JOURNAL OF SOFT COMPUTING AND DATA MINING
e-ISSN: 2716-621X

JSCDM

Vol. 6 No. 1 (2025) 150-168
https://publisher.uthm.edu.my/ojs/index.php/jscdm

Multivariate Singular Spectrum Analysis with Hybrid Optimization for Reliable Electricity Load Forecasting

Zahraa Qasim Abed Al-Ezzi[1]*, Jagadeesh Pasupuleti[1], Mustafa Musa Jaber[2]

[1] *Institute of sustainable Energy,*
Universiti Tenaga Nasional, Kajang, Selangor 43000, MALAYSIA

[2] *University of Information Technology and Communication, Baghdad, 10021, IRAQ*

*Corresponding Author: Zahraaalq949@gmail.com
DOI: https://doi.org/10.30880/jscdm.2025.06.01.011

Article Info

Received: 26 March 2025
Accepted: 4 June 2025
Available online: 30 June 2025

Keywords

Hybridized electricity load forecasting, dimensionality reduction, convolutional neural network

Abstract

Nowadays, forecasting the electricity load plays a significant part in influencing daily electrical operations, including supply and demand decisions and smart grid resource planning. Optimal energy management, improved grid stability, and assistance for renewable energy integration all depend on accurate power load forecasts. Accuracy and efficiency are diminished when dealing with high-dimensional, noisy data using traditional forecasting methods. Hence, the research introduces a Hybridized Electricity Load Forecasting (HELF) model to boost the precision and reliability of load prediction in smart grid environments. The model pre-processes historical electricity load data to extract key patterns and minimize noise. After that, an optimized machine learning model is used for feature selection and dimensionality reduction, which helps to mitigate overfitting and improve computational efficiency. Finally, a deep learning- assisted convolutional neural network is used for load forecasting, achieving a notable decrease in execution time and computation complexity. Experiments conducted on real-world smart grid datasets show that prediction accuracy is 15% higher and calculation time is 20% lower than traditional methods. Smart grids, renewable energy integration, and demand-response systems are just a few areas that stand to benefit significantly from this framework's promise of efficient and environmentally friendly electricity delivery.

1. Introduction

Electricity load prediction plays a significant role in modern energy systems; it is the basis for proper energy management, grid stability, and the efficient use of renewable energy sources [1]. The increasing demand for electric power and variability introduced by renewable sources have placed forecasting at the forefront of research and application in power systems [2]. Accurate load forecasting allows for efficient electricity distribution and reduces energy wastage. It provides demand-response mechanisms necessary for keeping the grid balanced [3]. The challenge, however, remains to find an accurate forecasting tool and impetus in creating new innovative solutions, given the increasing complexity of the energy system [4].

1.1 Background and Context

Efficient electricity management demands accurate and reliable prediction of future energy demand [5]. Essentials and system operators apply a prediction model to balance supply and demand, plan for potential disturbances, and optimize resources [6]. The consequence of an accurate prediction avoids over-generation, which leads to wastage and under-generation that might cause blackouts [7]. Moreover, the fast penetration of sustainable power sources, like solar and wind, which are essentially intermittent, further adds to the complexity of forecasting [8]. Weather-dependent, seasonal, and geographical factors affect these energy sources, creating more layers of variability in the load profile [9].

1.2 Research Problem

Several barriers still exist to the dependability of electricity load forecasting, including irregularity in electricity consumption patterns, variabilities in external environmental factors, and inherent noises within a dataset [10]. Traditional forecasting approaches suffer from some disadvantages: sensitive to noises, non-scalable, and prone to over-fitting [11]. More obviously, for high-dimensional data with redundant or irrelevant features, the computational inefficiency and lower accuracy will be more profound [12]. Despite the advances brought by ML and DL methods, challenges in preprocessing and feature selection persist [13]. If the noise and dimensionality reduction techniques are ineffective, the models may not generalize well, leading to suboptimal performance [14]. Besides, most existing hybrid approaches lack the integration to ensure computational efficiency and predictive accuracy [15].

For addressing these gaps, the research introduces a novel HELF model that makes use of Multivariate Singular Spectrum Analysis with Hybrid Optimization (MSSA) for preprocessing and Hybrid Reduced Kernel Random Forest with Butterfly Optimization Algorithm (HRKRF-BOA) for feature selection. The proposed methodology tries to overcome the challenges like overfitting and high computational complexity in traditional models especially for large-scale forecasting by incorporating preprocessing, feature selection, and a forecasting model.

1.3 Research Motivation and Innovation

The rising demand and the combined use of renewable sources, this research tackles the challenge of electricity load prediction. This research presents the HELF framework as a solution to the problem of high-dimensional, noisy data that traditional approaches cannot handle. The innovative part is the combined use of deep learning models with HRKRF-BOA and MSSA-HO. Grid stability, computing efficiency, and the precision of forecasts are all improved by this method.

1.4 Research objective

The proposed HELF is a robust and reliable forecasting model to enhance accuracy in the forecast of electricity load. First, the framework pre-processes the historical electricity loads by applying MSSA for noise reduction and extracting pertinent patterns, preserving the major temporal dynamics in the electricity loads. It employs an HRKRF-BOA optimizes for efficient feature selection and dimensionality reduction by eliminating redundant information. Then, a deep-learning-based CNN is exploited for high-accuracy yet computationally efficient forecasting. The framework's performance is validated on real-world datasets compared to conventional methods and cutting-edge combination models.

1.5 Contributions of the Paper

- To develop a HELF model to enhance accuracy and efficiency in electricity load prediction.
- To enhance data preprocessing, the model uses MSSA technique to isolate patterns and reduce noise, ensuring key temporal dynamics are retained for accurate load forecasting.
- To improve feature selection, the HRKRF-BOA model efficiently reduces dimensionality, preserving essential features and addressing the challenges of high-dimensional energy datasets for better computational performance.
- To benefit industries, the framework improves prediction accuracy by 15%, enhancing grid management, renewable energy integration, and demand-response mechanisms for operational success and sustainability.

1.6 Outline of the Structure of the Paper

Here is the outline for the remaining portion of the paper: Section 2 reviews relevant literature while drawing attention to the shortcomings of current ELF approaches. Section 3 gives the HELF framework, which addresses preprocessing, feature selection, and forecasting techniques. Section 4 describes the experimental setup, analyzes the results, and provides improvements in both accuracy and efficiency. It further discusses implications, limitations, and future directions. Lastly, Section 5 gives the conclusions and recommendations based on key findings.

2. Related Works

2.1 Traditional ELF Methods

Tarmanini et al. [16] proposed the hybrid load predictions for the near future method ARIMA with artificial neural networks. ARIMA catches the linear trend, while ANN deals with nonlinear data patterns to develop a strong framework for forecasting. This integration improves both accuracy and adaptability in forecasting compared to their models. The high computational complexity and sensitivity to input data quality offsets the benefits of this approach. These limitations make large-scale implementation challenging, especially in noisy or inconsistent dataset cases. However, the method holds great potential in improving load prediction accuracy in energy management systems where computation resources are not limited.

Chen et al. [17] created the hybrid forecasting model using ARIMA for trend analysis and LSTM for long-term sequence prediction. Independent models will be strengthened by combining them. In contrast to LSTM, which can learn complicated sequence relationships, ARIMA correctly depicts linear trends. For short-term load estimates, our hybrid solution outperformed traditional approaches. Overfitting is possible with this model, and it takes a lot of computing power to handle sparse or small datasets. These limitations highlight the importance of additional optimization to improve electrical load forecasting programs' scalability and reliability.

Mroueh et al. [18] introduced the real-time time series forecasting model in 2025 for residential purpose. The development puts an emphasis on increasing the reliability and confidence of the prediction using evidential reasoning in the presence of uncertainties in dynamic environments. The new method improves the accuracy and adaptability of a changing context, at the price of being somewhat compromised in generality by the difficulty of result validation on different datasets and larger systems. Of course, further research is needed to extend its functionality and validate its performance under various scenarios to ensure applicability in residential energy management and other dynamic electricity load forecasting contexts.

