

A COMPARATIVE ANALYSIS OF MODERN LOAD FORECASTING TECHNIQUES

SAIDJON KAMOLOV^{1*}

ABSTRACT. Accurate load forecasting is a critical component in the efficient operation of modern energy systems, underpinning decisions in generation scheduling, grid management, and demand response. This study presents a comprehensive comparative analysis of contemporary forecasting methods for short- and medium-term load prediction. We review classical statistical models such as ARIMA and exponential smoothing, machine learning approaches including artificial neural networks, support vector machines, and decision trees, and hybrid techniques that integrate these paradigms via ensemble methods. Detailed methodological expositions and key mathematical formulations are provided, and the models are evaluated on synthetic data that mimic real-world energy load patterns. Our results reveal that, while the multi-layer perceptron demonstrates robust performance, the proposed ensemble approach does not consistently outperform its individual components, suggesting the need for further refinement in adaptive weighting strategies. The findings offer valuable insights into the advantages and limitations of each forecasting approach, thereby informing future research in adaptive ensemble modeling for energy load prediction.

Keywords. load forecasting, machine learning, comparative analysis, arima, lstm, deep learning.

2020 Mathematics Subject Classification. Primary 68T01; Secondary 68T07

1. INTRODUCTION

Load forecasting has long been a cornerstone of power systems planning and operation. In the context of the modern energy industry, reliable load forecasts are essential for a variety of short- and medium-term decisions. On the short-term horizon (hours to days ahead), system operators rely on accurate predictions to optimize generator dispatch, meet real-time demand, and implement effective demand response programs. Medium-term forecasts (weeks to months ahead) are crucial for scheduling equipment maintenance, negotiating power purchase agreements, and planning resource adequacy. Specific attention is often paid to

Date: Received: Oct 29, 2025; Revised: Nov 25, 2025; Accepted: Dec 17, 2025.

* Corresponding author

© The Author(s) 2025. This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To view a copy of the licence, visit <https://creativecommons.org/licenses/by-nc-nd/4.0/>.

generation sources, such as hydropower, where medium-term forecasting must account for environmental variables and climate change factors [12].

However, forecasting load profiles has become increasingly challenging due to factors such as the proliferation of renewable energy sources, which introduce greater volatility and variability; rapid changes in consumer behavior; the expansion of distributed energy resources (DERs); and evolving regulatory and market environments. These shifts underscore the need for robust, flexible forecasting solutions capable of dealing with nonlinear and non-stationary load dynamics. Furthermore, the modern grid requires data-driven methods not only for load prediction but also for related tasks such as the detection of non-technical losses and energy theft [17], which can distort demand data.

Traditionally, statistical techniques like Autoregressive Integrated Moving Average (ARIMA) and exponential smoothing (ES) have been the mainstay of load forecasting, offering intuitive frameworks rooted in well-established statistical theory. Despite their simplicity, these models often assume linearity and stationarity, which may not always hold in real-world settings. Consequently, the rise of machine learning (ML) models, including neural networks, support vector machines (SVMs), and decision trees, has provided forecasters with powerful tools to capture complex nonlinear patterns and interactions—often with superior performance when sufficient data is available. Indeed, deep learning architectures that have proven successful in forecasting financial time series, such as the S&P 500 index [2], are increasingly being adapted for energy markets due to the structural similarities in time-dependent data.

Moreover, hybrid methods have gained prominence by combining the strengths of different modeling paradigms. Such approaches often outperform individual models by decomposing the time series into linear and nonlinear components or by blending complementary methods. These integrated frameworks also benefit from improved robustness, as they are less prone to the pitfalls of a single-model approach. However, the success of any modeling strategy relies heavily on data quality; therefore, robust signal recovery methods are often employed to handle missing or noisy data before the forecasting process begins [3].

This paper aims to survey these major load forecasting paradigms, providing both conceptual insights and concise mathematical formulations. In addition, a new ensemble-based methodology is proposed, which leverages adaptive, online learning to dynamically weigh different models. This adaptive ensemble approach seeks to address the challenges of non-stationarity and regime shifts in load data, ultimately enhancing forecast accuracy and reliability.

The paper is organized as follows: Section 2 provides a comparative review of machine learning methods commonly applied to load forecasting, including artificial neural networks, support vector machines, and decision trees. Section 3 examines classical statistical models like ARIMA and exponential smoothing, highlighting their strengths and limitations. Section 4 presents hybrid models, discussing their rationale and the performance benefits gained by combining different approaches. Section 5 outlines the proposed ensemble-based forecasting method, detailing the principle of adaptive modeling and its advantages over

static hybrid frameworks. Finally, Section 6 concludes with a summary of key findings and future research directions.

