RAIRO-Oper. Res. 60 (2026) 1959–1979
https://doi.org/10.1051/ro/2026054

RAIRO Operations Research
www.rairo-ro.org

MACHINE LEARNING OPTIMIZATION FOR ENERGY CONSUMPTION PREDICTION

Walid Hariri[1], Ahmed Boulemden[2,*] and Fujo Dickson Baraka[3]

Abstract. Global energy use has increased significantly over the last few decades, with residential structures accounting for a sizable amount of this usage. Therefore, creating trustworthy instruments for assessing and predicting energy use has become crucial in the global endeavor to improve sustainability, ultimately leading to more effective energy management strategies and a reduction in greenhouse gas emissions. Machine learning (ML) techniques have demonstrated high accuracy in energy usage prediction tasks. Using the publicly accessible KAG energy dataset, we assess and contrast ten ML and deep learning (DL) models to forecast energy usage in smart buildings, including Extra Trees Regression (ExtraTr), Long Short-Term Memory (LSTM), Multi-layer Perceptron (MLP), XG Boost Regressor, Gradient Boosting (GBoost), Convolutional Neural Network (CNN), Random Forest Regressor (RF), Elastic Net (ElNet) Regressor, Polynomial Regressor, and Support Vector Regresssor (SVR). The obtained results were compared with those of ARIMA model. According to our experimental findings, LSTM outperforms other models in capturing temporal dependencies in energy consumption data, demonstrating its superior ability to capture long-term patterns and fluctuations. This suggests that the recurrent nature of LSTMs, which allows them to retain information about past energy usage, is crucial for accurate forecasting in smart buildings.

Mathematics Subject Classification. 68T07, 68T20, 90C59, 68T05.

Received January 30, 2025. Accepted May 8, 2026.

1. Introduction

Reliable energy forecasting is essential for effective energy management, as it supports optimal resource allocation and the development of resilient energy systems capable of handling demand fluctuations. By reducing the mismatch between energy production and consumption [1], accurate forecasting contributes to maintaining a balanced and sustainable energy ecosystem [2]. In modern urban infrastructures, particularly smart buildings, advanced technologies such as Internet of Things (IoT) sensors, automated control systems, and intelligent energy management platforms enable continuous monitoring and optimization of energy usage. In this context, precise energy consumption prediction facilitates adaptive control of building subsystems, including lighting,

Keywords. Energy consumption, optimization, prediction, smart building, machine learning.

[1] LABGED Laboratory, Computer Science Department, Badji Mokhtar-Annaba University, P.O. Box 12, Annaba 23000, Algeria.
[2] LRI Laboratory, Computer Science Department, Badji Mokhtar Annaba University, P.O. Box 12, Annaba 23000, Algeria.
[3] Computer Science Department, Badji Mokhtar Annaba University, P.O. Box 12, Annaba 23000, Algeria.
*Corresponding author: ahmed.boulemden@univ-annaba.dz

heating, ventilation, and air conditioning, thereby improving operational efficiency while reducing unnecessary energy usage and associated costs.

In recent years, machine learning (ML), particularly deep learning (DL), has emerged as a transformative tool in enhancing the accuracy of energy consumption predictions [3]. Unlike traditional statistical approaches, which often rely on linear assumptions, deep learning models can analyze vast, complex datasets and uncover nonlinear patterns, trends, and interactions that might otherwise go unnoticed. These models excel in capturing the intricate relationships between environmental variables, building operations, and occupant behaviors, enabling more robust and reliable energy forecasts.

In this study, we explore the potential of ML, with a focus on DL techniques, to optimize the prediction of energy consumption in buildings. We emphasize the integration of environmental variables such as temperature, humidity, and solar radiation to further enhance prediction accuracy. By doing so, this research contributes to the development of more sustainable, energy-efficient building systems and lays the groundwork for future advancements in energy management.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the proposed approach and presents each model in detail. Section 4 reports the experimental results and analyzes the performance of the evaluated models in comparison with ARIMA. Section 5 discusses the obtained results, and Section 6 concludes the paper and outlines future research directions.

2. Related works

Energy consumption prediction has experienced substantial research over the past 20 years, driven by the increasing need for grid stability and building efficiency. This section classifies state-of-the-art methodologies based on their underlying modeling techniques – rather than building typology or specific energy end-uses – into two primary data-driven categories. The first category comprises *single models*, which include classical statistical approaches and time-series analysis. The second category focuses on *combined models*, encompassing ensemble methods and hybrid architectures.

2.1. Single models

These are data-driven strategies for tackling forecasting problems that use just one prediction algorithm. Below, we explore some instances of published work utilizing each model.

(a) Statistical regressions: For instance, the study by Amber *et al.* [4] examines Germany's energy supply and demand, analyzing the consumption of each sector, and uses energy balance sheet data to explore correlations with factors such as population and industrial growth through a regression model. These techniques use historical data to establish relationships between various factors (*e.g.*, weather, occupancy) and energy usage, and then project future energy consumption based on these relationships.

(b) K-Nearest Neighbours (KNN): KNN for optimization of energy consumption prediction involves identifying the most similar past data points (neighbors) to a given input and using their energy usage to estimate the current prediction. It leverages historical data to forecast future energy needs based on similarity measures. Valgaev *et al.* [5] created a kNN model to predict the overall electrical consumption of mixed-use buildings of various sizes and aggregated end-user load 24 h in advance. Daily profiles from smart metering were created using an Irish energy database that included approximately 6000 low-voltage buildings; day-type was also determined.

(c) Artificial Neural Network (ANN): Inspired by the neural networks found in the human brain, an ANN is a computational model that uses data to identify patterns and resolve complex problems through learning from data [6]. To improve early stage energy usage estimates for residential structures, Elbeltagi *et al.* [7] suggested an ANN model. The approach incorporates and automates commercial tools into a smooth workflow, based on simulations of several design alternatives using random inputs. Energy usage is precisely predicted by the validated ANN model. In [8], ANN and genetic algorithms were utilized for Galapagos plugin optimization.

(d) Time Series: Optimizing the prediction of energy consumption through time series analysis entails looking for patterns and trends in historical energy usage data. Recurrent neural networks (RNN) and long short-term memory (LSTM) employ these patterns to predict future energy usage based on historical behavior. For example, Jiang *et al.* [1] suggested an LSTM model to predict industrial robot energy consumption (EC) using joint motion variables, without necessitating robot parameters. By optimizing the trajectory, an adaptive genetic algorithm minimizes EC without requiring hardware changes by improving the time-variant scaling function. In order to forecast building energy consumption for residential and commercial buildings using historical data, occupancy patterns, and weather conditions, Mishra *et al.* [9] proposed an LSTM model and compared it with random forests (RF), decision trees, and linear regression.

The authors of [10] used RNN to predict and analyze overall power consumption and renewable energy output over an 11-year period. Feature engineering and thorough exploratory data analysis were conducted to investigate the relationship between yearly power use and renewable energy output. In contrast to traditional statistical approaches such as the AutoRegressive Integrated Moving Average (ARIMA) model as shown in [11], which assumes linear relationships and requires stationarity in the data, deep learning-based time series models (RNN/LSTM) can capture nonlinear dependencies, long-term temporal correlations, and complex interactions among multiple variables. While ARIMA remains effective for short-term linear forecasting with limited features, LSTM-based models are better suited for high-dimensional, non-stationary, and multivariate energy consumption datasets.

2.2. Combined models

Combined models focus on forecasting technique optimization to increase prediction accuracy. Several single algorithms (ensemble models) or optimization strategies are integrated into an amalgamated modeling framework. The authors of [10] developed a method to analyze and forecast total power consumption and renewable energy production over eleven years using RNN.

(a) Ensemble models: In Florida, USA, a comparison of regression trees, SVRs, and RFs was conducted to estimate electricity usage in two institutional buildings, one of which is a LEED building [12]. Factors entered included humidity, time of day, pressure, type of workday, WS, pressure, precipitation SR, and projected tenancy based on daily operations and class schedule. Twenty percent of the data collected during a typical operating year was used for model testing, while the remaining eighty percent was used for training.

(b) Hybrid models: In this work, "improved models" refers to individual models that have been improved through optimization techniques. In [13], a different study focused on a 5 min prediction horizon and offered a hybrid methodology that incorporates LSTM and CNN. Their detailed analysis included a comparison with industry standard methodologies including ARIMA, Deep Neural Networks (DNN), Random Forest, and Light Gradient Boosting Machine. According to the study, the CNN-LSTM hybrid model performs better than any other model, proving its superiority.

3. The proposed methods

This section focuses on our suggested approaches to the problems in the field of energy consumption prediction. We provide a comprehensive overview of the main ML and DL regression-based approaches employed, as well as an explanation of the ten strategies employed in our suggested model.