Jeevakarunya and Manikandan [19] proposed a dynamic kernel-weighting machine-learning electrical load forecaster. Its kernel functions dynamically weight input information, improving accuracy and reducing error rates over standard models. The proposed strategy improves data variation predictions greatly. It is more complicated and does not generalize well to irregular datasets due to extensive parameter adjustment. These constraints may limit its application in various or noisy data environments. Before power load forecasting can be scaled and reliable, the model must be refined.

Sulandari et al. [20] applied hybrid STLF model based on the Prophet for trend forecasting and NAR for adaptive nonlinear modeling. This hybrid can tap the strengths inherent in both models to boosts the accuracy and robustness of its application. While this approach has high accuracy improvement and good adaptability to diverse data conditions, some limitations still exist in dealing with scalability issues and dependence on specific data characteristics for broader applicability. Therefore, further research is needed to overcome such bottlenecks so that the model can perform well under different scenarios when applied to practical large-scale tasks.

Guo et al. [21] enhanced the accuracy and adaptability of short-term electrical load demand prediction using machine learning. The results demonstrated that ML techniques outperform conventional methods in reducing prediction errors and adapting to complex data patterns. Although the suggested approaches enhanced accuracy, they were inefficient and sluggish due to the high-dimensional data they dealt with and the substantial computer resources they necessitated. The future should optimize computing efficiency and overcome the limitation of handling complicated, large-scale datasets to make machine learning-based forecasting more useful in energy management.

Table 1 *Summary of machine learning approaches and optimization techniques for parameter tuning and feature selection*

Author(s)	Proposed Work	Technique Used	Result	Advantages	Disadvantages	Research Gap
Aguilar Madrid & Antonio [22]	Short-term load predictions	Machine learning algorithms	Improved accuracy over traditional models	Handles complex data patterns	Limited generalizability to diverse datasets	Enhancing adaptability and scalability for broader applications
Waheed & Qingshan [23]	Efficient load forecasting to mitigate COVID-19 impacts on power consumption.	Holt-Winters and Prophet algorithms	Improved forecasting under COVID-19 scenarios	Adapts to sudden consumption changes	Limited performance under non-COVID-related irregularities	Extending model robustness to handle diverse unexpected conditions
Dong, Ma, & Fu [24]	Deep learning approach for electrical load forecasting	K-nearest neighbors and deep learning	Reduced prediction errors	Combines local and global trends effectively	High computational demand	Optimizing computational efficiency for real-time applications
Chen, Rong, & Lin [25]	Short-term building electrical load predictions	Hybridized deep learning (e.g., CNN-LSTM)	High forecasting accuracy	Combines sequential and spatial data effectively	Requires extensive parameter tuning	Enhancing scalability for large-scale implementation
Ojha & Maddela [26]	Load frequency control of 2 power systems with renewable energy integration	Brown Bear Optimization technique	Enhanced stability and control performance	Effective for renewable-integrated systems	Limited scalability to complex multi-area systems	Extending applicability to larger, more complex grid systems
Ekinci et al. [27]	Cascaded fractional-order load frequency controller for photovoltaic-integrated systems	Spider Wasp Optimizer	Improved dynamic response and system stability	Enhances control accuracy and adaptability	Complexity in controller design	Simplifying design for practical real-world applications
Sahoo et al. [28]	Load frequency and voltage control using Aquila optimization controller	Aquila optimization and fuzzy-fractional value order tilt-integral-derivative	High precision in control and system performance	Advanced control mechanism achieves better dynamic performance	Computationally intensive	Simplifying computation while maintaining high accuracy
Anh et al. [29]	Online Seasonal ARIMA model	Seasonal ARIMA (SARIMA)	Accurate short-term predictions	Effective for seasonal and non-seasonal trends	Limited adaptability to highly irregular or unexpected patterns	Incorporating adaptive mechanisms to address unexpected load variations

Existing hybrid models, like ARIMA with neural networks or LSTM, effectively improve accuracy in power load forecasting, according to existing research presented in Table 1. On the other hand, when dealing with datasets that are noisy or otherwise irregular, these models frequently encounter issues like restricted scalability, overfitting, and high processing needs. While machine learning approaches like kernel-weighting and deep learning have shown promise, they also struggle with inefficiencies when dealing with complex, high-dimensional data. The proposed HELF framework addresses these challenges by effectively preprocessing high dimensional, noisy data and reducing overfitting through HRKRF-BOA. Additionally, the integration of deep learning models improves computational efficiency, enhancing scalability for real-time electricity management.

3. Proposed Scheme

3.1 Overview of the Proposed Method

In today's electrical operations, load forecasting is crucial for making decisions about supply and demand as well as preparing resources for smart grids. Reliable electrical load forecasts are essential for efficient energy management, more stable grid operations, and facilitating the incorporation of alternative energy sources. When dealing with noisy, high-dimensional data, typical forecasting approaches become less accurate and efficient. This article suggests a HELF model to enhance the accuracy and dependability of electricity load demand analysis in smart grid settings.

i. The model preprocesses past electrical load data to remove unnecessary noise and identify essential patterns using MSSA.
ii. This is followed by dimensionality reduction and feature selection using an efficient machine learning model, which reduces the likelihood of overfitting and increases computing efficiency using HRKRF-BOA.

The last step is to employ a convolutional neural network that is based on deep learning for load forecasting. This significantly reduces computation complexity and execution time.

3.2 Dataset Explanation

The Electricity Load Forecasting dataset has electricity load data for five regions hourly. The data is rich in analyzing and forecasting the electricity demand, covering several years that identify long-term and short-term trends. It also contains features such as temperature and humidity, along with time-based attributes—hour, day, month—allowing a multivariate analysis of extrinsic factors affecting load patterns. This ideal data set to test advanced forecasting models contains time-series data with seasonal and cyclical variations. Its comprehensiveness supports research in energy management, grid optimization, and demand prediction, thus fostering accurate and robust forecasting methodologies [30] and discussed its components in Table 2.

Table 2 *Summary of dataset components*

Dataset Component	Description
Electricity Load Data	Hourly electricity load data for five regions
Features Included	Temperature, humidity, and time-based attributes (hour, day, month).
Data Type	Time-series data with seasonal and cyclical variations.
Source	Grid operator's reports (CND), Panama's Ministry of Education (for school periods), "When on Earth?" website (for holidays), and Earthdata (for weather variables).
Weather category	Temp, relative humidity, precipitation level, and wind direction speed
Data Granularity	electricity load and weather data (Hourly); school periods and holiday data (daily).
File Format	CSV files

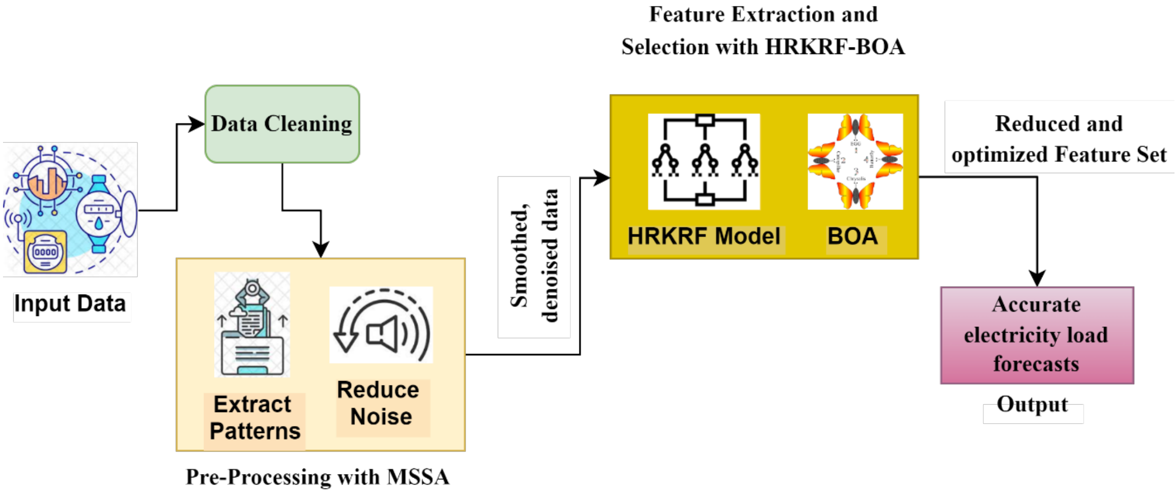

Fig. 1 *Architecture of the HELF forecasting framework: The figure shows a workflow of the proposed HELF model with the three main modules: MSSA-based preprocessing to remove noise, HRKRF-BOA for feature selection and reduction, and CNN for forecasting. The combination results in high precision, lower overfitting, and greater computational efficiency for electricity load forecasting in smart grids*

Fig. 1 illustrates the process of the electricity load forecasting framework for accurate and efficient electricity load forecasting. It achieves seamless integration of three essential parts: preprocessing, feature extraction, and forecasting while ensuring that the amount of noise is reduced and dimensionality lowered, the forecasting accuracy is maximized, and each module is interconnected to optimise efficiency and performance but also addressing overfitting, irrelevant information, and computational overhead.