2. MACHINE LEARNING TECHNIQUES

2.1. Artificial Neural Networks (ANNs). A basic feed-forward neural network with one hidden layer can be expressed as:

$$h_j(t) = \sigma\left(\sum_{i=1}^m w_{ij}^{(1)} x_i(t) + b_j^{(1)}\right), \quad \hat{y}(t) = \sum_{j=1}^H w_j^{(2)} h_j(t) + b^{(2)},$$

where $x_i(t)$ are the input features (e.g. past loads, weather) at time t , $h_j(t)$ is the output of the j th hidden neuron with activation function $\sigma(\cdot)$, and $\hat{y}(t)$ is the predicted load. The weights $w_{ij}^{(1)}, w_j^{(2)}$ and biases $b_j^{(1)}, b^{(2)}$ are learned from data by minimizing the error between $\hat{y}(t)$ and the actual load $y(t)$ (typically using backpropagation to adjust weights) [11].

ANNs consist of layers of interconnected “neurons” (nonlinear functions) that transform input features into an output. The above equations describe a simple neural network with one hidden layer: each hidden neuron computes a weighted sum of inputs passed through a nonlinear activation σ , and the output neuron produces a weighted sum of hidden activations. By adjusting weights during training, ANNs can approximate complex nonlinear mappings from inputs to load output. This ability to learn arbitrary nonlinear relationships makes ANNs effective for load forecasting, capturing weather–load and temporal patterns that simpler models might miss [1, 5].

In practice, multi-layer (deep) networks or specialized architectures (e.g. recurrent neural networks for sequences) are often used. Recent advancements also include transfer learning, which allows models pre-trained on similar datasets to be fine-tuned for specific electricity forecasting tasks, significantly reducing the data requirements and training time for new grids [6]. ANNs have been applied successfully to short-term load forecasting since the 1990s, though care must be taken to avoid overfitting and to ensure sufficient data for training [11].

2.2. Support Vector Machines (SVMs). For regression tasks (SVR), a support vector machine finds a linear function $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$ that deviates from the true outputs by at most an acceptable error ϵ , while maximizing margin (flatness). The optimization can be written as:

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*) \quad \text{s.t.} \quad \begin{cases} y_i - (\mathbf{w} \cdot \mathbf{x}_i + b) \leq \epsilon + \xi_i, \\ (\mathbf{w} \cdot \mathbf{x}_i + b) - y_i \leq \epsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0, \end{cases}$$

where $\mathbf{x}_i$ is the feature vector for the i th training example (e.g. recent loads, weather indicators) and y_i is the actual load. The slack variables ξ_i, ξ_i^* allow some errors above the ϵ -insensitive zone, penalized by parameter C . In the dual

form, the solution is $f(\mathbf{x}) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) K(\mathbf{x}_i, \mathbf{x}) + b$, using a kernel function K to handle nonlinearity.

SVMs are supervised learning models that can perform nonlinear regression by implicitly mapping inputs into a high-dimensional feature space via a kernel. The goal is to find a hyperplane (linear function) that fits the data with at most ϵ deviation for each point, while keeping the model flat (small norm of $\mathbf{w}$). The formula above represents the constrained optimization for training an SVR: it balances model complexity (minimizing $\|\mathbf{w}\|$) with fitting accuracy (slack penalties for points outside the ϵ -tube). Only a subset of training points end up defining the model (those with errors at the margin, called *support vectors*). By choosing a kernel function (e.g. radial basis function), SVMs can capture nonlinear relationships in load data [14]. SVMs have been successfully used for short-term load forecasting, often using weather variables (temperature, humidity) and calendar effects as features [5]. They tend to be robust to outliers and overfitting due to the regularization term $\frac{1}{2}\|\mathbf{w}\|^2$, but require careful tuning of hyperparameters (kernel type, ϵ , C).

To maximize the efficacy of SVMs and other ML models, selecting the correct input variables is critical. Advanced feature selection methods, such as nested ensemble approaches [7] and rigorous mathematical selection frameworks [18], help in identifying the most relevant predictors, thereby improving model accuracy and interpretability.