3.1. Framework overview

The conceptual framework diagram in Figure 1 explains our methodical approach to forecasting energy usage in a smart building using several ML models. In order to comprehend patterns of energy use in the building, it starts with a dataset in a .csv file format that includes historical data parameters like temperature, humidity, and other weather factors. This dataset is subjected to exploratory data analysis, where cross-validation and data visualization are crucial steps in revealing insights and guaranteeing the accuracy of the data.

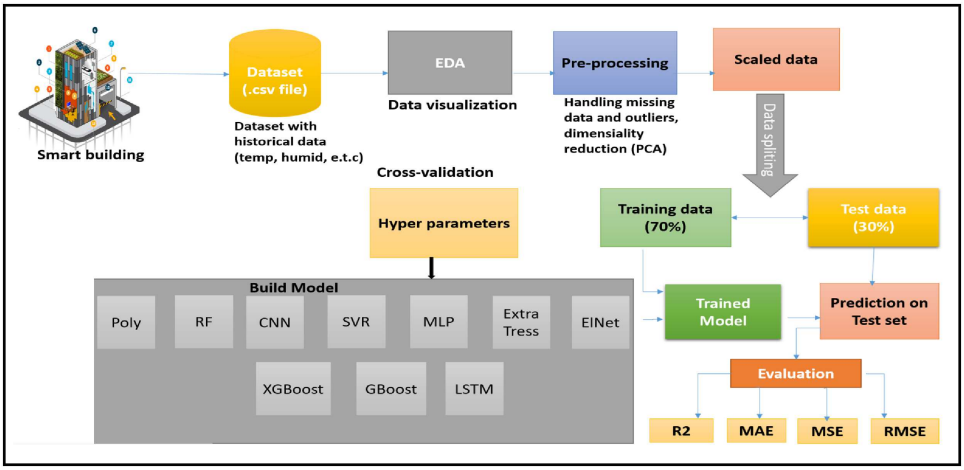


FIGURE 1. Conceptual framework.

The next phase is pre-processing, which involves handling missing data and outliers, reducing dimensionality through Principal Component Analysis (PCA), and splitting the dataset into training (70%) and test data (30%). This step is critical to prepare the data for effective model training and testing. The framework then presents an array of ten algorithms for building predictive models, where each algorithm is trained independently on the training set to create a trained model, which is then used to make predictions on the test set. The final step in this process is Evaluation, where metrics such as R-squared value (R^2), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) are used to assess the performance of the models. This comprehensive approach ensures that the most accurate and robust model is selected for forecasting energy consumption, ultimately aiding in the optimization of energy use within smart buildings.

3.2. Machine learning prediction methods

Ten ML methods for energy consumption prediction are covered in this section. A variety of input variables are used in the models, such as relative humidity, indoor and outdoor temperatures, appliance energy usage, and other pertinent meteorological factors. These features are integrated into the models to improve the accuracy of the prediction and provide insight into how various environmental factors influence energy use.

3.2.1. Extra trees regression

Originally presented by Pierre Geurts, Damien Ernst, and Louis Wehenkel [14], Extra Trees Regression reduces variance by averaging decision tree outputs, producing additional trees from dataset subsamples, and selecting the best split for each node from random splits drawn from the *max_features* selected features. In contrast to standard RF, Extra Trees use the entire learning sample to train each tree and apply random top-down splitting. Equation (1) represents the prediction $\hat{y}$ of the Extra Trees Regression model as the average of the predictions from M decision trees (DT).

$$\hat{y} = \frac{1}{M} \sum_{m=1}^{M} T_m(\mathbf{x}). \tag{1}$$

Where:

- $\hat{y}$: The predicted output.
- M: The total number of trees in the ensemble.

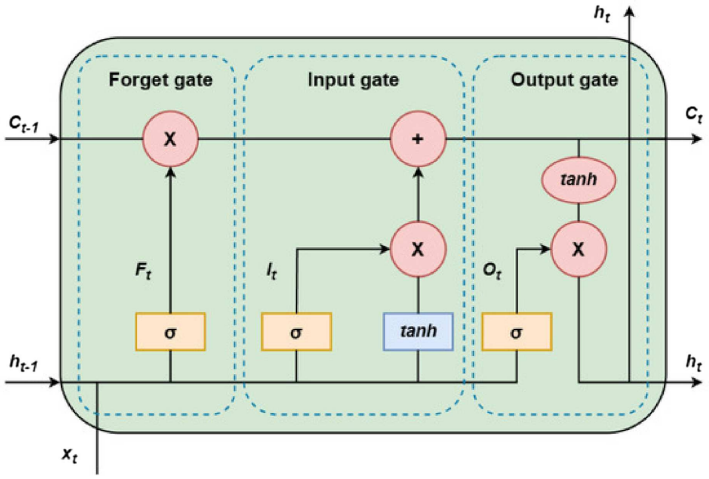


FIGURE 2. Proposed LSTM cell architecture.

– $T_m(\mathbf{x})$: The output of the m-th DT for input $\mathbf{x}$.
– $\mathbf{x}$: The input feature vector.

3.2.2. Long short-term memory (LSTM)

These are one kind of RNN that can learn long-term dependencies. Since their introduction in 1997, they have developed into significant elements of the field of deep learning, which deals with sequential data analysis [15]. LSTM excels in identifying and simulating dependencies in sequential data because it has specialized memory cells that can detect long-range dependencies. As illustrated in Figure 2, an LSTM cell is made up of three separate gates. While the forget gate determines which information from the previous cell state should be forgotten, the input gate selects the new data that should be added to the current cell state. The output gate regulates what information is displayed to the next cell by eliminating less important information. The functions of an LSTM cell to forecast energy usage are described by equation (2).

$$\begin{aligned} \mathbf{C}_t &= \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t \\ \mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{C}_t). \end{aligned} \tag{2}$$

Where:

– $\mathbf{C}_t$: Cell state vector at time step t.
– $\mathbf{h}_t$: Hidden state vector at time step t.
– $\mathbf{f}_t$: Forget gate activation vector.
– $\mathbf{i}_t$: Input gate activation vector.
– $\tilde{\mathbf{C}}_t$: Candidate cell state vector.
– $\mathbf{o}_t$: Output gate activation vector.
– $\odot$: Element-wise multiplication.
– tanh: Hyperbolic tangent activation function.

3.2.3. Multi-layer perceptron regressor (MLPRegressor)

Frank Rosenblatt first proposed the basic concept of a perceptron in 1958, and in the 1980s, MLP with backpropagation training and hidden layers became popular [16]. The identity function is employed in the absence of an activation function in the output layer, resulting in continuous outputs with the square error serving as the loss function.

To avoid overfitting, the MLPRegressor uses a regularization term throughout its iterative training process. Each of its many nodes and activation functions which send outputs to later layers consists of input, hidden, and output layers. Equation (3) simplifies the presentation while capturing the key functions of the MLPRegressor.

$$\hat{y} = f(\mathbf{W}_2 \cdot g(\mathbf{W}_1 \cdot \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2). \tag{3}$$

Where:

- $\hat{y}$: The predicted output.
- $\mathbf{x}$: The input feature vector.
- $\mathbf{W}_1$: The weight matrix for the input to hidden layer.
- $\mathbf{W}_2$: The weight matrix for the hidden to output layer.
- $\mathbf{b}_1$: The bias vector for the hidden layer.
- $\mathbf{b}_2$: The bias vector for the output layer.
- $g(\cdot)$: The activation function for the hidden layer (*e.g.*, ReLU, sigmoid).
- $f(\cdot)$: The activation function for the output layer, which is the identity function in this case.

3.2.4. XG boost regressor

An XG Boost Regressor is a powerful gradient boosting (GB) algorithm that builds an ensemble of DT in a sequential manner [17]. It incorporates regularization to avoid overfitting and optimizes for performance by minimizing a differentiable loss function. It is frequently employed in ML applications because of its reputation for accuracy and speed.

3.2.5. Gradient boosting regressor

Instead of using conventional residuals, it employs pseudo-residuals, which are based on boosting in a functional space [18]. It offers a prediction model in the form of a group of weak models, usually straightforward DT. Gradient-boosted trees enable optimization of any differentiable loss function by using DT as weak learners.

3.2.6. Convolutional neural network (CNN)

CNNs, a regularized type of feed-forward neural network, learn feature engineering on their own through filters (or kernel) optimization. Regularized weights are preferred over fewer connections because they avoid the increasing and disappearing gradients that were seen in previous neural networks during backpropagation [19,20].

