ENERGY CONSUMPTION FORECASTING MODELS FOR SMART GRIDS: A STATE-OF-THE-ART REVIEW AND APPLICATION PERSPECTIVES.

Abstract

Energy consumption forecasting plays a crucial role in optimizing the operation of modern power systems, particularly in the context of smart grids and renewable energy integration. Accurate prediction models enable utilities and policymakers to plan generation, balance demand, and improve energy efficiency.

This paper presents a comprehensive review of forecasting approaches ranging from traditional statistical methods (ARIMA, SARIMA) to machine learning algorithms (SVM, Random Forest, KNN) and deep learning architectures (ANN, MLP, LSTM). The main contribution of this review is to highlight the strengths, limitations, and application contexts of these models according to the nature of available data, the time horizon, and computational constraints.

A comparative synthesis of recent studies (2020–2024) is provided, focusing on evaluation metrics such as RMSE, MAE, MAPE, and R^2. Finally, perspectives are proposed for hybrid and data-driven approaches, particularly for developing countries where data scarcity and climatic variability remain major challenges.

1. INTRODUCTION

Energy demand forecasting has become an essential task for ensuring the reliability, sustainability, and efficiency of power systems. The increasing integration of renewable energy sources—especially solar and wind—has introduced significant variability in generation, requiring more accurate and adaptive forecasting tools.

Traditional statistical models such as ARIMA and SARIMA have long been used for time series forecasting because of their simplicity and interpretability. However, these models struggle to capture nonlinear relationships and external factors such as weather or human behavior.

With the emergence of artificial intelligence, machine learning and deep learning models now offer powerful alternatives capable of handling large and complex datasets. The challenge is no longer only to predict future demand but also to understand the dynamic interactions among consumption patterns, environmental conditions, and socio-economic factors.

This review aims to present an updated synthesis of energy consumption forecasting models, classify them into major methodological families, and identify the trends and research gaps that remain to be addressed.

2. METHODOLOGY OF THE REVIEW

The review was conducted according to a structured approach inspired by the PRISMA methodology. The main steps included:

- **Definition of the research scope:**
  The review focuses on forecasting models applied to energy consumption (electricity, building load, national demand, etc.) rather than generation forecasting. The main objective was to identify the different methodological approaches (statistical, machine learning, and deep learning) and to compare their performance and applicability.
- **Search strategy:**
  A documentary search was conducted using the following major scientific databases: **ScienceDirect**, **IEEE Xplore**, **SpringerLink**, **Google Scholar**, and **Scopus**.
  The search was based on combinations of keywords such as:
  "energy consumption forecasting," "ARIMA," "machine learning," "deep learning," "ANN," "LSTM," "smart grid," and **"hybrid model."**
- **Selection criteria:**
  - **Inclusion criteria:** peer-reviewed publications from 2020 to 2024, focusing on energy consumption prediction using quantitative models, and presenting measurable performance indicators (RMSE, MAE, MAPE, or R^2).
  - **Exclusion criteria:** studies dealing exclusively with renewable generation forecasting (e.g., PV or wind output), purely theoretical models without validation data, or publications lacking performance comparison.
- **Data extraction and classification:**
  Each selected article was analyzed according to the following parameters:
  - Forecasting model or algorithm used (ARIMA, SARIMA, SVM, RF, ANN, LSTM, etc.);
  - Type and source of data;
  - Forecasting horizon (short, medium, or long term);
  - Evaluation metrics;
  - Main findings and limitations.

  The information collected was then organized into summary tables to facilitate comparison between models.
- **Analysis and synthesis:**
  The models were grouped into three main families — statistical, machine learning, and deep learning approaches — and analyzed according to their advantages, drawbacks, and context of application. Comparative results were interpreted in terms of accuracy, computational cost, data requirements, and adaptability to developing-country contexts.

3. STATISTICAL MODELS FOR ENERGY CONSUMPTION FORECASTING

Statistical models have historically been the foundation of energy consumption forecasting, particularly because of their interpretability and low computational requirements. These models establish mathematical relationships between past and future values of a time series, often assuming stationarity and linearity of the data.

3.1 Autoregressive Integrated Moving Average (ARIMA) Model

3.1.1. Principle and Mathematical Formulation

The Autoregressive Integrated Moving Average (ARIMA) approach, also known as the Box–Jenkins model, was developed by George Box and Gwilyn Jenkins.

It is a widely used time series forecasting method, particularly applied in electric load prediction.

The model combines Autoregressive (AR) and Moving Average (MA) processes. In other words, ARIMA (p, d, q) relies on three main components:

- **Autoregression (AR):** described by the parameter (p); this component captures the dependence between a current observation and previous lagged observations.

In the context of load forecasting, it corresponds to predicting future electricity consumption for the next hour or anticipating an upcoming consumption peak.

It is expressed as:

The following equation represents the autoregressive (AR) process used in time series modeling:

$$Y_t = \alpha Y_{t-1} + \alpha Y_{t-2} + \cdots + \alpha Y_{t-p} + \varepsilon_t \qquad (1)$$

Where:

• α : autoregressive coefficient

• Y_t : value of the variable at time t

• ε_t : white noise or random error term

Integration (I): represented by parameter (d), defines the differencing degree for achieving stationarity.

It is governed by:

$$Y_t = (1 - B)^d X_t = \varepsilon_t (2)$$

3. Moving Average (MA): represented by the parameter (q).

It is given by:

$$Y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta q \varepsilon_{t-} q (3)$$

(Fathin et al., 2021; Pierre et al., 2023; Szostek et al., 2024)

3.1.2. Applications and Performance of the ARIMA Model

According to Fathin et al. (2021), using non-stationary hourly energy consumption data from Kaggle.com, the ARIMA model achieved an RMSE = 753.98 and an accuracy of 94.7% in consumption forecasting. Similarly, Monia&Jaleleddine (n.d.) used data from the National Institute of Statistics of Tunisia covering 1979–2008, and predicted energy demand from 2008 to 2020, showing an expected increase of 17.14%.

3.1.3. Limitations of the ARIMA Model

The ARIMA model encounters difficulties when modeling **nonlinear relationships**, as it assumes **linearity** between past and future observations.

This becomes problematic for datasets with nonlinear dependencies.

