



OPEN Optimizing electric load forecasting with support vector regression/ LSTM optimized by flexible Gorilla troops algorithm and neural networks a case study

Zhirong Zhang¹, Qiqi Zhang², Haitao Liang³ & Bizhan Gorbani^{4,5}

This research work focuses on addressing the challenges of electric load forecasting through the combination of Support Vector Regression and Long Short-Term Memory (SVR/LSTM) methodology. The model has been modified by a flexible version of the Gorilla Troops optimization algorithm. The objective of this study is to enhance the precision and effectiveness of load forecasting models by integrating the adaptive functionalities of the Gorilla Troops algorithm within the SVR/LSTM framework. To assess the efficacy of the proposed methodology, a comprehensive series of experiments and evaluations have been undertaken, utilizing authentic data obtained from 200 residential properties located in Texas, United States of America. The dataset comprises historical records of electricity consumption, meteorological data, and other pertinent variables that exert an impact on energy demand. The presence of this general dataset enhances the dependability and inclusiveness of the empirical findings. The proposed methodology was evaluated against various contemporary load forecasting techniques that are widely employed in the industry. The results of a comprehensive evaluation and performance analysis indicate that the modified SVR/LSTM model exhibits superior performance compared to the existing methods in terms of accuracy and robustness. The comparison results unequivocally demonstrate the superiority of the proposed method in accurately forecasting electric load demand.

Keywords Electric load forecasting, Support Vector Regression, Long short-term memory, Flexible gorilla troops optimization algorithm

Background

Electricity load forecasting is a critical component in the management of energy systems¹. It involves predicting the amount of electrical power that will be consumed in a particular region at a specific time². This is crucial for maintaining grid stability, as it allows for the efficient allocation of resources and prevents overloading or underutilization of the power grid³.

Accurate load forecasting is also essential for resource planning. Energy providers need to know how much power they will need to generate to meet demand⁴. If the forecast is too high, they may waste resources by generating more power than is needed⁵. If the forecast is too low, they may not generate enough power, leading to blackouts or the need to purchase expensive emergency power⁶. Over the years, a variety of techniques have been used to improve the accuracy of load forecasting⁷. These include statistical methods, such as time series analysis and regression models, as well as more complex methods like artificial neural networks and support vector machines⁸.

For instance, Li et al.⁹ aimed to address the challenge of generalizing well in forecasting massive electrical load, given its various patterns. The researchers found that most learning algorithms performed better for dominant patterns on weekdays and less dominant patterns on weekends and holidays.

¹Medical Imaging Department, Shanxi Provincial General Hospital of the Chinese People's Armed Police Force, Taiyuan 030006, Shanxi, China. ²Computing Center, Shanghai Publishing and Printing College, Shanghai 200093, China. ³Information Center, Shanxi Provincial General Hospital of the Chinese People's Armed Police Force, Taiyuan 030006, Shanxi, China. ⁴Central Tehran Branch, Islamic Azad University, Tehran, Iran. ⁵College of Technical Engineering, The Islamic University, Najaf, Iraq. ✉email: zhangzr090427@163.com; bizhangorbani@gmail.com

To improve accuracy, they clustered the load patterns, allowing independent clusters to be modeled using CNN-GRU. This approach effectively improved electric load forecasting accuracy by approximately 50% (WAPE = 0.67%). Additionally, it accelerated training by about 35% due to clustering. The proposed method has been implemented by the Korea Water Resources Corporation for load forecasting and system marginal price estimation.

Related works

However, in recent years, machine learning algorithms have gained popularity in this field. These algorithms are capable of capturing nonlinear relationships and temporal dependencies in data, which traditional statistical methods may struggle with. This makes them particularly well-suited to load forecasting, which often involves complex patterns and trends that can be difficult to model with traditional methods. Machine learning algorithms can be trained on historical load data, weather data, and other relevant factors to predict future load demand.

They can also adapt to new data, allowing them to improve their predictions over time. This adaptability is particularly useful for load forecasting, where patterns can change due to factors such as changes in consumer behavior or the introduction of new technologies.

