Advances in Sustainable Energies & Environment, 1 (2026) 1, 18-28

Advances in Sustainable Energies & Environment

Journal homepage: https://A-SEE.MazUST.ac.ir/

Research Article

Enhanced Electricity Load Forecasting Using HHO-Optimized LSTM Networks

Hossein Elahi, Mohammad Mehdi Pourhashem Kallehbasti*, Jamshid Pirgazi

Department of Electrical and Computer Engineering, University of Science and Technology of Mazandaran, Iran

ARTICLE INFO

Article history:
Received: 2025-03-15
Revised: 2025-05-15
Accepted: 2025-05-29
Published online: 2026-01-01

Keywords:
Load forecasting;
Deep learning;
Harris Hawks Optimization;
LSTM.

ABSTRACT

The analysis and forecasting of customer electricity consumption remain among the primary challenges in the power generation industry. Over the past decades, extensive research has been conducted to enhance the accuracy and efficiency of analysis and forecasting methodologies in this domain. With the rapid progress in computer science and artificial intelligence, machine learning algorithms have emerged as robust tools for predicting customer electricity consumption, attracting significant attention from researchers. This paper proposes a hybrid method based on Harris Hawks Optimization (HHO) algorithm and Long Short-Term Memory (LSTM) neural network for load forecasting using time series data. The HHO algorithm is employed to optimize the hyperparameters of the LSTM network, including the number of LSTM units, learning rate, and number of layers. The dataset consists of electricity consumption records, weather conditions, and temporal variables from Panama for the period spanning 2015 to 2020. The evaluation, conducted using RMSE, MAPE, MAE, and MSE metrics, indicates that the proposed HHO-LSTM model outperforms conventional methods such as Support Vector Machines (SVM), linear regression, and basic neural networks. The model achieves a MAPE of 0.08% and an RMSE of 27.36. This approach offers a promising solution for optimizing energy production and distribution planning within smart power systems.

1. Introduction

Electricity has permeated nearly every aspect of modern society, providing substantial benefits to humanity. Effective and systematic planning of energy consumption is essential to ensure the comfort and security of human communities, whereas inadequate planning often leads to sustained losses and adverse consequences. Inaccurate predictions of power demand can result in deficient planning by authorities, leading to significant economic losses for power distribution companies and unnecessary energy waste. Consequently, precise forecasting of electricity consumption is a pivotal challenge in power system planning. Research conducted by Hobbs et al. [1] has demonstrated that a 1% reduction in load forecasting error for a 10-gigawatt power plant can result in annual economic savings of up to $1.6 million.

Initially, traditional energy consumption forecasting methods were introduced by various experts and researchers. These methods are advantageous due to their simplicity in calculation and implementation. However, a significant limitation lies in their low predictive accuracy, which restricts management authorities and decision-makers from formulating logical and precise strategies [2].

The rapid evolution of power systems, driven by the privatization of the electricity industry and the expansion of renewable energy sources, has rendered traditional forecasting approaches increasingly inadequate. These methods are unable to provide generalizable and interpretable insights for future demand. Developing models capable of predicting demand for the upcoming days and hours and offering timely and reliable predictions is, therefore, indispensable. Accurate daily energy consumption forecasting supports distribution companies in maintaining operational reserves while simultaneously reducing costs for both companies and their customers [3].

The study presented in [4] evaluates regression and machine learning models for electrical load forecasting in commercial buildings. The primary objective is to assess the advantages, limitations, and real-world performance of these

*** Corresponding author.**
E-mail address: *pourhashem@mazust.ac.ir*

Cite this article as:

Elahi, H., Pourhashem Kallehbasti, M.M., & Pirgazi, J. (2026). Enhanced Electricity Load Forecasting Using HHO-Optimized LSTM Networks. *Advances in Sustainable Energies & Environment.* https://doi.org/10.22034/a-see.2025.2055995.1002

models. Accurate prediction of commercial building loads contributes to reducing energy consumption, peak demand, and greenhouse gas emissions while enhancing energy management and economic efficiency. Nevertheless, forecasting energy consumption in commercial buildings, particularly for systems such as heating, ventilation, and air conditioning (HVAC), poses substantial complexity [5]. Factors such as weather conditions, occupancy patterns, and activity schedules significantly influence energy consumption, making precise predictions challenging. Regression models, including simple linear regression (SLR), multivariate linear regression (MLR), and simulation-based models [6], are straightforward and do not require specialized expertise. Machine learning models, such as artificial neural networks (ANN), support vector machines (SVM), and regression trees, excel at identifying complex, nonlinear relationships within data [7].

In [8], utilizing real-world data from the Tyree Energy Technologies Building and the University of New South Wales (UNSW) campus in Australia, the study demonstrates that the selection of models should correspond to the available resources, specific requirements, and the complexity of the environment. Machine learning models are particularly effective in sophisticated applications involving complex datasets, whereas regression models are more suitable for simpler analyses.

A practical study [9] investigated a fuzzy interactive regression method for short-term load forecasting (STLF), which outperformed both simple fuzzy regression and multiple linear regression. By incorporating variable interactions, such as temperature with time/day/month and time of day with weekday, and employing hourly load and temperature data from 2005-2007, the proposed method enhanced forecasting accuracy.

In [10], the integration of the chaotic ant swarm (CAS) algorithm with support vector regression (SVR) effectively addressed challenges encountered by classical methods (e.g., ARIMA) and intelligent methods (e.g., neural networks) in managing nonlinear data and severe fluctuations. An analysis of Taiwan's data (1981-2000) demonstrated the superior accuracy of the SVR-CAS model, which achieved the lowest MAPE. These studies highlight the importance of integrating intelligent techniques, such as fuzzy and chaotic methods, with forecasting models to avoid local optima and effectively handle data ambiguity and instability, thereby advancing power system optimization.

In [11], a hybrid CNN-LSTM model was developed for short-term load forecasting in Bangladesh. By combining CNN's capability for feature extraction with LSTM's [12] strength in capturing temporal dependencies, this hybrid approach outperformed standalone models such as LSTM, RBFN, and XGBoost in MAE, RMSE, MAPE, and R^2 metrics, providing an efficient solution for power systems exhibiting nonlinear fluctuations. Yang et al. [13] proposed the Variational Mode Decomposition (VMD)- Variational Auto Encoder (VAE)-LSTM [14] model for short-term electrical load forecasting. This model employed data decomposition (VMD), noise and dimensionality reduction (VAE) [15], and temporal forecasting (LSTM), achieving remarkable accuracy with MAPE values of 1% and 0.8% on real-world Chinese datasets, surpassing other methods. This framework presents an effective solution for managing complex and noisy data in power systems.