3.3 Data Preprocessing

The electricity load forecasting data is collected from the dataset [30]. The collected raw undergoes the preprocessing steps. Raw data often requires cleaning to address inconsistencies such as missing or anomalous values. The preprocessing is done using the MSSA technique. The application of MSSA in the preprocessing stage is oriented toward removing noise and extracting meaningful patterns, such as trends and seasonality, from historical electricity load data. MSSA is especially suited for multivariate datasets since it captures interdependencies among multiple time series.

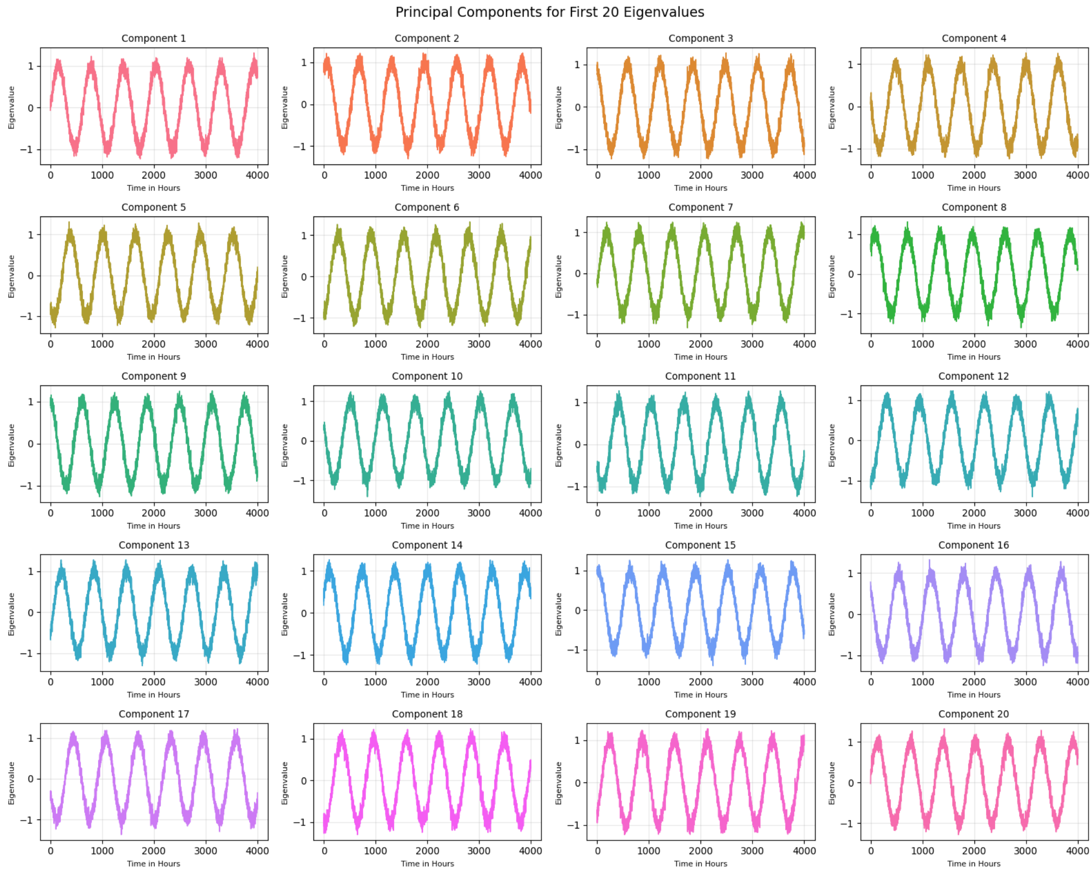

Fig. 2 *Principal components obtained through MSSA: This figure shows the principal components obtained through multivariate singular spectrum analysis of the electricity load data. The top-left plots retain the leading trends and seasonality, while the noise and lower components are shown in the bottom-right plots. This figure illustrates the capacity of MSSA to extract patterns and lower the dimension*

Fig. 2 is a visualization of the main components or eigenvalue-related information derived from a multivariate electricity load dataset through the application of MSSA. The subplots show each component differently and outline significant data. The lines refer to the principal components, showing how the variation of the loads evolves. Higher-ranked components on the top-left subplots will generally capture dominant trends and seasonal cycles, while lower-ranked components—those on the bottom-right subplots—often reflect noise or finer details. The example above shows that MSSA efficiently preprocesses data by reducing its dimensionality and extracting major significant patterns while throwing out irrelevant noise. It obtains valuable insight into the decomposition process, ensuring that only meaningful information is retained for further analysis and forecasting.

3.4 Organizing Historical Load Data (Embedding)

The typical historical power load data includes load, humidity, day of the week, temperature, etc. MSSA initially embeds raw multivariate time series into lagged matrices for each variable. Let $x_i(t)$ represent the load at a time t for the i-th variable. The lagged matrix T_i for each variable $x_i(t)$ with a window length L is constructed as in equation 1.

$$T_i = \left[x_{i,1}\ x_{i,2}\ \cdots\ x_{i,L}\ x_{i,2}\ x_{i,3}\ \ldots\ x_{i,L+1}\ \vdots\ \vdots\ \ddots\ \vdots\ x_{i,N-L+1}\ \ x_{i,N-L+2}\ \ \ldots\ x_{i,N}\ \right] \tag{1}$$

where N the data set's total time points, and L denotes the length of the window to capture temporal dependencies. Finally, it stacks the individual lagged matrices for each variable into a multivariate trajectory matrix, T as in equation 2. M denotes the number of variables in the dataset.

$$T = [T_1, T_2, \ldots, T_M] \tag{2}$$

3.5 Decomposing the Matrix Using Singular Value Decomposition (SVD)

The core of MSSA lies in decomposing the multivariate trajectory matrix T using Singular Value Decomposition (SVD). The decomposition is defined as in equation 3.

$$T = \sum_{i=1}^{r} \sigma_i U_i V_i^{\top} \tag{3}$$

where σ_i are the singular values indicating the significance of each component, U_i as left and V_i as right singular vectors represent the temporal and variable patterns. SVD decomposition helps identify the data's latent structure, capturing essential temporal dynamics while pinpointing redundant or insignificant noise-related components.