It also requires the **series to be stationary**, meaning its statistical properties (mean and variance) remain constant over time.

The model struggles to capture **seasonal behaviors** unless extended to the **Seasonal ARIMA (SARIMA)** form.

(**Learn Statistics Easily, 2024, August 29 – "What is the ARIMA Model – Complete Guide."**)

3.2 SARIMA Model

3.2.1 Principle and Mathematical Formulation

The Seasonal ARIMA (SARIMA) model is highly suitable for studying time series that display both short-term and long-term dependencies along with seasonal patterns. It extends the ARIMA structure by integrating seasonal terms. It is written as:

$$\text{ARIMA}(p, d, q) \times (P, D, Q)_m \qquad (4)$$

(p, d, q) represents the non-seasonal part, $(P, D, Q)_m$ represents the seasonal part of the model.

3.2.2 Applications and Strengths of the SARIMA Model

Yin et al. (2023) applied the SARIMA model to forecast medium- and long-term electricity demand in Yunnan Province, China, based on 2008–2018 data exhibiting strong seasonality. The study predicted electricity consumption for 2019–2020. When compared with Holt–Winter, LSTM, and ARIMA models using the MAPE metric, SARIMA achieved a MAPE of 6.05%, versus 9.18%, 10.67%, and 13.87% respectively.

Similarly, HamsaHadi Mohammed, Aziza Asem, and Hazem El-Bakry (2024) used SARIMA to forecast UK energy consumption from 2009 to 2023 (UK National Grid data, 30-minute intervals). The SARIMA model achieved a MAPE of 13.84%, indicating moderate accuracy.

3.2.3 Limitations of the SARIMA Model

Despite its performance, SARIMA presents several theoretical and practical constraints:

- **Stationarity requirement:** the series must be stationary, implying complex preprocessing (differencing, outlier removal) that can be error-prone (**Wang et al., 2021**).
- **Assumption of regular seasonality:** performance decreases sharply when the series exhibits irregular or evolving seasonal patterns (**Wang et al., 2021**).
- **Difficulty handling random fluctuations:** SARIMA performs poorly when random variations dominate the time series (**Andoh et al., 2021**).
- **Linearity limitation:** as a linear model, it cannot capture complex nonlinear dependencies, which restricts its use for dynamically interacting variables (**Hamsa et al., 2024**).

*3.3 **Comparison between Traditional Statistical Models (ARIMA and SARIMA)***

Table 1. Comparison between traditional statistical models (ARIMA and SARIMA)

Model	Strengths	Weaknesses
ARIMA	• Suitable for stationary time series • Simple to implement • Provides good medium-term accuracy	• Difficulty in handling nonlinear relationships • Requires rigorous preprocessing • Limited performance with irregular data
SARIMA	• Incorporates seasonality • Performs well on seasonal data	• Requires strong stationarity • Inefficient for irregular data • Ineffective for nonlinear relationships

4. NEURAL NETWORK-BASED MODELS

4.1 Artificial Neural Networks (ANN)

Artificial Neural Networks (ANN) represent a class of machine learning models inspired by the structure and function of the human brain. They are composed of interconnected processing units called neurons, which are organized in layers and capable of learning complex nonlinear relationships from data. ANN models are increasingly applied in energy forecasting because they can adapt to time-varying patterns and integrate multiple influencing variables.

4.1.1 Principle and Mathematical Representation

During the training of the ANN, the output of the output layer is compared with the target, and the difference between the two is calculated as an error. This error is then reduced to an acceptable level, and the weights are updated until satisfactory prediction results are achieved. Activation Functions

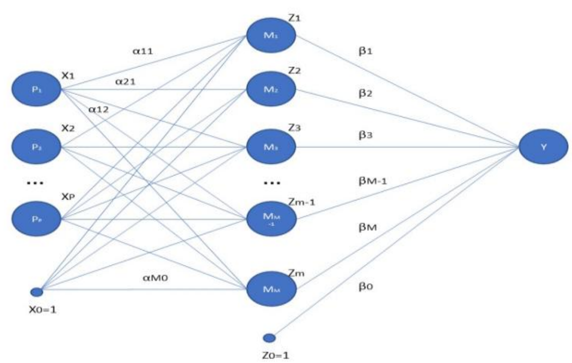

Figure 1. Architecture of the Artificial Neural Network (ANN)

4.1.2 Applications of ANN in Energy Consumption Forecasting

According to Ferrero Bermejo et al. (2019), ANN-based models offer several advantages, such as:
• A high correlation coefficient.
• Rapid adaptation and flexibility to behavioral patterns (including pattern recognition capability and error tolerance, even in the presence of missing data or noise).
• Better suitability for complex and nonlinear problems.

However, the work of **Dahmani et al. (2023)** highlighted two major limitations of ANN models:

- **Determination of network size:**
  This refers to choosing the appropriate number of layers and the number of neurons per layer.
  An insufficient number of hidden neurons leads to difficulties during the learning phase, whereas an excessive number results in long training times with only marginal improvement in prediction accuracy.
  Furthermore, estimating the synaptic weights becomes more complex.
  To determine the optimal architecture, it is generally necessary to perform multiple experiments and evaluate the estimation error each time.

- **Obtaining optimal synaptic weights:**
  The optimization of weights remains a challenge, as the learning process may converge slowly or get trapped in local minima depending on initialization and data characteristics.

4.2 The LSTM Model

The backpropagation algorithm is the most widely used supervised learning method for training Multilayer Perceptron (MLP) models. It consists in adjusting the weights of the network to minimize the prediction error between the desired output and the actual output produced by the model. The process takes place in two main phases: the forward propagation and the backward propagation.

4.2.1 Principle and Architecture

The **Long Short-Term Memory (LSTM)** network is a type of **Recurrent Neural Network (RNN)** proposed by **Hochreiter and Schmidhuber (1997)**, designed to learn **long-term dependencies** in sequential data. The key idea behind the LSTM architecture is the introduction of a **memory unit** called the *cell state*, which can store information over long periods and selectively **forget** or **update** information as needed.

This improvement allows LSTM networks to better capture **long-term dependencies** when processing **time series data**, thereby overcoming the limitations of traditional RNNs, which often suffer from **vanishing** or **exploding gradient problems** when dealing with long sequences (*Rajagukguk et al., 2020b; Y. Zhu, 2023*).

