



OPEN Evaluating machine learning algorithms for energy consumption prediction in electric vehicles: A comparative study

Izhar Hussain^{1,2✉}, Kok Boon Ching^{1✉}, Chessda Uttraphan³, Kim Gaik Tay⁴ & Adil Noor¹

An accurate energy consumption prediction becomes crucial with increasing electric vehicle usage for effective power grid management. This research examined the performance of eleven machine learning models for this purpose: Ridge Regression, Lasso Regression, K-Nearest Neighbors, Gradient Boosting, Support Vector Regression, Multi-Layer Perceptron, XGBoost, CatBoost, LightGBM, Gaussian Processes for Regression (GPR) and Extra Trees Regressor, considering real historical data from Colorado. The models were evaluated using different metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), R^2 , Root Mean Squared Error (RMSE) and Normalized Root Mean Squared Error (NRMSE), with visual analyses through scatter plots and time series plots. The best model observed was the Extra Trees Regressor, which had an MAE of 0.5888, an MSE of 3.2683, R^2 value of 0.9592, RMSE of 1.8078 and NRMSE of 0.020. Gradient Boosting and KNN also returned good results, although they were slightly more dispersed. Nevertheless, while non-linear models like MLP, XGBoost, CatBoost, LightGBM and linear models such as Ridge and Lasso Regression offer valuable insights, they exhibit shortcomings in estimating energy, especially at extreme levels, highlighting limitations in capturing complex non-linear interactions. This study focuses on their applicability to energy projections to demonstrate how well ensemble and non-linear models may capture intricate patterns in time series. These cutting-edge machine learning techniques might greatly enhance energy demand predictions.

Keywords Energy consumption prediction, Machine learning algorithms, Electric vehicles

Nomenclature

EV	Electric Vehicle
HEVs	hybrid electric vehicles
ML	Machine Learning
SVR	Support Vector Regression
SVM	SUPPORT VECTOR Machine
KNN	K-Nearest Neighbors
NNs	Neural Networks
GPR	Gaussian Process for Regression
CatBoost	Categorical Boosting
XGBoost	eXtreme Gradient Boosting
LightGBM	Light Gradient Boosting Machine
SGD	Stochastic gradient descent
FedAvg	Federated averaging
DRL	deep reinforcement learning
LSTM	Long Short-Term Memory

¹Departement of Electrical Engineering, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, Johar 86400, Malaysia. ²Department of Electronics Engineering Benazir Bhutto Shaheed, University of Technology and Skill Development, Khairpure mirs, Khairpur 66020, Pakistan. ³Department of Computer Engineering, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia. ⁴Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, Johar 86400, Malaysia. ✉email: izharhussain@bbsutsd.edu.pk; bckok@uthm.edu.my

ConvLSTM	Convolutional Long Short-Term Memory
BiConvLSTM	Bidirectional Convolutional Long Short-Term Memory
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
ARIMA	AutoRegressive Integrated Moving Average
TFC	Total Fuel Consumption
MAE	Mean Absolute Error
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
NRMSE	Normalized Root Mean Squared Error
MAPE	Mean Absolute Percentage Error
CRPS	Continuous Ranked Probability Score
RBF	radial basis function

The rising popularity of electric vehicles (EVs) is driving a transition in transportation toward electric vehicles (EVs), which presents both opportunities and problems for energy management systems. With the increasing use of EVs, it is critical to accurately predict their charging requirements to preserve the efficiency and stability of the grid. Accurate predictions enable utility companies to anticipate peak demand, alleviate congestion, and enhance the integration of renewable energy resources. These measures collectively contribute to the advancement of a more sustainable future. According to the latest machine learning algorithms, energy consumption prediction has experienced a revolution, which provides strong tools for deciphering complex trends in huge data sets. Unlike conventional statistical methods, models such as Gradient Boosting Machines, Neural Networks, and Ridge Regression promise to increase prediction accuracy by utilizing past data to comprehend complex relationships between variables and offer insightful information about future energy demand. Even with these developments, choosing the best models and evaluating their effectiveness across various datasets and scenarios still present challenges^{1,2}.

This research presents a novel approach by thoroughly evaluating and comparing Eleven machine learning models: Ridge Regression, Lasso Regression, K-Nearest Neighbors, Gradient Boosting, Support Vector Regression, Multi-Layer Perceptron, XGBoost, CatBoost, LightGBM, GPR and Extra Trees Regressor, for predicting energy consumption in electric vehicles (EVs). Unlike previous studies that often concentrate on individual or similar models, this investigation rigorously benchmarks each model's predictive performance using error metrics. The findings ultimately highlight the Extra Trees Regressor as the most effective model for accurately capturing the complex, non-linear energy consumption patterns associated with EV data. This approach addresses a significant gap in the existing literature, showcasing the benefits of ensemble learning for precise and scalable energy demand predicting, which is vital for effective real-time grid management and sustainable energy planning in light of the increasing adoption of EVs.

The prime objective of this article is to evaluate and contrast the performance of different ML models in predicting energy consumption using historical data from Colorado. Specifically, the study aims to:

1. **Assess Model Accuracy:** The research will assess the extent to which seven different machine learning models—namely, the Extra Trees Regressor, Gradient Boosting, the Support Vector Regression (SVR), K-Nearest Neighbors (KNN), Ridge Regression, XGBoost, CatBoost, LightGBM, GPR and Lasso Regression—predict energy consumption. Evaluation measures such as MAE, MSE, R-squared (R^2), RMSE and NRMSE will be used to assess each model's accuracy and dependability.
2. **Analyze Prediction Patterns:** Scatter plots and time series graphs will be used in the study to demonstrate and examine each model's prediction patterns. This method will assist in identifying any notable inconsistencies or deviations from the models and give an extensive understanding of how well the models reflect real energy consumption patterns. These visual aids will make comparing the models' efficacy and performance in properly predicting energy usage easier.
3. **Identify Superior Models:** This study aims to determine the most effective machine learning algorithms for predicting energy usage. The investigation will evaluate which models demonstrate superior precision and dependability, making them ideal for practical applications in energy consumption prediction. The primary objective is to identify algorithms that consistently accurately capture and project energy patterns.

Accomplishing these goals allows the study to enhance the understanding of machine learning applications in energy prediction and offer practical recommendations for improving power grid operations amid rising energy consumption prediction in terms of EVs.

Literature review

Researchers have developed models, namely ConvLSTM and BiConvLSTM, to predict energy usage at electric vehicle (EV) charging stations. Both models outperform traditional forecasting methods, with BiConvLSTM achieving a mean absolute error (MAE) of 4.89 kW, compared to 5.34 kW for ConvLSTM. However, due to its complexity, BiConvLSTM is less suitable for real-time applications that rely on smaller data sets. Additional methods evaluated in the study included artificial neural networks (ANN), long short-term memory (LSTM), and autoregressive integrated moving average (ARIMA) for forecasting EV charging demand³. The prediction accuracy increased by incorporating external variables like the weather and special day indications. While LSTM is excellent for making short-term predictions, ARIMA proved flexible when working with large datasets. The research pointed out weaknesses due to its dependence on historical data, which might not accurately reflect how dynamic EV uptake and usage trends are. Regarding prediction accuracy, ARIMA models with exogenous

variables performed better at the national and local levels than LSTM models. The expansion of EVs, which imposes more stress on electrical networks, highlights the implication of advanced prediction techniques⁴. Research indicates that machine learning (ML) models outperform traditional statistical models in accuracy, particularly when analyzing complex atmospheric data. These ML models require a substantial amount of historical data and must account for external factors. Notably, the Long Short-Term Memory (LSTM) model identified intricate charging patterns, achieving a mean squared error (MSE) of 0.0022 and a mean absolute percentage error (MAPE) of 5.7%. According to the article⁵, LSTM deep learning methods perform better than other approaches, considering huge data sets and prediction instantly. However, the actual applicability of these models may be limited because of their strong reliance on enormous computing resources and high-quality information. Compared to traditional models such as ARIMA, which recorded an RMSE of 5.34 kW and a MAPE of 6.1%, the LSTM model obtained an RMSE of 4.65 kW and a MAPE of 5.2%. The ANN method effectively predicted energy consumption, whereas ensemble models performed well in predicting session length. The ANN model achieved an R^2 of 0.85, an MAE of 23.1 kWh, and an RMSE of 32.7 kWh⁶.

A different approach included combining similar driving behaviours using K-means clustering and afterwards applying two-layer neural networks (NNs), stochastic gradient descent (SGD), federated averaging (FedAvg), and federated averaging with clustering (FedAGC) to the models⁷. With an RMSE of 25.6 kWh and a CRPS of 0.345, the Neural Network with FedAGC demonstrated the greatest performance⁸. A technique utilising deep reinforcement learning (DRL) and computer vision was created for energy management in hybrid electric vehicles (HEVs)⁷. The researchers combined a DRL algorithm with a Convolutional Neural Network (CNN) to maximise energy distribution decisions. While keeping an average speed of 50 km/h, the DRL model improved Total Fuel Consumption (TFC) by 15% compared to traditional models. The researchers in⁹ assessed a range of AI models, such as CNNs for deep learning and RF and SVM for machine learning. In energy demand prediction, CNNs performed well, achieving a 4.2% MAPE and a 5.3% RMSE. Managing the charging of electric vehicles may improve energy efficiency in smart networks. It predicts charging demand and grid load utilising machine learning models and optimisation methods like Particle Swarm optimisation, Genetic methods, and Linear Programming. Grid stability index and total energy cost are examples of evaluation measures. According to the study, the PSO algorithm has the most significant grid stability index, while the LP method has the lowest energy cost⁷. The use of AI and ML in the development of hybrid electric vehicles (HEVs) on the use of neural networks (NN) for prediction maintenance and reinforcement learning (RL) for energy management optimisation. The challenges with computational complexity and real-time processing needs that emerge when incorporating AI into HEV systems. The evaluation criteria include predictive maintenance accuracy, model training time, and energy efficiency. The NN-based predictive maintenance system had 94% accuracy, and the RL model increased energy efficiency by 12%⁸. The authors propose using four Support Vector Machine (SVM) kernel functions to recognise power theft occurrences. Using this method, the user's category may be predicted by examining patterns in power use. This approach utilises the capabilities of the polynomial, sigmoid, linear, and radial basis function (RBF) kernel functions¹⁰.

