees [41]. To decrease overfitting and increase accuracy, it integrates predictions from several trees. A method for group learning that produces a lot of decision trees during training and outputs the class mode (classification) or mean prediction (regression) of each tree. Robust and versatile, suitable for various tasks, and effective at handling complex connections.

Random Forest is a versatile and widely-used ensemble learning method for classification and regression tasks. It works by constructing multiple decision trees during training and outputting the mode of the classes (for classification) or the mean prediction (for regression) of the individual trees. Random Forests enhance the predictive performance of decision trees by combining the outputs of multiple trees trained on different subsets of the data and feature sets. This ensemble approach improves accuracy, reduces overfitting, and provides insights into feature importance, making it a powerful tool for various machine-learning tasks. Random Forests belong to the ensemble learning family, where the predictions of multiple models (decision trees, in this case) are combined to improve overall performance and robustness. Each tree in the forest is a decision tree. In this flowchart-like structure, each internal node represents a feature (or attribute), each branch represents a decision rule, and each leaf node represents an outcome (or class label). During the training phase, Random Forest uses bootstrap aggregating, or bagging. Multiple subsets of the training data are created by random sampling with replacement. Each subset is used to train a different decision tree. This helps reduce the variance and overfitting typically associated with individual decision trees.

When splitting nodes in each decision tree, Random Forests introduce additional randomness by selecting a random subset of features. This means that each split is based on only a random selection of features rather than all features, promoting diversity among the trees and improving model robustness. Each decision tree is trained on a different subset of the data and grows to its maximum depth without pruning. The goal is for each tree to learn different patterns from the data due to the variations in the bootstrap samples and the random feature selection. For classification tasks, when a new input is fed into the Random Forest, it is passed down to each decision tree in the forest. Each tree provides a classification, and the forest outputs the class that receives the majority vote. Each tree provides a numerical prediction for regression tasks, and the forest outputs the average of these predictions.

By averaging multiple trees, Random Forests reduce the risk of overfitting, which is common with individual decision trees. Combining multiple trees generally results in better predictive performance and robustness to noise in the data. Random Forests can estimate the importance of different features in the prediction task, which is useful for understanding the underlying data.

Training many deep trees can be computationally intensive and require significant memory, especially with large datasets. While individual decision trees are easy to interpret, the overall Random Forest model, an ensemble of many trees, can be complex and less interpretable.

4. Results and Discussion:

To accurately evaluate predictive models' performance, the dataset will be divided into training and testing sets using a suitable splitting strategy. Before splitting the data, the dataset will be randomized to ensure that the samples are randomly shuffled. This helps prevent any bias or systematic patterns in the data that could affect model performance. The dataset will be partitioned into two subsets: training and testing sets. The training set will comprise most of the data (e.g., 70-80%), while the testing set will consist of the remaining portion (e.g., 20-30%). Suppose the dataset is imbalanced or contains categorical target variables. In that case, a stratified splitting approach may be employed to ensure that the distribution of classes is preserved in both the training and testing sets. This helps prevent potential biases in model evaluation, particularly for classification tasks. A temporal splitting approach may be used if the dataset exhibits temporal dependencies or time-series patterns. In this case, the dataset is divided into training and testing sets based on a chronological time frame, ensuring that training data precedes testing data.

This approach helps evaluate the model's generalization ability to unseen future data. In addition to a single train-test split, cross-validation techniques such as k-fold cross-validation or time-series cross-validation may be employed for robust model evaluation. These techniques involve repeatedly splitting the data into multiple folds, training the model on different subsets, and averaging the performance metrics across folds to obtain more reliable estimates of model performance. Sometimes, a separate validation set may be used for hyperparameter tuning and model selection. This validation set is distinct from the training and testing sets. It is used to assess the performance of different model configurations before finalizing the chosen model for evaluation on the testing set.

By following these steps, we ensure that the predictive models are trained on a representative subset of the data and evaluated on unseen data to assess their generalization performance accurately. This approach helps mitigate overfitting and provides reliable estimates of model performance in predicting energy consumption patterns in OPEC countries. To ensure the accuracy and reliability of predictive models for energy consumption in OPEC countries, the dataset will systematically be divided into training and testing sets. Initially, the dataset will be randomized to eliminate inherent biases or patterns. Following randomization, the dataset will be partitioned into two subsets, with the majority allocated to the training set (typically 70-80%) and the remainder reserved for the testing set. This partitioning ensures that the models are trained on a representative sample of the data and evaluated on unseen data, enabling an accurate assessment of their generalization performance. Suppose the dataset is imbalanced or contains categorical target variables, a stratified splitting approach will be employed to preserve the distribution of classes or target variable values in both sets.