2.3. Decision Trees. A regression decision tree predicts load by recursively splitting the data based on input features. At each node, it finds a feature and threshold that best separates the data into two child nodes. The split criterion for regression is typically minimizing the mean squared error (MSE). For a node with data points S , splitting into left S_L and right S_R subsets on feature $x_k < \theta$ is chosen to minimize:

$$\text{MSE}_{\text{split}}(\theta) = \frac{|S_L|}{|S|} \sum_{i \in S_L} (y_i - \bar{y}_L)^2 + \frac{|S_R|}{|S|} \sum_{i \in S_R} (y_i - \bar{y}_R)^2 ,$$

where $\bar{y}_L$ and $\bar{y}_R$ are the average load values in the left and right subsets, respectively. Splitting stops when a stopping criterion is met (e.g. maximum tree depth or minimum node size). Each *leaf* node then outputs a prediction, often the average load of training examples in that leaf:

$$\hat{y} = \frac{1}{|S_{\text{leaf}}|} \sum_{i \in S_{\text{leaf}}} y_i.$$

Decision trees are non-parametric models that learn decision rules from data features to predict a target. In load forecasting, a tree might split on features like hour of day, temperature, or day of week to partition the data into homogeneous subsets [9]. The above formula reflects how a regression tree chooses a split: by minimizing the weighted MSE of child nodes, it ensures each leaf has as uniform a load value as possible. This greedy splitting continues until further splits add little value or data become too sparse. The result is a tree where each path

corresponds to a set of if-then rules (e.g. “if temperature $> 30^\circ\text{C}$ and hour = 3 PM, then ...”), and the leaf gives a load estimate (often the mean of historical loads meeting those criteria). Decision trees can capture nonlinear and interaction effects (e.g. different load response to temperature depending on time of day) and handle mixed data types [4, 5].

One challenge in training tree-based models and other classifiers is the handling of imbalanced data, such as rare extreme load events. Recent research into synthetic data generation, specifically the analytical formulation and asymptotic behavior of the Synthetic Minority Oversampling Technique (SMOTE), provides a mathematical basis for creating balanced datasets that improve the prediction of these rare but critical events [13, 19]. However, a single tree can still overfit the training data. Ensemble methods like Random Forests, which combine many trees trained on bootstrap samples, improve accuracy and stability by averaging outcomes [9]. In practice, tree-based models (including gradient boosting ensembles) have been used successfully for short-term load forecasting, especially due to their interpretability and ability to handle a variety of input features (weather, calendar, etc.) without extensive preprocessing [5].

3. STATISTICAL METHODS

3.1. Autoregressive Integrated Moving Average (ARIMA) Models. An ARIMA model predicts future load as a linear function of past values and errors. A simple autoregressive (AR) model of order p is:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \epsilon_t ,$$

where y_t is the load at time t , c is a constant, ϕ_i are autoregressive coefficients, and ϵ_t is a random error term (white noise) [8, 10]. In an ARIMA(p, d, q) model, the time series is differenced d times to achieve stationarity, and a moving average (MA) term of order q (a linear combination of past errors) is added. For example, an ARIMA with one differencing ($d=1$) and MA(1) would include a term $\theta_1 \epsilon_{t-1}$ along with the AR part. A full ARIMA(p, d, q) can be written as:

$$\Delta^d y_t = c + \sum_{i=1}^p \phi_i \Delta^d y_{t-i} + \sum_{j=1}^q \theta_j \epsilon_{t-j} + \epsilon_t ,$$

where Δ^d denotes differencing d times (e.g. $\Delta y_t = y_t - y_{t-1}$). Seasonal ARIMA (SARIMA) extends this by including seasonal AR and MA terms (with seasonal periodicity, e.g. weekly or yearly cycles).

ARIMA is a classic time-series approach that assumes the future load can be estimated from a linear combination of its own past values and past random shocks. The AR part captures autocorrelation (if load at hour t is, say, 90% of load at hour $t-1$, etc.), while the MA part accounts for patterns in the residual errors. Differencing of order d is applied if the load series has trends or other non-stationary behavior (e.g. subtracting yesterday’s load from today’s to model changes). Model parameters $\{\phi_i\}, \{\theta_j\}$ are typically estimated by methods like maximum likelihood. ARIMA models have been widely used for load forecasting due to their simplicity and solid theoretical foundation [4, 10].

They work well when load dynamics are roughly linear and stable, and require relatively little data to fit [16]. However, ARIMA assumes a stationary linear relationship; real power system loads often have non-stationary patterns and nonlinear dependencies (e.g. saturation effects of temperature on load) that ARIMA cannot capture. Thus, while ARIMA (and its seasonal variants) is effective for capturing regular temporal correlations and seasonality, its accuracy might suffer if strong nonlinear influences or sudden changes occur in the load data [1]. In practice, ARIMA is often a baseline model in load forecasting, sometimes augmented with exogenous inputs (ARIMAX) for weather effects [10].