Machine learning (ML) algorithms have been extensively applied to wind power prediction, showcasing diverse approaches and datasets. Random Forest, AdaBoost, and k-Nearest Neighbors were evaluated using hourly wind farm data from Texas over one year, with Random Forest achieving the lowest Mean Absolute Percentage Error (MAPE) of 0.0079, followed by AdaBoost (MAPE=2.78) and k-Nearest Neighbors (MAPE=16.33)¹¹. Similarly, Decision Tree, LightGBM, and Extra Tree were utilized with California solar farm data (2019–2021), where Extra Tree demonstrated superior performance with a Root Mean Square Error (RMSE) of 2.851, outperforming LightGBM (RMSE=4.48) and Decision Tree (RMSE=5.0002)¹². Advanced techniques like CNN-LSTM combined with Bayesian Optimization, Gradient Boosting, and Ensemble models (Gradient Boosting and XGBoost) were tested on wind turbine meteorological data, with CNN-LSTM achieving the best performance (MSE=6.8) within a computation time of 450 s¹³. Another study employed LightGBM, Gradient Boosting Regressor, AdaBoost, Elastic Net, Lasso, and an Ensemble model using SCADA system data, where the Ensemble model (LightGBM and AdaBoost) achieved the lowest RMSE of 11.78¹⁴. Furthermore, a comparison using the Sotavento wind farm dataset (Galicia, Spain) revealed that LightGBM achieved R^2 scores of 0.942 (train) and 0.939 (test), closely followed by Gradient Boosting Regressor, Random Forest, and k-Nearest Neighbors with comparable R^2 values¹⁵. Lastly, LightGBM applied to SCADA system data from Turkey achieved exceptional results with an R^2 of 1.0 for power prediction, RMSE of 4.27, and MAE of 2.71¹⁶. Recent studies on renewable energy prediction and charge consumption have utilized various machine learning algorithms and evaluation metrics, as summarized in Table 1. Algorithms such as Random Forest, LightGBM, Gradient Boosting, CNN-LSTM, and Ensemble models have been widely applied to datasets from wind farms, solar farms, EV charging stations and SCADA systems. Evaluation metrics, including MAPE, RMSE, MSE, R^2 , and MAE, highlight the performance of these models.

These findings highlight the versatility and effectiveness of ML algorithms, such as Random Forest, LightGBM, Gradient Boosting, and Ensemble methods, in addressing the challenges of renewable energy prediction. Their success provides valuable insights for evaluating algorithms in other domains, such as electric vehicle power consumption prediction.

Applications for predicting charging consumption (electric vehicle)

Infrastructure planning and optimization Predicting the energy consumption demands for electric vehicle (EV) charging is critical for effective station placement. Infrastructure design requires an identification of the location and timing of demand peaks. To locate EV charging stations effectively, advanced methods emphasise determining the technological feasibility¹⁷.

Reference	Existing Approches	Metric Values	Insights
3	ConvLSTM, BiConvLSTM	ConvLSTM: MAE = 5.34 kW, RMSE = 6.78 kW BiConvLSTM: MAE = 4.89 kW, RMSE = 6.21 kW	LSTM excels for short-term predictions; ARIMA is flexible with large datasets. Challenges include dependence on historical data.
4	LSTM, ARIMA	ARIMA: better at national/local levels LSTM: suitable for short-term predictions	ML models incorporating atmospheric data perform better but require large datasets. Highlights EV growth's stress on grids and prediction importance.
5	LSTM	MSE = 0.0022, MAPE = 5.7% LSTM: RMSE = 4.65 kW, MAPE = 5.2% ARIMA: RMSE = 5.34 kW, MAPE = 6.1%	LSTM handles complex charging patterns effectively but relies on significant computational resources and data quality.
6	ANN	$R^2 = 0.85$, MAE = 23.1 kWh, RMSE = 32.7 kWh	Effective in energy consumption prediction. Ensemble models excel in predicting session length.
8	Neural Network, FedAGC	RMSE = 25.6 kWh, CRPS = 0.345	FedAGC demonstrated the best performance for clustered driving behavior models.
7	DRL, CNN	TFC reduction = 15%	DRL combined with CNN optimizes energy distribution decisions, maintaining an average speed of 50 km/h.
9	CNN, RF, SVM	CNN: MAPE = 4.2%, RMSE = 5.3%	CNNs performed well in energy demand prediction; ML models improve energy efficiency in smart grids.
7	PSO, LP, Genetic Algorithms	PSO: highest grid stability index LP: lowest energy cost	ML models optimize EV charging demand and grid load, evaluated by grid stability index and total energy cost.
8	NN, RL	NN: 94% accuracy, RL: 12% energy efficiency	Explores computational challenges and real-time processing for HEVs using AI/ML models.
10	SVM (polynomial, sigmoid, linear, RBF)	High accuracy in power theft recognition	Predicts user categories by analyzing power usage patterns; utilizes different SVM kernel functions.
11	Random Forest, Ada Boost, K Neighbors	Random Forest (MAPE = 0.0079), Adaboost (MAPE = 2.78), KNearest Neighbors (MAPE = 16.33)	Random Forest achieved the best performance using hourly wind farm data from Texas.
12	Decision Tree, LightGBM, Extra Tree	LightGBM (RMSE = 4.48), Decision Tree (RMSE = 5.0002), Extra Tree (RMSE = 2.851)	Extra Tree performed best on California solar farm data (2019–2021).
13	MLP + Bayesian Optimization, Gradient Boosting, CNN-LSTM, Ensemble (GB & XGB)	MSE; Best: CNN-LSTM (MSE = 6.8, 450 s computation)	CNN-LSTM demonstrated superior performance using wind turbine meteorological data.
14	LightGBM, Gradient Boosting Regressor, Ada Boost, Elastic Net, Lasso, Ensemble	RMSE; Best: Ensemble (LightGBM & Ada Boost, RMSE = 11.78)	Ensemble model (LightGBM + AdaBoost) achieved the best performance on SCADA system data.
15	LightGBM, Gradient Boosting, Random Forest	$R^2 = 0.942$ (train), 0.939 (test) for LightGBM, $R^2 = 0.934$ (Train), 0.933 (test) for GB Regressor, $R^2 = 0.986$ (train), 0.926 (test) for Random Forest, $R^2 = 0.949$ (train), 0.924 (test) for KNeighbors Regressor	LightGBM showed superior performance with Sotavento wind farm data (Galicia, Spain).
16	LightGBM	$R^2 = 1.0$ for power prediction, RMSE = 4.27, MAE = 2.71	LightGBM achieved exceptional results using SCADA system data from Turkey for power prediction.

Table 1. Summary of key studies with algorithms used and their evaluation metrics.

Grid load management The unpredictable nature of EV charging behaviour can result in demand variations during peak hours; hence, predicting the charging demand is crucial for grid stability. To control loads, avoid grid overloads, and assure a steady power supply, grid operators can use precise prediction models for energy demand predictions¹⁸.

Energy cost optimization Energy suppliers may utilise EV charging demand projections to optimise energy procurement costs. By acquiring energy at off-peak hours and storing it for later use, providers can minimise overall costs and increase profitability, as recent research has shown that machine learning models have demonstrated potential in predicting charging behaviours based on numerous criteria¹⁹.

Environmental impact and policy making Policymakers use predictions for EV charging to develop regulations and incentives that promote the adoption of electric vehicles and address their environmental consequences. By predicting charging requirements, renewable energy may be integrated sustainably and improve environmental policy and urban planning choices²⁰.

Improvement of user experience Enhancing the electric vehicle (EV) owner experience requires accurate demand predictions. Service providers could ensure sufficient infrastructure for charging by predicting high demand times and places, which decreases wait times and improves customer experience. Dynamic pricing models can also be supported by real-time data-driven methods, which incentivise customers to charge their EVs during off-peak hours, balancing demand and enhancing satisfaction²¹.

Data description

The statistics about stations of EV charging in Colorado were gathered using automated systems that monitor different factors connected to EV charging sessions. These systems are placed at charging stations owned by the city and record specific details about every charging session. The gathered information encompasses:

- **Station and Location Information:** The documentation encompasses specific data on the geographical placement of each charging station, incorporating the precise address, the municipality in which it is situated, and the associated postal code.
- **Session Details:** Commencement and completion times, along with the length of every charging session.
- **Energy Metrics:** The electricity used during the session (in kWh).
- **Environmental Impact:** Assessments of the reductions in greenhouse gas (GHG) emissions and gasoline usage demonstrate the environmental advantages of utilizing electric vehicles instead of conventional gasoline-powered vehicles.

The management systems of charging stations automatically compile data for real-time monitoring. This aggregated data, available on public platforms like Boulder’s open data portal, includes information on energy usage, charging duration, cost savings, and greenhouse gas emissions. The dataset was last updated on September 7, 2023. A statistical summary, likely generated by the describe() function in pandas, offers an overview of central tendency, dispersion, and shape for each column in the dataset. The meaning of each line for each column is explained below.

Count The count of non-empty values in the column.

Mean The column’s average value.

Std It measures the variation or dispersion of a set of values. A low standard deviation indicates that the values are close to the mean, while a high standard deviation indicates that the values are spread over a broader range.

Min The minimum value of the entire column.

25% The value below which 25% of the data falls is indicated by the first quartile (Q1).

50% The second quartile (Q2), commonly called the median, represents the point at which 50% of the data falls below.

75% Remember that the third quartile (Q3) shows the value at which 75% of the data falls below.

Max The maximum value of the entire column.

Table 2 gives a statistical overview of a dataset that includes information about energy consumption, greenhouse gas savings, gasoline savings, and object identifiers, possibly in evaluating energy efficacy or environmental impact across different postal codes or regions. The median energy consumption in Colorado is around 10 kWh. The relatively small IQR indicates that most data is clustered around the median. A relatively small IQR indicates low variability and suggests that most energy consumption data points skew closely to the median value of 10 kWh. This clustering implies that most users have similar energy consumption patterns, likely reflecting standard usage patterns or consistent vehicle types. However, many outliers are data points that significantly exceed the other data. These outliers likely represent EVs with very high energy consumption, perhaps due to factors such as the large size of battery storage, as in Fig. 1.

The correlation heatmap in Fig. 2 shows how the features in the dataset are related to each other. Charging Time, Energy Consumption and GHG Savings are strongly connected, meaning longer charging times lead to higher energy use and more environmental benefits. Gasoline Savings is also closely linked to GHG Savings. On the other hand, features like Station Name, Address, and City have little to no impact on energy-related metrics. Temporal details like Start and End Times and encoded identifiers also show weak relationships. Overall, the heatmap helps identify the most important features for understanding energy use and environmental savings.

Data preprocessing

The Python programming language is frequently utilized in data preprocessing or data cleaning pipelines to ensure that date and time columns are appropriate for assessment or modelling. Converting these columns

Statistic	Zip_Postal_Code	Energy_kWh	GHG_Savings_kg	Gasoline_Savings_gallons	ObjectID	ObjectID2
count	148,136	148,136	148,136	148,136	148,136	148,136
mean	80,302.58	8.46	5.30	1.06	37,134	74,068
std	1.12	8.87	5.99	1.11	21,556	42,763
min	80,301	0.00	0.00	0.00	0	1
25%	80,302	2.80	1.59	0.35	18,517	37,035
50%	80,302	6.22	3.70	0.78	37,034	74,069
75%	80,303	11.39	6.92	1.43	55,550	111,102
max	80,305	92.69	65.81	11.63	77,941	148,136

Table 2. The statistical metrics for each column of the dataset.