A review of existing literature on electricity consumption forecasting reveals a wide array of approaches, including neural networks, linear regression, LSTM models, and machine learning techniques such as SVM. Findings indicate that machine learning significantly enhances forecasting accuracy, while parameter optimization and feature engineering play a crucial role. Nevertheless, current forecasting methods exhibit limitations, including sensitivity to weather fluctuations, which often leads to prediction inaccuracies. Although deep learning models excel in identifying complex patterns, challenges such as model complexity, extensive data requirements, high computational costs, interpretability issues, sensitivity to initial parameters, and the need for precise input tuning remain. To address these limitations, this study proposes a hybrid method that combines LSTM neural networks with HHO algorithm for parameter tuning. This approach overcomes existing constraints and enhances both accuracy and robustness in power system load forecasting.

The proposed HHO-LSTM model differs structurally from traditional methods like SVM, linear regression, and basic neural networks by incorporating the Harris Hawks Optimization algorithm to optimize the hyperparameters of the LSTM network. While conventional models rely on fixed or manually tuned parameters, our hybrid approach dynamically adjusts key parameters such as the number of LSTM units and learning rate, enhancing forecasting accuracy. This structural innovation allows the model to better handle complex, multidimensional time series data, making it more effective for energy load forecasting in smart grid systems.

The main novelty of this study lies in the integration of the HHO algorithm with a LSTM network for electricity load forecasting. Unlike prior works that rely on conventional optimization methods such as Genetic Algorithms or Particle Swarm Optimization, this research is the first to systematically apply HHO for deep learning hyperparameter tuning in the energy domain. The proposed hybrid LSTM-HHO framework enables precise adjustment of model parameters, including the number of layers, neurons, batch size, and learning rate, significantly improving prediction accuracy. Furthermore, a two-stage feature selection process combining Recursive Feature Elimination (RFE) and Random Forest ensures the model utilizes only the most impactful features, enhancing both efficiency and performance. Experimental results demonstrate the superior accuracy of the proposed model (e.g., MAPE as low as 2.17%) over other evolutionary and hybrid models, validating its effectiveness across short- and long-term forecasting horizons. This work provides a scalable and intelligent forecasting approach applicable to modern energy management systems.

2. Proposed Method

To predict customer consumption, this study proposes a method based on the integration of HHO [16] algorithm with LSTM deep learning model. The forecasting process is divided into four stages. The first stage entails data collection and preprocessing. The second stage focuses on feature selection through the use of the Rotation Forest method. In the third stage, the LSTM hyperparameters are optimized using the LSTM-HHO method. Finally, in the fourth stage, the model is trained and evaluated. Figure 1 presents an overview of the main stages of the proposed approach, which will be elaborated upon in subsequent sections. The primary research gap addressed in this study is the challenge of selecting appropriate hyperparameters for the LSTM network, a critical factor for accurately capturing the sequential dependencies inherent in time-series data. To tackle this issue, the HHO algorithm an optimization technique inspired by the foraging and hunting behavior of hawks is employed to effectively fine-tune the LSTM hyperparameters.

In this study, HHO algorithm was employed to determine the optimal values for eight hyperparameters of LSTM network. These hyperparameters include the number of LSTM units, epochs, batch size, LSTM layers, dense neurons, learning rate, optimizer type, and activation function. These parameters are crucial for the performance and accuracy of the LSTM network, and their appropriate tuning can significantly enhance the model's predictive capabilities.

2.1 Dataset Preparation

This study addresses the prediction of daily energy consumption using historical data and environmental features. A dataset from Panama’s national electricity distribution company is used, which includes hourly energy demand records over a 24-hour period spanning from 2015 to 2020 [17,18]. The dataset comprises 48,048 rows, corresponding to 2002 days of energy demand, and incorporates various features such as temperature, humidity, air pressure, holidays, and school schedules to account for environmental and social influences on energy consumption. The proposed model employs 15 features per day and leverages historical data from the preceding 30 days to improve the accuracy of predicting the target day’s energy consumption. In addition to the current day’s features, historical data are integrated into the model to simulate the conditions expected on the predicted day. This approach facilitates the identification of hidden patterns in energy consumption and provides insight into long-term trends and fluctuations over time. Extensive experimentation led to the selection of a 30-day temporal window, allowing the model to utilize 450 distinct features (15 features × 30 days) for generating more precise daily energy consumption forecasts. This methodology has the potential to optimize energy management, reduce operational costs, and improve efficiency in electricity distribution networks.

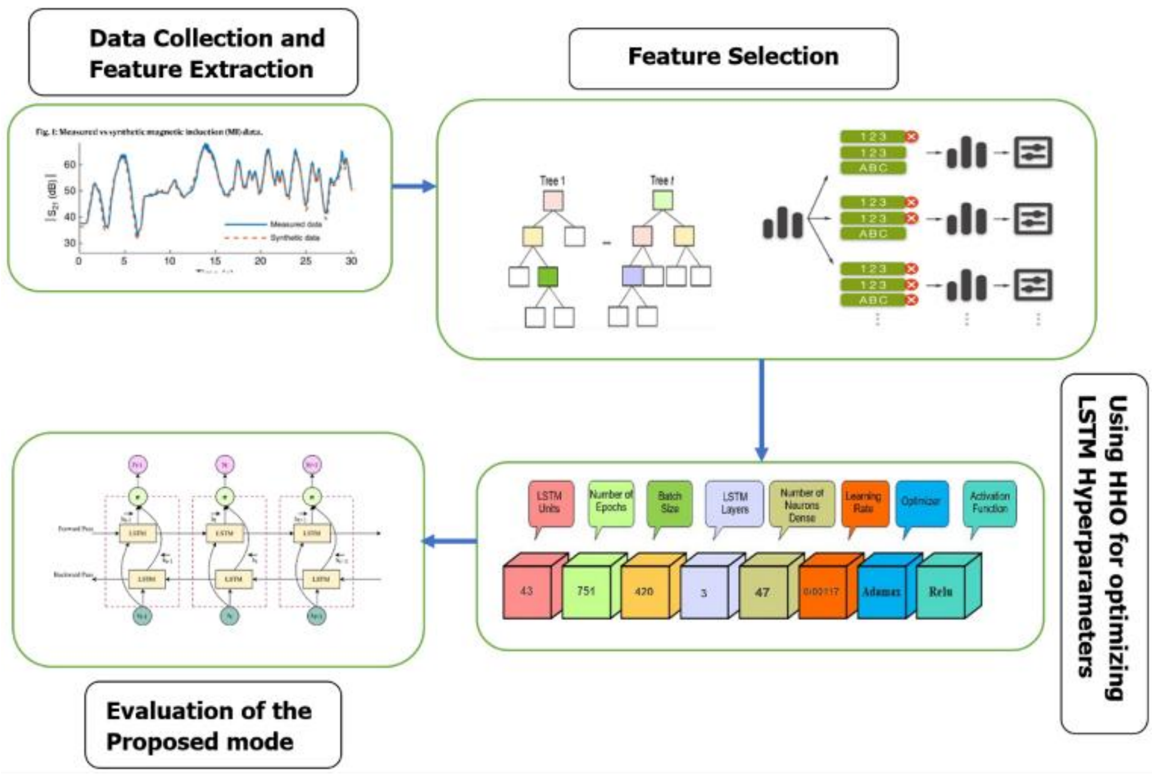

Fig. 1. Overview of the proposed methodology for load forecasting

2.2 Feature selection

In load prediction, the selection of appropriate features is critical for improving model accuracy and reducing complexity. The aim of this process is to identify features that have the most significant predictive impact while eliminating redundant ones that could potentially impair model performance. To achieve this, a two-stage hybrid approach is adopted, combining Recursive Feature Elimination (RFE) [19] with the Random Forest algorithm. This integration ensures an optimal balance between feature relevance and computational efficiency.