Each LSTM cell involves **three types of gates** that control the flow of information and the cell state:

- **Forget Gate:**
  It outputs a number between 0 and 1, where **1 means "completely keep this information"** and **0 means "completely forget it."**
  This gate determines which information from the previous cell state should be retained or discarded.

- **Input (Memory) Gate:**
  This gate decides which new information will be stored in the cell.
  First, a **sigmoid layer**, called the **input gate layer**, determines which values will be updated.
  Then, a **tanh layer** creates a vector of **candidate values** that can be added to the cell state.
- **Output Gate:**
  This gate determines what the cell will output.
  The output value is computed based on both the **cell state** and the **filtered and newly added information** (**Siami-Namini et al., 2018**).

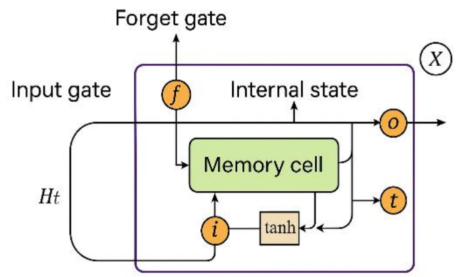

Figure 2. Architecture of the Long Short-Term Memory (LSTM) Network

4.2.2 Applications and Strengths

Nyangaresi (2024) proposed a method for forecasting energy consumption in smart grids using **Long Short-Term Memory (LSTM)** networks, comparing their performance with other approaches such as statistical models like **ARIMA** and machine learning algorithms such as **Decision Trees**.
The results demonstrated that LSTM networks are highly effective in predicting energy consumption with a high level of accuracy.
Indeed, LSTMs are less sensitive to long-term dependencies and are well suited for modeling **nonlinear relationships**.

Compared with other models, the LSTM network achieved a **Mean Absolute Error (MAE)** of **5.62** and a **coefficient of determination (R^2)** of **0.89**, whereas the Decision Tree model achieved **8.85** and **0.78**, and ARIMA recorded **12.6** and **0.68**, respectively.

Furthermore, **Siami-Namini et al. (2018)** compared the forecasting accuracy of **ARIMA** and **LSTM** models in time series prediction.
Both techniques were implemented and applied to a **financial dataset**, and the results showed that LSTM significantly outperformed ARIMA, reducing error rates by **84% to 87%**.

In addition, **Joy et al. (2024)** presented a new approach to forecasting **daily solar irradiance** in **Dhaka, Bangladesh**, using a **Long Short-Term Memory (LSTM)** neural network that effectively captures temporal dependencies in solar data.
The model utilized **historical meteorological data** and **sunshine duration** covering the period **2000–2022**, achieving a **Root Mean Square Error (RMSE)** of **43.48 W/m²**, outperforming the **Random Forest Regressor** model, which had an RMSE of **46 W/m²**.

4.2.3 Limitations

Despite its strengths, the **LSTM model** still presents several limitations:

- **High computational cost:**
  LSTMs require considerable processing time, which raises concerns when deployed on a large scale or in environments with limited computational resources.
  Moreover, due to the scarcity of diverse datasets, the model's ability to generalize to extreme scenarios—such as sudden demand surges or severe weather conditions—remains limited.
- **High sensitivity to poor performance due to improper hyperparameter configuration:**
  For this reason, **Dhake et al. (2023)** proposed two new algorithms for **hyperparameter optimization** in LSTM networks, along with a **data decomposition technique** based on the **Fast Fourier Transform (FFT)**.
  These algorithms were trained on raw data collected from **22 sites** of a solar power plant in southern India, comprising **154,277 solar irradiance entries (W/m²)** as well as net received irradiance.
  The results showed a significant improvement in model performance, with the **goodness-of-fit** increasing from **81.20% to 95.23%**, and a **53.42% reduction in RMSE** for 90-minute-ahead forecasting after applying the optimized training workflow.

4.3 The MLP Model

4.3.1 Principle and Architecture

The simplest architecture of a neural network is the **Multilayer Perceptron (MLP)**.
It is also known as a **feed-forward neural network**, in which all neurons in each layer are directly connected to all neurons in the subsequent layer, as shown in **Figure 3**.
The MLP consists of multiple perceptrons organized in a layered structure (including **input**, **hidden**, and **output** layers) capable of approximating future values based on a given input—particularly in **renewable energy forecasting applications** (**Rahman et al., 2021**).

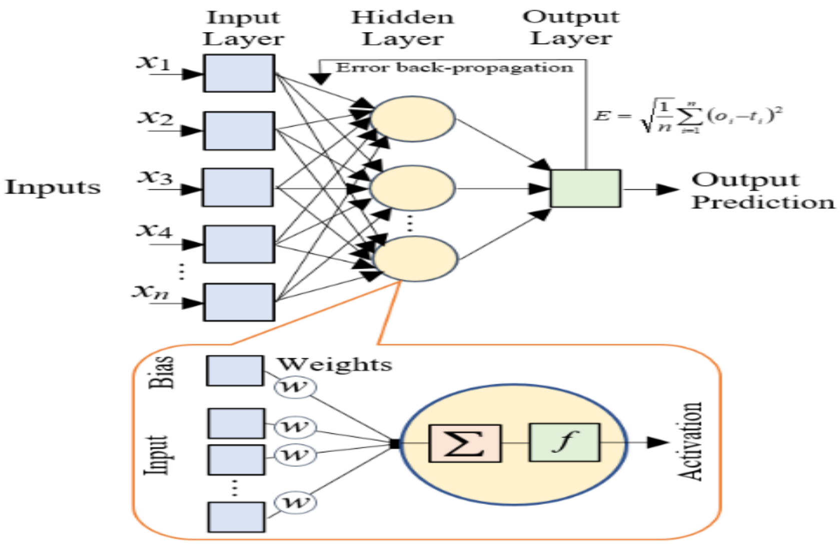

Figure 3. Structure of the Multilayer Perceptron (MLP)

In the MLP network, information flows **only from the input layer to the output layer**.
Each layer contains a variable number of neurons, and the output layer represents the number of outputs of the system (**Kahaji et al., 2013**).
The signal produced by the output layer depends on the **weight matrix**, **bias terms**, **activation function**, and the **vector of output signals** (**Drałus et al., 2023**).