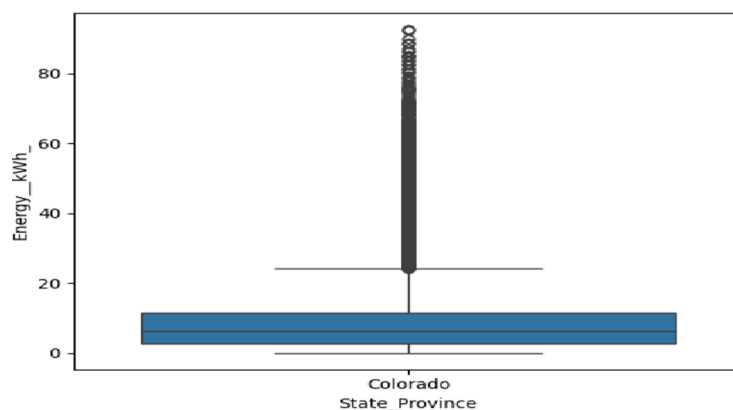


Fig. 1. Box Plot of energy consumption.

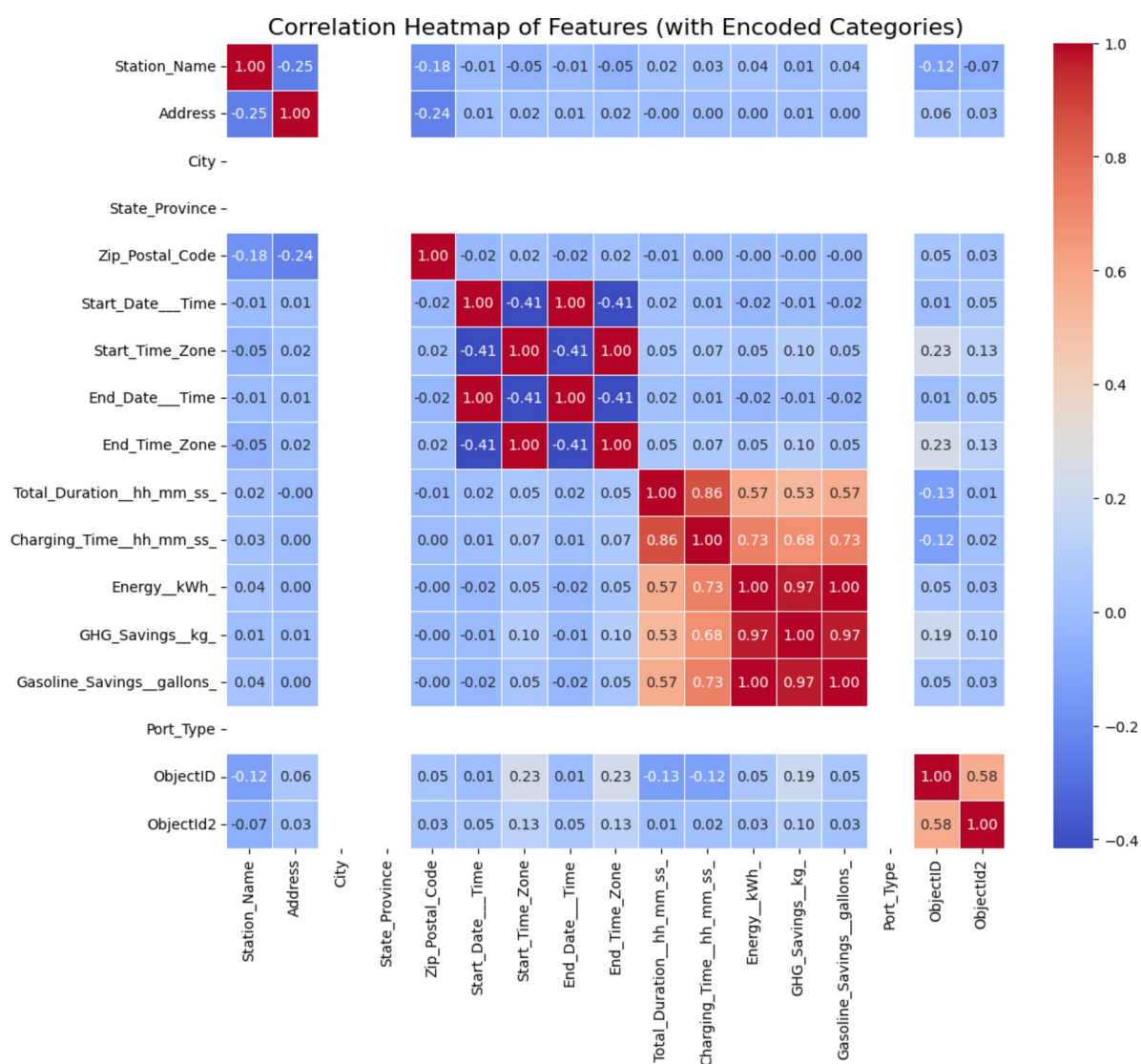


Fig. 2. Heatmap of the Colorado dataset.

to DateTime format makes it possible to conduct date and time-related operations, such as calculating time differences or grouping data by date and checking for null values. During the initial data preprocessing step, the dataset's start and end date columns were converted into a DateTime format. This procedure introduced several null values, 7820 entries, caused by erroneous date and time data in these columns. A strategy involving removing rows displaying these null values was implemented to rectify this anomaly and improve the dataset's integrity. After handling missing data, calculate the total time and time to charge, both in seconds. A method separates the temporal components from the preprocessing stage's start date and time entries. This method permitted accurately retrieving the hour, day of the week, and month, adding more temporal aspects to the information.

Defining features and target variable

To compare ML models in predicting energy consumption, quantified in kilowatt-hours (Energy kWh). A set of input features has been selected across all models. The information includes the charging duration, charging time, start time of the charging session, day of the week and month when the charging started. These features were chosen due to their influence on energy consumption rates during the charging process. The presence of these factors is based on the idea that they are important determinants of the amount of energy in kilowatt-hours. Total Duration and Charging Time capture session, specific details directly affecting energy usage. Temporal features such as Start Hour, Start Day of Week, and Start Month account for variations in energy demand influenced by time of day, weekly patterns, and seasonal effects. This ensures that each model is evaluated using the same criteria. This approach allows for an equitable and focused comparison of the predictive capabilities of each ML model regarding their accuracy in estimating Energy kWh.

Methodology

The general architectural diagram for energy prediction with different ML models is shown in Fig. 3. Data collection was done at the start, and then data preprocessing and exploratory data analysis were carried out. Feature extraction and feature selection are also part of this research, and according to the features selected, different algorithms were trained.

Model building

Different Machine Learning models such as Ridge Regression, Lasso Regression, Gradient Boosting, Support Vector Machine(SVM), K-Nearest Neighbors, Multi-Layer Perceptron and Extra Trees Regressor are utilized to predict future values based on historical data.

Ridge regression model

Ridge Regression, also called L2 regularization, is a method used to address multicollinearity in linear regression models. This includes introducing a penalty that is the same as the square of the size of the coefficients to the loss function. By imposing a penalty on the coefficient size, Ridge regression mitigates some issues associated with Ordinary Least Squares²². The ridge coefficients minimize the sum of squares of residuals with a penalty, as illustrated in Eq. 1.

$$J(\theta) = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p \theta_j^2 \quad (1)$$

Where

$J(\theta)$ represents the cost function which the Ridge Regression model aims to minimize.

y_i signifies the actual target value.

\hat{y}_i Signifies the predicted value by the model.

θ_j represents the coefficients of the model (excluding the intercept).

λ (lambda) is the regularization parameter that controls the shrinkage applied to the coefficients.

p is the number of features (parameters) in the model.

Lasso regression

Lasso Regression(Least Absolute Shrinkage and Selection Operator) is an alternative regularization technique that employs L1 regularization. This adds a penalty equal to the absolute value of the coefficients. Unlike other methods, the Lasso produces a linear model that estimates sparse coefficients. This attribute proves advantageous in specific scenarios as it tends to favour solutions with fewer non-zero coefficients, reducing reliance on many features. As a result, Lasso and its variations hold significant importance in compressed sensing. Under specific conditions, it can precisely retrieve the set of non-zero coefficients¹⁴. Mathematically, it involves a linear model with an additional regularization term. The objective function to minimize is represented in Eq. 2:

$$J(\theta) = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p |\theta_j| \quad (2)$$

Where

$J(\theta)$ is the cost function the Lasso Regression model tries to minimize.

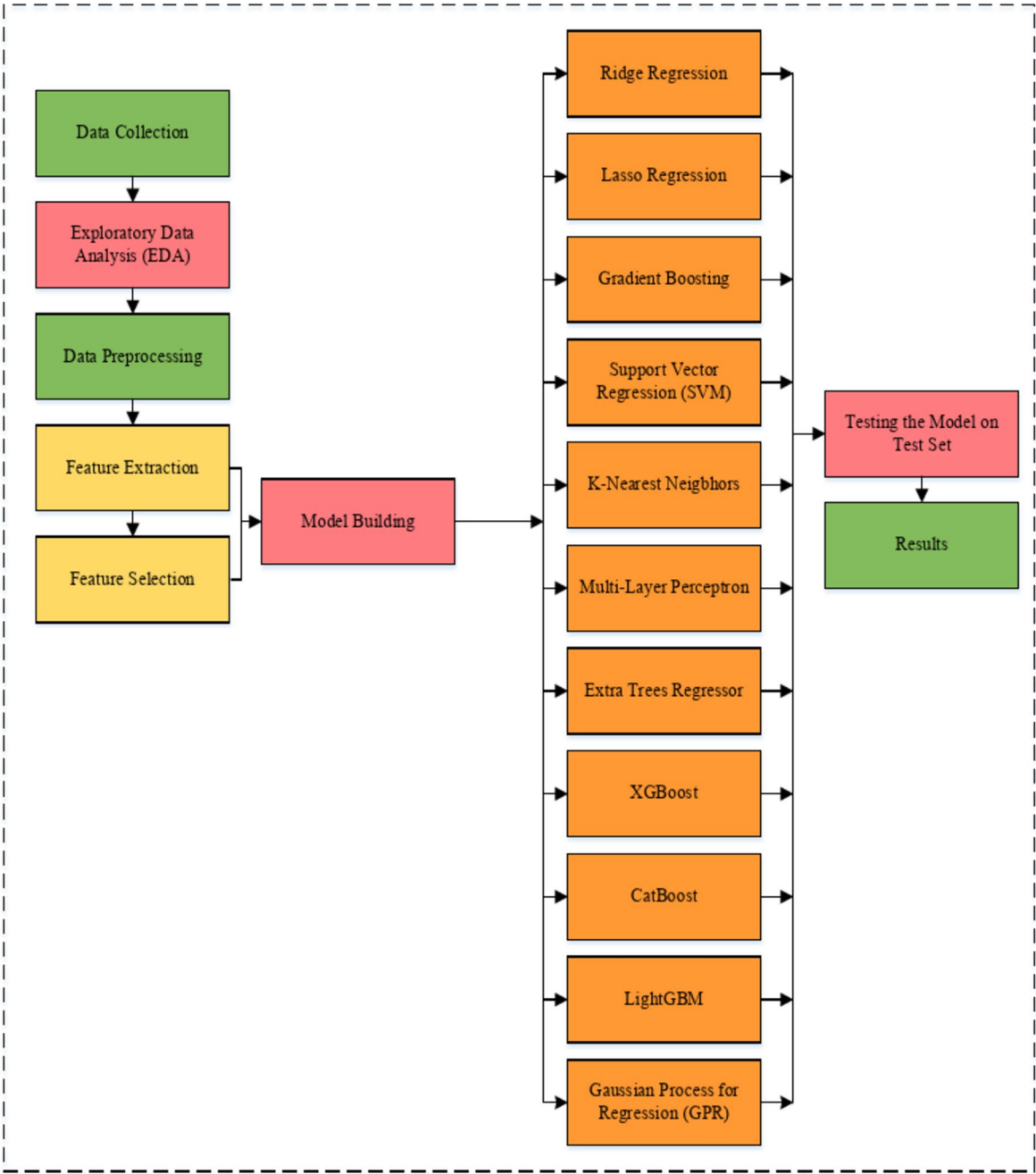


Fig. 3. General Architectural Diagram for Energy Prediction with Different ML Models.

y_i presents the actual target value.
 \hat{y}_i presents the predicted value by the model.
 θ_j represents the coefficients of the model (excluding the intercept).
 λ (lambda) is the regularization parameter that controls the shrinkage applied to the coefficients.
 p is the number of features (parameters) in the model.