- **Removing less important features with REF method**

RFE method operates using a machine learning model and systematically removes features with minimal impact on the model's performance. Initially, all available features are input into the model, and its performance is evaluated. Features with the least influence on accuracy are identified and eliminated. This process is repeated iteratively until the feature set reaches an optimal size. By the end of this stage, the top 100 features with the most significant impact on the model's accuracy are selected. Employing RFE effectively removes irrelevant and noisy features, enabling the model to focus on impactful data. This approach not only improves prediction accuracy but also reduces the model's complexity.

- **Effective feature selection with the random forest algorithm**

Following the application of RFE and the selection of the top 100 features, Random Forest algorithm is used to further assess the selected features. Random Forest, an ensemble-based model composed of independently constructed decision trees, evaluates the importance of features based on their contribution to the prediction process. In the initial step, Random Forest calculates the importance of each feature and determines which features are most critical to the model's accuracy. From the 100 features identified in the first stage, the 20 features with the greatest impact on prediction accuracy are selected, while those with minimal influence are excluded. This ensures that the model focuses exclusively on essential data, thereby improving both efficiency and precision.

2.3 Optimization of LSTM model parameters based on HHO

After identifying influential features for load forecasting through RFE and Random Forest, the subsequent stage involves model optimization. Among the most critical steps in developing an accurate and efficient predictive system is hyperparameter tuning. In deep learning models, particularly LSTM networks, the appropriate selection of hyperparameters significantly influences model performance. Key LSTM hyperparameters, including the number of LSTM units, epochs, batch size, LSTM layers, dense neurons, learning rate, optimizer type, and activation function, have a profound impact on prediction accuracy and training duration.

In this study, HHO algorithm is employed for hyper-parameter optimization. Inspired by the cooperative hunting behavior of Harris hawks, HHO is a metaheuristic algorithm designed to efficiently explore and optimize the parameter space. Its capacity for performing global searches and identifying optimal parameter configurations makes it particularly effective for complex problems characterized by vast search spaces. The general workflow of hyperparameter optimization using HHO is depicted in Figure 2, with detailed explanations provided in the following sections.

In this paper, the HHO algorithm is employed to optimize the hyperparameters of the LSTM network for electricity load forecasting. The selection of HHO is driven by its key advantages in addressing challenges associated with complex data and enhancing prediction accuracy. First, HHO is a metaheuristic algorithm inspired by the intelligent cooperative hunting behavior of Harris hawks. It effectively balances exploration (global search) and exploitation (local search), preventing the model from being trapped in local optima while comprehensively exploring the hyperparameter space. This capability is critical for tuning sensitive LSTM hyperparameters, such as the number of layers, learning rate, and activation function type. Second, energy consumption data, influenced by environmental factors (e.g., temperature, humidity, holidays) and characterized by nonlinearity and volatility, require a model capable of capturing intricate temporal patterns. By precisely optimizing LSTM parameters, HHO enhances the model's ability to learn these patterns, significantly reducing forecasting errors.

2.4 Particle initialization

To select the hyperparameters of the LSTM network and initiate the optimization process, it is essential to define the search space and the structure of each hawk. This search space includes the most critical hyperparameters for the LSTM network. Tables 1 and 2 present a detailed list of these hyperparameters, including the number of LSTM units, the maximum number of training iterations, and batch sizes. These hyperparameters play a significant role in shaping the performance and behavior of the LSTM network. These ranges were selected to balance computational efficiency with model performance, ensuring that the search space was broad enough to capture optimal configurations while avoiding unnecessary complexity. The chosen limits reflect practical values observed in similar studies and were refined through multiple trial runs to identify boundaries that lead to stable and accurate model convergence.

Next, to generate each particle (hawk), the random hyperparameters are determined. Specifically, m numbers are selected during each iteration. The structure of the particles in the proposed method is outlined in Table 3.

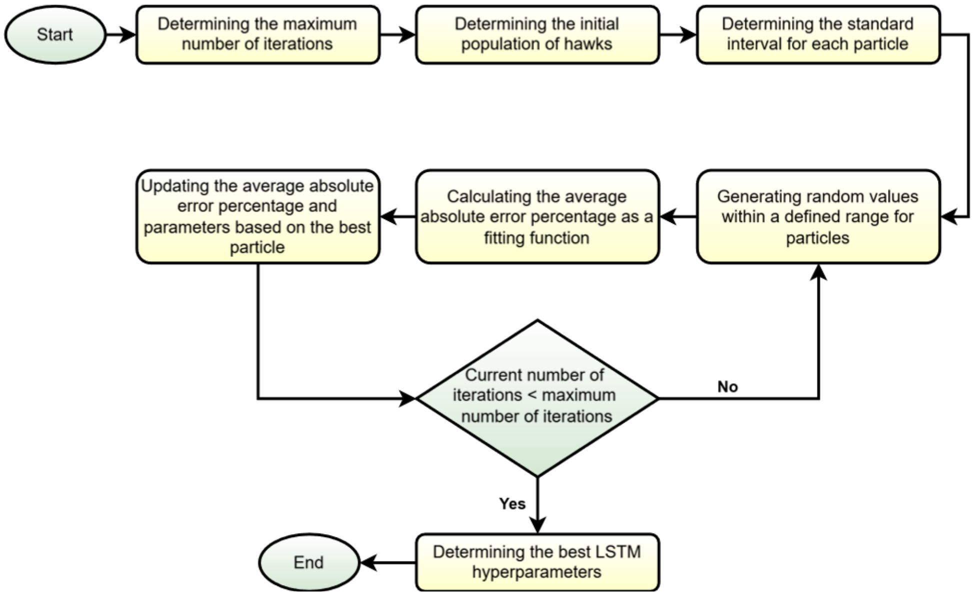

Fig. 2. Flowchart of the HHO algorithm for optimizing LSTM hyperparameters

Table 1. The numerical search space of hyperparameters of LSTM network space

Parameter	Min.	Max
LSTM units	1	100
Epochs	5	1000
Batch size	8	500
LSTM Layers	1	5
Dense Neurons	1	50
Learning Rate	0.0001	0.1

Table 2. LSTM hyperparameter batch search.