Additionally, if temporal dependencies are present, a temporal splitting approach will be utilized, ensuring that the training set contains data from earlier periods while the testing set comprises data from later periods. Finally, cross-validation techniques such as k-fold cross validation or time-series cross-validation may be applied to assess model stability and generalization ability further. Following these systematic steps, we aim to develop predictive models that effectively capture energy consumption patterns in OPEC countries and provide reliable energy planning and policy-making insights.

We will employ hyperparameter tuning and rigorous model validation techniques to optimize the performance of predictive models for energy consumption in OPEC countries. Hyperparameters, including model-specific, regularization, learning rates, and kernel parameters, will be tuned to enhance model performance and prevent overfitting. Model validation will involve a train-validation split of the training data, allowing us to evaluate various hyperparameter configurations and select the optimal model. Additionally, cross-validation techniques such as k-fold cross-validation or time-series cross-validation may be utilized to assess model stability and generalization ability across different subsets of the data. Performance metrics such as mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), coefficient of determination (R-squared), or classification accuracy will be employed to evaluate model performance quantitatively. By systematically tuning hyperparameters and performing robust model validation, we aim to develop predictive models that accurately capture energy consumption patterns in OPEC countries and provide actionable insights for energy planning and policy-making.

The performance of various machine learning methods was evaluated during the training phase, with each model assessed based on multiple metrics, including R^2, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Relative Absolute Error (RAE), and Root Relative Squared Error (RRSE). Among the models, Random Committee achieved the highest performance with an R^2 of 0.9999, MAE of 0.7411, RMSE of 1.0509, RAE of 1.2199%, and RRSE of 1.2399%, resulting in a total ranking score of 50 and the top rank overall. Lazy.Kstar also performed well with an R^2 of 0.9995 and a total ranking score of 44, securing the second rank—conversely, models such as lazy.LWL showed relatively poor performance, with the lowest R^2 of 0.8857 and the highest errors across all metrics, leading to the lowest total ranking score of 5 and rank of 10. Other models like MLP, Simple Linear Regression, and SMOreg exhibited strong R^2 values (0.9964, 0.9957, and 0.9956, respectively) but varied in their ranking scores, placing them in the middle of the performance spectrum. Gaussian Processes, Random Subspace, and Random Forest also demonstrated high accuracy, with Random Forest achieving an R^2 of 0.9990 and a total ranking score of 38, earning it the third rank. These results highlight the superior predictive capabilities of ensemble methods like Random Committee and

Random Forest in the context of energy consumption prediction for OPEC countries.

Various machine learning models were evaluated during the testing phase to determine their predictive accuracy and reliability. Lazy.Kstar emerged as the top-performing model with an R^2 of 0.9965, MAE of 5.5216, RMSE of 9.8002, RAE of 8.6381%, and RRSE of 10.1235%, achieving the highest total ranking score of 46 and securing the first rank overall. Simple Linear Regression also demonstrated strong performance with an R^2 of 0.9948 and a total ranking score of 44, earning it the second rank. SMOreg ranked third with an R^2 of 0.9943 and a total ranking score of 42. Despite its high R^2 value of 0.9949, lazy.IBK scored lower in other metrics, resulting in a fourth-place rank with a total ranking score of 35. The MLP model showed decent performance with an R^2 of 0.9944 and a ranking score of 33, placing it fifth. Random Forest and Gaussian Processes, with R^2 values of 0.9914 and 0.9888, respectively, achieved moderate performance with ranking scores of 22 and 21, placing them sixth and seventh, respectively. Despite having an R^2 of 0.9928, Random Subspace ranked eighth with a total score of 15. The Random Committee model, with an R^2 of 0.9525, showed weaker performance in the testing phase, resulting in a ninth-place rank with a total score of 12. Lazy.LWL was the least effective model, with a significantly lower R^2 of 0.8800 and high error metrics, leading to the lowest total ranking score of 5 and ranking tenth. These results underscore the varying effectiveness of different machine learning models in predicting energy consumption for OPEC countries, with Lazy.Kstar and Simple Linear Regression emerged as the most reliable models during the testing phase.