CNNs, are typically composed of four major components that work together to mimic the way in which the human brain interprets patterns and characteristics in images: Convolutional, pooling, fully linked, and Rectified Linear Unit (ReLU) layers are among the types of layers. A proposed CNN architecture is depicted in Figure 3. It employs a number of convolutional and pooling layers to extract progressively abstract features from the data, which are subsequently employed for prediction by the fully connected layers. For jobs where comprehending intricate patterns is essential, such as predicting energy use, this method works incredibly well. In summary, the convolutional layers apply filters (kernels) to input data, capturing local patterns and features. Pooling layers downsample the feature maps, reducing spatial dimensions. Fully connected layers connect the extracted features to the output layer for classification or regression. Finally, a number of convolutional and pooling layers extracts progressively abstract features from the data, which are subsequently employed for prediction by the fully connected layers. For jobs where comprehending intricate patterns is essential, such as predicting energy use, this method works incredibly well .

3.2.7. Random forest regressor (RF)

By averaging the anticipated energy consumption of the many randomized DT, it generates output. It is frequently applied to classification and regression-related issues. The concept of RF was initially presented by Breiman in 1984 [21]. It also performs exceptionally well with both small and big sample sizes and has the advantage of requiring less hyperparameter modifications [22]. The n-estimators hyperparameter regulates the random forest's decision tree count.

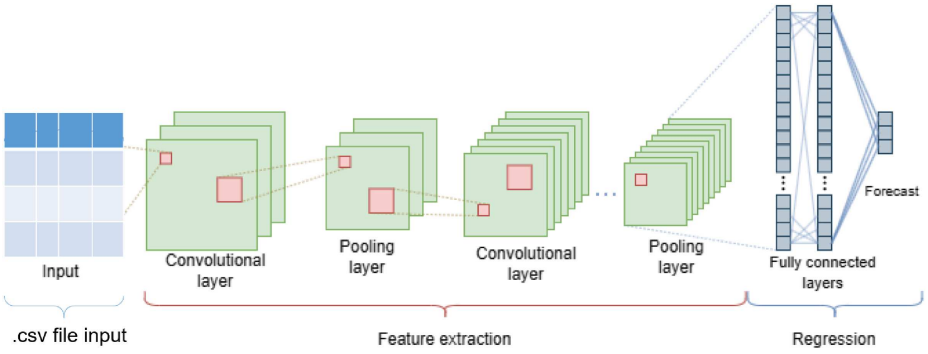


FIGURE 3. Proposed CNN architecture that uses multiple convolutional and pooling layers to extract increasingly abstract features from the data, after which it is used by the fully connected layers for prediction.

3.2.8. Elastic net regressor (ElNet)

Originally released in 2005 According to [23], this model is a regularized regression technique used in statistics, specifically in the fitting of logistic or linear regression (LR) models. It combines in a linear way the L1 and L2 penalties of the lasso and ridge approaches. Even when additional covariates are linked to the outcome, lasso can only select n covariates when $p > n$ (the number of covariates is more than the sample size) and it usually chooses one covariate from any group of highly correlated covariates. Furthermore, given strongly correlated covariates, ridge regression typically performs better even when $n > p$. The ElNet adds an extra e^2 penalty term to the lasso, extending it.

Sometimes, the estimated coefficients of the naïve version of ElNet are multiplied by $(1+\lambda_2)$ in order to rescale the coefficients and increase prediction performance [23]. Here are some instances where the ElNet method has been used: Machine support vector [24] Learners using metrics [25] Optimizing a portfolio [26] Prognosis for cancer [27].

3.2.9. Polynomial regressor

An nth degree polynomial in x is used in polynomial regression, a sort of regression analysis used in statistics, to represent the relationship between the independent variable x and the dependent variable y. Polynomial regression can be used to fit a nonlinear relationship between the value of x and the corresponding conditional mean of y, denoted as $E(y|x)$. Polynomial regression is a linear statistical estimation issue even though it fits a nonlinear model to the data because the regression function $E(y|x)$ is linear in the unknown parameters that are estimated from the data. Multiple linear regression is therefore considered to be a specific instance of polynomial regression.

The independent variables that result from the polynomial expansion of the "baseline" variables are known as higher-degree terms. Certain variables are also used in classification settings [28].

Most often, the least squares approach is used to fit polynomial regression models. As long as the Gauss–Markov theorem is met, the least-squares approach reduces the variance of the unbiased estimators of the coefficients. Legendre published the least-squares approach in 1805, and in 1809, Gauss published it. The first polynomial regression experiment design was published in a paper by Gergonne [29, 30] in 1815. Regression analysis was developed in the 20th century with a stronger focus on design and inference concerns thanks in large part to polynomial regression [31]. In contemporary times, non-polynomial models have emerged as a useful alternative to polynomial models in some problem classes.

Regression analysis is used to model the expected value of a dependent variable, y, in terms of the value of an independent variable, x, or vector of independent variables. In a basic linear regression, the model ϵ is an

unobserved random error with a zero mean that is conditioned on a scalar variable x. For each unit rise in the value of x, the conditional expectation of y in this model increases by β_1 units.

$$y = \beta_0 + \beta_1 x + \epsilon. \tag{4}$$

Such a linear correlations might not hold in many situations. For example, we may find that the yield of a chemical synthesis grows proportionately with each unit rise in temperature if we represent the yield in terms of the temperature at which it happens. We might recommend a quadratic model of the following type in this case:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon. \tag{5}$$

In this model, as the temperature increases from x to $x+1$ units, the expected yield varies by $\beta_1, \beta_2(2x+1)$. This is seen by subtracting the equation in x from the equation in $x+1$, where x is substituted with x+1, to see this. One can determine the impact of minute variations in x on y by computing the total derivative with respect to x, or $\beta_1, 2\beta_2 x$. The yield change relies on x, therefore even though the model is linear in the parameters that need to be estimated, the relationship between x and y is nonlinear. The anticipated value of y can be modeled as an nth degree polynomial to create the general polynomial regression model.

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \cdots + \beta_n x^n + \epsilon. \tag{6}$$

From an estimating perspective, these models are all conveniently linear since the regression function is linear with respect to the unknown parameters $\beta_0, \beta_1, \ldots$ Therefore, the computational and inferential issues of polynomial regression can be fully handled using multiple regression techniques for least squares analysis. $x, x^2, \ldots$ are treated as separate independent variables in a multiple regression model in order to achieve this.

3.2.10. Support vector regresssor (SVR)

First introduced in 1996, fnding a function that minimizes the prediction error and roughly represents the relationship between the input variables and a continuous target variable is the aim of SVR [32]. Because training points that fall outside of the margin are ignored by the cost function used to construct the model, the support vector classification model only requires a fraction of the training data. Comparably to Support Vector Machine (SVMM), the SVR model only requires a portion of the training set since any training set that is near the model prediction is not taken into account by the cost function that was employed to build the model [33]. To train the original SVR, one must solve:

minimize $\frac{1}{2}\|w\|^2$
according to $|y_i - \langle w, x_i \rangle - b| \leq \varepsilon$

where the training sample, x_i, has a target value of y_i. The forecast for that sample is ε, a free parameter that serves as a threshold, requiring all predictions to fall within a range of the right predictions. The forecast is the inner product plus intercept $(w, xi) + b$. Generally, slack variables are added to the above to allow for approximation in case the above problem is not attainable and to account for errors. For the data that lie inside the *ε-insensitive* tube, the hyperplane provides the best fit. Figure 4 summarizes the differences between SVM and SVR.

Theory behind SVR: Finding the optimal fit that properly predicts the target variable while lowering complexity to prevent overfitting is the aim of SVMs. In order to accomplish this, a *ε-insensitive* tube is defined around the hyperplane. This tube allows for a certain amount of variance between the actual and anticipated target values, balancing the model's complexity and generalization capability. Mathematically speaking, SVR is a convex optimization problem. Finding a function $f(x)$ that is as flat as possible for all training data with a maximum divergence of ε from the real targets is the goal of the task. The function's flatness lowers the chance of overfitting by indicating that it is less sensitive to slight variations in the input data. In the best fit function $f(x) = (w_x) + b$, a small value of w for linear functions is referred to as flatness. Figure 5 displays the Univariate linear SVR (with zero tolerance for error) while Figure 6 shows univariate linear SVR (allowing for errors).

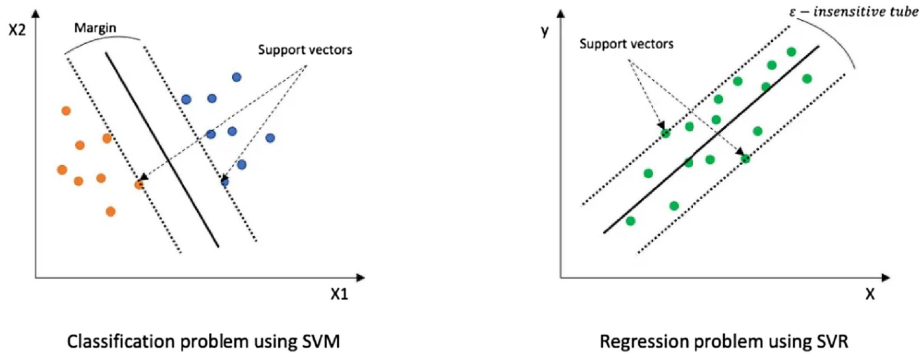


FIGURE 4. The differences between SVM and SVR summary.