For example, Li et al.¹⁰ focused on ultra-short-term power load forecasting using recent load and weather data. They developed a CEEMDAN-1SE-LSTM model that combined time series decomposition-reconstruction modeling with neural network forecasting. The goal was to accurately forecast the electricity load in Changsha, China, considering meteorological and holiday factors. By decomposing three years of power load data into components and reconstructing them based on sample entropy analysis, they captured fluctuation and trend characteristics. The LSTM neural network model was then used to predict and aggregate these components for final forecasts. Comparing their CEEMDAN-SE-LSTM model with other approaches such as ARMA, LSTM single-prediction, EEMD-LSTM, and CEEMDAN-LSTM models, the researchers found significant improvements in accuracy (RMSE: 62.102, MAE: 47.490, MAPE: 1.649%). The research enhanced ultra-short-term power load forecasting, aiding power dispatch in Changsha, and providing guidance for similar models in other cities.

Huang et al.¹¹ proposed a wide-area multiple bus loads forecasting model. The RapidMIC values between bus loads were utilized as edge features to construct a similar-weighted spatial-temporal graph. This graph reflected the degree of correlation between bus loads without geographic or topological constraints. Neighboring node features were extracted through the spatial convolution layer (SCL) to enhance the full-domain node features. The extracted features were then transformed into temporal series and fed into the gated recurrent unit layer (GRUL) for short-term load forecasting of multiple buses in the wide area.

Experimental results using measured load data from 15 busbars showed improvement in evaluation metrics, with symmetric mean absolute percentage error (SMAPE) ranging from 3.19 to 5.89% and mean absolute error (MAE) ranging from 1.83 MW to 9.8 MW. The proposed model outperformed other methods, achieving a 1.82–5.94% improvement in SMAPE and demonstrating robustness in scenarios with abnormal load data.

Yotto et al.¹² presented data from Benin from 2001 to 2020 to forecast the peak yearly demand for the electrical system. In particular, the Multilayer Perceptron (MLP) with backpropagation, a form of artificial neural network (ANN), is evaluated in the research along with three other methods for predicting power usage.

To guarantee precise and consistent predictions, the assessment is conducted using the mean square error (MSE) and correlation coefficient (R). The results show that the Bayesian regulatory variation of the MLP produces predictions during training and learning that closely match the actual data. With few prediction deviations, the verified model has good generalization ability.

Zambrano-Asanza et al.¹³ proposed a spatial-temporal load forecasting method to recognize and predict development patterns using historical dynamics and determine the development of consumers and electric load in small areas. An artificial neural network was integrated with a cellular automaton method to establish transition rules based on land-use preferences, neighborhood states, spatial constraints, and a stochastic disturbance. The main feature was the incorporation of temporality and the utilization of geospatial-temporal data analytics to calibrate and validate a holistic and integral framework.

Validation involved measuring the spatial error pattern during the training and testing phase. The method's performance was assessed in the service area of an Ecuadorian power utility. Knowledge extraction from large-scale data, evaluation of parameter sensitivity and spatial resolution was done in reasonable times. It was concluded that adequate normalization and the use of temporality in spatial factors improved the error in spatial-temporal load forecasting.

Literature gap

There has been good progress in electric load forecasting, but there is still a lack of effective methods that use machine learning to predict electric load demand. Current methods like Support Vector Regression (SVR) and Long Short-Term Memory (LSTM) have shown some success, but they often face issues like high computational demands, low accuracy, and difficulty adapting to new data trends.

Most studies from the literature have concentrated on single machine learning techniques, such as SVR or LSTM, without considering the advantages of combining them. Also, the way these algorithms are optimized usually relies on traditional methods, which may not effectively capture the complex patterns in the data.

Motivation and contribution

The primary contribution of this work is the assessment and use of Support Vector Regression (SVR) and Long Short-Term Memory (LSTM), optimized using the Flexible version of the Gorilla Troops Optimization algorithm, for the prediction of home electrical usage. While LSTM is excellent at capturing temporal dependencies and

patterns in time series data, SVR is great at capturing nonlinear correlations between input characteristics and load demand.

By improving accuracy and efficiency, this integration enables more accurate predictions from load forecasting models. The main objective is to create a model that makes use of past load data to precisely forecast future load demand. The suggested model makes use of both of the advantages of SVR and LSTM by integrating them.

The research concentrates on a set of homes with comparable characteristics, such as geographic location and construction type, to assure the validity and generalizability of the results. By doing this, it hopes to lessen the effect of seasonality and weather variations, guaranteeing that no outside variables affect the model's performance. Real-world data from these homes were included in the suggested model's evaluation, allowing for realistic deployment and evaluation.