The combined performance of various machine learning models across training and testing phases highlights their overall effectiveness in predicting energy consumption for OPEC countries. Lazy.Kstar emerged as the top performer with a total score of 90, securing the first rank due to its consistently high R^2 values and low error metrics in both phases. Simple Linear Regression also demonstrated robust performance, achieving a total score of 72 and the second rank, followed closely by SMOreg, which attained a total score of 69 and ranked third. The MLP and Random Committee models achieved a total score of 62, tying for fourth place, showcasing their strong predictive capabilities. Lazy.IBK and Random Forest, with total scores of 60 each, were ranked sixth, indicating moderate performance. Despite its competitive results in some metrics, Gaussian Processes ranked eighth with a total score of 39. Random Subspace, with a total score of 26, ranked ninth. Finally, Lazy.LWL was ranked tenth with consistently low performance metrics and a total score of 10. These rankings underscore the varying effectiveness of machine learning models, with ensemble methods like Lazy.Kstar and Random Committee generally perform well, while models like Lazy.LWL lag in predictive accuracy and reliability.

Table 1. The result of different machine learning methods in a training phase

Proposed models	Network results					Ranking the predicted models					Total ranking score	Rank
	R^2	MAE	RMSE	RAE (%)	RRSE (%)	R^2	MAE	RMSE	RAE (%)	RRSE (%)		
Simple Linear Regression	0.9957	3.6926	7.8738	6.0779	9.2897	6	5	5	6	6	28	5
Gaussian Processes	0.9917	8.2645	11.8929	13.6031	14.0316	4	3	3	4	4	18	8
MLP	0.9964	3.8162	7.1862	6.2814	8.4785	7	4	6	5	7	29	4
SMOreg	0.9956	3.0873	8.0767	5.0815	9.5291	5	6	4	7	5	27	6
lazy. IBK	0.9676	1.7232	2.4020	20.0855	25.2651	2	8	9	3	3	25	7
lazy.Kstar	0.9995	1.3113	2.7147	2.1584	3.2029	9	9	8	9	9	44	2
lazy.LWL	0.8857	29.3897	39.5106	48.3746	46.6160	1	1	1	1	1	5	10
Random Subspace	0.9761	17.0357	24.2137	28.0403	28.5682	3	2	2	2	2	11	9
Random Committee	0.9999	0.7411	1.0509	1.2199	1.2399	10	10	10	10	10	50	1
Random Forest	0.9990	2.2571	4.3641	3.7152	5.1489	8	7	7	8	8	38	3

Table 2. The result of different machine learning methods in the testing phase

Proposed models	Network results					Ranking the predicted models					Total ranking score	Rank
	R^2	MAE	RMSE	RAE (%)	RRSE (%)	R^2	MAE	RMSE	RAE (%)	RRSE (%)		
Simple Linear Regression	0.9948	5.4106	9.8607	8.4644	10.1860	8	9	9	9	9	44	2
Gaussian Processes	0.9888	11.0340	15.9029	17.2617	16.4276	3	4	5	4	5	21	7
MLP	0.9944	6.3164	10.5044	9.8814	10.8509	7	6	7	6	7	33	5
SMOreg	0.9943	4.9416	10.4019	7.7307	10.7451	6	10	8	10	8	42	3
lazy. IBK	0.9949	5.5859	10.7424	8.7386	11.0968	9	7	6	7	6	35	4
lazy.Kstar	0.9965	5.5216	9.8002	8.6381	10.1235	10	8	10	8	10	46	1
lazy.LWL	0.8800	36.9461	51.5478	57.7988	53.2485	1	1	1	1	1	5	10
Random Subspace	0.9928	12.7976	19.5344	20.0206	20.1788	5	2	3	2	3	15	8
Random Committee	0.9525	11.4795	30.6585	17.9586	31.6700	2	3	2	3	2	12	9
Random Forest	0.9914	8.0524	16.4844	12.5973	17.0282	4	5	4	5	4	22	6