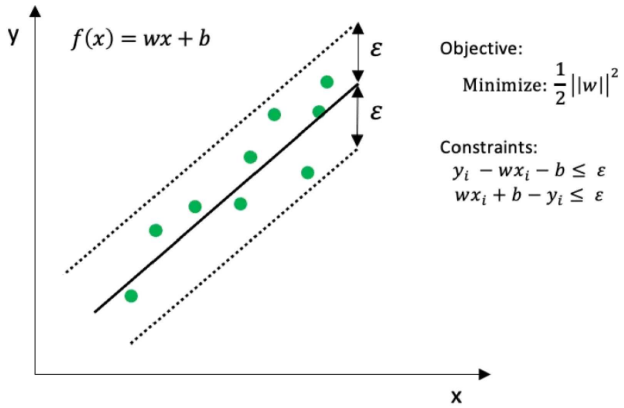


FIGURE 5. Univariate linear SVR (with zero tolerance for error).

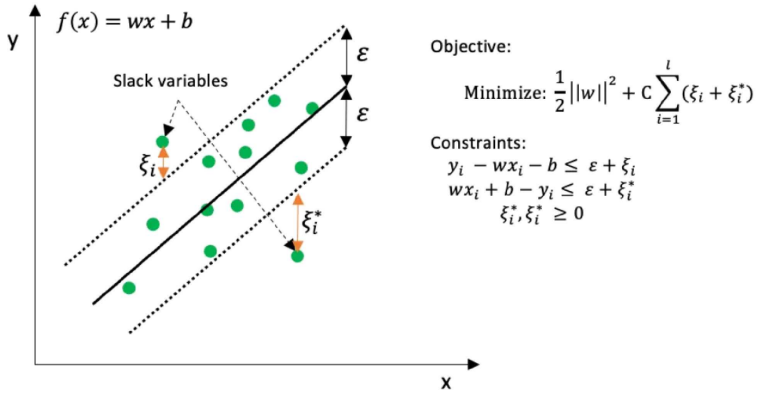


FIGURE 6. Univariate linear SVR (allowing for errors).

By utilizing a variety of kernel functions to address non-linear regression, SVR is applicable to problems involving both linear and non-linear regression. Kernel aids in locating function $f(x)$ in a higher-dimensional space where a LR problem can be solved when the data is not linearly separable. The linear, polynomial, sigmoid, and radial basis function (RBF) kernel functions are commonly used in SVR.

3.2.11. ARIMA

The Autoregressive Integrated Moving Average (ARIMA) model is a statistical approach widely used for time-series regression and forecasting tasks such as energy consumption prediction. It models the temporal dependency of the series by combining autoregressive (AR) and moving average (MA) components applied to a differenced version of the data to ensure stationarity. An ARIMA model is denoted as ARIMA (p, d, q), where p represents the autoregressive order, d denotes the degree of differencing required to make the series stationary, and q corresponds to the moving average order. In the context of energy consumption prediction, ARIMA captures linear temporal correlations between past observations and current demand levels. The model estimates future values based on previous energy consumption patterns and past prediction errors. Despite its simplicity, ARIMA remains a strong baseline for time-series forecasting problems in energy management applications.

4. Experimental results and discussion

The experiments were performed using the Google Colaboratory cloud platform. The local machine used to access the environment was equipped with 16 GB RAM, a 2.5 GHz CPU, and Windows 11. The training and evaluation of the models were executed using the GPU resources provided by Google Colaboratory, specifically an NVIDIA Tesla T4 accelerator. This section covers every step of the experimental procedure, including data collection, pre-processing, and visualization to help with dataset understanding and variable correlation analysis. After that, we go into detail on the parameter configurations for the ML regression techniques that were covered, and we show the dataset's experimental outcomes.

4.1. Dataset description

Using the publicly accessible KAG energy data complete dataset [34], we created and assessed our algorithms. Among the attributes contained in the dataset are temperature, humidity, time of day, and various appliance power levels. The information used includes humidity and temperature measurements from nine rooms in a house that are connected to a wireless ZigBee sensor network with several features and observations scattered across 19 735 rows and 29 columns. It also includes climatic and meteorological data from the airport at Chievres, Belgium, such as humidity point, wind speed, visibility, temperature, pressure, and humidity. Over the course of 4.5 months, from 11 January 2016 to 26 May 2016, energy data was recorded using M-Bus energy meters from a smart building at 10 min intervals. The dataset's correlation between environmental factors is displayed in Figure 7.

- **Temperature**: There is a positive correlation between the target appliances and every temperature variable from T1 to T9 and T_out. The high relationships between interior temperatures can be explained by the HRV device, which stabilizes indoor temperatures. Notably, T6 exhibits a substantial correlation with T_out, and T9 is strongly correlated with T3, T5, T7, and T8. Because T6 and T9's data is redundant, they can be removed from the training set.
- **Weather attributes**: The correlation values between visibility, Tdewpoint, and Press_mm_hg are poor.
- **Humidity**: For humidity sensors, there are no occurrences with noticeably high correlation (> 0.9).
- **Random variables:** have no significance to the model.

tr_scr (Training Score) and tst_scr (Testing Score): Indicates how well the model fits the training data and how it performs on unseen data respectively, while R^2 (Coefficient of Determination) measures the proportion of variance in the dependent variable predictable from the independent variables. **MAE** (Mean Absolute Error)

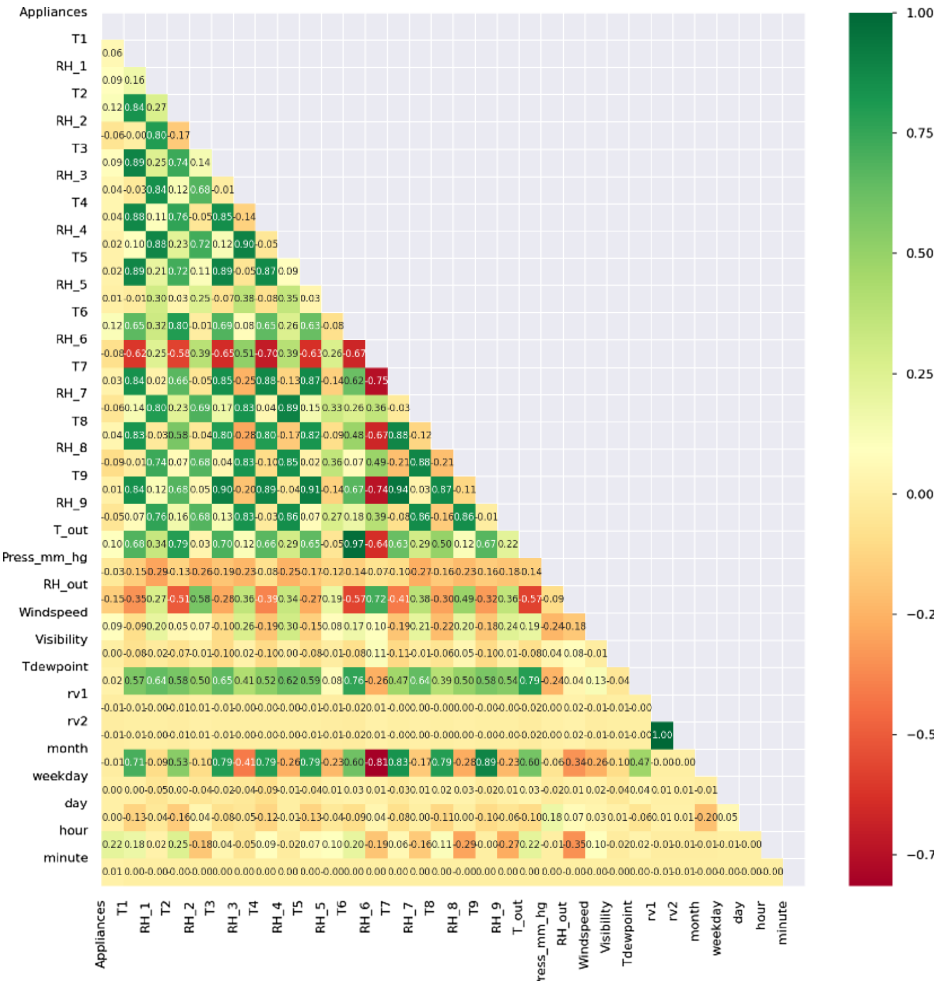


FIGURE 7. Variables correlation.

measures the average magnitude of errors in a set of predictions, without considering their direction. **MSE** (Mean Squared Error) measures the average of the squares of the errors, giving more weight to larger errors. **RMSE** (Root Mean Squared Error) is the square root of MSE, providing an error metric in the same units as the target variable.