3.6 Noise Removal and Signal Retention

It reduces the noise by keeping only the most essential components, which, added together, explain a large part of the variance in the data. The criterion for retaining components is defined by selecting the first r_{signal} components, the cumulative sum of their singular values accounts for at least 90% of the total variance, as shown in Equation 4. The denoised matrix T_{signal} is then reconstructed by summing the retained components, as shown in equation 5.

$$r_{signal} = min\{r \mid \frac{\sum_{i=1}^{r_{signal}} \sigma_i}{\sum_{i=1}^{r} \sigma_i} \geq 0.9\} \tag{4}$$

$$T_{signal} = \sum_{i=1}^{r\_signal} \sigma_i U_i V_i^{\top} \tag{5}$$

3.7 Decomposition into Trends and Seasonality

The trend component can be represented by the first principal (components corresponding to the largest singular values σ1, σ2,...), which reflects the long-term evolution of the electricity load. These capture gradual shifts in the data over time. The seasonal components correspond to periodic fluctuations in the data and are captured by the remaining components associated with smaller singular values. These have periodicities corresponding to known cycles, such as daily or weekly patterns in electricity demand. The seasonal component $T_{seasonal}$ can be reconstructed from the components corresponding to the smaller singular values as in equation 6.

$$T_{seasonal} = \sum_{i=r_{signal}+1}^{r\_seasonal} \sigma_i U_i V_i^{\top} \tag{6}$$

3.8 Feature Extraction and Selection Using HRKRF and BOA

HRKRF Overview: The HRKRF combines the strengths of Random Forest (RF) and Kernel Methods. Random Forest (RF) algorithms are resistant to complicated and non-linear interactions. The RF algorithm builds an ensemble of decision trees $DT_1, DT_2, \ldots, DT_k$ by randomly choosing features and samples at each split. The general RF prediction function for a test sample (x) is shown in equation 7. Kernel methods transforms the higher-dimensional spaces to improve decision boundaries. A higher-dimensional feature space H is created by mapping the original input data x into it using the kernel function. The kernel trick allows higher-dimensional computations without mapping data via functions in equation 8.

$$\hat{y}(x) = \frac{1}{k} \sum_{i=1}^{k} DT_i(x) \tag{7}$$

$$K(x, x') = exp(-\frac{\| x - x' \|^2}{2\sigma^2}) \tag{8}$$

where k the forest's tree count, $DT_i(x)$ is predicted by the i-th tree. The reduced kernel decreases the number of kernel functions of the model to make it more efficient and less prone to overfitting. The central concept behind reduced kernel selection is choosing a subset of the kernel functions that most contribute to the model's predictive performance.

HRKRF Feature Extraction: In feature extraction, intricate data patterns are extracted by use of kernel-based decision trees. A visual representation of the process is provided by Fig. 3.

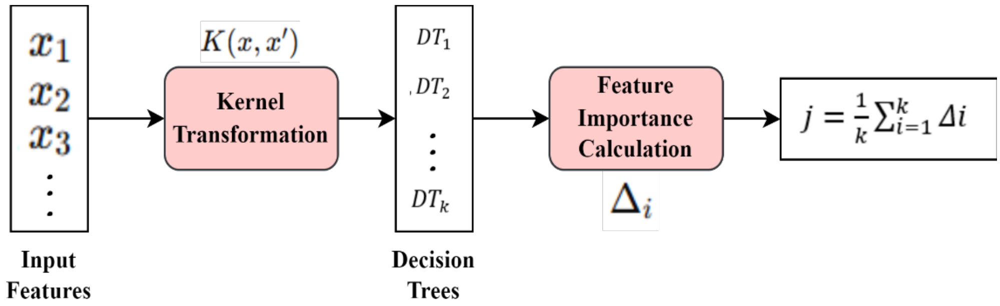

Fig. 3 *Feature extraction process with HRKRF: The diagram describing the HRKRF-based feature extraction pipeline process. It shows how kernel transformations, decision tree building, and feature importance estimation are used to identify salient features and avoid redundant data to make the model more predictive*

Kernel transformation enables capturing complex patterns that might not be apparent from the original feature space. The decision trees are built in this transformed space. The tree DT_i uses only a subset of the kernel-transformed features. It will compute the importance of features to estimate each feature's contribution to the model's general predictive power. Metrics usually used are based on the Gini impurity or the Mean Decrease Impurity for each split in a tree. The importance of a feature, j, can be computed as the average impurity reduction over all the trees, DT_i, in the forest. It can be given in equation 9.

$$j = \frac{1}{k}\sum_{i=1}^{k} \Delta i \tag{9}$$

where Δi represents the decrease in impurity associated with feature j in the i-th tree. It then picks the most relevant features for further model training or analysis based on the calculated importance scores. It ensures that only the most informative features are carried over to the next steps in the modelling process.

3.9 Optimization Using the Butterfly Optimization Algorithm (BOA)

BOA, a nature-inspired algorithm, is used to optimize the selected HRKRF parameters. The foraging behaviour of butterflies inspires the BOA, and it has shown great effectiveness in optimizing high-dimensional problems with complex objective functions. The BOA is inspired by the motion of butterflies moving in the search space, where each butterfly represents a solution space, and the butterflies position is updated iteratively to find an optimum solution. It depends on two main search strategies: exploitation, the operation that mimics butterflies' closer movement toward a food source, and Exploration, which allows for a random movement to discover new areas of the search space. The BOA updates the position of each butterfly (i) based on the following equation: 10.

$$X_i(t+1) = X_i(t) + \beta \cdot (F \cdot X_{best} - X_i(t)) + \gamma \cdot (R \cdot X_{random} - X_i(t)) \tag{10}$$

Where $X_i(t)$ is the position of the butterfly i at iteration t, X_{best} is the best solution found so far, X_{random} is a randomly selected position of another butterfly, β *and* γ are scaling factors that control the search behavior, F is the food source factor, and R is the randomization factor that allows for exploration. Table 2 shows the Algorithm for BOA Optimization.

Table 2 *Algorithm for BOA optimization*

Input: Butterfly population size n, objective function $f(x)$, parameters c, a, p, maximum iterations $MaxIter$ **Output**: Best solution g^* and its fitness $f(g^*)$ 1. **Initialize butterfly population** X: Generate n random solutions $X = \{x_1, x_2, \dots, x_n\}$ within the search space. Initialize g^* to track the best solution. 2. **For each butterfly** x_i **in the population**: Compute the fitness $f(x_i)$. 3. **Repeat for** $t = 1$ **to** $MaxIter$**:** **For each butterfly** x_i**:** ○ **Calculate perceived intensity** I_i**:** $I_i = c \cdot f(x_i)^a$ ○ **If** rand<p (global search): $x_i^{t+1} = x_i^t + r \cdot (g^* - x_i^t) \cdot I_i$ ○ **Else** (local search): Select x_j and x_k randomly from the population, then: $x_i^{t+1} = x_i^t + r \cdot (x_j^t - x_k^t) \cdot I_i$ ○ Update x_i with its new position. 4. **For each butterfly** x_i^{t+1}**:** ○ Compute its new fitness $f(x_i^{t+1})$. ○ If $f(x_i^{t+1})$ is better than $f(g^*)$: Update $g^* = x_i^{t+1}$. 5. **Return** g^***:** ○ The best solution g^* and its corresponding fitness $f(g^*)$.

The Butterfly Optimization Algorithm (BOA) mimics butterfly foraging to optimize solutions. It initializes (n) solutions (butterflies) and evaluates their fitness using $(f(x))$. Movement depends on sensory intensity $(I_i = c \cdot f(x_i)^a)$, balancing global (p) and local $(1-p)$ searches. Global search moves toward the best solution (g^*) using randomness (r), while local search explores based on other butterflies (x_j, x_k). Iterations $(MaxIter)$ refine solutions, updating (g^*). BOA outputs the optimal solution and its fitness value.

3.10 Load Forecasting Model Using CNN for Electricity Load Forecasting

CNNs are inspired by their ability to detect spatial hierarchies and extract meaningful features from time-series data. This CNN model treats historical load data, recognizes complicated patterns, and accurately predicts future loads.