Parameter	Value
Optimizer	{'adam', 'rmsprop', 'sgd', 'adadelta', 'adagrad', 'adamax', 'nadam'}
Activation Function	{'relu', 'sigmoid', 'tanh'}

Table 3. Particle structure

Activator function	Relu
Optimizing	Adam
Learning Rate	0.001
Number of Dense Nuron	64
Number of LSTM layers	2
Size of Batch	32
Number of Epochs	100
LSTM Units	128

2.5 Applying the Harris Hawk algorithm (HHO)

HHO algorithm operates through three distinct phases: exploration, transition, and exploitation, where hawks simulate mathematical relationships to target prey. In this framework, the optimal particle (prey) in each generation is determined using a fitness function, with the remaining hawks converging toward it. At the start of every iteration, the algorithm selects one of these phases based on the prey's remaining energy. Random values, including perching chance, prey escape

2.6 Calculating the fitness of each particle

In this study, evaluation metrics are essential for assessing the accuracy and efficiency of predictive models and play a key role in selecting appropriate hyperparameters [20]. For

probability, residual energy, and leap strength, are generated to guide phase selection.

Notably, the prey's energy, escape probability, and leap strength diminish over iterations, illustrating their dependence on algorithmic cycles. Based on the active phase, all particle positions are updated, followed by a recalculation of the fitness function. If updated parameters exceed allowable bounds or result in duplicates, they are randomly reinitialized to ensure thorough exploration of the search space. At the conclusion of each phase, any hawk exhibiting a superior fitness value compared to the current prey replaces it as the new target.

The Mean Absolute Percentage Error (MAPE) for each hawk is calculated five times, with the average recorded as the final fitness value. To further enhance exploration and improve the searchability of the feature space, the two fittest hawks from each generation are selected as parents, with crossover and mutation operators generating two offspring to replace the least-fit hawks from the previous generation. Figure 3 provides an overview of the workflow for the proposed method of optimizing LSTM hyperparameters.

2.7 Stop conditions of the algorithm

The termination of the algorithm is determined by several criteria. One criterion involves a predefined number of iterations specified by the user. In this study, an optimal iteration count of 100 or 300 was identified based on an evaluation of the results and the specific characteristics of the algorithm. Another termination condition is the stagnation in the average detection rate of particles across consecutive generations. If the average accuracy rate of particles remains constant over multiple iterations, the algorithm terminates.

optimizing the hyperparameters of the LSTM neural network and the HHO algorithm, multiple metrics, including MAE, RMSE, and MAPE, are employed. MAPE is particularly advantageous for analyzing time series data and evaluating prediction accuracy relative to actual values, as it expresses error in percentage terms, offering ease of interpretation compared to other metrics. In this research, MAPE is utilized to assess forecasting accuracy by quantifying error as a percentage of actual values, thus reflecting the model's precision in real-world scales [21].

In addition, MAE and RMSE are applied to evaluate prediction performance. MAE computes the average magnitude of prediction errors, providing a straightforward interpretation of the model's effectiveness. RMSE, on the other hand, measures the dispersion of residuals or the closeness of predicted values to actual values, with lower RMSE indicating higher model accuracy. The selection of these metrics is guided by the specific objectives and requirements of the problem being addressed.

3. Evaluation of results

This section assesses the performance of the forecasting model by analyzing the training process, reviewing the datasets, and emphasizing the model's precision in predicting load amounts. The results are further compared with alternative methods to highlight the model's advantages.

3.1 Model Training

Upon completing the hyperparameter optimization of the LSTM model using HHO, the final stage focuses on training and evaluating the load forecasting model. This phase represents the conclusion of the model development process and is essential for evaluating the model's efficiency and accuracy. The LSTM model is trained with optimized hyperparameters and subsequently tested on a designated dataset. The dataset is divided into 70% training data, 15% validation data, and 15% test data, a split also applied during the hyperparameter tuning phase. To ensure statistical significance in the prediction results, k=5 cross-validation is employed during evaluation. Training data, fundamental to machine learning, consist of labeled (or unlabeled, for unsupervised learning) examples that allow the model to learn patterns, relationships, and features [22,23]. These data are used to adjust the model's weights and parameters, enabling it to identify underlying patterns effectively. Validation data, which differ from training data, are used during training to fine-tune the model. In this study, 15% of the dataset is allocated to validation. Test data, fully independent of both training and validation datasets, are reserved to evaluate the final performance of the model. This approach assesses the model's ability to generalize to unseen data, with 15% of the dataset designated for this purpose. The distribution of the dataset is depicted in Figure 4.

During the training phase, the LSTM model is trained using features selected through Recursive Feature Elimination (RFE) and the Random Forest method, combined with hyperparameters optimized by the HHO algorithm. The training dataset consists of historical load consumption time-series data, which is input into the model. Through these data, the LSTM identifies temporal patterns and consumption trends. The training process involves updating the neural network weights using the Adam optimization algorithm. Following this phase, the model's generalization ability is assessed by evaluating it on a holdout test dataset that remains unseen during training. This step measures the model's performance in forecasting load consumption for previously unobserved data, ensuring its reliability and accuracy in practical applications.