In the output layer of the network, the output of the **j**[th] neuron is defined by the following equation:

$$y_j(t) = f_2\left[\sum_{k=0}^{K} w_{jk}^{(2)} f_1\left[\sum_{i=0}^{I}\left(w_{ki}^{(1)} x_i(t) + b_k^{(1)}\right)\right] + b_j^{(2)}\right] \qquad (5)$$

where :

- xi(t) represents the input to the **i**[th] neuron,
- $y_j(t)$ represents the output of the **j**[th] neuron,
- $w_{ki}^{(1)}$and$w_{jk}^{(2)}$ are the weights in their respective layers,
- $b_k^{(1)}$ and $bj^{(2)}$are the biases of neurons in successive layers,
- f (t) corresponds to the activation function applied to the input data sample.

The **MLP structure** is particularly advantageous for **electricity consumption forecasting**, as it:

- effectively models **nonlinear relationships**,

- offers high **flexibility** due to a wide range of **hyperparameters** (such as network architecture and activation functions),
- can **handle missing data** efficiently,
- and has the ability to **learn and adapt** to changing patterns in complex datasets (**Cordeiro-Costas et al., 2023**).

4.3.2 Application and Results in Energy Forecasting

Niazai et al. (2022) compared the **MLP** and **ARIMA** models using time series forecasting data collected between **April 2011 and July 2022** to predict **monthly electricity consumption** in the **Kunduz province** for the period **2022–2025**.
The **Mean Absolute Percentage Error (MAPE)** and **Mean Absolute Error (MAE)** metrics were used to evaluate and compare the predictive performance of both models.
The results showed that the **MLP model achieved an accuracy of 82%**, whereas the **ARIMA model** achieved a lower accuracy of **78.5%**.

In another study, **Lee et al. (2023)** applied **MLP**, **linear SVR**, **RBF-SVR**, and **polynomial SVR** algorithms to identify the most accurate model for predicting **electricity** and **LNG consumption** in a **food manufacturing plant**.
The **MLP model** demonstrated the **highest prediction accuracy** for electricity consumption, achieving a **Coefficient of Variation of the RMSE (CvRMSE)** of **17.35%** and an **R^2 of 0.84**.
Moreover, MLP was also selected among the four models as the most suitable for forecasting **LNG consumption**.

Across **eight case analyses** (four electricity consumption models and four LNG consumption models), the **CvRMSE** ranged between **12.52% and 22.10%**, and **R^2** values ranged between **0.71 and 0.88**, confirming that **MLP-based models are highly applicable** for predicting energy consumption in industrial systems, particularly in the **target food processing plant**.

Finally, **Jannah et al. (2023)** compared the predictive capabilities of the **Multilayer Perceptron (MLP)** and the **Convolutional Long Short-Term Memory (CNN-LSTM)** networks for **photovoltaic (PV) energy output forecasting**.
Both models were trained using **13 input features** from datasets collected across **10 PV sites** in **Hebei Province, China**, spanning **300 days** (from **July 1, 2018** to **June 13, 2019**).
The **CNN-LSTM model** demonstrated superior performance, achieving **Mean Absolute Error (MAE)**, **Mean Squared Error (MSE)**, and **Root Mean Squared Error (RMSE)** values of **0.088**, **0.051**, and **0.227**, respectively, compared to **0.260**, **0.156**, and **0.395** for the MLP model.

These findings highlight the **potential of CNN-LSTM architectures** to enhance**solar energy forecasting** and facilitate improved management of **renewable energy resources**.

4.4 Comparison between Neural Network Models

Table 2. Comparison between neural network models (ANN, LSTM, MLP)

Model	Strengths	Limitations
ANN	• Automatically learns nonlinear relationships. • Adapts well to noisy data.	• Long training time. • Sensitive to network configuration (number of layers and neurons).
LSTM	• Captures long-term dependencies. • Handles nonlinearities effectively.	• High computational cost and processing time. • Sensitive to hyperparameter configuration.
MLP	• Simple structure. • Flexible for modeling nonlinear relationships.	• Less accurate than certain hybrid models such as **CNN-LSTM**.

5. SUPERVISED MACHINE LEARNING MODELS

5.1 The SVM Model

5.1.1 Principle and Architecture of the SVM Model

The **Support Vector Machine (SVM)** was introduced by **Vapnik (1995)**, based on **Statistical Learning Theory (SLT)** and the principle of **Structural Risk Minimization (SRM)**.
It can be applied not only to **nonlinear regression estimation problems** but also to **time series forecasting tasks**, such as **energy consumption prediction** (**Ahmad et al., 2014**).

The operational principle of the SVM involves the construction of a **hyperplane** that separates data into distinct categories.
This hyperplane is selected to **maximize the margin** between the two classes — the margin being the distance between the hyperplane and the **nearest data points** of each class, known as **support vectors**.
The SVM algorithm searches for the hyperplane with the **largest possible margin**, thus achieving the best generalization performance (**Pokharel&Ghimire, 2023**).

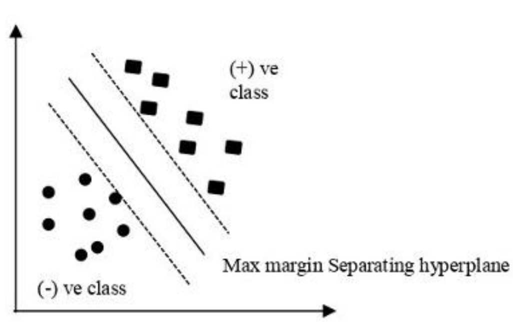

Figure 5. Diagram of the Support Vector Machine (SVM) Model

According to **Guido et al. (2024)**, the mathematical representation of this hyperplane can be expressed as follows:

$$w^T \phi(x_i) + b = 0 \quad (6)$$

where:

- w is the **weight vector**,
- x is the **input feature vector**,
- b is the **bias term**.

To maximize the margin between the hyperplane and the nearest data points (support vectors), the optimization problem can be formulated as:

$$min \frac{1}{2} ||w||^2 + C \sum_{i=1}^{l} \xi_i \quad (7)$$

Subject to:

$$\begin{aligned} & y_i(w^T \phi(x_i) + b) - 1 + \xi_i \geq 0 \quad i = 1, \ldots, l \\ & \xi_i \geq 0 \qquad\qquad i = 1, \ldots, l \end{aligned} \quad (8)$$

where:

- yi is the **class label** associated with the data point xix_ixi,
- w is the **weight vector**,
- b is the **bias term**.