Gradient boosting
Gradient Boosting is a powerful ensemble learning technique combining multiple weak learners, typically decision trees, to create a strong predictive model. This is achieved by creating models one after another, with each subsequent model rectifying the mistakes of the one before it¹³. This process is mathematically represented in Eq. 3.

$$F_m(x) = F_{m-1}(x) + h_m(x) \tag{3}$$

Where

$F_m(x)$ is the model at iteration m .

$F_{m-1}(x)$ is the model from the previous iteration $m-1$.

$h_m(x)$ is the weak learner (or weak model) added at iteration m to correct the errors of the previous model.

Support vector machine(SVM)

Support Vector Machine (SVM) is a supervised learning model for classification and regression tasks. Its primary function is identifying a hyperplane that effectively separates data points into distinct classes while maximizing the margin. Support Vector Regression (SVR) extends these principles to regression problems. Both SVM and SVR models depend on a subset of the training data, as their cost functions ignore points beyond the margin or those with predictions close to their targets^{23,24}. This can be mathematically expressed in Eq. 4.

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \max(0, 1 - y_i(w^T x_i + b)) \quad (4)$$

Where

w is the weight vector.

x_i is the feature vector for the i -th training example.

y_i is the label for the i -th training example, which is typically $y_i \in \{-1, +1\}$ for binary classification.

b is the bias term (intercept).

The value of C is a regularization parameter that determines the balance between maximizing the margin and minimizing the classification error.

K-Nearest neighbors

KNN is a simple, non-parametric algorithm used for regression and classification. It determines the majority class or average among the k -nearest data points to a query point. The label is calculated for regression as the mean of the nearest neighbors' labels. The value of k is an integer, and uniform weights are typically used, giving equal influence to all nearby points. Weights can be adjusted, with options like 'uniform' for equal weights and 'distance' for weights inversely proportional to distance¹⁵. Custom weight functions can also be defined. The mathematical form is as in Eq. 5.

$$\hat{y} = \frac{1}{k} \sum_{i \in N_k} y_i \quad (5)$$

Where.

\hat{y} The predicted value for the query point.

k : The number of nearest neighbors considered.

N_k : The set of k nearest neighbors to the query point.

y_i The actual target values (labels) of the i -th nearest neighbor.

Multi-Layer perceptron

MLP falls under the feedforward artificial neural networks category and comprises at least three layers of nodes: an input layer, a hidden layer, and an output layer. It is a supervised learning algorithm that learns a function: $R^m \rightarrow R^o$ by training on a dataset, where m is the number of dimensions for input and o is the number of dimensions for output. Given a set of features $X = x_1, x_2, \dots, x_m$ and a target y , it can learn a non-linear function approximator for either classification or regression. Between the input and output layers, neural networks differ from logistic regression in that one or more non-linear, hidden layers can exist²⁵. Mathematically, the linear combination is shown in Eq. 6, and the activation function is in Eq. 7.

$$z = W^T x + b \quad (6)$$

Where.

W^T This is the transposition of the weight vector W . Each weight W_j in W corresponds to the weight or strength of the connection between the input feature x_j and the neuron.

x The input vector represents the features or data fed into the neuron.

b The bias term allows the model to fit the data even when all input features are zero. It shifts the activation function to the left or right.

z The linear combination of the inputs, weights, and bias. This is also known as the “pre-activation” value.

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

Where.

$\sigma(z)$ This represents the activation function applied to the linear combination z .

a The output of the neuron after applying the activation function.

The activation function $\sigma(z)$ leads non-linearity into the model, allowing the network to learn complex patterns.

Extra trees regressor

Extra Trees, also called Extremely Randomized Trees, are ensemble learning method that predicts from multiple decision trees. Unlike conventional decision trees, extra trees utilize random splits during tree construction, leading to an improved reduction in variance¹². Mathematically, it is denoted as in Eq. 8.

$$\hat{y} = \frac{1}{M} \sum_{m=1}^M T_m(x) \quad (8)$$

Where

\hat{y} is the predicted value for the feature vector x .

M is the total number of trees in the ensemble.

$T_m(x)$ is a prediction made by the m -th tree in the ensemble for the input x .

Light GBM

LightGBM, or Light Gradient Boosting Machine, is an efficient gradient-boosting framework for speed and scalability. It uses histogram-based learning, which groups continuous features into discrete bins, reducing computational complexity and memory usage¹⁵. The iterative model-building process is represented in Eq. 9.

$$F_m(x) = F_{m-1}(x) + h_m(x) \quad (9)$$

Where $F_m(x)$ is the model at iteration m , $F_{m-1}(x)$ is the previous model, and $h_m(x)$ is the weak learner added at iteration m . LightGBM is well-suited for large datasets with high feature dimensions and offers excellent performance with minimal tuning.

Gaussian process for regression

Gaussian Process for Regression (GPR) is a non-parametric Bayesian approach that models the underlying data distribution as a Gaussian process. It provides predictions and confidence intervals by learning a probability distribution over possible functions²⁶. The prediction at any point is given by Eq. 10.

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \quad (10)$$

Where \mathcal{GP} denotes a Gaussian process, which is a collection of random variables, any finite number of which follow a joint Gaussian distribution $m(x)$ is the mean function, and $k(x, x')$ is the kernel function defining the covariance between points x and x' . GPR is particularly effective for small to medium-sized datasets where uncertainty quantification and smooth predictions are essential.

XGBoost

XGBoost, or Extreme Gradient Boosting, is a highly efficient and scalable implementation of gradient boosting. It incorporates advanced features like regularization, parallel processing, and tree pruning, making it faster and more accurate than traditional gradient boosting methods¹⁴. The iterative process of building the model can be expressed as in Eq. 11.

$$F_m(x) = F_{m-1}(x) + h_m(x) \quad (11)$$

Where $F_m(x)$ is the model at iteration m , $F_{m-1}(x)$ is the previous model, and $h_m(x)$ is the weak learner added at iteration m to reduce residual errors. XGBoost's ability to handle missing data and its efficient memory usage make it a popular choice for regression and classification tasks.

CatBoost

CatBoost is a gradient-boosting algorithm that handles categorical data efficiently without requiring extensive preprocessing. It uses ordered boosting, a technique that reduces overfitting by calculating leaf values based on permutations of the dataset. The model iteratively improves predictions by adding decision trees and correcting errors in previous iterations²⁷. This process can be expressed by Eq. 12.

$$F_m(x) = F_{m-1}(x) + h_m(x) \quad (12)$$

Where $F_m(x)$ is the model at iteration m , $F_{m-1}(x)$ is the previous model, and $h_m(x)$ is the weak learner added at iteration m to reduce errors. CatBoost excels in achieving high accuracy with minimal hyperparameter tuning and is particularly effective for datasets with a mix of numerical and categorical features.

Training and testing the model

Data split is important for evaluating model performance. The training and test sets are the two subsets into which the dataset is divided. The model trains from the training set, which helps the algorithm understand the data's structures, interactions, and patterns. By using the training data, the model can adjust its parameters to reduce mistakes and improve accuracy. Usually between 70% and 80% of the total dataset, the training set varies depending on the particular use case and data availability. In this case, 80% of all algorithms comprise training data. The model's performance is evaluated using the test set following training. Usually comprising 20–30% of the entire dataset, 20% is the baseline for all algorithms. This includes data that the model was not exposed to during training. It is possible to correctly predict the model performance by evaluating it using a test set.

Evaluation metrics

Examining the efficacy of ML models involves utilizing various metrics to evaluate how closely the model's predictions match the actual values. Multiple commonly used evaluation metrics for regression models were utilized to evaluate each model.

Mean absolute error (MAE)

The Mean Absolute Error (MAE) measures the average size of errors in a set of predictions, irrespective of their direction. It calculates the average absolute differences between the predicted and actual values. MAE is a metric for assessing the average disparity between predicted and actual values. Its mathematical representation is articulated in Eq. 13.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (13)$$

Where.

n The quantity/number of data points (observations).

y_i The actual value for the i -th observation.

\hat{y}_i The predicted value for the i -th observation.

$|y_i - \hat{y}_i|$ The absolute error for the i -th observation.

The MAE is a simple metric that interprets the average error. It is especially valuable when you must assess the precision of predictions using the same unit as the target variable. However, it does not apply heavier penalties to larger errors than smaller ones.

Mean squared error (MSE)

The Mean Squared Error (MSE) calculates the average of the squared errors or variances – in other words, the gap between the estimate and the estimated value. MSE gives more weight to larger errors because it involves squaring the differences. MSE measures the average squared difference between predicted and actual values. Mathematically, it can be described using Eq. 14.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (14)$$

Where.

n The number of data points (observations).

y_i The actual value for the i -th observation.

\hat{y}_i The predicted value for the i -th observation.

$y_i - \hat{y}_i$ The squared error for the i -th observation.

MSE is widely used because it emphasizes larger errors more than smaller ones, making it more sensitive to outliers. It is commonly used as a loss function in regression models. The units of MSE are the square of the original data units, which can sometimes make interpretation less intuitive.

R-squared (R^2)

The coefficient of determination, commonly known as R-squared, is a statistical metric that indicates the proportion of the variance in the dependent variable explained by the independent variables in a regression model. This can be expressed mathematically using Eq. 15.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (15)$$

Where.

n The number of data points (observations).

y_i The actual value for the i -th observation.

\hat{y}_i The predicted value for the i -th observation.

\bar{y}_i The mean of the actual values y_i .

$\sum_{i=1}^n (y_i - \hat{y}_i)^2$ The sum of squared residuals (errors), also known as the **Residual Sum of Squares (RSS)**.

$\sum_{i=1}^n (y_i - \bar{y}_i)^2$ The total sum of squares is also known as the **Total Sum of Squares (TSS)**.

R-squared values range from 0 to 1. A value of 1 indicates that the model perfectly explains the variance, while 0 indicates that the model does not describe any of the variance. However, R-squared can sometimes be misleading, particularly with non-linear models or when comparing models with different numbers of predictors.