3.2 Review of the model training process

Given the multiple parameters in the proposed method, training and validation loss curves were plotted to track the learning process and ensure proper model training. The vertical axis represents the loss value (MSE), while the horizontal axis shows the number of training epochs.

```
Input    D: Dataset;
Output   S: The selected hyperparameters for LSTM
// HHO Initialization

N_var = ٤٠ // Number of hyperparameters to optimize
N_pop = ١٠ // Population size

For i= ١ to N_pop
    Hawk [i].Position= Select N_var hyperparameters from a predefined range
    Hawk [i].Fitness= Evaluate LSTM performance using MAEP
//HHO Main Loop
    While (stopping condition is not met) do
    Hawks =Sort (Hawk. Fitness);   //Sort Ascending by fitness of Hawk.
    Rabbit.Position = Hawks [١].position   // The position of best Hawk
    For i = ١ to N_pop do
     Update the initial energy E٠        E٠[i]=٢rand()-١        //
       J[i]=٢(١-rand())          // Update the initial  jump strength J
        E[i]=٢ E٠ [i](١- t/T)         // Update the E
       if (|E[i]|≥ ١) then        // Exploration phase
           if q >= ٠,٥
                   Hawk [i].Position = RandomHawk.Position – r١| RandomHawk.Position – ٢r٢ Hawk
    [i].Position |
    Hawk [i].Position=GRASP(Hawk [i].Position,D,RF);
      if  q < ٠,٥
                   Hawk [i].Position = (Rabbit.Position – AvrageHawks.position) – r٣ (LB + r٤(UB-
    LB))
    if (|E[i]|< ١) then          // Exploitation phase
    if (r ≥٠,٥ and |E[i]|≥ ٠,٥) then                  // Soft besiege
           Hawk [i].Position  =Δ Hawk [i].Position- E | J Rabbit.Position – Hawk [i].Position |
    else if (r ≥٠,٥ and |E[i]|< ٠,٥) then              // Hard besiege
          Hawk [i].Position = Rabbit.Position -E| Δ Hawk [i].Position |
    else if (r < ٠,٥ and |E[i]|≥ ٠,٥) then             // Soft besiege with progressive
    rapid dives
           if  F(Y) < F(X(t)) then
                       Hawk [i].Position = Y.position
           if  F(Z) < F(X(t)) then
                       Hawk [i].Position = Z.position
    else if (r < ٠,٥ and |E[i]|< ٠,٥) then             // Hard besiege with progressive
    rapid dives
         if  F(Y) < F(X(t)) then
                       Hawk [i].Position = Y.position
           if  F(Z) < F(X(t)) then
                       Hawk [i].Position = Z.position
     Pr١ , Pr٢ =Select_best(Hawks)        //select ٢ Hawks with the best fitness

    Child١ , Child٢ = crossover_ mutation (Pr١, Pr٢)

    substitute ٢ children instead of ٢ samples with the worst fitness

S= Hawks[١].position;          //final subset
```

Fig. 3. The pseudocode of the proposed algorithm

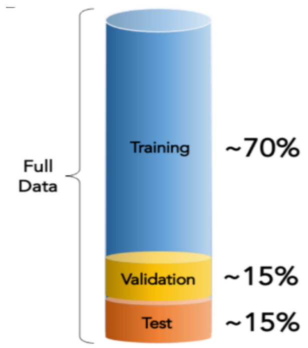

Fig. 4. Data distribution for training, validation, and testing phases

The overall trend indicates a steady decline in both training and validation losses as the number of epochs increases, reflecting gradual learning and improved performance. During the initial training phase, the loss decreases sharply; however, the rate of decline slows in later epochs, indicating convergence where parameter updates have diminishing effects.

A key element of this analysis is detecting overfitting, which arises when the model becomes overly tailored to the training data, limiting its ability to generalize. In this study, the validation loss decreases consistently and remains closely aligned with the training loss, confirming the absence of significant overfitting and demonstrating robust generalization. As depicted in Figure 5, the validation loss initially exhibits variability and diverges from the training loss but stabilizes after epoch 30, with oscillations reducing. By the final epochs, the reduction in loss becomes negligible, and both curves plateau, indicating the model has reached its minimum error. Further training provides little benefit. This analysis validates that the model is well-trained, achieving improved performance on both training and validation data. The stable, downward loss trajectory indicates the model is close to the optimal point, eliminating the necessity for additional training.

In the proposed method, the optimization of LSTM hyperparameters using the HHO algorithm necessitates the definition of initial values for key parameters, including population size (number of hawks) and maximum iterations. These parameters were empirically determined through extensive experimentation to ensure effective optimization and favorable model learning outcomes. The selected values play a critical role in influencing the algorithm's efficiency and the model's predictive accuracy. Parameter tuning was guided by principles derived from the natural predatory behavior of Harris hawks. For example, an excessively large hawk population can lead to resource scarcity, reflecting ecological balance in nature, while an optimal iteration count represents the equilibrium between prey exhaustion and successful hunting. These biologically inspired principles were incorporated into the HHO framework to establish parameter settings. The finalized hyperparameters, such as the "number of hawks generated via mutation and crossover operators" (representing new samples produced through random variations and recombination of top-performing hawks in each iteration) and the "fitness evaluation frequency per hawk" (indicating how frequently the fitness function is computed to evaluate a hawk's suitability for LSTM hyperparameter optimization), are summarized in Table 4.

3.3 Datasets

A correlation matrix is a statistical tool used to represent the linear relationships between features within a dataset. Each element Pij in the matrix corresponds to the correlation coefficient between two features i and j. The matrix is square, with its dimensions reflecting the total number of features, and its values range from -1 (indicating a perfect negative correlation) to +1 (indicating a perfect positive correlation), while 0 indicates no linear relationship between the features. As shown in Figure 6, the correlation between the features and labels is notably low, indicating weak or near-zero correlations.

This implies a lack of strong linear relationships between specific features and the labels, suggesting that these features may have limited informative value. To address this, feature selection methods have been applied to identify more impactful features.

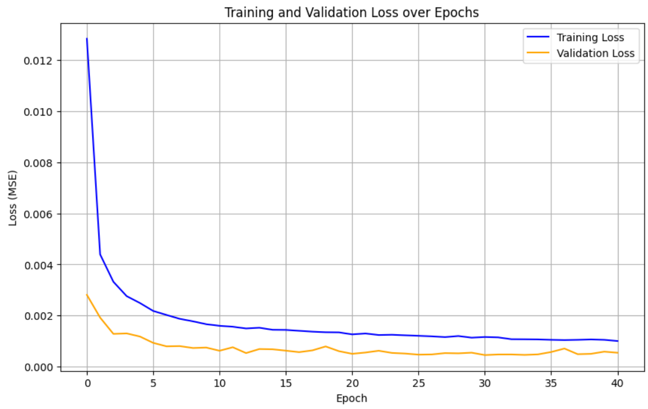

Fig. 5. Error graph on training and validation data in the proposal method

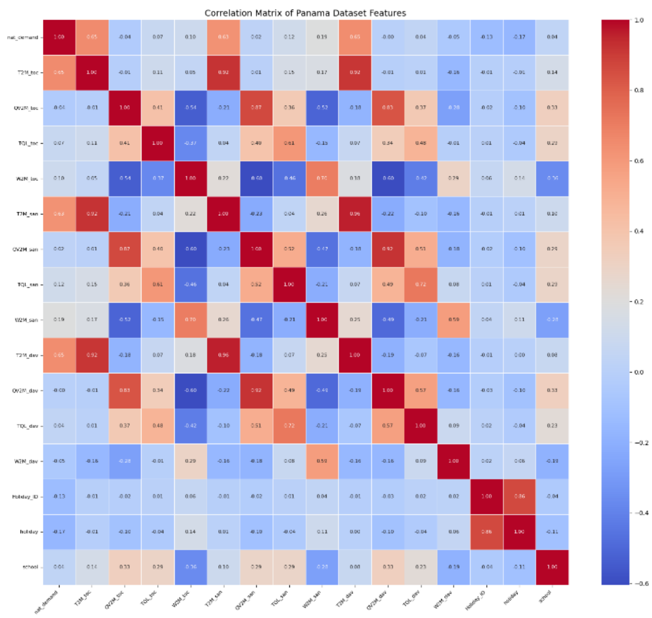

Fig. 6. Correlation matrix between features in the dataset

Table 4. HHO algorithm hyperparameter values

Parameter name	The right amount
Number of hawks per generation	5
Number of hawks produced using mutation and intersection operators	4
Number of times each hawk's fitness is calculated	5
Number of Repeat	500