5.1.2 Applications and Strengths

High forecasting accuracy:

Qiong Li et al. (2010) applied four different modeling methods—**Backpropagation Neural Network (BPNN)**, **Radial Basis Function Neural Network (RBFNN)**, **General Regression Neural Network (GRNN)**, and **Support Vector Machine (SVM)**—to predict the **annual energy consumption** of **59 residential buildings** in China. The results showed that **SVM** achieved the **best performance**, with an **RMSE of 2.395** and an **MRE of 1.895** on the test dataset, followed by **GRNN (RMSE = 5.237, MRE = 4.912)**. In contrast, the **BPNN (RMSE = 14.462, MRE = 13.635)** and **RBFNN (RMSE = 12.440, MRE = 11.204)** models exhibited much higher errors.

These results indicate that **SVM** and **GRNN** are highly effective for **annual energy consumption forecasting** in buildings.

Flexibility and ease of implementation:

SVMs have been successfully applied in various fields, including **image retrieval**, **regression problems**, **fault diagnosis**, and **text detection** (**Zendehboudi et al., 2018**).

Their **versatility** and **robustness** make them suitable for a wide range of applications in energy systems and beyond.

Model complexity control:

SVMs incorporate **regularization parameters** to control model complexity and prevent overfitting.

The algorithm is also **less sensitive to hyperparameter selection** than many other models, which simplifies the tuning process (**Y. Liu et al., 2024; Zendehboudi et al., 2018**).

5.1.3 Limitations of the SVM Model

- **Dependence on kernel selection:**

The performance of SVM models largely depends on the **choice of kernel function** and its parameters, which can significantly affect prediction accuracy.

Additionally, selecting the **optimal subset of input features** plays a critical role in determining the quality of the model's predictions (**Park & Yang, 2024**).

- **High computational cost:**

One of the main drawbacks of SVMs is their **high computational burden**, stemming from the **constrained optimization problem** that must be solved during training.

This can considerably increase processing time for large datasets.

To address this issue, an alternative approach called the **Least Squares Support Vector Machine (LSSVM)** was developed.

The **LSSVM** converts the inequality constraints of the original SVM into **equality constraints**, simplifying computation (**Ahmad et al., 2014**).

For example, **Xuemei et al. (2009)** applied **LSSVM** to improve the **computational efficiency** of **building cooling load prediction**.

A comparison between the models showed that **LSSVM (RMSE = 11.84)** outperformed **BPNN (RMSE = 5.86)**, providing better prediction accuracy and faster training time (**Ahmad et al., 2014**).

Not effective for large-scale datasets:

Traditional SVMs are generally **inefficient when handling very large amounts of data**, due to their computational and memory demands.

5.2 Random Forest

5.2.1 Principle and Architecture

Random Forest (RF) is an **ensemble learning** method used for both **regression** and **classification** tasks by constructing multiple independent decision trees.

It is primarily based on the idea of **Bagging (Bootstrap Aggregating)**, which employs a random bootstrap sampling method to select data subsets.

The key concept behind bagging is to average the results of several noisy but approximately unbiased base models in order to **reduce variance**.

Decision trees, being noisy and weakly biased models when deep enough, are therefore **excellent candidates** for bagging (**Dudek, 2022; D. Liu & Sun, 2019**).

For the weighting aspect, the Random Forest algorithm uses **uniform sampling**, in which all prediction functions have **equal weights**, improving sample quality and facilitating **parallel computation** of each prediction function (**D. Liu & Sun, 2019**).

5.2.2 Applications and Results

Studies by **Pokharel&Ghimire (2023b)** and **Sarswatula et al. (2022)** on **energy consumption modeling** using machine learning approaches demonstrated that the **Random Forest (RF)** model effectively **reduces overfitting** compared to individual decision trees and **automatically handles missing values** without requiring data normalization.

Overfitting occurs when a model fits the training data too closely, thus limiting its ability to generalize and perform efficiently on new data.

Their results showed that **RF achieved the best performance**, with an **R^2 of 0.869**, outperforming models such as **K-Nearest Neighbor (KNN)** and **Support Vector Machine (SVM)**.

Moreover, **Random Forest** operates **without extensive hyperparameter tuning** and is one of the **fastest machine learning algorithms**, providing strong predictive capabilities for **regression problems** (**Shin & Woo, 2022**).

However, the model exhibits limitations when applied to **time series forecasting**, especially when major variations are **internally dependent** (e.g., influenced by lagged events or past states).

In such cases, most target variations result from **endogenous temporal effects**, which Random Forest struggles to capture effectively (**T. Zhu, 2020**).

Additionally, because RF combines predictions from multiple decision trees, it is **less prone to overfitting** than many other machine learning algorithms (**Pokharel&Ghimire, 2023b**).

5.3 Decision Tree

5.3.1 Principle and Architecture

In machine learning, the **Decision Tree algorithm** is one of the most widely used techniques.
It generates a **tree-like flow diagram** that divides data samples into distinct classes.
A decision tree consists of a **root node**, **internal nodes**, and **terminal nodes (leaves)** (**Chockalingam, 2018**).

- **Internal nodes** represent a **test** on a given attribute,
- **Branches** represent the outcomes of these tests, and
- **Terminal nodes (leaves)** represent the final **class labels or decisions** after evaluating all attributes.

The path from the root to the leaf node represents a **classification rule**.
At each node, the tree splits further depending on the attribute's condition.
To predict an outcome, the decision path follows from the **root node** (beginning) to the **leaf node** (end), which contains the **final decision output** (**Arora et al., 2020**).

There are several well-known Decision Tree algorithms, including:
Iterative Dichotomizer 3 (ID3), **Successor of ID3 (C4.5)**, **Classification and Regression Tree (CART)**, **Random Forest**, and **XGBoost** (**Charbuty&Abdulazeez, 2021**).
When applying decision trees to **time series analysis**, the **CART** algorithm is most commonly used.