Root mean squared error (RMSE)

RMSE measures the average magnitude of errors between predicted and observed values in a dataset. It is the square root of the average squared differences between predictions and actual observations, indicating how far predictions deviate from actual values on average. The mathematical equation is as (16)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (16)$$

Where.

n The number of data points (observations).

y_i The actual value for the i -th observation.

\hat{y}_i The predicted value for the i -th observation.

Normalised root mean squared error (NRMSE)

NRMSE normalizes RMSE to make it easier to interpret across datasets with different ranges. It is often normalized by the range, mean, or standard deviation of the observed values, converting RMSE into a dimensionless quantity. This can be expressed mathematically using Eq. 17.

$$NRMSE = \frac{RMSE}{\max(y) - \min(y)} \quad (17)$$

Other normalisation forms can include dividing by the mean or standard deviation of observed values.

Results and discussion

This research used seven distinct machine-learning algorithms to predict energy usage by analyzing historical data from Colorado. The efficacy of the models was assessed using MAE, MSE, R^2 , RMSE and NRMSE values, as shown in Table 3, offering valuable insights into their accuracy and dependability. The performance of the ML models in predicting energy consumption was evaluated through scatter plots comparing actual energy consumption to predicted values, as well as time series plots illustrating the models' ability to track actual energy consumption over time.

The **Extra Trees Regressor (ETR)** outperformed all other models significantly, with an MAE of 0.5888, MSE of 3.2683, R^2 of 0.9592, RMSE of 1.8078 and NRMSE of 0.020. These results showed that ETR accurately predicted energy consumption, capturing 95.92% of the variance in the data. The ETRs handle large datasets and model complex relationships, contributing significantly to their outstanding performance. The time series plot for the ETR in Fig. 4(a) demonstrates that the predicted values closely align with the actual energy consumption trends, showing minimal deviation across most data points. This alignment indicates that the model accurately captures the underlying patterns in the data. Figure 4(b) shows a scatter plot indicating a strong correlation between actual and predicted values. The data points are closely clustered along the diagonal line, confirming the model's effectiveness.

K-Nearest Neighbors (KNN) and Gradient Boosting performed well compared to other models. KNN had an MAE of 1.8517, an MSE of 9.6826, R^2 of 0.8792, RMSE of 3.111 and NRMSE of 0.034, while Gradient Boosting had an MAE of 2.0306, MSE of 9.3121, R^2 of 0.8838, RMSE of 3.052 and NRMSE of 0.033. These models displayed strong predictive capabilities but were slightly less resilient than the ETR. The time series plot for KNN in Fig. 5(a) showed that the predicted values closely matched the actual energy consumption patterns.

S/No	Model	MAE	MSE	R^2	RMSE	NRMSE
01	Extra Tree	0.5888	3.2683	0.9592	1.8078	0.020
02	KNN	1.8517	9.6826	0.8792	3.111	0.034
03	Gradient Boosting	2.0306	9.3121	0.8838	3.052	0.033
04	SVR	1.9984	12.4641	0.844	3.530	0.039
05	Ridge Regression	2.2076	13.2877	0.8342	3.645	0.040
06	Lasso Regression	2.2057	13.2980	0.8340	3.646	0.040
07	MLP	3.7244	48.8066	0.3909	6.986	0.077
08	XGBoost	1.868	7.929	0.901	2.8159	0.031
09	CatBoost	1.915	8.415	0.894	2.901	0.032
10	LightGBM	1.959	9.068	0.886	3.011	0.0334
11	Gaussian Process Regression	4.872	80.0259	0.202	8.945	0.143

Table 3. Results of evaluation metrics for all eleven models.

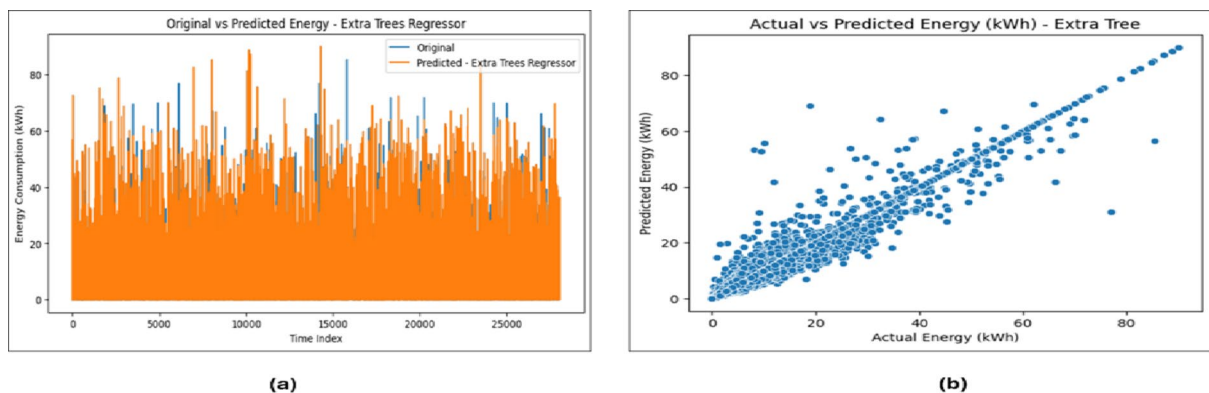


Fig. 4. (a) Original vs. Predicted Energy with time index by Extra Trees Regressor. (b) Actual vs. Predicted Energy by Extra Trees.

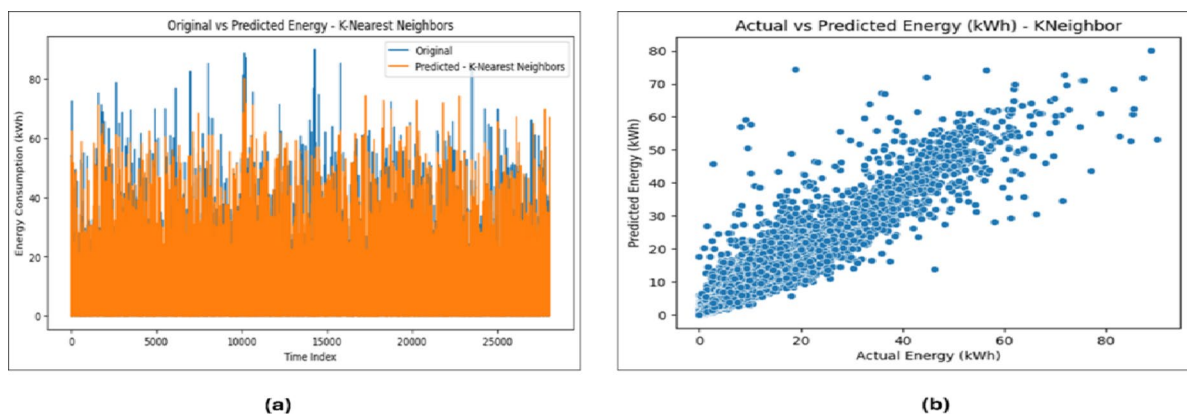


Fig. 5. (a) Original vs. Predicted Energy with time index by K-Nearest Nighbors. (b) Actual vs. Predicted Energy by K-Nearest Nighbors.

However, there were more noticeable differences between the ET and GB models. The scatter plot in Fig. 5(b) indicates a robust clustering around the diagonal line, which is slightly more spread out than GB. This suggests good performance with minor variation. The Gradient Boosting time series plot in Fig. 6(a) indicated a close alignment with actual values despite slightly more deviations than the Extra Trees model, capturing most peaks and troughs of the energy consumption data. The scatter plot shown in Fig. 6(b) displayed a strong correlation between the actual and predicted values, with slightly more dispersion than the Extra Trees Regressor, indicating minor prediction errors. The Support Vector Regression (SVR) model demonstrated satisfactory performance with a mean absolute error (MAE) of 1.9984, mean squared error (MSE) of 12.4641, R^2 of 0.8444, RMSE of

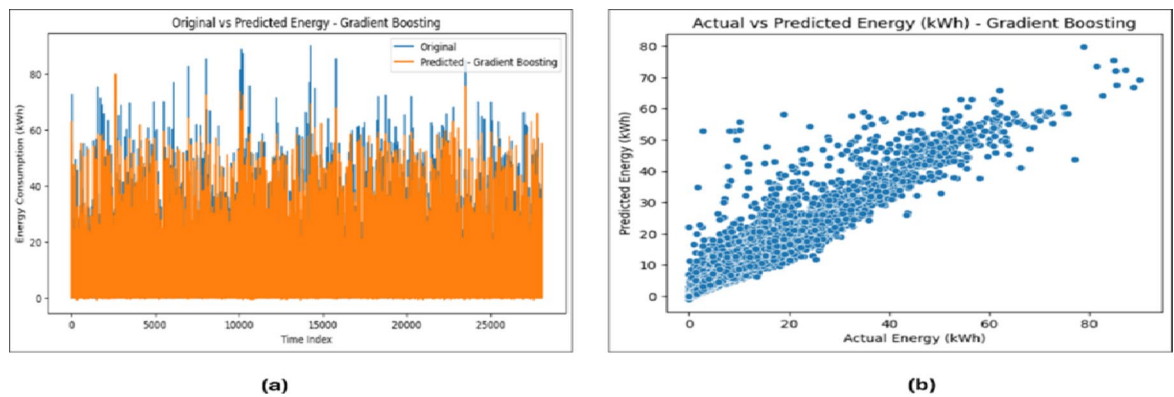


Fig. 6. (a) Original vs. Predicted Energy with time index by Gradient Boosting. (b) Actual vs. Predicted Energy by Gradient Boosting.

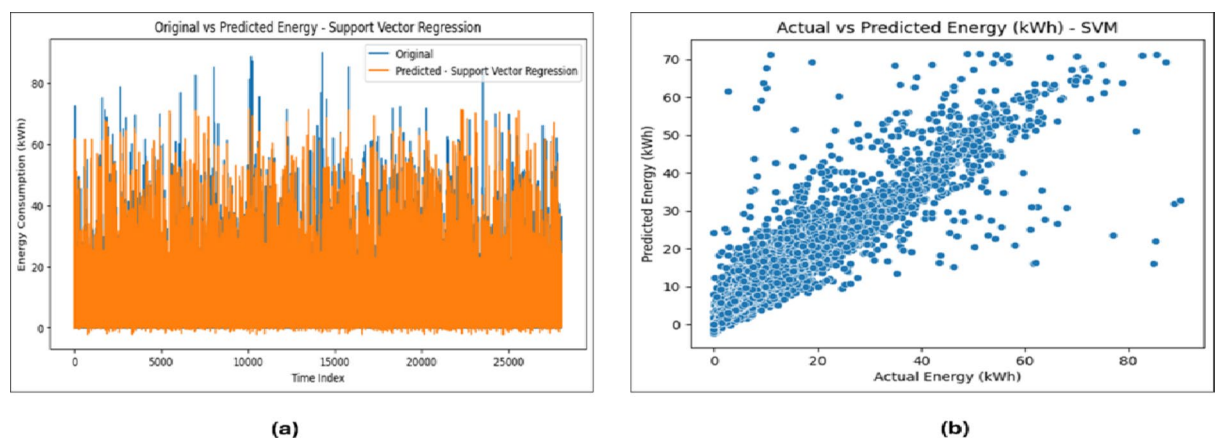


Fig. 7. (a) Original vs. Predicted Energy with time index by SVM. (b) Actual vs. Predicted Energy by SVM.