Figure 7 illustrates the MAPE values across various forecasting horizons. Energy consumption was initially lower in the early years and gradually increased over time, leading to the division of forecasting intervals into four distinct periods. The results show that periods 1 and 2 have the lowest average MAPE values, indicating higher accuracy and reduced error rates during these intervals. An exception is observed in period 4, where the 90-day forecasting horizon shows a notable reduction in error compared to periods 1–3. This highlights the model's enhanced precision during this interval and its ability to effectively capture long-term trends. Furthermore, as depicted in the figure, the next-day (EOD) forecast consistently produces the lowest error within each period. In contrast, the error increases as the forecasting horizon extends further from the current day. Across all periods except period 4, the 90-day and 30-day forecasts exhibit the highest error rates, emphasizing the challenges associated with long-term predictions under typical conditions.

Figure 8 presents the violin plots of evaluation metrics across forecasting horizons ranging from 1 to 30 days. The plots reveal that, for short- to medium-term forecasts, the MAPE tends to converge around 0.6, approximately at the center of the data distribution. The probability density for MAE and RMSE values is concentrated near 25 and 30, respectively, with these metrics showing a skew toward the upper half of the data distribution. Overall, the results demonstrate that the model maintains relatively stable and satisfactory performance in most scenarios. However, certain instances of high-error outliers are observed, likely caused by anomalous data points within the dataset.

3.4 Forecasting the amount of load using the LSTM-HHO model

To assess the effectiveness of the proposed method under various conditions, the predicted values were compared with the actual values. This evaluation allows for a direct measurement of the model's accuracy in capturing demand levels within the forecasted time intervals. Figure 9 depicts the graph of actual electricity consumption demand for subscribers. The vertical axis represents demand magnitude, with values up to 500 units, while the horizontal axis indicates time, spanning up to 2500 time units.

Figure 10 displays the visualization of subscriber consumption predictions produced by the LSTM-HHO model. Figure 11 provides a comparative representation of actual versus predicted demand, showcasing the performance of the LSTM-HHO model examined in this study. The vertical axis represents the demand quantity, while the horizontal axis indicates time, divided into five 500-unit intervals ranging from 0 to 2500.

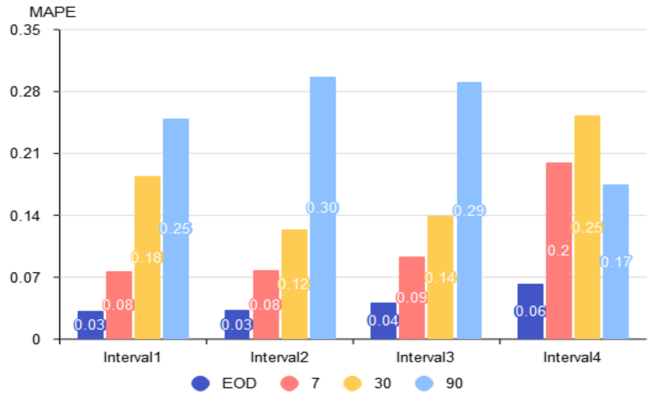

Fig. 7. MAPE values at different time horizons to evaluate the model's forecast accuracy

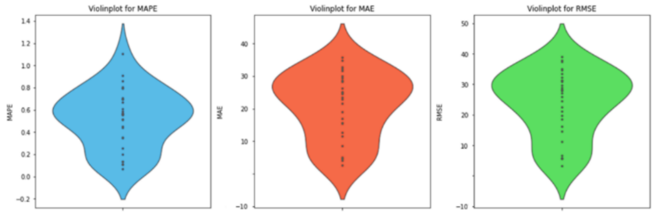

Fig. 8. Violin plot of evaluation criteria including (a) MAPE (b) MAE, (c) RMSE for time horizons of 1 to 30 days.

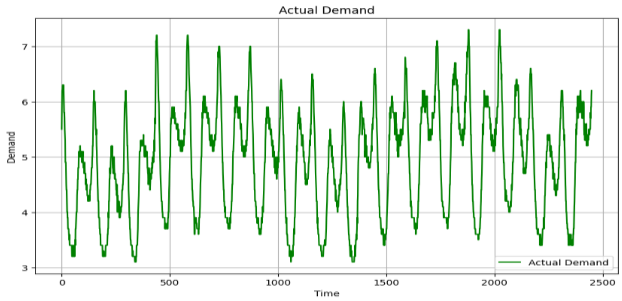

Fig. 9. Actual electricity consumption demand of subscribers over time

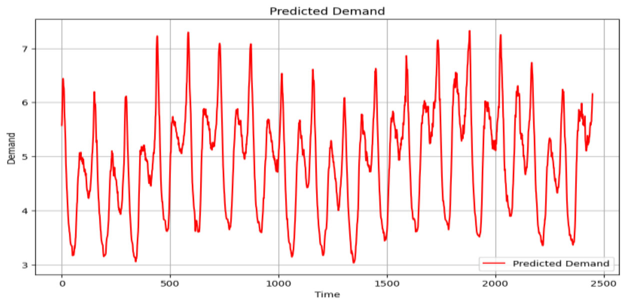

Fig. 10. Subscriber electricity consumption predictions using the LSTM-HHO model

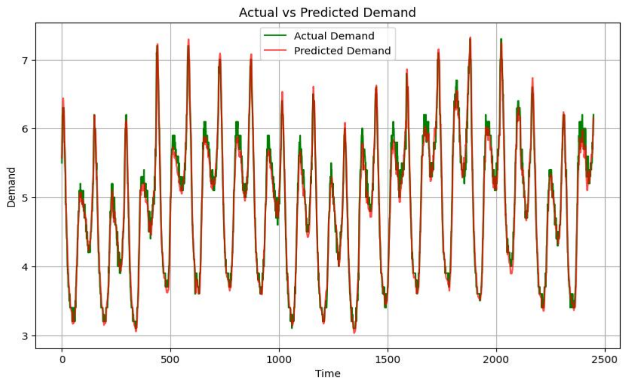

Fig. 11. Performance of the LSTM-HHO model

Figure 12 illustrates the forecast performance over an 8-day period, encompassing 200 data points. The results demonstrate that the proposed model achieves satisfactory accuracy in its predictions.