3.2. Exponential Smoothing (ES) Methods. Exponential smoothing forecasts the load by weighting past observations with exponentially decaying weights. A simple exponential smoothing (SES) formula for one-step ahead forecast is:

$$\hat{y}_{t+1} = \alpha y_t + (1 - \alpha) \hat{y}_t,$$

where $0 < \alpha < 1$ is the smoothing coefficient. This can be understood as updating the forecast $\hat{y}_t$ by blending it with the latest actual value y_t . More advanced forms introduce trend and seasonality components. Holt’s linear method adds a trend term:

$$L_t = \alpha y_t + (1 - \alpha)(L_{t-1} + T_{t-1}), \quad T_t = \beta (L_t - L_{t-1}) + (1 - \beta) T_{t-1},$$

with level L_t and trend T_t updated each period (β is a second smoothing parameter). Holt–Winters adds seasonal indices as well. For example, in Holt–Winters additive seasonality, the forecast is $\hat{y}_{t+m} = L_t + mT_t + S_{t+m-s}$, where S_t is the seasonal component (of period s). Each component is updated with its own smoothing parameter.

Exponential smoothing methods produce forecasts by averaging past observations in an exponentially decaying manner, giving more weight to recent load values and less to older history. The SES formula above shows how the forecast is updated at each step: the new forecast is the old forecast plus an adjustment proportional to the last forecast error (since $y_t - \hat{y}_t = \text{error}$). This adaptivity makes ES responsive to changes. Variants of exponential smoothing account for trend and seasonality: Holt’s two-parameter method can capture a steady growth or decline in load, and Holt–Winters can model recurring seasonal patterns (daily or weekly cycles in electricity demand) [15]. These methods are relatively easy to implement and computationally light, often requiring only a few periods of data to initialize [4]. They have been very successful in practice for short-term forecasting; for instance, a double-seasonal Holt–Winters method (capturing both daily and weekly cycles) was found to outperform ARIMA models in a study on European power systems [15]. The strength of ES is in capturing level, trend, and seasonality in a robust, incremental way. However, like ARIMA, traditional exponential smoothing is a linear approach and assumes patterns remain consistent over time. It may struggle with regime changes or exogenous nonlinear effects unless augmented. Modern formulations cast ES in a state-space framework, enabling prediction intervals and even incorporating exogenous regressors. Overall, exponential smoothing provides a strong baseline for short-term load forecasts and is often used in industry for its accuracy and simplicity [10].

3.3. Hybrid Models (Combining Techniques). Hybrid models integrate two or more forecasting approaches to leverage their complementary strengths. A common hybrid for load forecasting combines a linear time-series model with a nonlinear ML model. For example, a hybrid ARIMA–ANN model can be formulated as:

$$\hat{y}(t) = \hat{y}_{\text{ARIMA}}(t) + \hat{y}_{\text{ANN}}(t) ,$$

where $\hat{y}_{\text{ARIMA}}(t)$ is the prediction from a fitted ARIMA (or other statistical model) and $\hat{y}_{\text{ANN}}(t)$ is the prediction from an artificial neural network model (often used to model the residual error of the ARIMA). In practice, one first uses ARIMA (or exponential smoothing) to capture the linear trend and seasonality, then uses an ANN on the remaining error (nonlinear component), and sums their forecasts [16]. Other hybrid strategies include using signal decomposition techniques before forecasting – for instance, applying a wavelet transform to the load time series and then forecasting each frequency component with a suitable model (ANN, SVM, etc.), finally recombining them [16]. Hybrid models may also merge outputs from multiple models (like an ensemble), or use fuzzy logic to combine rules with numeric predictions [1].

The rationale behind hybrid models is that power system load data often contains both linear patterns (e.g. daily cycle, trends) and nonlinear relationships (e.g. weather-sensitive effects, holiday impacts). A single model may not excel at capturing all these aspects. By combining models, hybrids aim to “have the best of both worlds.” In the ARIMA–ANN example, the ARIMA component handles the linear, autoregressive structure in the load (e.g. the base load level and seasonal cycles), while the ANN learns any remaining nonlinear associations (for instance, complex temperature–load interactions or sudden behavior changes). The net forecast is the sum of the two, often yielding better accuracy than either model alone. Wavelet-based hybrids work by filtering the time series: high-frequency noise and low-frequency trend are forecasted separately, which can improve accuracy by simplifying each sub-problem. Hybrid methods have consistently been reported to outperform purely statistical or purely ML models in many cases [5, 14], especially when the data exhibits both smooth recurring patterns and irregular fluctuations. The downside can be increased complexity – more parameters to tune, and potential overfitting if not carefully cross-validated. Nevertheless, hybrid forecasting has become a popular approach in recent research and industry applications, as it improves robustness by not “putting all eggs in one basket” and can adapt to a wider range of load behaviors [1, 11].