4.2. Results

This section presents the experimental results and evaluates the effectiveness of the proposed models across the tested dataset.

4.2.1. LSTM

Figure 8 displays an LSTM model's training and validation loss over 70 epochs. Effective learning and convergence are indicated by both losses first decreasing substantially before flattening off. Good model performance and generalization are shown by the near proximity of training and validation losses, which suggests little overfitting.

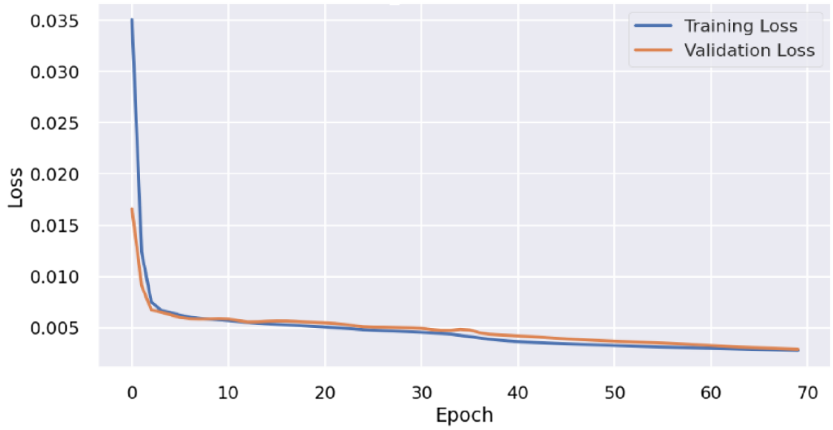


FIGURE 8. LSTM train and validation loss.

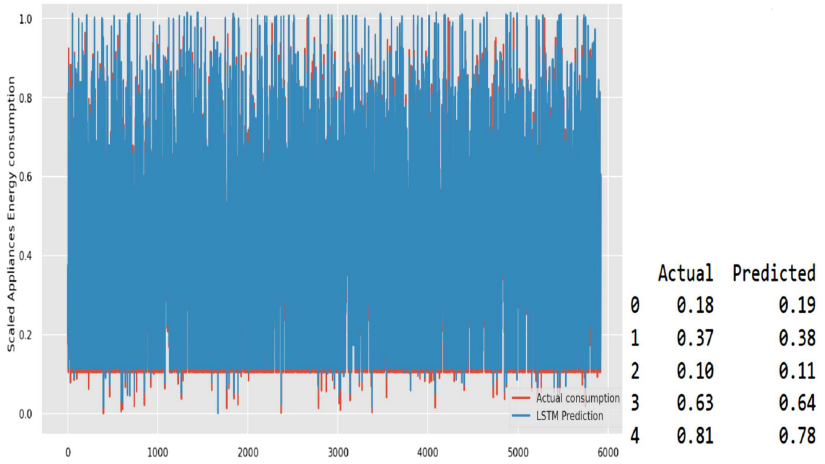
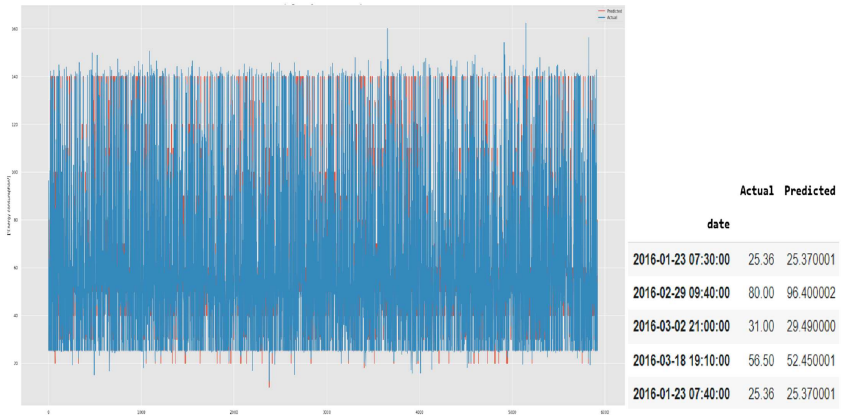


FIGURE 9. LSTM energy consumption prediction.

The performance of the LSTM model is displayed in Figure 9. It is simpler to compare the actual and forecast energy consumption values since the y-axis, which is labeled "Scaled Appliance Energy Consumption," ranges from 0 to 1.0. Time steps or data points are represented by the x-axis, which runs from 0 to 7000. Over this range, the actual and predicted values closely match, indicating that the model's accuracy holds steady over time. Furthermore, the table on the graph's right side contrasts particular actual and expected values, demonstrating how closely the forecasts match the actual numbers.

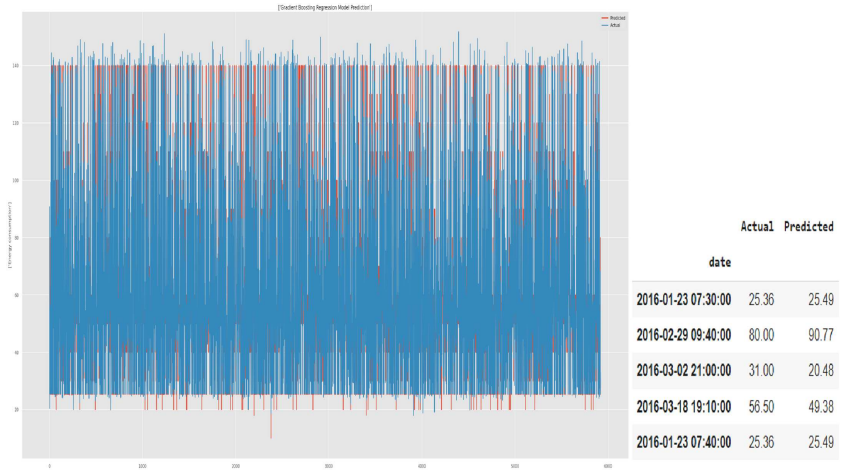
4.2.2. XGBoost

The effectiveness of an XGBoost Regressor in forecasting energy use over time is shown in Figure 10. Dates are shown on the X-axis, and energy usage is displayed on the Y-axis. Good model performance is indicated by the tight relationship between the blue lines (actual values) and the red lines (predicted values). Deviations draw attention to areas that need work. The accuracy of the model is demonstrated by the numerical values for actual and expected consumption in the accompanying table.



date	Actual	Predicted
2016-01-23 07:30:00	25.36	25.370001
2016-02-29 09:40:00	80.00	96.400002
2016-03-02 21:00:00	31.00	29.490000
2016-03-18 19:10:00	56.50	52.450001
2016-01-23 07:40:00	25.36	25.370001

FIGURE 10. XGBoost regression prediction.



date	Actual	Predicted
2016-01-23 07:30:00	25.36	25.49
2016-02-29 09:40:00	80.00	90.77
2016-03-02 21:00:00	31.00	20.48
2016-03-18 19:10:00	56.50	49.38
2016-01-23 07:40:00	25.36	25.49

FIGURE 11. Gradient Boosting regression prediction.

4.2.3. Gradient boosting

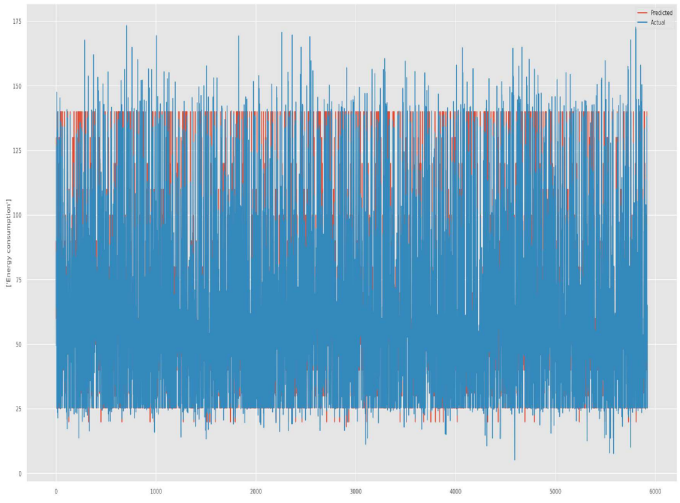
Using a Gradient Boosting model, Figure 11 displays the actual and anticipated energy use over a variety of dates and kilowatt-hours. Specific actual and expected numbers, such as 25.36 kWh and 25.49 kWh on 2016-01-23 at 07:30:00, are compared in the table on the right. The model's accuracy in forecasting energy usage is demonstrated by this close match.

4.2.4. MLPRegressor

Based on input features, the MLPRegressor model predicts energy usage with high accuracy. The actual consumption on March 20, 2016, at 2:00, was 48.0, whereas the projected figure was 36.22.