3.5 Comparison with other methods

The GA-LSTM, PSO-LSTM, and CSO-LSTM models combine Long Short-Term Memory (LSTM) neural networks with distinct optimization algorithms to improve forecasting performance. In the GA-LSTM model, a Genetic Algorithm optimizes LSTM parameters via evolutionary mechanisms such as mutation and crossover, thereby enabling adaptive parameter refinement. The PSO-LSTM model employs Particle Swarm Optimization, which simulates the collective movement of particles within a search space to dynamically identify optimal parameter values. The CSO-LSTM model relies on Chicken Swarm Optimization, leveraging the hierarchical flocking behavior of chickens. By partitioning the population into leader-following subgroups, the CSO-LSTM model attains a balanced trade-off between global exploration, which involves searching for diverse solutions, and local exploitation, which focuses on refining promising solutions, thus improving convergence and prediction accuracy. These hybrid approaches demonstrate the potential of metaheuristic optimization techniques to complement LSTM architectures when addressing complex time-series forecasting challenges.

Table 5 presents a comparison between the proposed method and alternative evolutionary approaches for LSTM optimization, evaluated using the metrics MAE, MAPE, and RMSE. The results indicate that the proposed method outperforms the other techniques, achieving MAE, MAPE, and RMSE values of 19.23, 0.08, and 27.36, respectively. These findings highlight the superior accuracy and precision of the proposed method in forecasting power consumption levels over the one-month period, surpassing the performance of existing evolutionary methods.

Compared to other hybrid models, the proposed LSTM-HHO method combined with F-score feature selection achieved superior performance, with the lowest MAPE (0.08) and RMSE (27.36). While methods like GA-LSTM and PSO-LSTM showed comparable MAPE values, their RMSE and MAE were significantly higher. The integration of F-score helped reduce input dimensionality, enhancing both accuracy and computational efficiency.

Table 6 compares the proposed LSTM-HHO model with other models, emphasizing its notable improvement in prediction accuracy. The proposed model outperforms LSTM, EMD-LSTM, SSA-LSTM, and STL-LSTM, in both 48- and 90-period horizons, achieving superior MAE and RMSE values. For the 48-period horizon, the LSTM-HHO model achieves an MAE of 20.17 and RMSE of 29.43, significantly outperforming the baseline LSTM model (MAE = 31.78, RMSE = 38.84). This reduction in error demonstrates the enhanced precision of the proposed model, attributed to its optimized hyperparameters. In the 90-period horizon, the LSTM-HHO model maintains strong performance, with MAE and RMSE values of 24.71 and 33.08, respectively, while the EMDHM model (MAE = 27.09) and SSA-LSTM model (MAE = 29.43, RMSE = 40.77) show comparatively weaker results. These findings further validate the superior accuracy of the LSTM-HHO model across different forecasting horizons.

However, as indicated in Table 6, the EMDHM model outperforms the proposed LSTM-HHO model in RMSE performance. For example, in the 48-period horizon, the EMDHM model achieves an RMSE of 28.08, and in the 90-period horizon, it attains an RMSE of 32.85, exceeding the proposed model's RMSE values of 29.43 (48-period) and 33.08 (90-period). This discrepancy suggests that the EMDHM model delivers more precise forecasts with lower error margins in these horizons. The EMDHM model's advantage in RMSE is likely due to its integration of Empirical Mode Decomposition (EMD) and Detrended Fluctuation Analysis (DFA), which facilitate effective modeling of long-term trends and high-frequency fluctuations in the data.

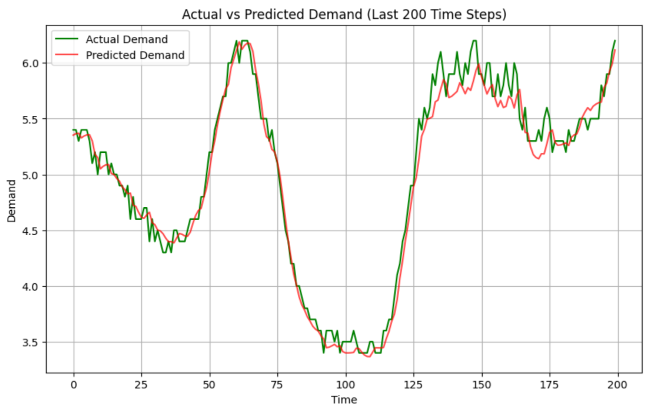

Fig. 12. Performance of the LSTM-HHO model for 200 data equivalents to 8 days

Table 5. Comparison of the proposed model with other evolutionary methods based on the evaluation metrics MAE, MAPE, and RMSE.

Model	MAPE	MAE	RMSE
GA-LSTM	0.11	22.56	45.72
PSO-LSTM	0.11	37.72	51.09
CSO-LSTM	0.11	24.69	47.76
FSS + LSTM-HHO	0.1	21.18	29.78
Proposed LSTM-HHO	**0.08**	**19.23**	**27.36**

Table 6. Comparing efficiency of the proposed method with other methods

	MAE		RMSE		
	Course length		Course length		
Model	48	90	48	90	**Ref**
LSTM	31.78	32.19	38.84	40.63	[20]
EMD-LSTM	30.48	32.92	39.03	39.34	[24]
SSA-LSTM	29.43	31.76	34.96	40.77	[24]
STL-LSTM	36.09	35.35	42.41	44.10	[24]
EMDHM	23.55	27.09	28.08	32.85	[24]
Proposed LSTM-HHO	**20.17**	**24.71**	**29.43**	**33.08**	

4. Conclusion

Accurate and efficient forecasting of electricity demand is a critical component in modern power systems, particularly in the context of smart grids, renewable energy integration, and operational planning. This study introduced a novel hybrid forecasting model that combines the LSTM neural network with the Harris Hawks Optimization algorithm to address the complex challenges inherent in time series-based electricity load forecasting. Through the synergy of deep learning and metaheuristic optimization, the proposed model effectively captures both short-term variations and long-term dependencies in energy consumption data. The methodological framework of this study was structured into four main stages: data collection and preprocessing, feature selection using RFE and Random Forest, hyperparameter optimization using the HHO algorithm, and final model training and evaluation. The dataset, derived from Panama's national electricity distribution records (2015–2020), included diverse environmental, temporal, and social variables. The inclusion of features such as temperature, humidity, holidays, and school schedules enriched the model's input space and enabled the network to capture external influences on consumption patterns. One of the core contributions of this work lies in the comprehensive optimization of LSTM hyperparameters such as the number of layers, neurons, learning rate, batch size, and activation functions using the HHO algorithm. This bio-inspired algorithm, mimicking the predatory behavior of Harris hawks, provided a powerful search mechanism for navigating the vast hyperparameter space. Unlike conventional optimization approaches that are often prone to local minima or require intensive manual tuning, the HHO algorithm demonstrated strong global search capabilities and adaptive convergence, which significantly improved the model's training stability and forecasting accuracy.