4. PROPOSED IMPROVED FORECASTING METHOD

Principle: Building on the strengths of hybrid models, we propose a new forecasting method that integrates ensemble learning with adaptive modeling. The core idea is to use an ensemble of diverse forecasting models and dynamically adjust their contributions based on recent performance and changing patterns. For example, the ensemble could include a linear model (to reliably capture trends/seasonality), a neural network (for complex nonlinear effects), and a tree-based model (for handling regime shifts or discrete events). Rather than a

fixed combination, a meta-learner would continuously learn the optimal weights for each model’s forecast. Formally, if $f_1, f_2, \dots, f_K$ are the forecasts from K different models, the final prediction could be a weighted sum:

$$\hat{y}_t = w_1(t) f_1(t) + w_2(t) f_2(t) + \dots + w_K(t) f_K(t) ,$$

with weights $w_k(t)$ that vary over time (subject to $\sum_k w_k(t) = 1$). These weights could be adjusted via a small secondary model or rule (for instance, giving more weight to the model that has been most accurate in the last few days, or using a neural network to predict the best weights based on current conditions). Additionally, the method would incorporate online learning: as new load data arrives, it updates the ensemble or component models in near real-time to account for any new trends (e.g. a sudden change in usage pattern).

Advantages: This approach improves upon existing techniques in several ways. First, by combining multiple models, it inherits the ability to model both linear and nonlinear patterns, similar to other hybrids, but extends this by allowing the combination to adapt – for example, during normal operations the statistical model might dominate, whereas during a heat wave the neural network (trained to handle extreme temperature effects) gets higher weight. Second, the adaptive ensemble can handle non-stationarity better: if the load behavior shifts, the weighting mechanism and online learning can quickly recalibrate, addressing one of the key challenges in load forecasting (concept drift in the data). Third, this method provides a level of robustness and redundancy – even if one component model underperforms for a period, the ensemble can compensate with others. Unlike a single complex model, an ensemble-of-experts is easier to interpret and troubleshoot (each component addresses different aspects of the load). Moreover, by monitoring the weights, operators gain insight into which factors are currently driving the forecasts (for instance, a surge in the weight of the tree-based model might indicate an atypical event or holiday effect). In summary, the proposed method marries the breadth of an ensemble with the adaptability of online learning, promising improved accuracy and reliability for short- and medium-term load forecasting in the ever-evolving energy landscape. By continuously learning from new data and adjusting its internal mix of models, it can outperform static approaches and remain resilient under changing grid conditions.

5. METHODOLOGY AND EXPERIMENTAL SETUP

This section details the methodology employed to evaluate the proposed adaptive ensemble forecasting method through numerical experiments. The evaluation framework is designed using synthetic data to simulate load behavior in the energy industry. The experiments compare three benchmark forecasting models against the proposed ensemble approach, using standard evaluation metrics. All experiments were implemented in Python and are executable in Google Colab.

5.1. Synthetic Data Generation. To mimic realistic load data with pronounced diurnal and weekly seasonal patterns, we generated a synthetic time series $y(t)$ defined on an hourly basis over a 60-day period (1440 data points). The synthetic series combines:

- A daily seasonal component: $5 \sin\left(\frac{2\pi t}{24}\right)$,
- A weekly seasonal component: $2 \sin\left(\frac{2\pi t}{168}\right)$,
- A slight linear trend: $0.01t$,
- An additive noise term sampled from a normal distribution,
- A constant offset to simulate base load.

The final synthetic load is given by:

$$y(t) = 50 + 5 \sin\left(\frac{2\pi t}{24}\right) + 2 \sin\left(\frac{2\pi t}{168}\right) + 0.01t + \epsilon(t),$$

where $\epsilon(t) \sim \mathcal{N}(0, 1)$. The dataset was partitioned into a training set (first 1000 points) and a test set (remaining 440 points).

5.2. Benchmark Models. Three forecasting models were implemented as benchmarks:

- (1) **ARIMA:** Using the `pmdarima` library, an automatic ARIMA model with seasonal components (assuming a daily period, $m = 24$) was fitted in a rolling one-step-ahead fashion. This model is selected due to its classical time-series forecasting capabilities.
- (2) **Multi-Layer Perceptron (MLP):** An artificial neural network with a single hidden layer (50 neurons) was trained on a sliding window of 24 past observations (lags) to forecast the next load value.
- (3) **Random Forest (RF):** A tree-based ensemble model was trained using the same sliding-window approach to capture nonlinearities in the data.