3.530 and NRMSE of 0.039. While SVR performed well, it was marginally less effective than KNN and Gradient Boosting in this scenario. The time series plot in Fig. 7(a) indicates that the model closely follows the actual energy consumption, albeit with slightly more variability than the top-performing models. Furthermore, the scatter plot shown in Fig. 7(b) reveals a strong linear relationship, although with more dispersion than the leading models, suggesting that SVR is effective but may not capture the complexity of the data as effectively.

The Ridge and Lasso Regression results were similar, with MAE values around 2.2076 and 2.2057, MSE values of 13.2877 and 13.2980, R^2 values of 0.8342 and 0.8340, RMSE values of 3.645 and 3.646 and NRMSE values of 0.040 and 0.040 respectively. However, these models were less effective in capturing the dataset's complexities than non-linear models, resulting in lower performance. The line graphs in Figs. 8(a) and 9(a) for Lasso and Ridge Regression showed noticeable problems at higher energy consumption levels, revealing the limitations of the models. The scatter plots in Figs. 8(b) and 9(b) also demonstrated that the models struggled to accurately predict high energy consumption values, with data points widely spread out for extreme values.

The Multi-Layer Perceptron (MLP) performed poorly, with an MAE of 3.7244, MSE of 48.8066, R^2 of 0.3909, RMSE of 6.986 and NRMSE of 0.077. The suboptimal results of the MLP could be attributed to its hyperparameter sensitivity and the requirement for extensive fine-tuning. Figure 10(a) shows that the time series plot reveals substantial variations from actual energy usage, particularly at the extremes, indicating the model's difficulties in grasping underlying patterns. Furthermore, the scatter plot in Fig. 10(b) demonstrates a considerable spread of data points, especially at higher consumption levels, underscoring the model's inability to capture intrinsic data patterns effectively.

XGBoost, CatBoost, LightGBM, and Gaussian Process Regression (GPR) were evaluated as part of the study, with mixed results compared to the top-performing models like Extra Trees and KNN. XGBoost performed well, achieving a Mean Absolute Error (MAE) of 1.868, a Mean Squared Error (MSE) of 7.929, an R-squared (R^2) of 0.901, a Root Mean Squared Error (RMSE) of 2.8159, and a Normalized Root Mean Squared Error (NRMSE) of 0.031, showcasing its capability to model the data effectively. The time series plot shown in Fig. 11(a) for XGBoost indicated a close alignment with actual energy consumption values, capturing most peaks and troughs with minor deviations. The scatter plot in Fig. 11(b) indicates a robust clustering around the diagonal line, which is slightly more spread out than GB. CatBoost followed closely, with an MAE of 1.915, MSE of 8.415, R^2 of 0.894,

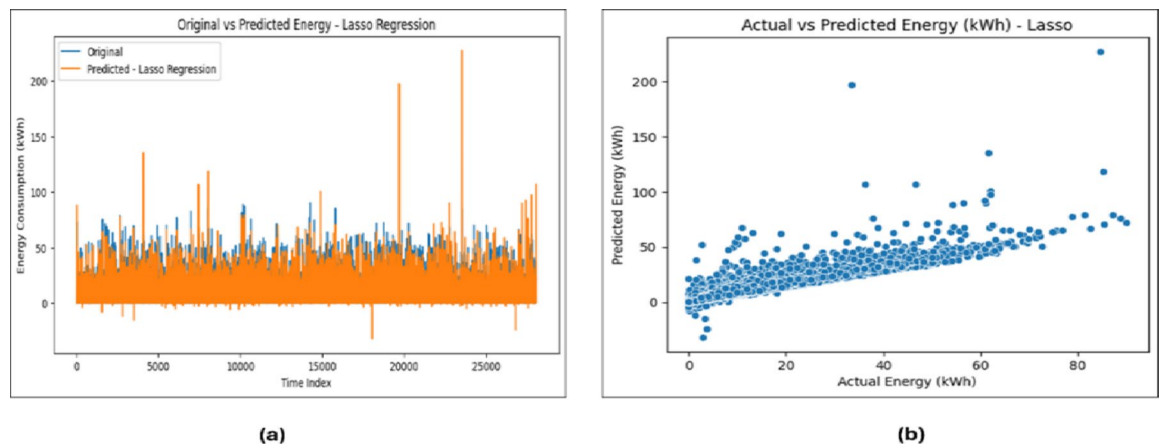


Fig. 8. (a) Original vs. Predicted Energy with time index by Lasso Regression. (b) Actual vs. Predicted Energy by Lasso Regression.

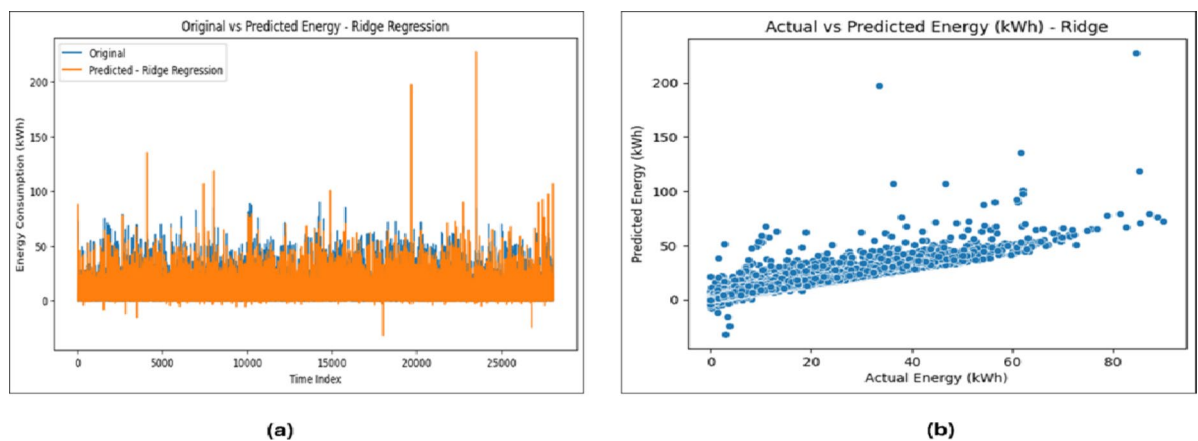


Fig. 9. (a) Original vs. Predicted Energy with time index Ridge Regression. (b) Actual vs. Predicted Energy by Ridge Regression.

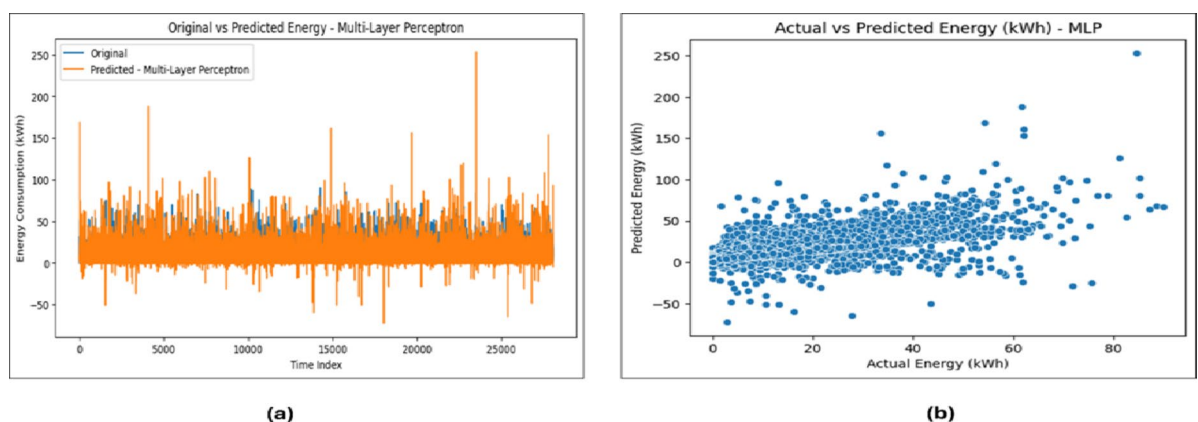


Fig. 10. (a) Original vs. Predicted Energy with time index MLP. (b) Actual vs. Predicted Energy by MLP.

RMSE of 2.901, and NRMSE of 0.032. While CatBoost performed slightly worse than XGBoost, its ability to handle categorical data effectively contributed to its relatively strong performance. LightGBM achieved slightly higher errors, with an MAE of 1.959, MSE of 9.068, R^2 of 0.886, RMSE of 3.011, and NRMSE of 0.0334, likely due to differences in its tree-splitting strategy. The time series plot in Fig. 12(a) indicates that the model closely

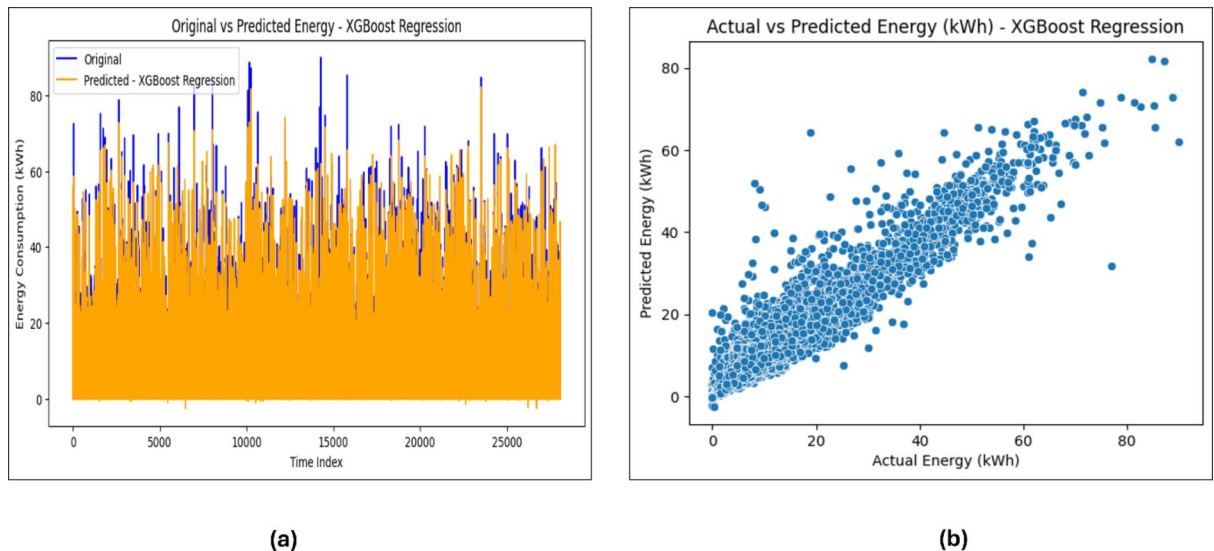


Fig. 11. (a) Original vs. Predicted Energy with time index XGBoost. (b) Actual vs. Predicted Energy by XGBoost.