Table 3. Ranking of training and testing

Proposed models	Network result										Total score	Total rank
	Training dataset					Testing dataset						
	R^2	MAE	RMSE	RAE (%)	RRSE (%)	R^2	MAE	RMSE	RAE	RRSE		
Simple Linear Regression	6	5	5	6	6	8	9	9	9	9	72	2
Gaussian Processes	4	3	3	4	4	3	4	5	4	5	39	8
MLP	7	4	6	5	7	7	6	7	6	7	62	4
SMOreg	5	6	4	7	5	6	10	8	10	8	69	3
lazy. IBK	2	8	9	3	3	9	7	6	7	6	60	6
lazy.Kstar	9	9	8	9	9	10	8	10	8	10	90	1
lazy.LWL	1	1	1	1	1	1	1	1	1	1	10	10
Random Subspace	3	2	2	2	2	5	2	3	2	3	26	9
Random Committee	10	10	10	10	10	2	3	2	3	2	62	4
Random Forest	8	7	7	8	8	4	5	4	5	4	60	6

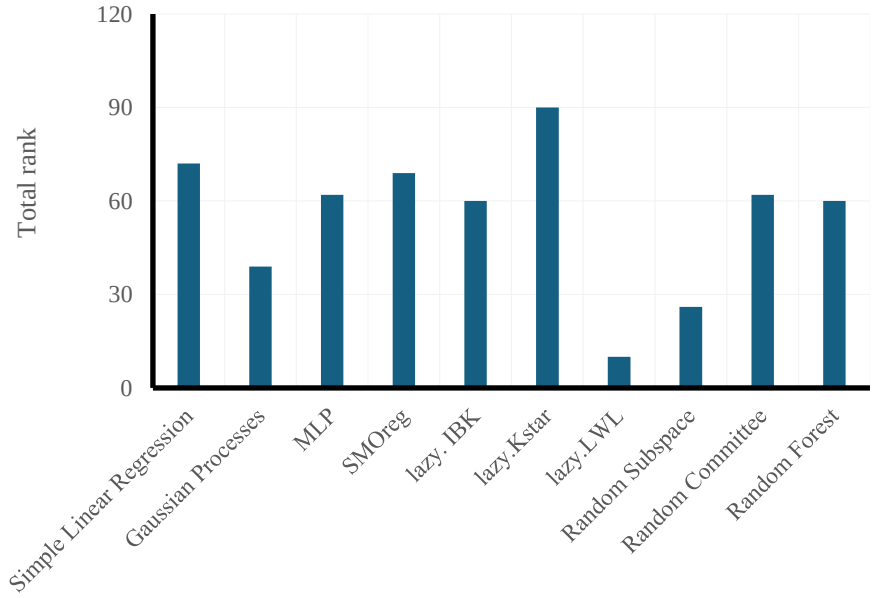


Figure 2. Machine learning ranking

5. Discussion:

The comprehensive analysis of various machine learning models for predicting energy consumption in OPEC countries yielded several key insights—the superior performance of Lazy.Kstar, which achieved the highest overall rank with a total score of 90, indicates that this model is highly effective in capturing complex patterns in the energy consumption data. Its consistently high R^2 values and low error metrics suggest it can provide reliable and accurate predictions. Simple Linear Regression, with a total score of 72, also demonstrated strong predictive capabilities, highlighting the utility of simpler models in certain contexts. The moderate performance of models like MLP and Random Committee, each with a total score of 62, underscores the importance of neural networks and ensemble methods in handling diverse datasets. However, the relatively lower ranks of models such as Gaussian Processes and Random Subspace, with scores of 39 and 26, respectively, suggest that while they are useful, they may not be as robust for this particular application—the consistently poor performance of Lazy.LWL, with a total score of 10, highlights its limitations and suggests that it may not be suitable for predicting energy consumption in OPEC countries. These insights emphasize the importance of selecting the appropriate model based on specific data characteristics and prediction requirements. Overall, the findings underscore the potential of advanced machine learning techniques to enhance energy consumption forecasting, thereby aiding in more informed energy planning and policy-making for OPEC countries.

Applying machine learning methods to predict energy consumption in OPEC countries offers several strengths and limitations. One of the main strengths is the ability of advanced algorithms like Lazy.Kstar and Random Committee capture complex patterns and interactions in the data, leading to highly accurate predictions. Additionally, using cross-validation techniques enhances the robustness and generalizability of the models. However, there are several limitations and potential sources of error. Firstly, the quality and completeness of the input data from sources such as www.eia.gov are crucial; any missing or inaccurate data can significantly affect the model's performance.