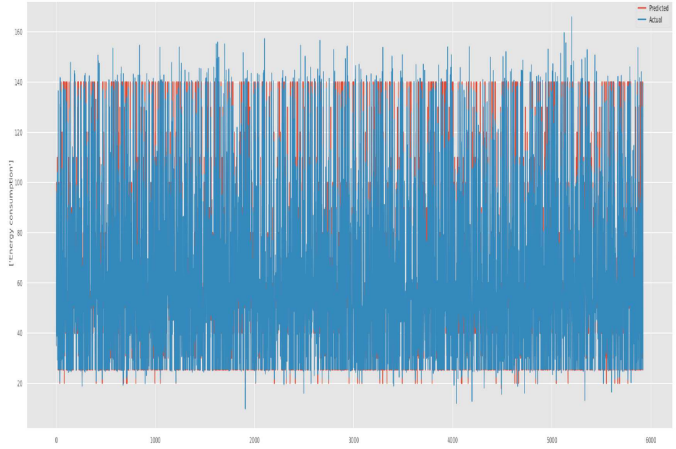
4.2.5. Extra trees

The Extra Trees Regressor model does well in predicting energy consumption, with predictions that closely match the data. More details about Extra Trees prediction results is given in Tables 1 and 2 below.



date	Actual	Predicted
2016-01-30 22:20:00	60.0	63.96
2016-04-11 11:30:00	90.0	85.47
2016-05-19 02:00:00	50.0	49.50
2016-02-14 20:30:00	130.0	126.82
2016-05-19 04:30:00	60.0	57.39

FIGURE 12. Polynomial regression prediction.



date	Actual	Predicted
2016-03-20 02:50:00	40.0	35.02
2016-04-16 20:00:00	100.0	96.06
2016-02-18 17:40:00	70.0	72.62
2016-01-19 20:20:00	56.5	76.17
2016-03-23 03:50:00	40.0	38.79

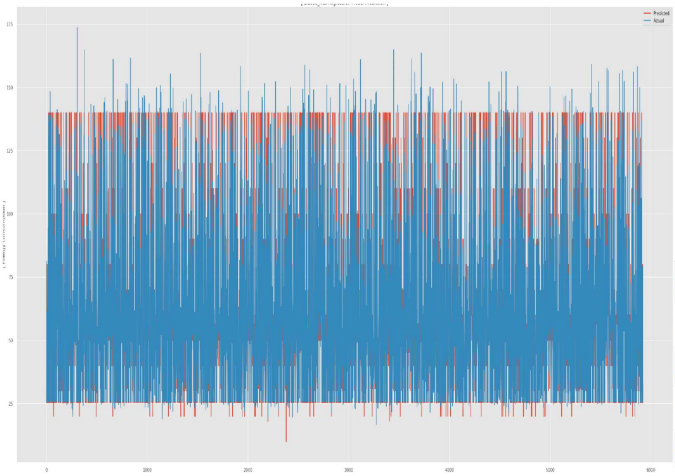
FIGURE 13. SVR regression prediction.

4.2.6. Polynomial regression

Figure 12 presents the energy consumption prediction by the polynomial regression model after cross validation. This method was selected because to its comprehensive exploration, automatic adjustment of hyperparameters, cross-validation, objective optimization, and its resilience and ease of use in hyperparameter optimization.

4.2.7. SVR

Figure 13 illustrates the energy consumption prediction performance of the SVR model. The time-series plot demonstrates that the predicted values (red) closely track the fluctuations of the actual data (blue), capturing the high-frequency peaks. As evidenced by the provided sample data points, the model achieves high precision in several instances, such as the prediction of 38.79 against an actual value of 40.0. While the regression line effectively follows the overall trend, slight deviations are observable in the extreme noise regions, though the model maintains a consistent fit across the tested temporal horizon.



date	Actual	Predicted
2016-01-23 07:30:00	25.36	25.75
2016-02-29 09:40:00	80.00	81.28
2016-03-02 21:00:00	31.00	25.51
2016-03-18 19:10:00	56.50	56.34
2016-01-23 07:40:00	25.36	25.75

FIGURE 14. Elastic Net regression prediction.

4.2.8. EINet

With respect to the training and test datasets, the model's overall performance has improved, as seen by reduced MSE and higher R-squared values. The ElNet model is now more precise and fits the data better thanks to the cross-validation and hyperparameter tuning procedures. Figure 14 displays the energy consumption prediction using the Elastic Net model after cross-validation, which demonstrates a consistent ability to capture the dataset's periodic trends. The regression plot shows the predicted values (red) aligning closely with the actual consumption peaks (blue), supported by numerical samples such as the 81.28 prediction for an 80.00 actual value. Despite minor deviations in high-variance segments, the model provides a computationally efficient and stable estimation across the testing period.

4.2.9. CNN

To put our CNN model into practice, we constructed a fully connected layer with 64 neurons and ReLU activation, a dropout layer with a 50% dropout rate, an output layer without an activation function because we are working on a regression task, a Max-pooling layer with a pool size of 2, a 1D convolutional layer with 32 filters and ReLU activation, and a Flatten layer that flattens the output from the convolutional layers. The 'adam' optimizer and the MSE loss function were then used to construct the model. The model was validated using 10-k-fold cross-validation, and the MSE we found was 1075.44. The training and validation loss of the CNN over 200 iterations is shown in Figure 15 while Figure 16 presents the energy consumption prediction by the CNN model after 10 k-fold cross-validation.

4.2.10. Random forest

The Random Forest Regressor is an ensemble model that averages predictions from multiple decision trees. Each tree's predictions are averaged to obtain the final outcome of regression tasks. Overfitting is reduced and prediction accuracy is raised by employing an ensemble approach. A discernible improvement can be seen in the RF Regressor's performance metrics prior to and following cross-validation.

Figure 17 presents the energy consumption prediction by the Random Forest Regression model after cross-validation.

4.2.11. ARIMA

In comparison to the advanced architectures presented in Table 2, the ARIMA model serves as a traditional statistical baseline that exhibits significantly higher error margins, characterized by an estimated RMSE of

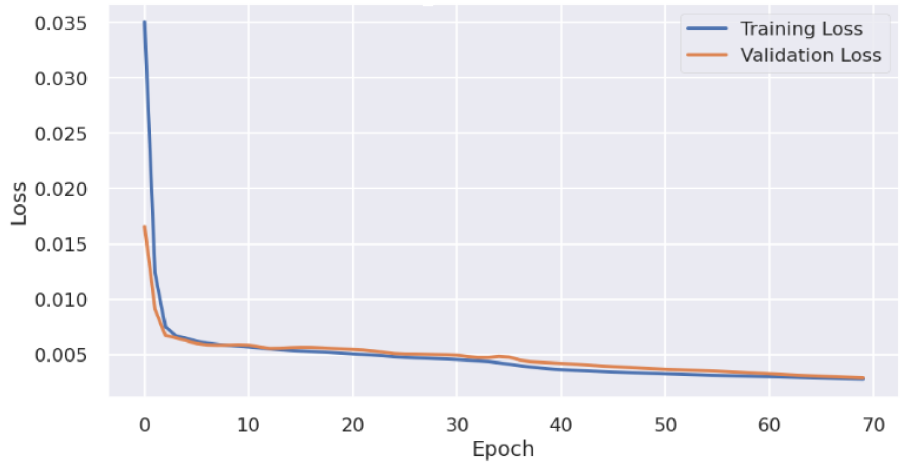
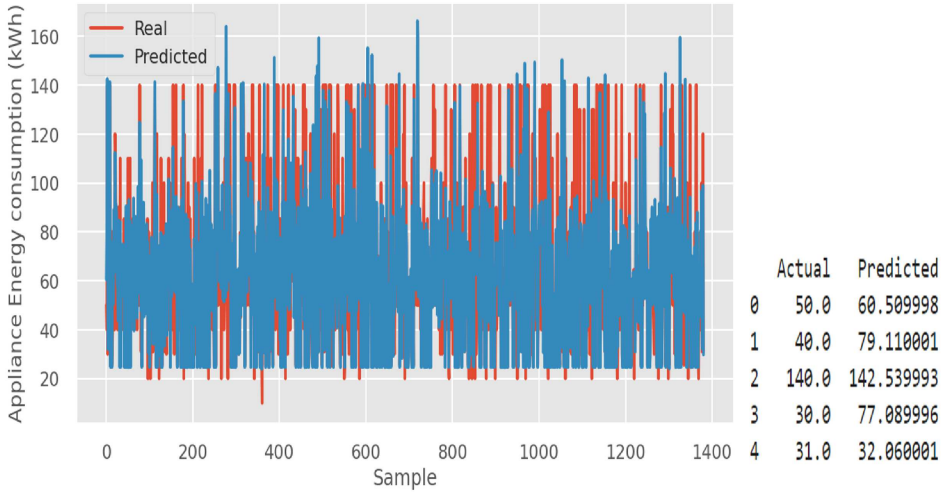


FIGURE 15. CNN train and validation loss.