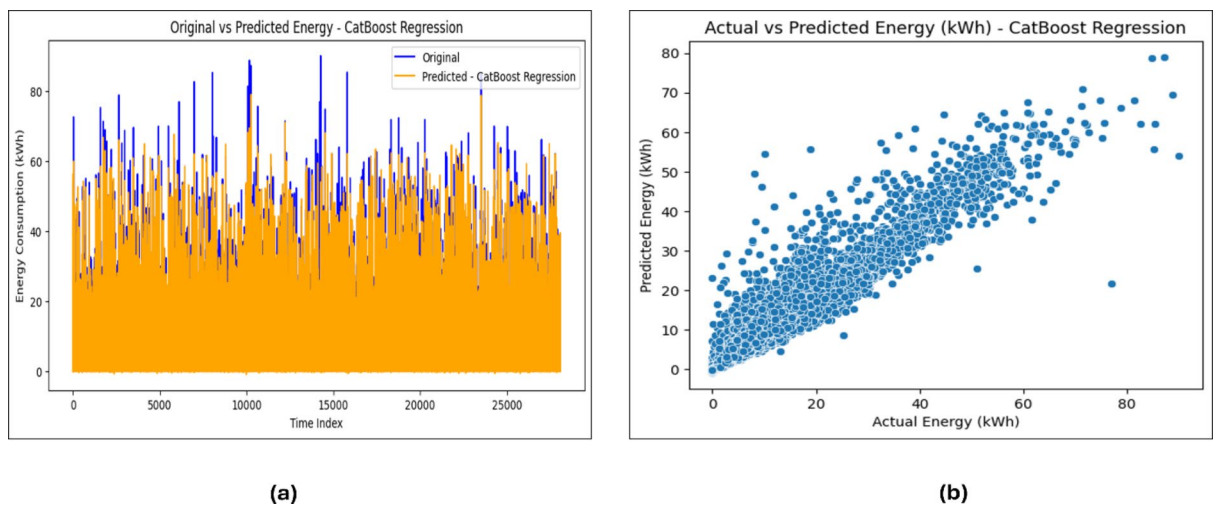


Fig. 12. (a) Original vs. Predicted Energy with time index CatBoost. (b) Actual vs. Predicted Energy by CatBoost.

follows the actual energy consumption, albeit with slightly more variability than the top-performing models. Furthermore, the scatter plot shown in Fig. 12(b) reveals a strong linear relationship, although with more dispersion than the leading models, suggesting that CatBoost is effective but may not capture the complexity of the data as effectively. Despite being efficient, LightGBM showed slightly less resilience in capturing complex patterns than XGBoost, CatBoost, KNN, SVR and ETR. The time series plot in Fig. 13(a) shows slightly larger deviations in the time series, indicating slightly lower accuracy compared to the XGBoost and CatBoost. The scatter plot in Fig. 13(b) reveals a correlation between actual and predicted energy consumption, suggesting that the LightGBM model generally captures the trend of energy usage. However, some points deviate from the ideal diagonal line, indicating instances where the model underpredicts energy consumption. In contrast, GPR performed the worst among the evaluated models, with an MAE of 4.872, MSE of 80.0259, R^2 of 0.202, RMSE of 8.945, and NRMSE of 0.143. The time series plot as shown in Fig. 14(a) for GPR, displayed significant deviations from the actual values, and the scatter plot as in Fig. 14(b) revealed a dispersed clustering, indicating that GPR struggled with the complexity of the dataset, likely due to its computational limitations and inability to handle larger datasets effectively.

The feature importance analysis, as shown in Fig. 15, visualized through permutation feature importance, highlights Charging_Time_seconds as the most influential predictor of energy consumption among the features selected. This suggests that the time spent charging plays a critical role in determining energy usage, aligning with the expected impact of direct charging duration on consumption. Total_Duration_seconds also

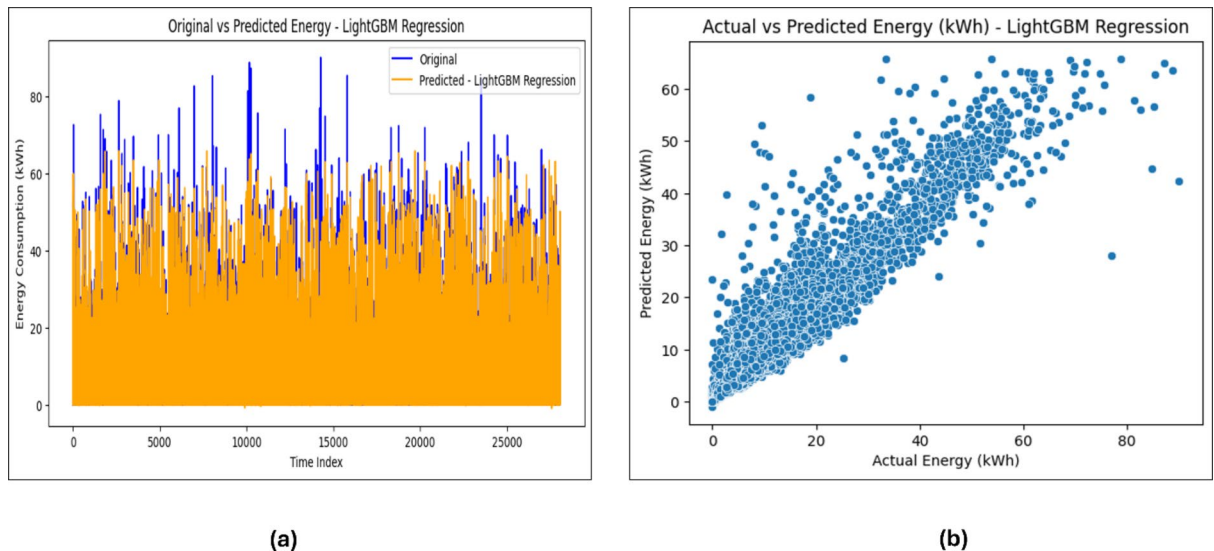


Fig. 13. (a)Original vs. Predicted Energy with time index LightGBM. (b) Actual vs. Predicted Energy by LightGBM.

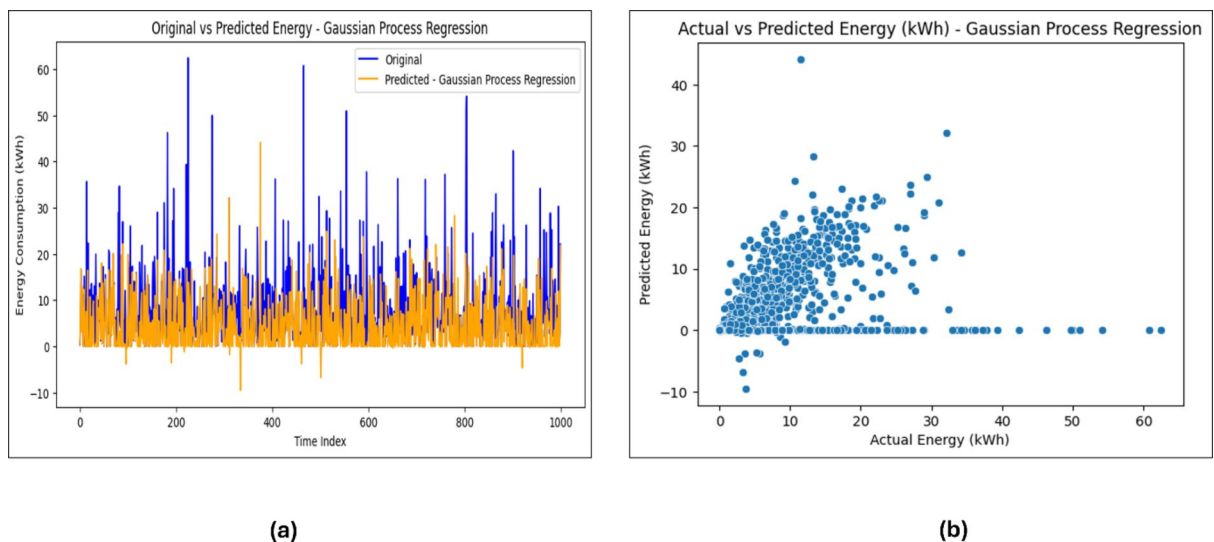


Fig. 14. (a)Original vs. Predicted Energy with time index GPR. (b) Actual vs. Predicted Energy by GPR.

contributes, though to a lesser extent, reflecting the influence of overall session time. Temporal features such as Start_Hour, Start_DayOfWeek, and Start_Month exhibit relatively low importance, indicating that while time-based patterns might exist, they are less predictive than session-specific characteristics. This analysis underscores the significance of session duration metrics in accurately predicting energy consumption in electric vehicles.

The Extra Trees Regressor demonstrated superior effectiveness in capturing the intricate, non-linear patterns inherent in the energy consumption data. This can be attributed to its ensemble structure, aggregating numerous decision trees to form a robust predictive model, reducing variance through random splits at each node. This approach allowed the model to manage complex relationships within the data, making it particularly well-suited for scenarios involving high-dimensional, heterogeneous features. The K-Nearest Neighbors (KNN) model also performed admirably, particularly in recognising and leveraging localized data patterns. Its non-parametric nature enabled it to classify data points based on their proximity to clusters with similar characteristics, which proved valuable for this dataset. However, KNN's heavy reliance on local data structures makes it less adaptable to extreme or sparse data points, potentially affecting its performance in datasets with significant variability.