5.3.2 Applications and Performance of the Model

One of the key advantages of the Decision Tree algorithm over other modeling techniques is its ability to produce **interpretable rules** or **logical statements**.
This **explainability**—due to its axis-parallel decision boundaries—is an important characteristic of decision trees.
Moreover, **classification can be performed without complex computations**, and the method can handle both **continuous and categorical variables**.

Decision Tree models also provide **clear insights** into the **relative importance of features** influencing prediction or classification outcomes (**Tso &Yau, 2007**).

5.3.3 Limitations

However, **Decision Tree induction** is generally **less effective** than neural networks when dealing with **nonlinear data** and is **sensitive to noisy datasets**. The technique is more suited for **categorical outcome prediction**, and unless clear trends or sequential patterns are present, decision trees are **less appropriate for time series forecasting** (**Tso &Yau, 2007**).

Yu et al. (2010) proposed a **Decision Tree-based method** for modeling the **energy demand of buildings**. This method was applied to **Japanese residential buildings** to **predict and classify Energy Use Intensity (EUI)** levels. The results showed that this approach achieved **93% accuracy on training data** and **92% on testing data**. It also automatically identified and classified the **most influential factors** affecting EUI levels and provided **combinations and thresholds** of significant variables, contributing to improved **building energy performance**.

These findings demonstrate that the Decision Tree approach stands out for its **simplicity**, **ease of interpretation**, and **applicability**, compared to other commonly used modeling techniques such as **regression** or **artificial neural networks (ANNs)**.

5.4 K-Nearest Neighbors (KNN) Algorithm

5.4.1 Principle and Architecture

The **K-Nearest Neighbors (KNN)** algorithm is a simple yet powerful **machine learning** technique. It was first proposed by **Thomas Cover** and **Peter Hart** in the late 1960s as a **pattern classification method**. The fundamental principle of KNN is that **similar data points tend to belong to the same class or group**. Therefore, the class of a given data point can be predicted based on the classes of its **nearest neighbors**.

Since its introduction, the **KNN algorithm** has been widely applied in various domains, including **image and speech recognition**, **natural language processing**, and **predictive modeling** (**Sahil Krishna et al., 2024**).

In general, the KNN algorithm computes the **distance** between the test sample and each training sample, then returns the **k closest samples** using a linear search method to identify the nearest neighbors. This allows KNN to effectively **map the relationship** between independent and dependent variable spaces. The computational complexity of KNN is **proportional to the size of the training dataset** for each test instance.

The distance between samples is typically calculated using the **Minkowski distance equation**:

$$\mathbf{d} = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p} \qquad (9)$$

where:

- xi and yi are the coordinates of the sampling points in the multidimensional space,
- d is the absolute distance (**Manhattan distance**) when p=1, and
- d is the linear distance (**Euclidean distance**) when p=2.

If the data distribution is **imbalanced**, the KNN algorithm applies **distance-based weighting**, assigning **higher weights to closer points** to minimize the influence of distant samples (**Hong et al., 2022**).

5.4.2 Applications and Model Results

Wahid & Kim (2016) applied the **K-Nearest Neighbor model** to predict **energy consumption** in **520 residential apartments** in **Seoul**.
Using a **60-40% training-to-testing data ratio**, the model achieved a **prediction accuracy of 95.96%**, indicating that KNN was highly effective in classifying apartment energy consumption levels.

Yesilbudak et al. (2013) developed a **KNN-based classification model** for **wind speed prediction**, using **object-oriented programming techniques**.
Their findings showed that the **choice of input parameters** strongly influenced forecast accuracy.
The best model configuration used **four input variables** — **wind direction**, **air temperature**, **atmospheric pressure**, and **relative humidity** — within a **4-tuple input space**, achieving optimal results for **k = 5** using the **Manhattan distance metric**, with performance indicators of **MAE = 0.594 m/s**, **MAPE = 5.695%**, and **NRMSE = 8.696%**.

Iheanetu&Obileke (2024) demonstrated that **KNN** could serve as a **fast, simple, and accurate tool** for **solar PV energy production forecasting**.
In their study, **MLPNN**, **CNN**, and **KNN** algorithms were compared using **hourly PV generation data** from **Grahamstown, Eastern Cape Province, South Africa**, a region prone to extreme weather conditions.
The **KNN algorithm** achieved **RMSE values ranging from 1.49% (best case) to 4.95% (worst case)** and **MAE values ranging from 0.85% to 2.74%**, significantly outperforming the other algorithms.

5.5 Comparison between Supervised Machine Learning Models

Table 3. Comparison between supervised machine-learning models

Model	Strengths	Limitations
SVM	• High accuracy for moderately complex datasets. • Controls model complexity (reduced overfitting).	• Sensitive to kernel selection. • High computational cost for large datasets.
Random Forest	• Handles noisy data effectively. • Reduces overfitting. • Fast execution and robust performance.	• Low performance on highly complex time series. • Difficulty in capturing temporal dependencies.
KNN	• Easy to implement. • Good performance on moderately sized datasets.	• Sensitive to the choice of the **k** parameter. • Inefficient for large datasets.

5.6 Summary of the Comparative Analysis of Prediction Models

Table 4. Summary of the comparative analysis of prediction model

Criteria	ARIMA / SARIMA	ANN / LSTM / MLP	SVM / Random Forest / KNN
Accuracy	Good for stationary time series.	Excellent for nonlinear data.	Varies depending on parameter selection.
ComputationalComplexity	Low to moderate.	High (especially for LSTM).	Moderate to high depending on data size.
Adaptability	Limited to regular and stationary series.	Highly adaptable (ANN, LSTM).	Flexible but parameter-sensitive.
Preferred Applications	Simple seasonalforecasting.	Complex long-termforecasting.	Categorical or mixed-type data prediction.

6. CONCLUSIONS

This review provided a comprehensive analysis of the main statistical, neural network, and supervised machine learning models used for **energy consumption forecasting**. The objective was to highlight the **principles,**

applications, strengths, and limitations of each modeling approach, as well as to compare their performance across various contexts of energy prediction.

Traditional statistical models such as **ARIMA** and **SARIMA** remain effective for **stationary and seasonal time series**, offering simplicity, interpretability, and reliable short- to medium-term predictions. However, they struggle to capture nonlinear dependencies and dynamic temporal variations that are typical of modern energy systems.