Moreover, the models' reliance on historical data may not fully account for future changes in energy consumption patterns due to unforeseen geopolitical, economic, or environmental factors. Another limitation is the potential overfitting of some models, despite efforts to mitigate this through regularization and cross-validation. Overfitting can result in models that perform well on training data but poorly on unseen data. Additionally, the computational complexity of

certain algorithms, such as neural networks and ensemble methods, may pose challenges regarding processing time and resource requirements. Lastly, the interpretability of complex models can be an issue, making it difficult to derive actionable insights and explain the predictions to stakeholders. Addressing these limitations requires ongoing refinement of models, incorporating more diverse and high-quality data, and developing methods to enhance model interpretability and robustness.

The findings from this analysis have significant implications for policymakers, energy analysts, and other stakeholders involved in energy planning and policy-making for OPEC countries. The ability of advanced machine learning models, particularly Lazy.Kstar and Random Committee, accurately predicting energy consumption can greatly enhance the precision of energy demand forecasts. This, in turn, enables more effective and informed decision-making regarding resource allocation, infrastructure investment, and strategic planning. For policymakers, the insights gained from these models can inform the development of robust energy policies that ensure a stable and sustainable energy supply, adapt to changing demand patterns, and mitigate the impacts of economic or geopolitical shifts. Energy analysts can leverage these predictive models to conduct more nuanced analyses, identify trends, and evaluate the potential impacts of different policy scenarios.

Furthermore, stakeholders such as energy producers and investors can use these predictions to optimize production schedules, plan for future capacity expansions, and manage financial risks more effectively. However, it is essential for all stakeholders to recognize the limitations and uncertainties inherent in predictive modeling, such as data quality and the potential for unforeseen future events, and to complement these models with qualitative insights and expert judgment. Overall, integrating machine learning predictions into the energy planning process can lead to more resilient and adaptive energy systems, benefiting both the OPEC countries and the global energy market.

6. Conclusions

In this study, we investigated the application of ten different machine learning algorithms to predict energy consumption in OPEC countries using historical data from the U.S. Energy Information Administration. Among the tested models, the Random Committee, lazy.Kstar, and Simple Linear Regression demonstrated the highest performance in both training and testing phases, with Random Committee achieving the top overall rank. These results highlight the potential of ensemble and instance-based learning methods in accurately modeling complex energy consumption patterns. The findings are significant for policymakers and analysts, as reliable forecasts can support informed decision-making in energy management and strategic planning. Future research can expand this framework by incorporating deep learning methods and external variables such as economic indicators, environmental policies, and geopolitical events to further enhance prediction accuracy.

References

[1] Akinrolabu, DR, MA Haruna, Organization of Petroleum Exporting Countries (OPEC) at 60: Challenges, Achievements and Prospects Towards Global Economic and Investment Development, Achievers Journal of Scientific Research, 3 (2021) 182-196,

[2] Khan, Abdullah, Haruna Chiroma, Muhammad Imran, Javed Iqbal Bangash, Muhammad Asim, Mukhtar F Hamza, Hanan Aljuaid, Forecasting electricity consumption based on machine learning to improve performance: A case study for the organization of petroleum exporting countries (OPEC), Computers & Electrical Engineering, 86 (2020) 106737,

[3] Ahmad, Tanveer, Hongcai Zhang, Biao Yan, A review on renewable energy and electricity requirement forecasting models for smart grid and buildings, Sustainable Cities and Society, 55 (2020) 102052,

[4] O'Dwyer, Edward, Indranil Pan, Salvador Acha, Nilay Shah, Smart energy systems for sustainable smart cities: Current developments, trends and future directions, Applied energy, 237 (2019) 581-597,

[5] Majid, MA4126072, Renewable energy for sustainable development in India: current status, future prospects, challenges, employment, and investment opportunities, Energy, Sustainability and Society, 10 (2020) 1-36,

[6] Okoye, Lawrence Uchenna, Alex Omankhanlen, Johnson I Okoh, Ngozi B Adeleye, Felix N Ezeji, Gideon K Ezu, Benjamin Ehikioya, Analyzing the energy consumption and economic growth nexus in Nigeria, International Journal of Energy Economics and Policy, 11 (2021) 378-387,