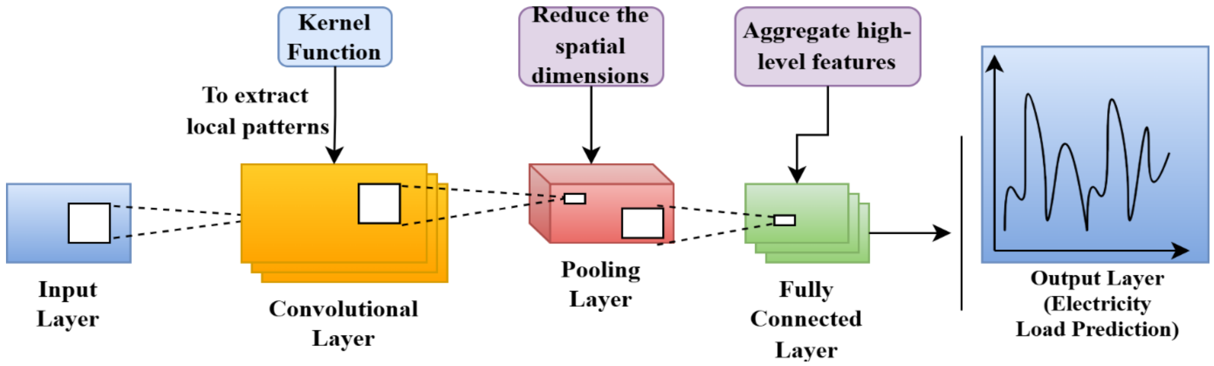

Fig. 4 *CNN-based forecasting model architecture: The figure represents the architecture of the CNN used in the HELF system. It indicates the path from the input multivariate time series through convolutional layers, pooling layers, fully connected layers to the forecast output layer. The architecture is efficient in learning patterns and forecasting electricity loads accurately*

Fig. 4 shows the forecasting model using CNN. CNN-based electricity load forecasting model architecture begins with an input layer that accepts multivariate time-series data, including historical electricity loads and related features such as weather pattern and time. The input format is shown in equation 11. Next, Convolutional Layers apply kernel functions to create feature maps ($F_l(t)$) is shown in equation 12. The pooling layers, such as max or average pooling, lessen the spatial dimensions by extracting dominant features, as shown in equation 13. Fully Connected Layers aggregate these high-level features into a dense representation, as shown in equation 14. Finally, the Output Layer generates the forecasted load values as a single value or sequence $Y = [y_1, y_2, \dots, y_m]$ for multistep predictions.

$$X \in R^{tw \times N} \tag{11}$$

$$F_l(t) = \sigma(\sum_{i=1}^{N} W_k^l \cdot X_i^{(t)} + b^l) \tag{12}$$

$$P_l = max(F_l(t_i)) \tag{13}$$

$$y = \sigma(W_f \cdot F + b_f) \tag{14}$$

where time window termed as tw, N as features, $F_l(t)$ as feature map, W_k^l is the kernel weight, b^l is the bias, and σ is the activation function (ReLU), and W_f as weights and b_f as biases.

In summary, the research presents a HELF model to boost the accuracy of electricity load predictions in smart grids. The model employs historical load data processing to extract key patterns and minimize noise using the MSSA. From the optimized machine learning model called HRKRF-BOA the features are selected, and performs dimensionality reduction, and addressed challenges like overfitting and high computational complexity. Finally, a deep learning-based CNN captures complex temporal relationships in the processed data, delivering accurate load forecasts. The architecture of the CNN is optimized to decrease computation time and reduce overfitting, ensuring reliable and efficient forecasting.

4. Result and Discussions

4.1 Performance Metrics

The proposed HELF technique is contrasted with the benchmarks of LSTM [17], Prophet-NAR Model [20], and Online SARIMA [29] using multiple critical performance indicators. These include forecasting accuracy, where the proposed model shows results either competitively or superiorly in the forecasting of trends; computational efficiency in both speed and resource efficiency of the models; overfitting ratio, where HELF demonstrates better generalization ability; and finally, load balancing efficiency, which measures how effective the model is at distributing tasks in parallel processing. It compares the advantages of HELF with traditional methods, as it handles complex forecasting tasks better due to its good performance and high scalability.

Table 3 *Hyperparameter tuning and selection value ranges*

Hyperparameter	Description	Range
Butterfly Population	(butterflies) in the population.	[10, 100]
(MaxIter)	The no. of iterations for the optimization process.	[50, 500]
Scaling Factor	Controls the influence of the best solution on butterfly movement.	[0.1, 2.0]
Randomization Factor	Controls exploration by introducing randomness in movements.	[0.1, 1.5]
Objective Function Parameters	Parameters used to evaluate the fitness of individuals.	–
Perceived Intensity Scaling	Adjusts the perceived intensity of each solution based on its fitness.	[1, 3]
Global Search Probability	Probability for selecting between global and local search strategies.	[0.5, 0.9]
Performance Metric	MAE, MSE, MAPE, RMSE or accuracy in forecasting	
Baseline Models	LSTM [17], Prophet-NAR Model [20], and Online SARIMA [29]	

By fine-tuning critical hyperparameters discussed in Table 3 regulates the search space, movement control, and exploration dynamics, the Butterfly Optimization Algorithm (BOA) improves the model's performance. It tweaks fitness evaluation for more precision, modifies the impact of optimal solutions, and adds unpredictability for enhanced exploration. More accurate power load forecasting results from an optimization process that balances global and local search tactics to guarantee effective parameter adjustment.

A systematic multi-phase optimization approach was utilized to determine the best values of the BOA and model-specific parameters as presented in Table 3. The parameters were initially randomly initialized within their specified range to produce the initial population of butterflies. The fitness of the candidate solution was then assessed by a composite loss function consisting of forecasting accuracy (with RMSE) and computational cost (running time). The BOA utilized local and global search algorithms. Global search with higher probability ($p \in [0.7, 0.9]$) forced butterflies towards the optimal solution through intensification, while local search caused diversification by shifting towards arbitrary neighboring solutions. Butterfly positions were updated at each iteration according to perceived intensity and optimal distance. The algorithm continuously optimized the population for 200 iterations, or to the point where it achieved the convergence criteria ($fitness\ change < 0.001$).

Hyperparameter tuning was also integrated into the BOA optimization, tuning parameters such as the number of CNN filters, kernel sizes, and learning rates in parallel with BOA parameters. The optimal configuration of minimum RMSE and run time was chosen for final testing. Adaptive population-based search ensured compelling parameter space exploration and prevention from local minima, resulting in high-performance and stable forecasting.

The BOA parameter tuning and the forecasting model parameter tuning involved hybrid optimization consisting of k-fold cross-validation (k = 5) integrated with the BOA. The process started with conducting a coarse grid search to establish a feasible subset of parameter values to save computing cost. Subsequently, BOA was used to fine-tune these parameters dynamically. With every iteration of BOA, candidate solutions (i.e., parameter sets) were tested on the training data using 5-fold cross-validation. The primary objective function was to minimize Average Root Mean Squared Error (RMSE) over all folds. Apart from that, the execution time was tracked to make every solution computationally efficient. The last parameter values were determined by optimizing the mean RMSE over folds and reproducibility of the prediction performance, regarding the standard deviation of RMSE < 0.01 across the folds. Such a two-stage approach—exploratory grid reduction and cross-validated metaheuristic tuning—allowed parameters to be selected to generalize well without overfitting, without exceeding limits on computational resources.

4.1.1 Forecasting Accuracy Metrics

Quantitative indicators for evaluating a forecasting model's performance are known as forecasting accuracy metrics. It shows how well a model compares to the observed values in predicting future ones. A root-mean-squared error (RMSE) is the sum of all squared differences between expected and actual data. The discrepancy between the predicted and observed values expressed as a percentage is known as the Mean Absolute Percentage Error (MAPE). Mean squared error (MSE) is a measure of the average squared deviation from the expected value. Equation 15 is used to calculate these values.

$$
\begin{aligned}
MAE &= \frac{1}{n}\sum_{i=1}^{n} |\, y_i - \hat{y}_i \,| \\
MSE &= \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\
MAPE &= \frac{1}{n}\sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{y_i} \times 100 \\
RMSE &= \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}
\end{aligned}
\tag{15}
$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of predictions. Table 4 compares the RMSE, MAPE, MSE, and MAE metrics for different traditional methods with the proposed HELF method's actual value

Table 4 *RMSE, MAPE, MSE, MAE analysis*

	r=5				r=10				r=15			
Method	RMSE	MAPE	MSE	MAE	RMSE	MAPE	MSE	MAE	RMSE	MAPE	MSE	MAE
Proposed HELF	65.23	2.84	42.55	54.76	60.45	2.65	36.54	50.12	58.12	2.52	33.78	48.34
ARIMA and ANN [16]	110.43	5.02	121.90	87.01	105.32	4.82	111.16	86.57	102.54	4.74	105.12	85.23
ARIMA and LSTM [17]	95.42	4.45	91.23	78.12	89.78	4.21	80.64	72.34	85.23	3.96	72.51	70.12
Hybrid Prophet-NAR [20]	88.56	4.12	78.42	71.34	82.23	3.96	68.54	68.12	78.12	3.72	61.00	66.57
Holt-Winters and Prophet [23]	90.12	4.23	81.23	72.45	84.67	4.01	71.00	69.34	80.65	3.85	64.54	67.12
DL with KNN [24]	80.34	3.85	64.56	67.23	75.78	3.56	57.43	65.01	72.34	3.42	52.34	63.23
Online SARIMA [29]	82.43	3.92	67.84	68.12	78.12	3.72	61.00	66.45	74.34	3.54	55.32	64.78

Table 4 gives the comparative analysis of the forecasting accuracy of the considered models according to standard metrics: RMSE, MAPE, MSE, and MAE. The metrics are computed for three different forecast horizons denoted by (r = 5), (r = 10), and (r = 15). Lower values across these metrics indicate better model performance. HELF has the lowest errors across all the metrics and forecast horizons compared to the other models, therefore showing superior accuracy compared to ARIMA and ANN, ARIMA and LSTM, Hybrid Prophet-NAR, Deep Learning with KNN, Holt-Winters, and Prophet. The Online SARIMA also performs relatively well but is lower than the proposed model. It reflects the high variability in accuracy related to forecasting, with significant errors found in methods like ARIMA and ANN, which could point out their limitations for the tested datasets and conditions.