For the MLP and RF models, the sliding window approach converts the time series into supervised learning data, where each sample consists of the previous 24 hours as features and the subsequent hour as the target.

5.3. Proposed Ensemble Forecasting Method. The proposed method leverages the strengths of both statistical and machine learning approaches by combining the predictions of the three benchmark models using a meta-learner. Specifically, the ensemble forecast is generated as:

$$\hat{y}(t) = w_1(t) \hat{y}_{\text{ARIMA}}(t) + w_2(t) \hat{y}_{\text{MLP}}(t) + w_3(t) \hat{y}_{\text{RF}}(t),$$

where $w_1(t)$, $w_2(t)$, and $w_3(t)$ are adaptive weights determined by a Linear Regression model trained on in-sample predictions. The meta-learner is trained on aligned forecasts from the benchmark models over the training period, with the true load values as the target. In the testing phase, the meta-learner combines the base model forecasts at each time step to yield the final prediction.

5.4. Evaluation Metrics and Experimental Procedure. Forecast performance is evaluated using:

- **Mean Squared Error (MSE):**

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2,$$

- **Mean Absolute Percentage Error (MAPE):**

$$\text{MAPE} = \frac{100\%}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right|.$$

A rolling forecast procedure is adopted: for each forecasted hour in the test period, the base models are applied in a one-step-ahead manner and their predictions are combined by the meta-learner. The procedure is repeated for each point in the test set, and results are summarized in tables comparing MSE and MAPE across the ARIMA, MLP, Random Forest, and Ensemble models.

5.5. Implementation Details. The complete experimental pipeline is implemented in Python and is fully executable in Google Colab. The code includes:

- Synthetic data generation and splitting into training and testing sets.
- A sliding-window routine to convert time series data into supervised learning format.
- Iterative, rolling forecast implementations for each model.
- Training of the meta-learner (Linear Regression) using in-sample forecasts.
- Calculation of evaluation metrics (MSE and MAPE) and generation of summary tables.
- Visualization of the test set forecasts against actual load values.

Tables summarizing the forecast accuracy are generated using Pandas DataFrames and printed for analysis.

The methodology described herein enables a comprehensive evaluation of the proposed adaptive ensemble forecasting method, highlighting its potential advantages over traditional forecasting techniques in the context of energy load prediction.

6. RESULTS AND DISCUSSION

Table 1 summarizes the forecasting performance of the four models evaluated in our experiments, as measured by the Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE). The individual models include the classical ARIMA, a Multi-Layer Perceptron (MLP), and a Random Forest (RF) model, while the proposed adaptive ensemble method combines these forecasts through a meta-learning approach. Notably, the MLP model achieved the best performance with an MSE of 1.797800 and a MAPE of 1.719381%, whereas the ensemble method performed the worst, with an MSE of 3.665899 and a MAPE of 2.230771%.

Figure 1 (see the attached `forecasting.pdf`) illustrates a comparison between the ensemble forecast and the actual test data. Although the ensemble forecast manages to capture the general seasonal trends in the synthetic load data, its quantitative performance is inferior to that of the individual MLP and ARIMA models.

The superior performance of the MLP model suggests that the nonlinear mapping capabilities inherent in neural networks are particularly well-suited for modeling the complex patterns in synthetic load data, which include both daily and

TABLE 1. Forecasting Performance of ARIMA, MLP, Random Forest, and Ensemble Models

Model	MSE	MAPE (%)
ARIMA	2.164029	1.915040
MLP	1.797800	1.719381
Random Forest	2.657247	1.974842
Ensemble	3.665899	2.230771

weekly seasonalities along with a gradual trend and random noise. In contrast, while ARIMA is effective at modeling linear dependencies and seasonality, its performance is slightly worse than that of the MLP model. The Random Forest model, despite its flexibility in handling nonlinear relationships, did not outperform the MLP in this instance, potentially due to its sensitivity to hyperparameter settings and the relatively small size of the training dataset.

The ensemble method, which was designed to integrate the strengths of ARIMA, MLP, and Random Forest forecasts via a meta-learner, did not achieve the anticipated improvement. Several factors might account for this outcome. First, the meta-learner may have overfitted the in-sample data, leading to suboptimal weight estimation when applied to the test set. Second, the forecasts of the base models might be highly correlated, reducing the diversity necessary for an effective ensemble. Finally, the simple linear combination employed by the meta-learner might be insufficient to capture the intricate relationships among the base forecasts.