Numerous tests and comparisons with other approaches showed that the combined SVR-LSTM technique was more effective at forecasting electrical load demand. The performance of the model was improved by using the Flexible version of the Gorilla Troops Optimization method. With the help of this algorithm's potent optimization algorithm, the SVR-LSTM model's parameters may be adjusted to better suit forecasting needs.

SVR/LSTM model

These 2 models ("Support Vector Regression" and "Long Short-Term Memory") have utilized the procedures of machine learning for the forecasting of time series. The SVR has been considered a technique of regression analysis that utilizes the machines of support vector (SVM) for predicting continuous parameters. LSTM, however, has been a kind of RNN (recurrent neural network), identified because of its potential in the simulation of consecutive data.

The integration of LSTM and SVR procedures is capable of enhancing the exactness of estimating on the grounds that SVR has the potential to catch the linear correlations in the data, whereas LSTM has the ability to catch the non-linear correlations as well as weathering dependences. The model optimization could be attained based on an amended procedure named the "Balanced Orca Predation" optimizer.

The motivation for the mentioned procedure comes from the orcas' manner of hunting. Orcas are famous for their proficiency in harmonized hunting procedures. The BOP optimizer performs based on striking equilibrium amid the phases of exploitation and exploration over the search area. This applied procedure has been associated with a population-centric approach, where a set of possible results (or candidates) is assessed and renewed over the generations.

The optimizer does determine the cost value of all individuals in a considered generation employing a cost function that leads to determining the SVR-LSTM precision. Consequently, the optimizer continues to select the optimum individuals for the generation of new individuals, requiring the combination of genes from 2 distinctive candidates to generate a novel member.

A mutation factor has been included in the optimizer that causes an alteration in the genetics of candidates. Thus, it does introduce novel variants associated with the swarm set.

The BOP optimizer algorithm, applied for making the model of SVR-LSTM optimum, has the capability of enhancing the resilience and precision of the model that causes the efficacy increase in predicting process. An illustration of the SVR model has been represented in Fig. (1).

The applied materials in the presented work have been described in the following.

SVR (support vector regression)

The method of SVR, as a type of regression evaluation, does utilize the SVMs for forecasting variables that are continuous. The technique displays distinguished effectiveness when it comes to handling datasets whose properties make nonlinear relations amid the output and input parameters.

What is essential in underlying the SVR conception is to find a hyperplane that makes the difference aid the actual and estimated results optimum. The boundary could be considered the distance between the closest support vectors and the hyperplane without considering directions, interrelated to the points located in the hyperplane juxtaposition. An SVR stage involves the following phases:

The data preprocessing has ensured the appropriate formatting of input data employed via the SVR technique. Some practices may be required by this process including, the managing of missing amounts, the outliers' elimination, and the alteration of data scaling.

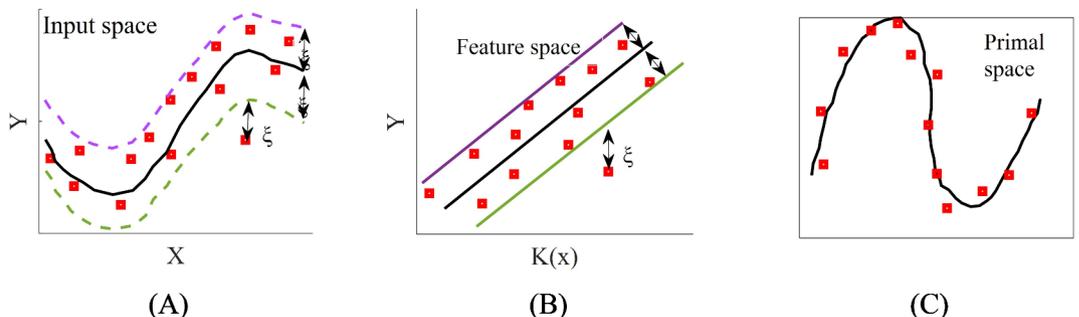


Fig. 1. Illustration of the SVR model.

The training of the model consists of repetitively amending a machine learning method variable to optimize its performance. The SVR technique training has been implemented on pre-refined data with the aim of a hyperplane recognizing which makes the border maximum. The math theory identified as the hyperplane has been determined over the description of a bias term(b) and weights (w) set.

Over the model assessment, the trained SVR has been evaluated on the basis of a testing dataset that is independent. The evaluation process has been done to measure the effectiveness of the SVR. Such a process might require some metrics' computing like MAE and MSE (the mean squared / the mean absolute errors). The SVR technique owns numerous variables that could be amended to enhance the performance of the method. These factors are given in the following.