Gradient Boosting likewise yielded strong predictive results, benefiting from its iterative, additive training process where each new model focuses on correcting the errors of previous iterations. This sequential refinement makes Gradient Boosting particularly adept at capturing data complexities. However, the model's dependency on parameter tuning for optimal performance suggests that careful optimization is required, especially to manage

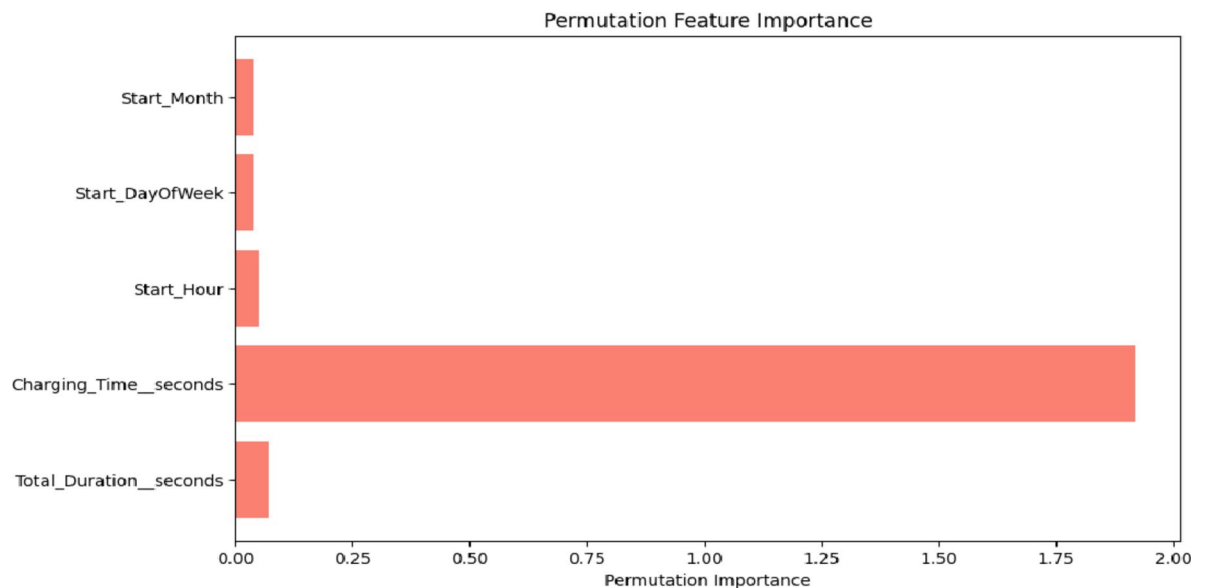


Fig. 15. Permutation Feature Importance for Predicting Energy Consumption.

outliers effectively and enhance its generalization to unseen data. In comparison, Support Vector Regression (SVR) displayed moderate performance, constrained by its assumptions of linear separability and sensitivity to outliers. While SVR is effective for certain types of continuous data, its rigidity in handling highly non-linear patterns limits its applicability. Overall, the ensemble and non-linear models, particularly Extra Trees, proved more adaptable and accurate for capturing the nuanced patterns in EV energy consumption data, highlighting the value of these approaches in this domain.

The implementation of the models proceeded smoothly without obstacles, indicating that the chosen algorithms were well-suited for the task at hand. Nevertheless, the underwhelming performance of the MLP suggests that future studies could explore more sophisticated neural network architectures or improved optimization techniques to enhance accuracy. These results establish a robust foundation for continued investigation and application of machine learning approaches in energy prediction. Such advancements can potentially improve power grid management and contribute to developing more sustainable energy systems.

XGBoost demonstrated predictive capabilities in capturing intricate patterns within the energy consumption data. Its ensemble structure and regularization techniques contributed to reducing overfitting and enhancing generalization, making it highly effective in modeling complex relationships. Similarly, CatBoost excelled due to its ability to handle categorical variables efficiently and its gradient boosting framework, which allowed it to adapt well to the dataset's non-linearities. LightGBM, with its innovative leaf-wise tree growth strategy, provided a balance between speed and accuracy, enabling it to process large datasets efficiently while maintaining reliable performance. However, Gaussian Process Regression (GPR) struggled to handle the complexity of the dataset. Its reliance on kernel functions for pattern recognition and the computational limitations of scaling with larger datasets limited its ability to effectively capture the variability and intricacies of energy consumption data.

Conclusions

The effectiveness of various machine learning algorithms in predicting energy consumption in Colorado using historical data was examined. Seven models were assessed, ranking them based on specific metrics such as MAE, MSE, and R^2 , along with the accuracy of their predictions compared to actual data. The Extra Trees Regressor algorithm emerged as the most precise, demonstrating minimal error in energy use predictions and effectively handling significant data variations. While the Gradient Boosting and K-Nearest Neighbors models also showed strong performance, they were slightly less reliable. This investigation underscores the significance of selecting appropriate algorithms for energy demand prediction, which is particularly relevant for managing power grids as electric vehicle adoption increases. Ensemble methods and non-linear models, which excel at handling complex time-series data, have shown remarkable effectiveness. The limitation of the work includes a focus on data exclusively from Colorado, which may limit the findings' applicability to other regions with different climates and driving behaviours. While the Extra Trees Regressor demonstrated high accuracy, its computational demands could hinder scalability for real-time applications. The study primarily focused on accuracy metrics, neglecting model interpretability and adaptability. Future research should explore advanced methodologies such as deep learning and hybrid models, broaden the dataset to include diverse demographics and weather conditions and utilize time-series analysis for better energy demand forecasting. Tailoring hyperparameter optimization and incorporating various machine learning models, including transformers, stacking and ensemble techniques, could improve predictive accuracy and model resilience as electric vehicle usage increases.

Software and packages used

The Scipy package in Python provides a range of statistical techniques, such as hypothesis testing, probability distributions, and correlation analysis. Moreover, statistical modelling, data analysis, manipulation, and date and time handling are performed using the Statsmodels, NumPy, and Pandas packages and the DateTime package in Python. This data preprocessing and model implementation was done on VS Code software.

Data availability

The Colorado datasets analysed during the current study are available in the [Colorado dataset] repository, City of Boulder Open Data (boulder.colorado.gov), which is used for energy consumption.

Received: 12 September 2024; Accepted: 18 March 2025

Published online: 08 May 2025

References

- Chen, Y. et al. *Artificial Intelligence/Machine Learning Technology in Power System Applications* (No. PNNL-35735). Pacific Northwest National Laboratory (PNNL), Richland, WA (United States). (2024).
- Steinborn, F. *Time series analysis and prediction of electricity load demand with machine learning techniques* (Doctoral dissertation). (2022).
- Kim, Y. & Kim, S. Forecasting charging demand of electric vehicles using time-series models. *Energies*, **14**(5), p.1487. (2021).
- Zhu, J. et al. Electric vehicle charging load forecasting: A comparative study of deep learning approaches. *Energies*, **12**(14), p.2692. (2019).
- Akshay, K. C., Grace, G. H., Gunasekaran, K. & Samikannu, R. Power consumption prediction for electric vehicle charging stations and forecasting income. *Scientific Reports*, **14**(1), p.6497. (2024).
- Shahriar, S., Al-Ali, A. R., Osman, A. H., Dhou, S. & Nijim, M. Prediction of EV charging behavior using machine learning. *Ieee Access*, **9**, 111576–111586 (2021).
- Wang, Y., Tan, H., Wu, Y. & Peng, J. Hybrid electric vehicle energy management with computer vision and deep reinforcement learning. *IEEE Trans. Industr. Inf.* **17** (6), 3857–3868 (2020).
- Thorgeirsson, A. T., Scheubner, S., Fünfgeld, S. & Gauterin, F. Probabilistic prediction of energy demand and driving range for electric vehicles with federated learning. *IEEE Open Journal of Vehicular Technology*, **2**, pp.151–161. (2021).
- Elkamel, M., Schleider, L., Pasilio, E. L., Diabat, A. & Zheng, Q. P. Long-term electricity demand prediction via socioeconomic factors—A machine learning approach with Florida as a case study. *Energies*, **13**(15), p.3996. (2020).
- Abro, S. A., Hua, L. G., Laghari, J. A., Bhayo, M. A. & Memon, A. A. Machine learning-based electricity theft detection using support vector machines. *International Journal of Electrical & Computer Engineering* (2088–8708), **14**(2). (2024).
- Malakouti, S. M. Estimating the output power and wind speed with ML methods: a case study in Texas. *Case Studies in Chemical and Environmental Engineering*, **7**, p.100324. (2023).
- Malakouti, S. M., Menhaj, M. B. & Suratgar, A. A. The usage of 10-fold cross-validation and grid search to enhance ML methods performance in solar farm power generation prediction. *Cleaner Engineering and Technology*, **15**, p.100664. (2023).
- Malakouti, S. M. et al. Advanced techniques for wind energy production forecasting: Leveraging multi-layer Perceptron + Bayesian optimization, ensemble learning, and CNN-LSTM models. *Case Studies in Chemical and Environmental Engineering*, **10**, p.100881. (2024).
- Malakouti, S. M. Improving the prediction of wind speed and power production of SCADA system with ensemble method and 10-fold cross-validation. *Case Studies in Chemical and Environmental Engineering*, **8**, p.100351. (2023).
- Malakouti, S. M. Babysitting hyperparameter optimization and 10-fold-cross-validation to enhance the performance of ML methods in Predicting Wind Speed and Energy Generation. *Intelligent Systems with Applications*, **19**, p.200248. (2023).
- Malakouti, S. M. Prediction of wind speed and power with LightGBM and grid search: case study based on scada system in Turkey. *Int. J. Energy Prod. Manage.* **2023 Vol. 8** (1), 35–40 (2023).
- Mastoi, M. S. et al. An in-depth analysis of electric vehicle charging station infrastructure, policy implications, and future trends. *Energy Reports*, **8**, pp.11504–11529. (2022).
- Wu, C., Jiang, S., Gao, S., Liu, Y. & Han, H. Charging demand forecasting of electric vehicles considering uncertainties in a microgrid. *Energy*, **247**, p.123475. (2022).
- Li, H. et al. July. Review of load forecasting methods for electric vehicle charging station. In *2022 IEEE/IAS Industrial and Commercial Power System Asia (I&CPS Asia)* (1833–1837). IEEE. (2022).
- Mahmoudi, E., dos Santos Barros, T. A. & Ruppert Filho, E. Forecasting urban electric vehicle charging power demand based on travel trajectory simulation in the realistic urban street network. *Energy Rep.* **11**, 4254–4276 (2024).
- Lipu, M. H. et al. Artificial intelligence approaches for advanced battery management system in electric vehicle applications: A statistical analysis towards future research opportunities. *Vehicles* **6** (1), 22–70 (2023).
- Shahzad, A., Amin, M., Emam, W. & Faisal, M. New ridge parameter estimators for the quasi-Poisson ridge regression model. *Scientific Reports*, **14**(1), p.8489. (2024).
- Hazarika, B. B. & Gupta, D. Density-weighted support vector machines for binary class imbalance learning. *Neural Comput. Appl.* **33** (9), 4243–4261 (2021).
- Hazarika, B. B. & Gupta, D. Density weighted twin support vector machines for binary class imbalance learning. *Neural Process. Lett.* **54** (2), 1091–1130 (2022).
- Pacal, I. A novel Swin transformer approach utilizing residual multi-layer perceptron for diagnosing brain tumors in MRI images. *Int. J. Mach. Learn. Cybernet.*, pp.1–19. (2024).
- Jin, B. & Xu, X. Forecasting wholesale prices of yellow corn through the Gaussian process regression. *Neural Comput. Appl.* **36** (15), 8693–8710 (2024).
- Vishwakarma, D. K. et al. Evaluation of catboost method for predicting weekly Pan evaporation in subtropical and sub-humid regions. *Pure. Appl. Geophys.* **181** (2), 719–747 (2024).

Acknowledgements

This research was supported by the Ministry of Higher Education (MOHE) of Malaysia through the Fundamental Research Grant Scheme (FRGS/1/2023/TK02/UTHM/02/1) with the grant code K458.

Author contributions

Izhar Hussain performed the programming and conducted the experiments to obtain the results presented in the manuscript. BC Kok supervised the entire research project, provided guidance throughout the study, and

was responsible for drafting the manuscript. Chessda reviewed the article, critically evaluated the models, and contributed to the analysis of results. Tey and Adil were involved in incorporating the mathematical formulations and theoretical aspects into the paper. Each author has made significant contributions to the research and manuscript preparation.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to I.H. or K.B.C.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025