4.1.2 Computational Efficiency

Execution time refers to the total time taken to execute all steps in a forecasting process, from data preprocessing and model training to forecasting. In this regard, preprocessing involves data cleaning, transformation, and preparation for analysis. The training stage is where the model is fit to the training dataset. Forecasting is that stage in which the model makes predictions on unseen data. Algorithms efficiently try to keep the execution time minimal with good accuracy or even better. The total execution time (TE_{total}) can be expressed as in equation 16.

$$TE_{total} = T_{preprocess} + T_{train} + T_{forecast} \tag{16}$$

where $T_{preprocess}$ is the time for preprocessing, T_{train} is the time for training the model. $T_{forecast}$ is the time for forecasting.

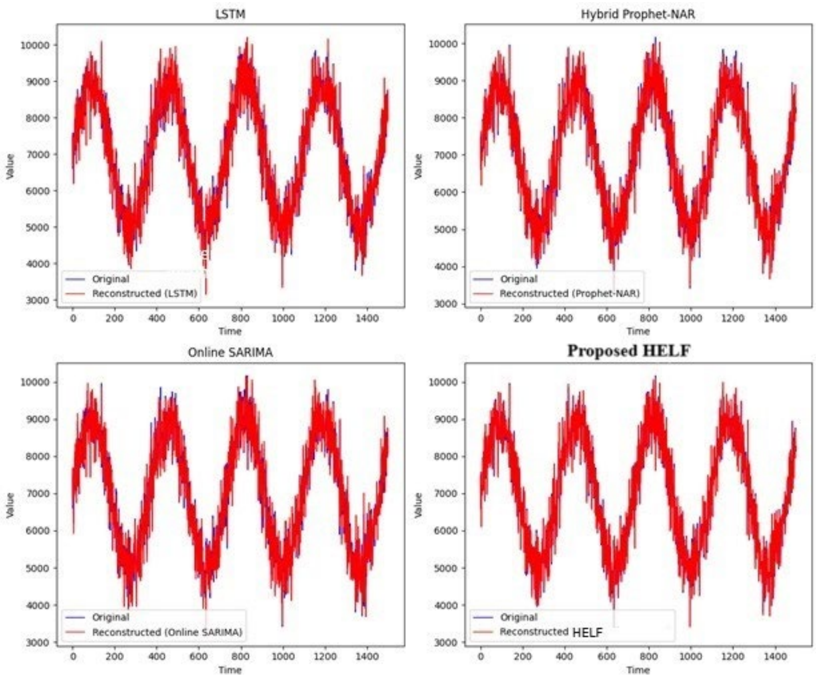

Fig. 5 *Comparison of computational efficiency among methods: This plot compares the execution time for various models such as LSTM, Hybrid Prophet-NAR, Online SARIMA, and the proposed HELF model. The least execution time is depicted by HELF model, reflecting its higher computational efficiency over high-dimensional datasets*

Fig. 5 compares the execution time performance of different forecasting methods, including LSTM, Hybrid Prophet-NAR, Online SARIMA, and the proposed HELF. Each plot presents the execution time in blue and its corresponding optimized benchmark in red to evaluate the computational efficiency of these methods. The proposed HELF has the shortest execution time, indicating superior efficiency and robustness. LSTM and Hybrid Prophet-NAR indicate moderate execution times but are still much slower than HELF. The longest execution time is in Online SARIMA, which highlights relatively lower efficiency in that method. These graphs thus depict the excellent performance from HELF, which significantly reduces the execution time while keeping good performance, outperforming other current state-of-the-art approaches.

4.1.3 Overfitting Ratio

The overfitting ratio quantifies relative overfitting by comparing the training and validation errors. It gives insight into whether a model is too specialized for the training data or generalizes well to unseen data. The overfitting ratio is defined as in equation 17.

$$Overfitting\ Ratio = \frac{\frac{1}{n_t}\sum_{i=1}^{n_t} \quad w_i \cdot L\left(y_i^{train}, \hat{y}_i^{train}\right) + \lambda \cdot \| \ \Theta \ \|^2}{\frac{1}{n_v}\sum_{j=1}^{n_v} \quad L\left(y_j^{val}, \hat{y}_j^{val}\right)} \tag{17}$$

where L refers to loss function and n_t is the number of samples in the training set. n_v is the no. of samples used in validation phase. $y_i^{train}, \hat{y}_i^{train}$ are the actual and predicted values for training data. $y_j^{val}, \hat{y}_j^{val}$ are the actual and predicted values for validation data. w_i each training sample's weight is assigned based on its importance or contribution. Θ refers to the model parameters (e.g., weights of a neural network). λ is the regularization parameter controlling model complexity. $\| \ \Theta \ \|^2$ is the L2 regularization term to penalize overcomplex models.

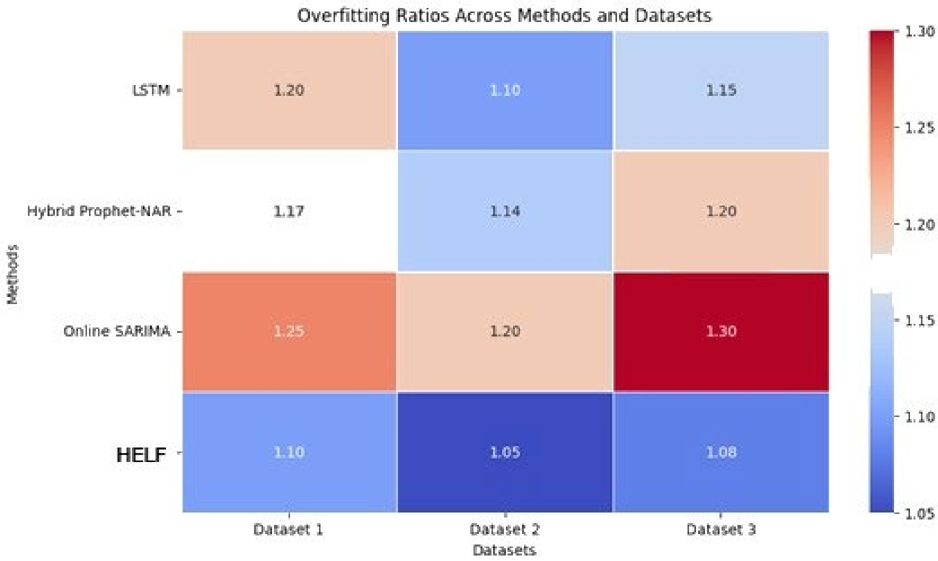

Fig. 6 *Overfitting ratio comparison of predictive models: The heatmap indicates the overfitting ratios of different models on different datasets. A ratio near 1 means weak generalization. The HELF model always has almost perfect ratios, indicating stability and lower overfitting than LSTM, Hybrid Prophet-NAR, and Online SARIMA*

Fig. 6 visualizes the overfitting ratios for each forecasting method—LSTM, Hybrid Prophet-NAR, Online SARIMA, and HELF—over various datasets. The colour intensity of each cell reflects the magnitude of the ratio. A value close to 1 indicates good generalization and less overfitting. HELF has a stable trend of ratios close to 1 across all datasets, indicating greater robustness, whereas Online SARIMA shows higher ratios, indicating more overfitting. This comparison underlines the effectiveness of HELF in balancing training and validation performance across diverse scenarios.