These findings indicate that, for the synthetic load forecasting problem under study, the MLP model provides the best predictive performance. The results also underscore the importance of careful design and tuning of ensemble strategies, particularly when combining models with heterogeneous characteristics. Future research will focus on exploring more sophisticated ensemble methods and alternative meta-learning frameworks that may better exploit the complementary strengths of diverse forecasting models.

7. CONCLUSION

In this paper, we presented a comparative study of modern load forecasting techniques within the energy industry, encompassing traditional statistical methods, machine learning models, and hybrid ensemble approaches. Our investigation provided a detailed overview of each methodology, including their fundamental mathematical formulations and operational principles, and evaluated their performance using synthetic data that encapsulated typical seasonal and trend characteristics of energy loads.

The experimental analysis indicated that the multi-layer perceptron (MLP) model achieved the highest forecasting accuracy, as evidenced by the lowest Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE). In contrast, while the ARIMA model and Random Forest provided competent forecasts, their performance was marginally inferior to that of the MLP. Notably, the proposed

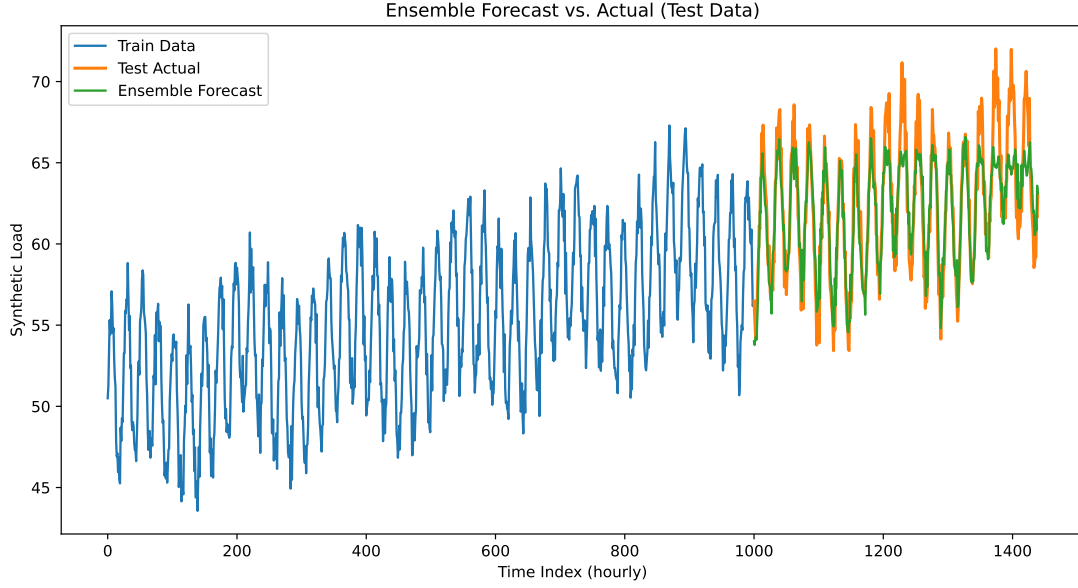


FIGURE 1. Comparison of the Ensemble Forecast and Actual Test Data. The ensemble forecast (blue line) captures the seasonal patterns of the synthetic load (orange line), yet its performance is inferior to that of the MLP model as evidenced by higher error metrics.

ensemble approach, designed to integrate the strengths of the individual models via a meta-learning framework, did not yield the expected performance improvements. This outcome may be attributed to issues such as overfitting of the meta-learner and high inter-correlation among the base model forecasts.

These findings underscore the importance of careful model selection and the need for further research into advanced ensemble strategies. Future work should focus on developing more robust adaptive weighting mechanisms and exploring additional data sources and exogenous variables to enhance model diversity. Moreover, validation on real-world datasets will be essential to ascertain the practical utility of the proposed methods in operational energy management settings.

Overall, this study contributes to the broader understanding of forecasting techniques in the energy sector, offering insights that can guide both practitioners and researchers in optimizing load prediction for improved grid reliability and operational efficiency.