[7] Ahmed, Zeeshan, Khalid Mohamed, Saman Zeeshan, XinQi Dong, Artificial intelligence with multi-functional machine learning platform development for better healthcare and precision medicine, Database, 2020 (2020) baaa010,

[8] Ahmad, Tanveer, Rafal Madonski, Dongdong Zhang, Chao Huang, Asad Mujeeb, Data-driven probabilistic machine learning in sustainable smart energy/smart energy systems: Key developments, challenges, and future research opportunities in the

context of smart grid paradigm, Renewable and Sustainable Energy Reviews, 160 (2022) 112128,
[9] Zhong, Shifa, Kai Zhang, Majid Bagheri, Joel G Burken, April Gu, Baikun Li, Xingmao Ma, Babetta L Marrone, Zhiyong Jason Ren, Joshua Schrier, Machine learning: new ideas and tools in environmental science and engineering, Environmental Science & Technology, 55 (2021) 12741-12754,
[10] Blondeel, Mathieu, Michael J Bradshaw, Gavin Bridge, Caroline Kuzemko, The geopolitics of energy system transformation: A review, Geography Compass, 15 (2021) e12580,
[11] Alzaabi, Fatima Rashed, Abid Mehmood, A Review of Recent Advances, Challenges, and Opportunities in Malicious Insider Threat Detection Using Machine Learning Methods, IEEE Access, 12 (2024) 30907-30927,
[12] Antonopoulos, Ioannis, Valentin Robu, Benoit Couraud, Desen Kirli, Sonam Norbu, Aristides Kiprakis, David Flynn, Sergio Elizondo-Gonzalez, Steve Wattam, Artificial intelligence and machine learning approaches to energy demand-side response: A systematic review, Renewable and Sustainable Energy Reviews, 130 (2020) 109899,
[13] Rind, Yousaf Murtaza, Muhammad Haseeb Raza, Muhammad Zubair, Muhammad Qasim Mehmood, Yehia Massoud, Smart energy meters for smart grids, an internet of things perspective, Energies, 16 (2023) 1974,
[14] Sharma, Vinod, A study on data scaling methods for machine learning, International Journal for Global Academic & Scientific Research, 1 (2022) 31-42,
[15] Bourdeau, Mathieu, Xiao qiang Zhai, Elyes Nefzaoui, Xiaofeng Guo, Patrice Chatellier, Modeling and forecasting building energy consumption: A review of data-driven techniques, Sustainable Cities and Society, 48 (2019) 101533,
[16] Narciso, Diogo AC, FG Martins, Application of machine learning tools for energy efficiency in industry: A review, Energy Reports, 6 (2020) 1181-1199,
[17] Gheibi, Omid, Danny Weyns, Federico Quin, Applying machine learning in self-adaptive systems: A systematic literature review, ACM Transactions on Autonomous and Adaptive Systems (TAAS), 15 (2021) 1-37,
[18] Rangel-Martinez, Daniel, KDP Nigam, Luis A Ricardez-Sandoval, Machine learning on sustainable energy: A review and outlook on renewable energy systems, catalysis, smart grid and energy storage, Chemical Engineering Research and Design, 174 (2021) 414-441,
[19] Burnett, J Wesley, L Lynne Kiesling, How do machines predict energy use? Comparing machine learning approaches for modeling household energy demand in the United States, Energy Research & Social Science, 91 (2022) 102715,
[20] Liu, Yang, Hongyu Chen, Limao Zhang, Xianguo Wu, Xian-jia Wang, Energy consumption prediction and diagnosis of public buildings based on support vector machine learning: A case study in China, Journal of Cleaner Production, 272 (2020) 122542,
[21] Liu, Xiaodong, Haode Sun, Shanshan Han, Shuyan Han, Shengnan Niu, Wen Qin, Piman Sun, Dexuan Song, A data mining research on office building energy pattern based on time-series energy consumption data, Energy and Buildings, 259 (2022) 111888,
[22] Ruiz, Luis Gonzaga Baca, MC Pegalajar, Rossella Arcucci, Miguel Molina-Solana, A time-series clustering methodology for knowledge extraction in energy consumption data, Expert Systems with Applications, 160 (2020) 113731,
[23] Bhuiyan, Md Monjur Hossain, Ahmed Nazmus Sakib, Syed Ishmam Alawee, Talayeh Razzaghi, Fueling the Future: A Comprehensive Analysis and Forecast of Fuel Consumption Trends in US Electricity Generation, Sustainability, 16 (2024) 2388,
[24] Fumo, Nelson, MA Rafe Biswas, Regression analysis for prediction of residential energy consumption, Renewable and sustainable energy reviews, 47 (2015) 332-343,
[25] Bourhnane, Safae, Mohamed Riduan Abid, Rachid Lghoul, Khalid Zine-Dine, Najib Elkamoun, Driss Benhaddou, Machine learning for energy consumption prediction and scheduling in smart buildings, SN Applied Sciences, 2 (2020) 297,
[26] Costa, Vinícius G, Carlos E Pedreira, Recent advances in decision trees: An updated survey, Artificial Intelligence Review, 56 (2023) 4765-4800,
[27] Schmidgall, Samuel, Rojin Ziaei, Jascha Achterberg, Louis Kirsch, S Hajiseyedrazi, Jason Eshraghian, Brain-inspired learning in artificial neural networks: A review, APL Machine Learning, 2 (2024),
[28] Li, Yikuan, Mohammad Mamouei, Gholamreza Salimi-Khorshidi, Shishir Rao, Abdelaali Hassaine, Dexter Canoy, Thomas Lukasiewicz, Kazem Rahimi, Hi-BEHRT: hierarchical transformer-based model for accurate prediction of clinical events using multimodal longitudinal electronic health records, IEEE journal of biomedical and health informatics, 27 (2022) 1106-1117,
[29] Sharifzadeh, Mahdi, Alexandra Sikinioti-Lock, Nilay Shah, Machine-learning methods for integrated renewable power generation: A comparative study of artificial neural networks, support vector regression, and Gaussian Process Regression, Renewable and Sustainable Energy Reviews, 108 (2019) 513-538,
[30] Mahesh, Batta, Machine learning algorithms-a review, International Journal of Science and Research (IJSR).[Internet], 9 (2020) 381-386, https://doi.org/10.21275/ART20203995.
[31] Marill, Keith A, Advanced statistics: linear regression, part I: simple linear regression, Academic emergency medicine, 11 (2004) 87-93,