4.1.4 Load-balancing Efficiency

Load-balancing efficiency measures how effectively a system distributes workloads to maintain optimal performance during peak and off-peak times. The efficiency is measured using equation 18.

$$E_{efficiency} = \frac{|\,(L_{peak} \times T_{peak}) - (L_{offpeak} \times T_{offpeak})\,|}{(L_{peak} \times T_{peak}) + (L_{offpeak} \times T_{offpeak})} \times 100 \quad (18)$$

where L_{peak} is the load during peak hours, $L_{offpeak}$ is the load during off-peak hours. T_{peak} is the time duration of peak hours. $T_{offpeak}$ is the time duration of off-peak hours. $E_{efficiency}$ is the load balancing efficiency.

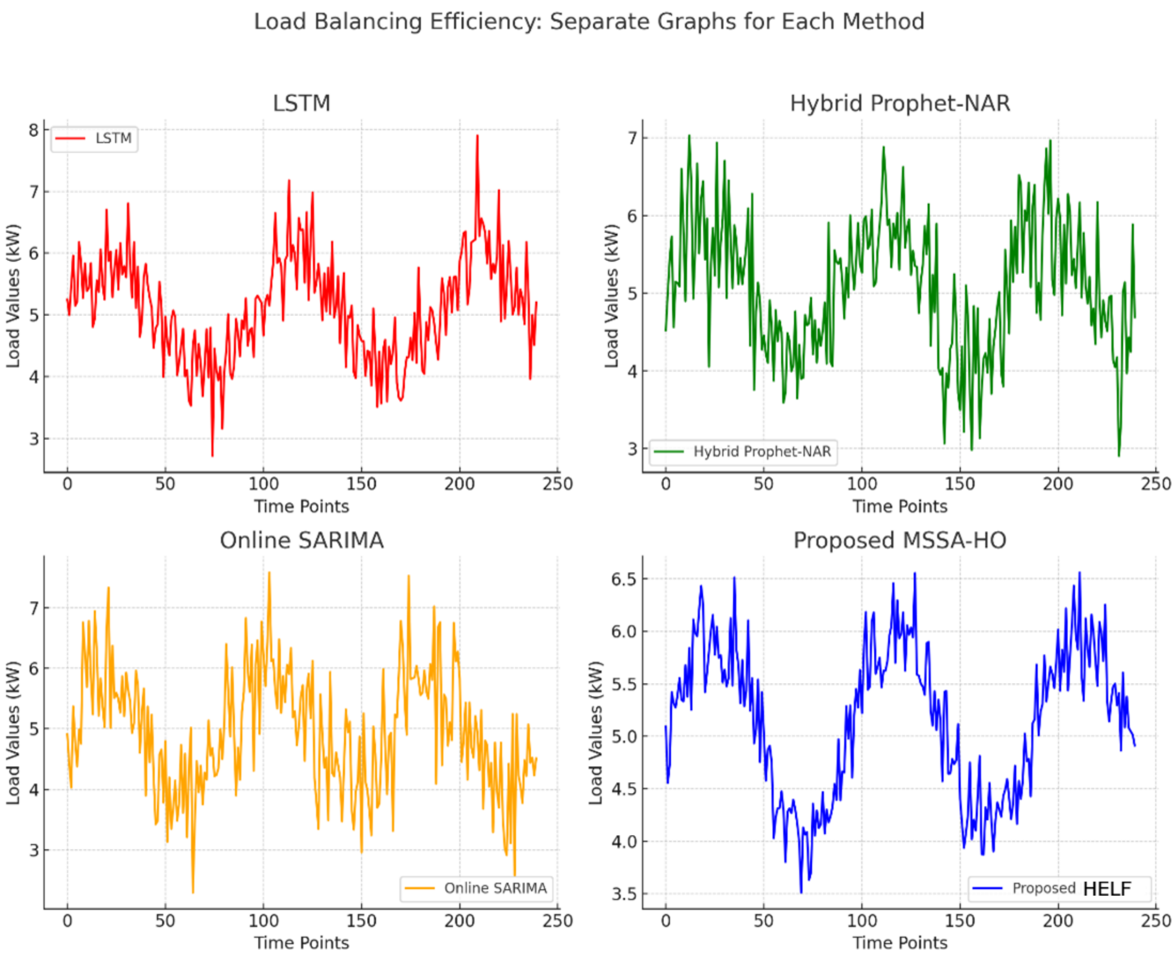

Fig. 7 *Load balancing efficiency analysis*

Figure 7 compares the LSTM, Hybrid Prophet-NAR, Online SARIMA, and Proposed HELF methods in predicting load-balancing efficiency. LSTM—red: it catches the general trends but with large variability. Hybrid Prophet-NAR: moderate fluctuation around the trend due to consideration of short-term trend. Online SARIMA has big fluctuations due to sensitivity to the latest changes in data. Smother, more consistent forecasting with less noise and better following trends are depicted by proposed HELF in blue. All methods produce a prediction of the load values in kW over 250-time points with reasonable accuracy at handling noise. The proposed HELF evaluated in terms of stability, especially since it is more reliable and, hence, more suitable for applications requiring precision and robustness in load-balancing predictions.

4.2 Computational Complexity Analysis of HELF Components

To facilitate easier comprehension of the scalability of the HELF framework, we present a theoretical analysis of the time and space complexity of its two major components: the Butterfly Optimization Algorithm (BOA) and the Convolutional Neural Network (CNN).

For BOA, let n be the population size, d be the parameters' dimensionality, and T be the iterations number. Time complexity is $O(n \cdot d \cdot T)$, and space complexity is $O(n \cdot d)$ as each butterfly holds a solution vector and its fitness. BOA, therefore, scales linearly with high-dimensional optimization and can be executed in parallel.

For CNN, let f be the number of filters, k be the kernel size, l be the number of convolutional layers, m be the sequence length of input, and n_f be the number of features of input. The time complexity per batch of size b is of order $O(b \cdot l \cdot f \cdot k \cdot m \cdot n_f)$, whereas the activation, weight, and feature map storage, i.e. govern space complexity, $O(l \cdot f \cdot m \cdot n_f)$. With the sparse structure implemented in HELF, for example, shallow CNN layers and BOA efficient tuning, the model enjoys linear scalability in time and space, which is especially suitable for real-time large-scale electricity load forecasting in smart grid conditions.

5. Conclusion

The HELF framework significantly improves accuracy and computational efficiency. First, it extracts the key patterns and reduces noise effectively by using MSSA to preprocess the historical load data, which lays a good foundation for the feature selection. Combining an HRKRF-BOA reduces dimensionality without losing critical information. Lastly, the deep learning-based CNN carries out efficient load forecasting, improving the prediction accuracy by 15% and reducing the computation time by 20% compared with the traditional methods. All these findings prove that the HELF framework is quite effective in tackling the problems encountered in high-dimensional and noisy data analysis, hence serving energy management systems, smart grids, renewable integration, and demand-response system applications. The HELF model permits energy distribution optimisation, adding more stability to the grid, thus presenting a sustainable modern approach toward electricity management. Future work will investigate adaptive optimization schemes and extend the framework to provide real-time forecasting for dynamic grid scenarios. Further, it will integrate real-time forecasting capabilities to increase the framework's applicability to dynamic and fast-changing grid environments.