In contrast, **neural network-based models** (such as **ANN**, **LSTM**, and **MLP**) demonstrate a high capacity to model **nonlinear and complex relationships**, making them well suited for **large and heterogeneous datasets**. Among them, **LSTM** architectures stand out for their ability to capture long-term temporal dependencies, although they require significant computational resources and careful hyperparameter tuning.

On the other hand, **supervised machine learning models** like **SVM**, **Random Forest**, and **KNN** offer a balance between accuracy, interpretability, and flexibility. These models are particularly useful when dealing with **mixed or categorical datasets**, and their performance depends largely on **parameter optimization** and **data structure**. While **Random Forest** efficiently reduces overfitting and handles noise, **SVM** achieves strong generalization at the cost of computational complexity, and **KNN** remains a simple yet robust baseline for smaller datasets.

Overall, no single model can be universally considered superior. The choice of the appropriate forecasting technique depends on the **nature of the data**, **the time horizon**, **the computational resources available**, and **the desired level of interpretability**.

Future research should focus on **hybrid and ensemble approaches**, combining the strengths of statistical, neural, and machine learning models to enhance **accuracy, adaptability, and generalization** in the face of increasingly variable and uncertain energy consumption patterns.

7. REFERENCES

[1] A. S. Ahmad, M. Y. Hassan, M. P. Abdullah, H. A. Rahman, F. Hussin, H. Abdullah, and R. Saidur, “A review on applications of ANN and SVM for building electrical energy consumption forecasting,” **Renewable and Sustainable Energy Reviews**, vol. 33, pp. 102-109, 2014.

[2] I. Arora, J. Gambhir, and T. Kaur, “Solar Irradiance Forecasting using Decision Tree and Ensemble Models,” **Proc. 2nd Int. Conf. on Inventive Research in Computing Applications (ICIRCA)**, pp. 675-681, 2020.

[3] B. Charbuty and A. Abdulazeez, “Classification Based on Decision Tree Algorithm for Machine Learning,” **Journal of Applied Science and Technology Trends**, vol. 2, no. 1, pp. 20-28, 2021.

[4] P. Chen, A. Niu, D. Liu, W. Jiang, and B. Ma, “Time Series Forecasting of Temperatures using SARIMA: An Example from Nanjing,” **IOP Conf. Ser.: Materials Science and Engineering**, vol. 394, p. 052024, 2018.

[5] A. L. Chockalingam, “Decision Tree Model for Predicting the Electricity Consumption of Home,” 2018.

[6] M. Cordeiro-Costas, D. Villanueva, P. Eguía-Oller, M. Martínez-Comesaña, and S. Ramos, “Load Forecasting with Machine Learning and Deep Learning Methods,” **Applied Sciences**, vol. 13, no. 13, p. 7933, 2023.

[7] K. Dahmani, G. Notton, R. Dizène, and C. Paoli, “Etat de l’art sur les réseaux de neurones artificiels appliqués à l’estimation du rayonnement solaire,” **Journal of Renewable Energies**, vol. 15, no. 4, 2023.

[8] H. Dhake, Y. Kashyap, and P. Kosmopoulos, “Algorithms for Hyperparameter Tuning of LSTMs for Time Series Forecasting,” **Remote Sensing**, vol. 15, no. 8, p. 2076, 2023.

[9] G. Drałus, D. Mazur, J. Kusznier, and J. Drałus, “Application of Artificial Intelligence Algorithms in Multilayer Perceptron and Elman Networks to Predict Photovoltaic Power Plant Generation,” **Energies**, vol. 16, no. 18, p. 6697, 2023.

[10] G. Dudek, “A Comprehensive Study of Random Forest for Short-Term Load Forecasting,” **Energies**, vol. 15, no. 20, p. 7547, 2022.

[11] M. R. Fathin, Y. Widhiyasana, and N. Syakrani, “Model for Predicting Electrical Energy Consumption Using ARIMA Method,” **Proc. 2nd Int. Seminar of Science and Applied Technology (ISSAT)**, Bandung, Indonesia, 2021.

[12] J. Ferrero Bermejo, J. F. Gómez Fernández, F. Olivencia Polo, and A. Crespo Márquez, “A Review of the Use of Artificial Neural Network Models for Energy and Reliability Prediction,” **Applied Sciences**, vol. 9, no. 9, p. 1844, 2019.

[13] R. Guido, S. Ferrisi, D. Lofaro, and D. Conforti, “An Overview on the Advancements of Support Vector Machine Models in Healthcare Applications: A Review,” **Information**, vol. 15, no. 4, p. 235, 2024.

[14] H. H. Mohammed, A. Asem, and H. El-Bakry, “Advanced Time Series Forecasting Models for Electricity Demand Prediction: A Comparative Study,” **Fusion: Practice and Applications**, vol. 15, no. 1, pp. 19-31, 2024.

[15] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” **Neural Computation**, vol. 9, no. 8, pp. 1735-1780, 1997.

[16] G. Hong, G.-S. Choi, J.-Y. Eum, H. S. Lee, and D. D. Kim, “The Hourly Energy Consumption Prediction by KNN for Buildings in Community Buildings,” **Buildings**, vol. 12, no. 10, p. 1636, 2022.

[17] K. Iheanetu and K. Obileke, “Short-Term Forecasting of Photovoltaic Power Using MLPNN, CNN and kNN,” **Computer Science and Mathematics**, 2024.

[18] N. Jannah, M. S. A. Hanifah, T. S. Gunawan, S. A. Zabidi, S. H. Yusoff, and S. N. M. Sapihie, “Comparative Analysis of MLP and CNN-LSTM Models for Solar Power Generation Forecasting,” **Proc. 9th IEEE Int. Conf. on Smart Instrumentation, Measurement and Applications (ICSIMA)**, pp. 284-289, 2023.

[19] R. Joy, Md. Sohan, A. Nahar, and A. Anannya, “Prediction of Daily Solar Irradiance for Solar Energy Approximation using LSTM Neural Network,” **Energy Proceedings**, vol. 51, 2024.

[20] A. Kahaji, R. Alaoui, S. Farhat, and L. Bouhouch, “Modèle de prédiction des performances énergétiques des alimentations photovoltaïques par l’architecture MLP,” 2013.