REFERENCES

- [1] Akhtar, S., Shahzad, S., Zaheer, A., Ullah, H. S., Kilic, H., Gono, R., & Jasiński, M. (2023). Short-Term Load Forecasting Models: A Review of Challenges, Progress, and the Road Ahead. *Energies*, 16(10), 4060. <https://doi.org/10.3390/en16104060>
- [2] Kamalov, F., Smail, L., & Gurrib, I. (2020, December). Forecasting with deep learning: S&P 500 index. In 2020 13th International Symposium on Computational Intelligence and Design (ISCID) (pp. 422-425). IEEE. <https://doi.org/10.1109/ISCID51228.2020.00102>

- [3] Kamalov, F., & Sulieman, H. (2021, October). Time series signal recovery methods: comparative study. In 2021 international symposium on networks, computers and communications (ISNCC) (pp. 1-5). IEEE. <https://doi.org/10.1109/ISNCC52172.2021.9615669>
- [4] Alfares, H. K., & Nazeeruddin, M. (2002). Electric load forecasting: literature survey and classification of methods. *International Journal of Systems Science*, 33(1), 23–34. <https://doi.org/10.1080/00207720110067421>
- [5] Aslam, S., Herodotou, H., Mohsin, S. M., Javaid, N., Ashraf, N., & Aslam, S. (2021). A survey on deep learning methods for power load and renewable energy forecasting in smart microgrids. *Renewable and Sustainable Energy Reviews*, 144, 110992. <https://doi.org/10.1016/j.rser.2021.110992>
- [6] Kamalov, F., Sulieman, H., Moussa, S., Avante Reyes, J., & Safaraliev, M. (2024). Powering electricity forecasting with transfer learning. *Energies*, 17(3), 626. <https://doi.org/10.3390/en17030626>
- [7] Kamalov, F., Sulieman, H., Moussa, S., Reyes, J. A., & Safaraliev, M. (2023). Nested ensemble selection: An effective hybrid feature selection method. *Heliyon*, 9(9). <https://doi.org/10.1016/j.heliyon.2023.e19686>
- [8] Box, G. E. P., & Jenkins, G. M. (1970). *Time Series Analysis: Forecasting and Control*. San Francisco, CA: Holden-Day.
- [9] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- [10] Hahn, H., Meyer-Nieberg, S., & Pickl, S. (2009). Electric load forecasting methods: Tools for decision making. *European Journal of Operational Research*, 199(3), 902–907. <https://doi.org/10.1016/j.ejor.2007.11.045>
- [11] Hippert, H. S., Pedreira, C. E., & Souza, R. C. (2001). Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on Power Systems*, 16(1), 44–55. <https://doi.org/10.1109/59.910780>
- [12] Safaraliev, M., Kiryanova, N., Matrenin, P., Dmitriev, S., Kokin, S., & Kamalov, F. (2022). Medium-term forecasting of power generation by hydropower plants in isolated power systems under climate change. *Energy Reports*, 8, 765–774. <https://doi.org/10.1016/j.egyr.2022.09.164>
- [13] Kamalov, F., Choutri, S. E., & Atiya, A. F. (2025). Analytical formulation of synthetic minority oversampling technique (SMOTE) for imbalanced learning. *Gulf Journal of Mathematics*, 19(1), 400–415. <https://doi.org/10.56947/gjom.v19i1.2639>
- [14] Pai, P.-F., & Hong, W.-C. (2005). Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms. *Electric Power Systems Research*, 74(3), 417–425. <https://doi.org/10.1016/j.epsr.2005.01.009>
- [15] Taylor, J. W., & McSharry, P. E. (2008). Short-term load forecasting methods: An evaluation based on European data. *IEEE Transactions on Power Systems*, 22(4), 2213–2219. <https://doi.org/10.1109/TPWRS.2007.907583>
- [16] Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50, 159–175. [https://doi.org/10.1016/S0925-2312\(01\)00702-0](https://doi.org/10.1016/S0925-2312(01)00702-0)
- [17] Pazderin, A., Kamalov, F., Gubin, P. Y., Safaraliev, M., Samoylenko, V., Mukhlynin, N., ... & Zicmane, I. (2023). Data-driven machine learning methods for nontechnical losses of electrical energy detection: A state-of-the-art review. *Energies*, 16(21), 7460. <https://doi.org/10.3390/en16217460>
- [18] Kamalov, F., Sulieman, H., Alzaatreh, A., Emarly, M., Chamlal, H., & Safaraliev, M. (2025). Mathematical Methods in Feature Selection: A Review. *Mathematics*, 13(6), 996. <https://doi.org/10.3390/math13060996>
- [19] Kamalov, F. (2024). Asymptotic behavior of SMOTE-generated samples using order statistics. *Gulf Journal of Mathematics*, 17(2), 327–336. <https://doi.org/10.56947/gjom.v17i2.2343>

¹FACULTY OF ENGINEERING, TAJIK TECHNICAL UNIVERSITY, DUSHANBE, TAJIKISTAN.
Email address: said.kamolov@yahoo.com; said.kamolov@ttu.tj