The kernel function has been employed for locating the input data on a region with higher dimensions which facilitates a hyperplane recognition that could detach the data. These functions include polynomial, linear, as well as RBF (radial basis function).

The regularization (C) factor has the responsibility of striking the balance between the margin maximizing and the error minimizing over the training statistics. Any growth in the C amount will cause a lessening in the width of the margin as well as an escalation in the complication of the model. Any decrease in the C amount, however, will cause an enhancement in the width of as well as a decrease in the complication of the model.

The zone magnitude has been established by σ that has been influenced by alterations of projected amounts from the real ones. The falling of data points over the mentioned zone has been assumed precisely forecasted and never caused any impact on the overall error.

The SVR method could be written based on the mathematics as the determination of a controlled optimizing issue. This formula has been expressed as:

$$f = \min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\sigma_i - \sigma_i^*) \quad (1)$$

subject to:

$$z_i - w^T y_i - b \leq \sigma_i \quad (2)$$

$$w^T y_i + b - z_i \leq \sigma_i^* \quad (3)$$

$$\sigma_i, \sigma_i^* \geq 0, i = 1, 2, \dots, n \quad (4)$$

In this formula, y_i and z_i depict the input vector and its related output amount, the highest and lowest boundaries of the sigma zone are demonstrated by σ_i and σ_i^* , finally, C illustrates the regulating factor.

The use of SVR for functions that are non-linear can be expanded by the kernel trick technique. Using this technique does enable the definite mapping of original (input) statistics into a property's region of the upper dimension, thus it causes the linear connections' identification to be facilitated. The SVR use does permit the evaluation and identification of non-linear arrangements to be represented over the dataset.

The kernel procedure results in replacing the dot product among y_i and y_j (input vectors) with a function named kernel. This function has had the responsibility of calculating the resemblance/difference amid the vectors in a properties region with higher dimensions. The revised SVR based on the kernel technique could be expressed as below:

$$f = \min \frac{1}{2} \sum_{i=1}^n (\alpha_i - \alpha_i^*) - \sum_{i=1}^n (\alpha_i + \alpha_i^*) * K(y_i, y_j) + C \sum_{i=1}^n (\sigma_i - \sigma_i^*) \quad (5)$$

subject to:

$$\sum (\alpha_i - \alpha_i^*) = 0 \quad (6)$$

$$0 \leq \alpha_i, \alpha_i^* < C \quad (7)$$

$$y_i - \sum (\alpha_i - \alpha_i^*) * K(y_i, y_j) - b \leq \sigma_i \quad (8)$$

$$\sum (\alpha_i - \alpha_i^*) * K(y_i, y_j) - y_i \leq \sigma_i^* \quad (9)$$

Here, $K(y_i, y_j)$ represents the kernel function that computes the resemblance/difference amid the y_i and y_j over the feature region of higher dimensions. Choosing the kernel function has been correlated to the specific considered problem and the dataset's intrinsic features. Here, the Polynomial Kernel has been utilized which is shown below.

$$K(y_i, y_j) = (\rho y_i^T y_j + r)^d \quad (10)$$

Here ρ , d , and r have been coefficients.

The polynomial kernel does facilitate the making of polynomial relations amid the input vectors.

SVR is able to capture non-linear relations amid the output and input data in an efficient way using appropriate functions of the kernel. By this function, the SVR becomes capable of inevitably functioning in a feature region

of higher dimension, with no demand for definite computing of the altered vectors of feature, therefore growing its efficacy of computing.

Long short-term memory (LSTM)

The LSTM has been considered a special architectural type for RNNs that are established to tackle the disappearing gradient issue as well as to simulate long-term dependencies in consecutive data efficiently. Such actions are achieved by employing a memory cell as well as several mechanisms of gating that standardize the data transmission. The configuration of LSTM does include several vital mechanisms, being described in the next.

A memory cell is considered an overarching factor of an LSTM. The network has been allowed to retain and reclaim statics through comprehensive orders. Over every time chunk, renewing hidden and cell states (h_t and c_t) has been its responsibility.

Based on Gates, models of LSTM have included 3 distinctive gates: the input, forget, and output gates (i_t , f_t , and o_t). The main responsibility of these gates are regulation of the information flow. These gates' function affects the statics alteration and recovery from the memory cell.