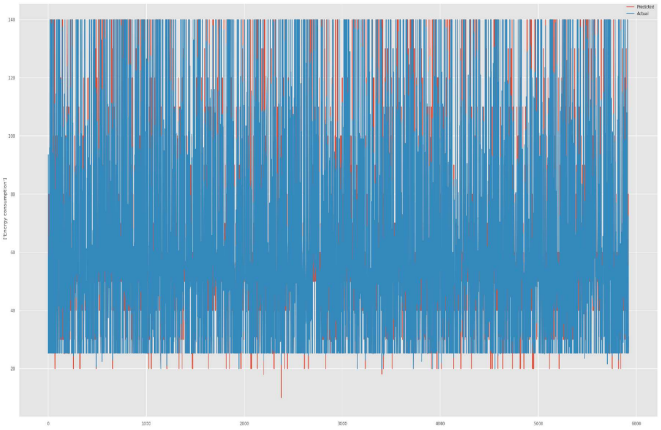


	Actual	Predicted
0	50.0	60.509998
1	40.0	79.110001
2	140.0	142.539993
3	30.0	77.089996
4	31.0	32.060001

FIGURE 16. CNN energy consumption prediction.

10.26 and a reduced R^2 of 0.71. While deep learning models such as LSTM and CNN effectively minimize loss by capturing complex non-linear patterns and long-term temporal dependencies, ARIMA predictive accuracy is constrained by its linear assumptions and univariate nature. Consequently, it fails to incorporate critical exogenous features, such as meteorological conditions or temporal metadata, and struggles to adapt to the high volatility and multi-seasonal fluctuations inherent in energy consumption data, leading to a substantial performance gap with an MSE of 105.4, which is higher than that of all the evaluated models, including EINet, which achieved an MSE of 76.798.

The following comparison evaluates traditional ARIMA models against related state-of-the-art approaches for both short-term and long-term predictions. While ARIMA is limited by linearity and stationarity assumptions, the applied machine learning and ensemble models effectively capture the nonlinearities and high-dimensional interactions inherent in complex, multivariate inputs such as weather variables, occupancy patterns, and external contextual factors. In contrast, the machine learning and ensemble models evaluated in this study naturally



date	Actual	Predicted
2016-01-23 07:30:00	25.36	25.36
2016-02-29 09:40:00	80.00	93.50
2016-03-02 21:00:00	31.00	31.00
2016-03-18 19:10:00	56.50	56.18
2016-01-23 07:40:00	25.36	25.36

FIGURE 17. Random forest regression prediction.

TABLE 1. Algorithm performance before cross-validation.

Metrics	tr-scr	tst-scr	R^2	MAE	MSE	RMSE
GBoost	0.939	0.928	0.93	6.30	89.855	9.479
XGBoost	0.984	0.959	0.96	4.45	51.382	7.168
LSTM	0.993	0.994	0.994	0.011	0.000362	0.019
ExtraTr	0.999	0.997	0.997	0.000005	0.00012	0.011
Poly	0.948	0.947	0.95	5.44	0.022	0.149
ElNet	0.836	0.831	0.83	10.43	17.706	4.55
RF	0.991	0.943	0.94	4.96	0.9801	0.99
SVR	0.93	0.93	0.93	6.00	30.91	5.56
CNN	0.78	0.77	0.77	1.51	0.62	0.40
MLP	0.99	0.991	0.991	0.016	0.00051	0.0227

handle nonlinear relationships and high-dimensional feature spaces without strict assumptions about data distribution or stationarity. For example, Azani *et al.* [35] evaluated LSTM and SVM models for electrical energy consumption forecasting using RMSE and prediction time as evaluation metrics. The reported results indicate that the ARIMA model can effectively predict electrical energy consumption over a one-year horizon. Compared with LSTM, ARIMA achieved a lower RMSE (7.659 *vs.* 11.418 for LSTM), while SVM obtained the lowest RMSE of 0.020. For long-term prediction, Sachin *et al.* [36] compared the ARIMA model with RNN and LSTM architectures for energy consumption prediction over four years for a single household. The results demonstrate that LSTM and RNN provide a lower Root Mean Square Error (RMSE) of 0.14 for both, whereas ARIMA achieved an RMSE of 0.46. This performance gap can be explained by the ability of LSTM and RNN models to capture non-linear dependencies and complex patterns within the energy consumption data, which the linear nature of the ARIMA model fails to adequately represent. Furthermore, ARIMA typically requires careful manual identification of parameters and differencing steps to achieve stationarity, whereas tree-based ensemble methods automatically adapt to data complexity through hierarchical partitioning. This flexibility enables models such as ExtraTrees and XGBoost to better capture irregular consumption dynamics and hidden interactions between predictors.

The results obtained before and after cross-validation are summarized in Tables 1 and 2, respectively. Prior to cross-validation, XGBoost outperforms GBoost with even higher scores (0.984 and 0.959) and smaller errors, whereas GBoost obtains high training and testing scores (0.939 and 0.928) with low error metrics. Overfit-

TABLE 2. Algorithm performance after cross-validation.

Metrics	tr-scr	tst-scr	R^2	MAE	MSE	RMSE
GBoost	0.96	0.94	0.943	5.47	71.03	8.428
XGBoost	0.99	0.962	0.96	4.245	46.965	6.853
Poly	0.948	0.947	0.95	5.42	66.25	8.14
ElNet	0.94	0.939	0.93	5.83	76.798	8.763
RF	0.944	0.94	4.96	71.24	8.44	2.905
SVR	0.96	0.96	0.96	4.42	51.34	7.17
CNN	0.76	0.76	0.76	2.93	0.01	0.75
LSTM	0.991	0.991	0.991	0.014	0.000520	0.023
ARIMA	0.72	0.70	0.71	8.12	105.4	10.26

ting is shown by the nearly perfect scores (0.999 and 0.997) displayed by ExtraTreesRegressor. Although they have a little greater error rate, LSTM and MLP also function effectively. Following cross-validation, ExtraTreesRegressor, LSTM, and MLP exhibit constant performance with tiny increases, XGBoost retains excellent performance with minor error reductions, and GBoost shows slight improvements (0.941 and 0.943). In general, cross-validation enhances model generalization; ExtraTreesRegressor is still susceptible to overfitting, although XGBoost and GBoost are dependable for energy prediction.

In terms of training time and computational cost, classical machine learning models such as Polynomial Regression and Elastic Net exhibit very short training times due to their simple optimization procedures. Ensemble models such as Random Forest, Extra Trees, Gradient Boosting, and XGBoost require moderately higher computational cost because they construct multiple decision trees sequentially or in parallel. In contrast, deep learning models including MLP, CNN, and LSTM involve iterative gradient-based optimization and multiple network layers, resulting in higher computational cost and longer training times. Among them, the LSTM model requires the highest computational resources due to its recurrent architecture designed to capture temporal dependencies. On the other hand, due to its limited number of parameters (p, d, q) and simpler statistical estimation process, ARIMA requires short training time ($\sim$1 s) with low computational cost. Table 3 summarizes the training time, computational cost (from lowest to highest) as well as the complexity of each model where:

- n = number of samples
- p = number of features
- E = number of epochs
- h = hidden units
- k = convolution kernel size.

5. DISCUSSION

This study evaluated several regression models for energy consumption prediction. Polynomial Regression (degree 2) showed stable behavior with similar training and test R^2 values, indicating limited overfitting. Elastic Net also demonstrated balanced performance ($R^2 \approx 0.83$) and good generalization due to its regularization capability. Tree-based ensemble models provided stronger predictive performance. Random Forest and Gradient Boosting achieved higher R^2 scores and lower errors, reflecting their ability to capture nonlinear relationships. After hyperparameter optimization, XGBoost further improved predictive accuracy, highlighting the importance of tuning for better generalization.

SVR delivered moderate results but remained less accurate than ensemble approaches. Overall, the optimized ExtraTrees regressor achieved the best performance, reaching $R^2 = 0.997$ and RMSE = 0.011. These results confirm that ensemble methods, particularly ExtraTrees, are well suited for modeling complex energy consumption patterns.

TABLE 3. Training time, computational cost, and theoretical complexity of the evaluated models.