[21] H. Lee, D. Kim, and J.-H. Gu, “Prediction of Food Factory Energy Consumption Using MLP and SVR Algorithms,” **Energies**, vol. 16, no. 3, p. 1550, 2023.

[22] D. Liu and K. Sun, “Random Forest Solar Power Forecast Based on Classification Optimization,” **Energy**, vol. 187, p. 115940, 2019.

[23] Y. Liu, Y. Xu, J. Huang, H. Liu, Y. Fang, and Y. Yu, “A Comparative Study of Intelligent Prediction Models for Landslide Susceptibility: Random Forest and Support Vector Machine,” **Frontiers in Earth Science**, vol. 12, p. 1519771, 2024.

[24] L. Monia and B. R. Jaleleddine, “ARIMA Forecasting of Energy Consumption in Tunisia,” Unpublished.

[25] A. J. Niazai, M. A. Shahpoor, A. R. Safi, and N. A. Noori, “Forecasting Monthly Electricity Consumption Using Machine Learning and Statistical-Based Models: A Case Study,” 2022.

[26] M. Park and H. Yang, “Comparative Study of Time Series Analysis Algorithms Suitable for Short-Term Forecasting in Implementing Demand Response Based on AMI,” **Computer Science and Mathematics**, 2024.

[27] A. A. Pierre, S. A. Akim, A. K. Semenyo, and B. Babiga, “Peak Electrical Energy Consumption Prediction by ARIMA, LSTM, GRU, ARIMA-LSTM and ARIMA-GRU Approaches,” **Energies**, vol. 16, no. 12, p. 4739, 2023.

[28] S. Pokharel and P. Ghimire, “Data-Driven ML Models for Accurate Prediction of Energy Consumption in a Low-Energy House: A Comparative Study of XGBoost, Random Forest, Decision Tree, and Support Vector Machine,” **Journal of Innovations in Engineering Education**, vol. 6, no. 1, pp. 12-20, 2023.

[29] M. M. Rahman, M. Shakeri, S. K. Tiong, F. Khatun, N. Amin, J. Pasupuleti, and M. K. Hasan, “Prospective Methodologies in Hybrid Renewable Energy Systems for Energy Prediction Using Artificial Neural Networks,” **Sustainability**, vol. 13, no. 4, p. 2393, 2021.

[30] R. A. Rajagukguk, R. A. A. Ramadhan, and H.-J. Lee, “A Review on Deep Learning Models for Forecasting Time Series Data of Solar Irradiance and Photovoltaic Power,” **Energies**, vol. 13, no. 24, p. 6623, 2020.

[31] S. H. Roodkoly, Z. Q. Fard, M. Tahsildoost, Z. Zomorodian, and M. Karami, “Development of a Simulation-Based ANN Framework for Predicting Energy Consumption Metrics: A Case Study of an Office Building,” **Energy Efficiency**, vol. 17, no. 1, p. 5, 2024.

[32] S. Krishna, I. Shafqat, I. Iqbal, A. Zaib, and M. Iqbal, “Comparative Analysis of KNN and SVM Algorithms,” 2024.

[33] S. A. Sarswatula, T. Pugh, and V. Prabhu, “Modeling Energy Consumption Using Machine Learning,” **Frontiers in Manufacturing Technology**, vol. 2, p. 855208, 2022.

[34] S.-Y. Shin and H.-G. Woo, “Energy Consumption Forecasting in Korea Using Machine Learning Algorithms,” **Energies**, vol. 15, no. 13, p. 4880, 2022.

[35] S. Siami-Namini, N. Tavakoli, and A. SiamiNamin, “A Comparison of ARIMA and LSTM in Forecasting Time Series,” **Proc. 17th IEEE Int. Conf. on Machine Learning and Applications (ICMLA)**, pp. 1394-1401, 2018.

[36] K. Szostek, D. Mazur, G. Drałus, and J. Kusznier, “Analysis of the Effectiveness of ARIMA, SARIMA, and SVR Models in Time Series Forecasting: A Case Study of Wind Farm Energy Production,” **Energies**, vol. 17, no. 19, p. 4803, 2024.

[37] C. Tarmanini, N. Sarma, C. Gezegin, and O. Ozgonenel, “Short-Term Load Forecasting Based on ARIMA and ANN Approaches,” **Energy Reports**, vol. 9, pp. 550-557, 2023.

[38] G. K. F. Tso and K. K. W. Yau, “Predicting Electricity Energy Consumption: A Comparison of Regression Analysis, Decision Tree and Neural Networks,” **Energy**, vol. 32, no. 9, pp. 1761-1768, 2007.

[39] F. Wahid and D. Kim, “A Prediction Approach for Demand Analysis of Energy Consumption Using K-Nearest Neighbor in Residential Buildings,” **International Journal of Smart Home**, vol. 10, no. 2, pp. 97-108, 2016.

[40] M. Yesilbudak, S. Sagiroglu, and I. Colak, “A New Approach to Very Short-Term Wind Speed Prediction Using K-Nearest Neighbor Classification,” **Energy Conversion and Management**, vol. 69, pp. 77-86, 2013.

[41] C. Yin, K. Liu, Q. Zhang, K. Hu, Z. Yang, L. Yang, and N. Zhao, “SARIMA-Based Medium- and Long-Term Load Forecasting,” **Strategic Planning for Energy and the Environment**, 2023.

[42] Z. Yu, F. Haghighat, B. C. M. Fung, and H. Yoshino, “A Decision Tree Method for Building Energy Demand Modeling,” **Energy and Buildings**, vol. 42, no. 10, pp. 1637-1646, 2010.

[43] A. Zendehboudi, M. A. Baseer, and R. Saidur, “Application of Support Vector Machine Models for Forecasting Solar and Wind Energy Resources: A Review,” **Journal of Cleaner Production**, vol. 199, pp. 272-285, 2018.

[44] T. Zhu, “Analysis on the Applicability of the Random Forest,” **Journal of Physics: Conf. Series**, vol. 1607, no. 1, p. 012123, 2020.

[45] Y. Zhu, “Stock Price Prediction Based on LSTM and XGBoost Combination Model,” **Transactions on Computer Science and Intelligent Systems Research**, vol. 1, pp. 94-109, 2023.