[32] MacKay, David JC, Introduction to Gaussian processes, NATO ASI series F computer and systems sciences, 168 (1998) 133-166,
[33] Taud, Hind, Jean-Franccois Mas, Multilayer perceptron (MLP), Geomatic approaches for modeling land change scenarios, (2018) 451-455,
[34] Al Rasyid, M Udin Harun, Isbat Uzzin Nadhori, Iwan Syarif, Idris Winarno, Fatin Furoida, Ata Amrullah, Anomaly detection in wireless body area network using Mahalanobis distance and sequential minimal optimization regression, 2021 International Seminar on Application for Technology of Information and Communication (iSemantic), IEEE, 2021, pp. 64-69.
[35] Vijayarani, S, M Muthulakshmi, Comparative analysis of bayes and lazy classification algorithms, International Journal of Advanced Research in Computer and Communication Engineering, 2 (2013) 3118-3124,
[36] Alshammari, Majdah, Mohammad Mezher, A comparative analysis of data mining techniques on breast cancer diagnosis data using WEKA toolbox, (2020),
[37] Panda, Mrutyunjaya, Ajith Abraham, Swagatam Das, Manas Ranjan Patra, Network intrusion detection system: A machine learning approach, Intelligent Decision Technologies, 5 (2011) 347-356,
[38] Skurichina, Marina, Robert PW Duin, Bagging, boosting and the random subspace method for linear classifiers, Pattern Analysis & Applications, 5 (2002) 121-135,
[39] Ganaie, Mudasir A, Minghui Hu, Ashwani Kumar Malik, Muhammad Tanveer, Ponnuthurai N Suganthan, Ensemble deep learning: A review, Engineering Applications of Artificial Intelligence, 115 (2022) 105151,
[40] Niranjan, A, DH Nutan, A Nitish, P Deepa Shenoy, KR Venugopal, ERCR TV: Ensemble of random committee and random tree for efficient anomaly classification using voting, 2018 3rd international conference for convergence in technology (I2CT), IEEE, 2018, pp. 1-5.
[41] Rigatti, Steven J, Random forest, Journal of Insurance Medicine, 47 (2017) 31-39,