Regression model	Training time	Computational cost	Complexity
1. Polynomial	∼0.4 s	Low	$O(n^2)$
2. ElNet	∼0.7 s	Low	$O(n \times p)$
3. SVR	∼6.5 s	Medium	$O(n^2 - n^3)$
4. RF	∼4.2 s	Medium	$O(\text{trees} \times n \log n)$
5. ExtraT	∼3.8 s	Medium	$O(\text{trees} \times n \log n)$
6. GBoost	∼8.9 s	Medium–High	$O(\text{trees} \times n \log n)$
7. XGBoost	∼7.5 s	Medium–High	$O(\text{trees} \times n \log n)$
8. MLP	∼22 s	High	$O(E \times n \times p)$
9. CNN	∼38 s	High	$O(E \times n \times k^2)$
10. LSTM	∼52 s	High	$O(E \times n \times h^2)$
ARIMA	∼1 s	Low	$O(n^2)$

Deep learning models such as LSTM and CNN achieved competitive or improved predictive performance; however, they require significantly longer training time and higher computational complexity compared to classical machine learning models. This trade-off highlights the balance between predictive accuracy and computational efficiency when selecting forecasting models.

While ARIMA remains computationally efficient and interpretable for simpler forecasting scenarios, the superior performance of ensemble methods in terms of R^2 and RMSE demonstrates their stronger capability for modeling complex energy consumption behaviors. Therefore, for smart building applications characterized by nonlinear patterns and multiple influencing factors, advanced ML models provide a more robust and scalable alternative to traditional statistical time series techniques.

6. CONCLUSION

This study provides a comprehensive comparison of ten machine learning optimization algorithms for predicting energy usage in smart buildings, both before and after cross-validation. Our multifaceted approach encompassed a range of models, from simpler regression techniques to more sophisticated deep learning architectures, showcasing the strengths of each and their applicability to diverse datasets. This broad selection of models allowed us to explore and potentially identify various patterns and relationships within the energy consumption data, representing a key strength of our methodology. To further enrich the comparison, the obtained results were also compared with those of the ARIMA model. Our findings reveal that ExtraTrees, LSTM, and ensemble models consistently achieved the highest training and testing scores, outperforming the other evaluated algorithms. These results offer valuable insights into the effectiveness and reliability of these models for accurate energy usage prediction in smart building environments.

Future work will focus on developing an online, continually learning model with online hyperparameter tuning to adapt to new data and maintain performance. We plan to explore advanced deep learning architectures, refine data pre-processing, and build robust real-time prediction systems. Improving model explainability, quantifying uncertainty, and ensuring scalability and sustainability are also key goals. These efforts will build upon our promising results to create more sophisticated and impactful energy prediction models for smart buildings.

DATA AVAILABILITY STATEMENT

The research data associated with this article are available in Kaggle, under the reference https://www.kaggle.com/datasets/loveall/appliances-energy-prediction.

References

[1] P. Jiang, Z. Wang, X. Li, X.V. Wang, B. Yang and J. Zheng, Energy consumption prediction and optimization of industrial robots based on LSTM. *J. Manuf. Syst.* **70** (2023) 137–148.

[2] M. Sarkar, A. Majumder, S. Bhattacharya and B. Sarkar, Optimization of energy cycle under a sustainable supply chain management. *RAIRO-Oper. Res.* **57** (2023) 2177–2196.

[3] L. Wang, D. Xie, L. Zhou and Z. Zhang, Application of the hybrid neural network model for energy consumption prediction of office buildings. *J. Build. Eng.* **72** (2023) 106503.

[4] S. Tabasi, A. Aslani and H. Forotan, Prediction of energy consumption by using regression model. *Comput. Res. Prog. Appl. Sci. Eng.* **2** (2016) 110–115.

[5] O. Valgaev, F. Kupzog and H. Schmeck, Low-voltage power demand forecasting using k-nearest neighbors approach, in 2016 IEEE Innovative Smart Grid Technologies-Asia (ISGT-Asia). IEEE (2016) 1019–1024.

[6] W. Hariri, H. Tabia, N. Farah, D. Declercq and A. Benouareth, Geometrical and visual feature quantization for 3D face recognition, in International Conference on Computer Vision Theory and Applications, Vol. 6. SciTePress (2017) 187–193.

[7] E. Elbeltagi and H. Wefki, Predicting energy consumption for residential buildings using ann through parametric modeling. *Energy Rep.* **7** (2021) 2534–2545.

[8] M. Ilbeigi, M. Ghomeishi and A. Dehghanbanadaki, Prediction and optimization of energy consumption in an office building using artificial neural network and a genetic algorithm. *Sustain. Cities Soc.* **61** (2020) 102325.

[9] A. Mishra, H.R. Lone and A. Mishra, DECODE: Data-driven energy consumption prediction leveraging historical data and environmental factors in buildings. *Energy Build.* **307** (2024) 113950.

[10] M.M.S. Yazdan, M. Khosravia, S. Saki and M.A. Al Mehedi, Forecasting energy consumption time series using recurrent neural network in tensorflow. Technical Note (2022).

[11] S. Li and R. Li, Comparison of forecasting energy consumption in Shandong, China using the ARIMA model, GM model and ARIMA-GM model. *Sustainability* **9** (2017) 1181.

[12] Z. Wang, Y. Wang, R. Zeng, R.S. Srinivasan and S. Ahrentzen, Random forest based hourly building energy prediction. *Energy Build.* **171** (2018) 11–25.

[13] A. Lotfipoor, S. Patidar and D.P. Jenkins, Short-term forecasting of residential electricity demand using CNN-LSTM, in 2nd IBPSA-Scotland uSIM Conference: Building to Buildings–Urban and Community Energy Modelling (2020).

[14] P. Geurts, D. Ernst and L. Wehenkel, Extremely randomized trees. *Mach. Learn.* **63** (2006) 3–42.

[15] S. Hochreiter and J. Schmidhuber, Long short-term memory. *Neural Comput.* **9** (1997) 1735–1780.

[16] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization – 1958 (2021).

[17] F.M. de Souza, J. Grando and F. Baldo, Adaptive fast XGBoost for regression, in Brazilian Conference on Intelligent Systems. Springer, Berlin (2022) 92–106.

[18] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, R. Tibshirani and J. Friedman, Boosting and additive trees. The Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer, Berlin (2009) 337–387.

[19] V.E. Balas, R. Kumar and R. Srivastava, Recent Trends and Advances in Artificial Intelligence and Internet of Things. Springer, Berlin (2020).

[20] R. Venkatesan and B. Li, Convolutional Neural Networks in Visual Computing: A Concise Guide. CRC Press (2017).

[21] L. Breiman, J. Friedman, R. Olshen and C. Stone, CART-Classification And Regression Trees. CRC Press (1984).

[22] G. Biau and E. Scornet, A random forest guided tour. *Test* **25** (2016) 197–227.

[23] H. Zou and T. Hastie, Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **67** (2005) 301–320.

[24] L. Wang, J. Zhu and H. Zou, The doubly regularized support vector machine. *Stat. Sin.* **16** (2006) 589–615.

[25] M. Liu and B.C. Vemuri, A robust and efficient doubly regularized metric learning approach, in Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part IV 12. Springer, Berlin (2012) 646–659.

[26] W. Shen, J. Wang and S. Ma, Doubly regularized portfolio with risk minimization, in Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 28 (2014).

[27] P. Milanez-Almeida, A.J. Martins, R.N. Germain and J.S. Tsang, Cancer prognosis with shallow tumor RNA sequencing. *Nat. Med.* **26** (2020) 188–192.

[28] Y.-W. Chang, C.-J. Hsieh, K.-W. Chang, M. Ringgaard and C.-J. Lin, Training and testing low-degree polynomial data mappings via linear SVM. *J. Mach. Learn. Res.* **11** (2010).

[29] J.D. Gergonne, The application of the method of least squares to the interpolation of sequences. *Hist. Math.* **1** (1974) 439–447.
[30] S.M. Stigler, Gergonne's 1815 paper on the design and analysis of polynomial regression experiments. *Hist. Math.* **1** (1974) 431–439.
[31] K. Smith, On the standard deviations of adjusted and interpolated values of an observed polynomial function and its constants and the guidance they give towards a proper choice of the distribution of observations. *Biometrika* **12** (1918) 1–85.
[32] H. Drucker, C.J. Burges, L. Kaufman, A. Smola and V. Vapnik, Support vector regression machines, in Advances in Neural Information Processing Systems, Vol. 9. MIT Press (1996).
[33] A.J. Smola and B. Schölkopf, A tutorial on support vector regression. *Stat. Comput.* **14** (2004) 199–222.
[34] Kag dataset, https://www.kaggle.com/datasets/loveall/appliances-energy-prediction. Accessed: 2024-06-12.
[35] N.W. Azani, C.P. Trisya, L.M. Sari, H. Handayani and M.R.M. Alhamid, Performance comparison of ARIMA, LSTM and SVM models for electric energy consumption analysis. *PREDATECS* **1** (2024).
[36] M. Sachin, M. Paily Baby and A. Sudharson Ponraj, Analysis of energy consumption using RNN-LSTM and ARIMA model, in *Journal of Physics: Conference Series*, Vol. 1716. IOP Publishing (2020) 012048.