Experimental results and performance comparisons underscored the effectiveness of the proposed model. Evaluation metrics, including Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE), revealed that the LSTM-HHO model consistently outperformed other models such as GA-LSTM, PSO-LSTM, and CSO-LSTM across multiple forecasting horizons. Specifically, the model achieved a MAPE of 0.08% and an RMSE of 27.36 on the primary dataset, reflecting high predictive precision. In comparative studies over 30- and 90-day horizons, the LSTM-HHO model achieved lower error margins compared to baseline LSTM and other hybrid models like STL-LSTM, EMD-LSTM, and SSA-LSTM. Furthermore, the model exhibited strong generalization capabilities, as evidenced by the stability of training and validation loss curves and the absence of significant overfitting. The use of 5-fold cross-validation and hold-out testing validated the model's robustness across unseen data. In addition, violin plots and period-based MAPE analysis confirmed the model's adaptability across various forecasting horizons and consumption scenarios.

5. Future Works

The proposed research opens several promising directions for future exploration and development. One area of focus is the incorporation of appropriate cost functions in regression models to improve the prediction of challenging samples. By employing cost functions tailored to handle these hard-to-predict cases, the model's robustness and overall accuracy can be enhanced.

Another potential avenue is the utilization of transformer-based network architectures, which have demonstrated exceptional performance in capturing long-range dependencies and complex patterns in sequential data. Adopting these advanced models could further enhance the predictive capabilities of the framework.

Finally, the integration of hybrid learning methods presents an exciting opportunity to improve model performance. By combining complementary approaches, hybrid learning can leverage the strengths of multiple methodologies, leading to more accurate and efficient predictions. These advancements have the potential to elevate the effectiveness of forecasting models and address existing challenges in electricity consumption prediction.

References

[1] Wang, N. and Li, Z., 2023. Short term power load forecasting based on BES-VMD and CNN-Bi-LSTM method with error correction. *Front. Energy Res*, *10*, p.1076529.

[2] Xiao, L., Shao, W., Liang, T. and Wang, C., 2016. A combined model based on multiple seasonal patterns and modified firefly algorithm for electrical load forecasting. *Appl. Energy*, *167*, pp.135-153.

[3] Dong, Y., Zhang, Z. and Hong, W.C., 2018. A hybrid seasonal mechanism with a chaotic cuckoo search algorithm with a support vector regression model for electric load forecasting. *Energies*, *11*(4), p.1009.

[4] Yildiz, B., Bilbao, J.I. and Sproul, A.B., 2017. A review and analysis of regression and machine learning models on commercial building electricity load forecasting. *Renew. Sustain. Energy Rev*, *73*, pp.1104-1122.

[5] Neto, A.H. and Fiorelli, F.A.S., 2008. Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption. *Energy Build*, *40*(12), pp.2169-2176.

[6] Nassif, N., 2012. Regression Models for Estimating Monthly Energy Consumptions in Schools in Hot and Humid Climates. *ASHRAE Trans*, *118*(1).

[7] Ahmad, A.S., Hassan, M.Y., Abdullah, M.P., Rahman, H.A., Hussin, F., Abdullah, H. and Saidur, R., 2014. A review on applications of ANN and SVM for building electrical energy consumption forecasting. *Renew. Sustain. Energy Rev*, *33*, pp.102-109.

[8] Green Building Council Australia, Green building council Australia (Ed.), Introducing Green Star, 2013.

[9] Hong, T. and Wang, P., 2014. Fuzzy interaction regression for short term load forecasting. *Fuzzy Optim. Decis. Mak*, *13*, pp.91-103.

[10] Hong, W.C., 2010. Application of chaotic ant swarm optimization in electric load forecasting. *Energy Policy*, *38*(10), pp.5830-5839.

[11] Rafi, S.H., Deeba, S.R. and Hossain, E., 2021. A short-term load forecasting method using integrated CNN and LSTM network. *IEEE Access*, *9*, pp.32436-32448.

[12] Alhussein, M., Aurangzeb, K. and Haider, S.I., 2020. Hybrid CNN-LSTM model for short-term individual household load forecasting. *IEEE Access*, *8*, pp.180544-180557.

[13] Yang, Y., Wang, Z., Gao, Y., Wu, J., Zhao, S. and Ding, Z., 2022. An effective dimensionality reduction approach for short-term load forecasting. *Electr. Power Syst. Res*, *210*, p.108150.

[14] Rayi, V.K., Mishra, S.P., Naik, J. and Dash, P.K., 2022. Adaptive VMD based optimized deep learning mixed kernel ELM autoencoder for single and multistep wind power forecasting. *Energy*, *244*, p.122585.

[15] Kingma, D.P. and Welling, M., 2013. *Auto-encoding variational bayes* [online].

[16] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. and Chen, H., 2019. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst*, *97*, pp.849-872.

[17] Madrid, A.E., 2021. Short-term electricity load forecasting (Panama case study). *Mendeley Data*, *1*.

[18] Al-Ja'afreh, M.A.A., Mokryani, G. and Amjad, B., 2023. An enhanced CNN-LSTM based multi-stage framework for PV and load short-term forecasting: DSO scenarios. *Energy Rep*, *10*, pp.1387-1408.

[19] Feng, C., Cui, M., Hodge, B.M. and Zhang, J., 2017. A data-driven multi-model methodology with deep feature selection for short-term wind forecasting. *Appl. Energy*, *190*, pp.1245-1257.

[20] Yin, C., Wei, N., Wu, J., Ruan, C., Luo, X. and Zeng, F., 2024. An Empirical Mode Decomposition-Based Hybrid Model for Sub-Hourly Load Forecasting. *Energies*, *17*(2), p.307.

[21] Sorkhi, A.G., Pirgazi, J. & Ghasemi, V. A hybrid feature extraction scheme for efficient malonylation site prediction. Sci Rep 12, 5756 (2022.

[22] Cawley, G.C., Talbot, N.L.C. Efficient approximate leave-one-out cross-validation for kernel logistic regression. Mach. Learn, 71, 243–264 (2008).

[23] Bürkner, PC., Gabry, J. & Vehtari, A. Efficient leave-one-out cross-validation for Bayesian non-factorized normal and Student-t models. Comput. Stat, 36, 1243–1261 (2021).

[24] Wei, N., Yin, C., Yin, L., Tan, J., Liu, J., Wang, S., Qiao, W. and Zeng, F., 2024. Short-term load forecasting based on WM algorithm and transfer learning model. *Appl. Energy*, *353*, p.122087.