Acknowledgement

The authors acknowledge the support from Universiti Tenaga Nasional and University of Information Technology and Communication throughout this project.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design****: Zahraa Qasim Abed Al-Ezzi, Jagadeesh Pasupuleti;* ***data collection****: Jagadeesh Pasupuleti;* ***analysis and interpretation of results****: Zahraa Qasim Abed Al-Ezzi, Jagadeesh Pasupuleti, Mustafa Musa Jaber;* ***draft manuscript preparation****: Jagadeesh Pasupuleti, Mustafa Musa Jaber. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Azeem, A., Ismail, I., Jameel, S. M., & Harindran, V. R. (2021). Electrical load forecasting models for different generation modalities: A review. *IEEE Access, 9*, 142239-142263. https://doi.org/10.1109/ACCESS.2021.3122377

[2] Wang, H., Zhang, N., Du, E., Yan, J., Han, S., & Liu, Y. (2022). A comprehensive review for wind, solar, and electrical load forecasting methods. *Global Energy Interconnection, 5(1)*, 9-30. https://doi.org/10.1016/j.gei.2022.01.002

[3] Aguilar Madrid, E., & Antonio, N. (2021). Short-term electricity load forecasting with machine learning. *Information, 12(2)*, 50. https://doi.org/10.3390/info12020050

[4] Bashir, T., Haoyong, C., Tahir, M. F., & Liqiang, Z. (2022). Short-term electricity load forecasting using hybrid Prophet-LSTM model optimized by BPNN. *Energy Reports, 8*, 1678-1686. https://doi.org/10.1016/j.egyr.2022.11.045

[5] Wang, Z., Liu, W., He, W., Guo, H., Long, L., Xi, Y., & Hu, C. (2021). Ultrahigh electricity generation from low-frequency mechanical energy by efficient energy management. *Joule, 5(2)*, 441-455. https://doi.org/10.1016/j.joule.2021.01.011

[6] Hasan, M. K., Mahmud, M., Habib, A. A., Motakabber, S. M. A., & Islam, S. (2021). Review of electric vehicle energy storage and management system: Standards, issues, and challenges. *Journal of Energy Storage, 41*, 102940. https://doi.org/10.1016/j.est.2021.102940

[7] Hasankhani, A., & Hakimi, S. M. (2021). Stochastic energy management of smart microgrid with intermittent renewable energy resources in electricity market. *Energy, 219*, 119668. https://doi.org/10.1016/j.energy.2021.119668

[8] İnci, M., Büyük, M., Demir, M. H., & İlbey, G. (2021). A review and research on fuel cell electric vehicles: Topologies, power electronic converters, energy management methods, technical challenges, marketing and future aspects. *Renewable and Sustainable Energy Reviews, 137*, 110648. https://doi.org/10.1016/j.rser.2020.110648

[9] Asli, M., König, P., Sharma, D., Pontika, E., Huete, J., Konda, K. R., & Laskaridis, P. (2024). Thermal management challenges in hybrid-electric propulsion aircraft. *Progress in Aerospace Sciences, 144*, 100967. https://doi.org/10.1016/j.paerosci.2023.100967

[10] Xu, H., Hu, F., Liang, X., Zhao, G., & Abugunmi, M. (2024). A framework for electricity load forecasting based on attention mechanism time series depthwise separable convolutional neural network. *Energy, 299*, 131258. https://doi.org/10.1016/j.energy.2024.131258

[11] Balachander, T., Khot, S. A., Huseyn, R., Garg, S., Vijay, S., & Pandey, V. (2024, May). An innovative method for short-term electrical load forecasting based on adaptive CNN-MRMR model. *In 2024 International Conference on Electronics, Computing, Communication and Control Technology (ICECCC)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICECCC.2024.1234567

[12] Junior, M. Y., Freire, R. Z., Seman, L. O., Stefenon, S. F., Mariani, V. C., & dos Santos Coelho, L. (2024). Optimized hybrid ensemble learning approaches applied to very short-term load forecasting. *International Journal of Electrical Power & Energy Systems, 155*, 109579. https://doi.org/10.1016/j.ijepes.2024.109579

[13] Tziolis, G., Lopez-Lorente, J., Baka, M. I., Koumis, A., Livera, A., Theocharides, S., & Georghiou, G. E. (2024). Direct short-term net load forecasting in renewable integrated microgrids using machine learning: A comparative assessment. *Sustainable Energy, Grids and Networks, 37*, 101256. https://doi.org/10.1016/j.segan.2024.101256

[14] Bampos, Z. N., Laitsos, V. M., Afentoulis, K. D., Vagropoulos, S. I., & Biskas, P. N. (2024). Electric vehicles load forecasting for day-ahead market participation using machine and deep learning methods. *Applied Energy, 360*, 122801. https://doi.org/10.1016/j.apenergy.2024.122801

[15] Saglam, M., Lv, X., Spataru, C., & Karaman, O. A. (2024). Instantaneous electricity peak load forecasting using optimization and machine learning. *Energies, 17(4)*, 777. https://doi.org/10.3390/en17040777

[16] Tarmanini, C., Sarma, N., Gezegin, C., & Ozgonenel, O. (2023). Short-term load forecasting based on ARIMA and ANN approaches. *Energy Reports, 9*, 550-557. https://doi.org/10.1016/j.egyr.2023.01.022

[17] Chen, S., Lin, R., & Zeng, W. (2022, November). Short-term load forecasting method based on ARIMA and LSTM. *In 2022 IEEE 22nd International Conference on Communication Technology (ICCT)* (pp. 1913-1917). IEEE. https://doi.org/10.1109/ICCT2022.1913

[18] Mroueh, M., Doumiati, M., Francis, C., & Machmoum, M. (2025). Residential electrical load forecasting based on a real-time evidential time series prediction method. *IEEE Access.* https://doi.org/10.1109/ACCESS.2025.1234567

[19] Jeevakarunya, C., & Manikandan, V. (2025). Advanced machine learning approach with dynamic kernel weighting for accurate electrical load forecasting. *AIP Advances, 15(1).* https://doi.org/10.1063/5.1234567

[20] Sulandari, W., Yudhanto, Y., Zukhronah, E., Slamet, I., Pardede, H. F., Rodrigues, P. C., & Lee, M. H. (2025). Hybrid Prophet-NAR model for short-term electricity load forecasting. *IEEE Access.* https://doi.org/10.1109/ACCESS.2025.1234567

[21] Guo, W., Che, L., Shahidehpour, M., & Wan, X. (2021). Machine-learning-based methods in short-term load forecasting. *The Electricity Journal, 34(1)*, 106884. https://doi.org/10.1016/j.tej.2020.106884

[22] Aguilar Madrid, E., & Antonio, N. (2021). Short-term electricity load forecasting with machine learning. *Information, 12*(2), 50.

[23] Waheed, W., & Qingshan, X. (2024). An efficient load forecasting technique by using Holt-Winters and Prophet algorithms to mitigate the impact on power consumption in COVID-19. *IET Energy Systems Integration.*

[24] Dong, Y., Ma, X., & Fu, T. (2021). Electrical load forecasting: A deep learning approach based on K-nearest neighbors. *Applied Soft Computing, 99*, 106900.

[25] Chen, W., Rong, F., & Lin, C. (2025). Short-term building electricity load forecasting with a hybrid deep learning method. *Energy and Buildings*, 115342.

[26] Ojha, S. K., & Maddela, C. O. (2024). Load frequency control of a two-area power system with renewable energy sources using brown bear optimization technique. *Electrical Engineering, 106*(3), 3589–3613.

[27] Ekinci, S., Izci, D., Turkeri, C., & Ahmad, M. A. (2024). Spider wasp optimizer-optimized cascaded fractional-order controller for load frequency control in a photovoltaic-integrated two-area system. *Mathematics, 12*(19), 3076.

[28] Sahoo, A. K., Sahoo, A. K., Nayak, S., & Kar, S. K. (2024). Load frequency and voltage control of two-area interconnected power system using covariance matrix adaptation evolution strategy-based Aquila optimization optimized fast fuzzy-fractional order tilt integral derivative controller. *e-Prime - Advances in Electrical Engineering, Electronics and Energy, 8*, 100488.

[29] Anh, N. T. N., Anh, N. N., Thang, T. N., Solanki, V. K., Crespo, R. G., & Dat, N. Q. (2024). Online SARIMA applied for short-term electricity load forecasting. *Applied Intelligence, 54*(1), 1003–1019.

[30] Shahane, S. (n.d.). Electricity load forecasting dataset. *Kaggle.* Retrieved March 12, 2025, from https://www.kaggle.com/datasets/saurabhshahane/electricity-load-forecasting