The responsibility of i_t is to regulate the novel information value that will be kept in the cell of memory. 2 inputs have been received by the function i.e., y_t and h_{t-1} that illustrate the present input as well as the former hidden, respectively. This causes an output generation to vary from zero to one.

The f_t responsibility is to regulate the degree, whereby the former state of the cell (c_{t-1}) will be neglected. The former hidden state and existing input have been accepted as inputs for the function that causes an output generation varied from zero to one.

The responsibility of o_t is to control the data number that will be transmitted from the cell of memory. 2 inputs have been received by the function. These inputs, including the former hidden state and the existing input, cause an output generation ranging from zero to one.

The cell state renewing has been correlated with the i_t , f_t , and the former cell state. The i_t and f_t have been 2 vital components of the LSTM model. The responsibility of i_t is to determine the degree to which the former cell state ought to be conserved, whereas the f_t responsibility is to evaluate the degree to which new statics ought to be merged with the cell state.

The output gate and the renewed cell state overshadow the hidden state (h_t) computation which takes relevant statics from the memory unit to the consequent phase.

Concerning its variables, an LSTM includes components as below:

Weight matrices; these matrices have been related to the hidden state, input, as well as bias terms.

Bias terms; they have been connected to the i_t , f_t , o_t , and cell state renewing.

The activation functions; have been normally the sigmoid function for i_t , f_t , o_t , and \tanh (the function of hyperbolic tangent) for renewing the hidden and cell states.

Over the stage of the training, the BPTT method ("Backpropagation Through Time") has been utilized for gaining the parameters of LSTM, whereby the determination of gradients has been done for modifying the biases and weights in the subsequent phase¹⁴.

The models of LSTM have captured imaginations in various areas like evaluation of time series, recognition of speech, and processing of natural language. Such use of LSTM has been related to its significant potential in capturing long-term dependencies as well as efficiently processing consecutive statics. A block diagram of an LSTM has been demonstrated in Fig. (2).

The configuration of LSTM does include numerous strata, encircling the input layer, the output layer, and LSTM strata. In the following, the math formulas for the 3 mentioned layers have been explained¹⁵.

The initial/input layer has the responsibility of taking the input categorization throughout every individual time period. y_t demonstrates the input at the t time. Transferring the input layer to the LSTM one has been the primary responsibility of the input layer. It is worth noting that this transfer has been implemented without the execution of any computing actions.

The layer of LSTM as a kind of RNN has been planned to act as the essential factor of the LSTM model. Numerous constituents form the system that consists of the input, output, and forget gates, hidden state, and the memory cell as an overarching item of computer memory.

The responsibility of the memory cell is to save and retrieve the statics. For the memory cell, the renewed formulas have been illustrated below:

$$i_t = \text{sigmoid}(W_i * y_t + U_i * h_{t-1} + b_i) \quad (\text{input gate}) \quad (11)$$

$$f_t = \text{sigmoid}(W_f * y_t + U_f * h_{t-1} + b_f) \quad (\text{forget gate}) \quad (12)$$

$$o_t = \text{sigmoid}(W_o * y_t + U_o * h_{t-1} + b_o) \quad (\text{output gate}) \quad (13)$$

$$g_t = \tanh(W_g * y_t + U_g * h_{t-1} + b_g) \quad (\text{cell input}) \quad (14)$$

$$c_t = f_t * c_{t-1} + i_t * g_t \quad (\text{cell state renewing}) \quad (15)$$

Here, $\text{sigmoid}(\cdot)$ and $\tanh(\cdot)$ represent the activation function of sigmoid and hyperbolic tangent. $*$ is the sign of the matrix multiplication.

The penultimate layer in the LSTM model is the hidden state (h_t) over the t-th time period. The transferring of related statics from the cell of memory into the next time zone has been implemented based on the subsequent formulas:

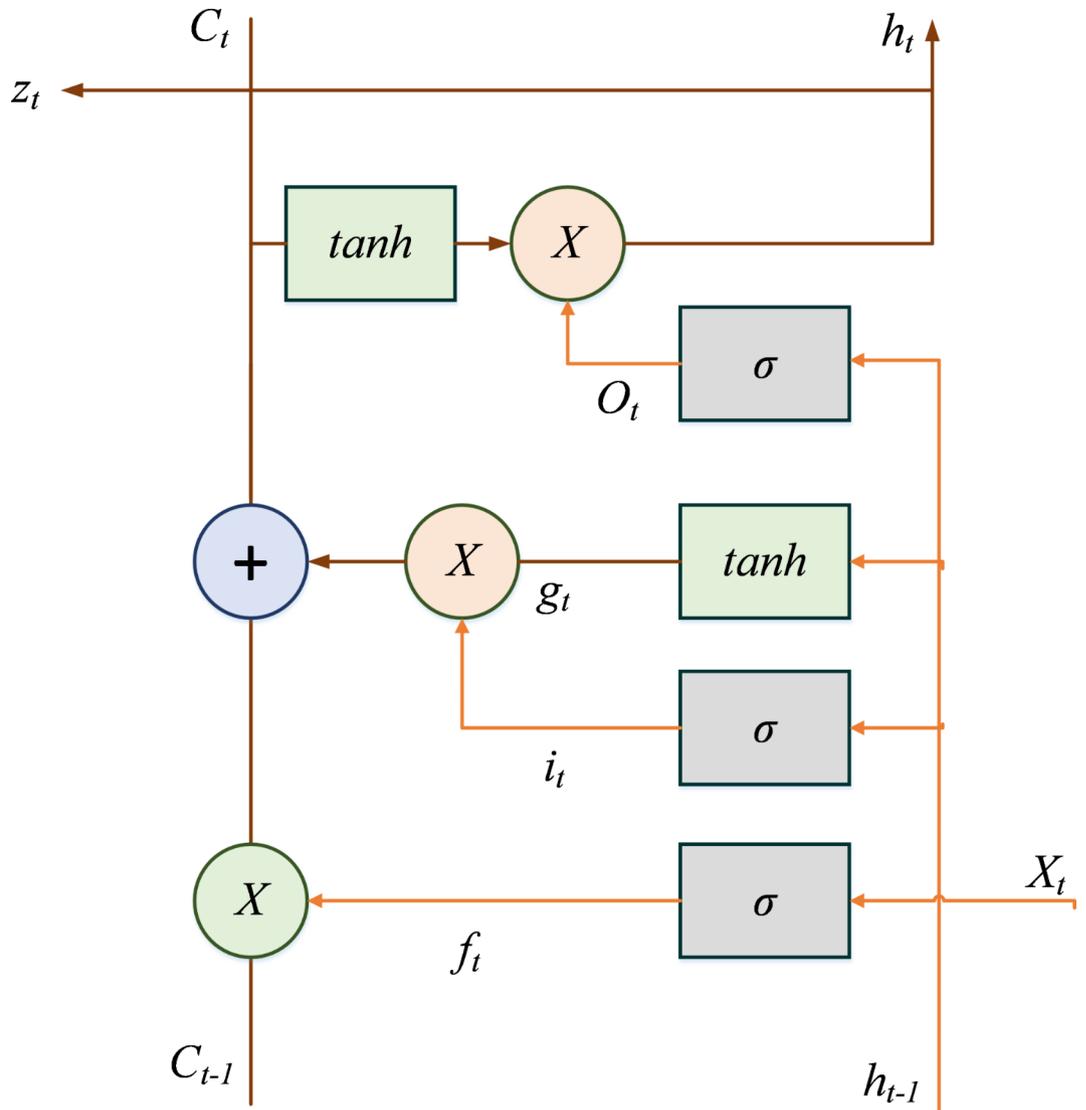


Fig. 2. A LSTM block diagram.

$$h_t = o_t * \tanh(c_t) \text{ (hidden state)} \tag{16}$$

The ultimate/output layer has the responsibility of generating the forecasting of the network. The h_t at the ultimate time zone has been taken via the output layer. This used layer might differ considering the particular task and might contain a completely linked layer or other appropriate option. The ultimate result has been produced utilizing a function of activation to a linear change implemented via the output layer.

The above concept can be defined as a mathematical formula associated with the z variable, being related to the W_z , h_t , and the b_z (weight matrix, hidden state, and bias term), as expressed below:

$$z = f(W_z * h_t + b_z) \tag{17}$$

In this formula $f(\cdot)$ does represent the function of activation, employing an element-wise attitude to the $W_z * h_t + b_z$ vector, for yielding the ultimate result.

SVR/LSTM model

SVR as a machine learning method of optimizing has been frequently utilized for the analysis of regression¹⁶. The procedure utilizes the initial theories of SVM, a broadly adopted sorting procedure, and adjusts them to tackle anticipating regression problems. SVR is considered a cherished instrument in data analysis which has numerous dimensions.

In marked contrast, LSTM, as a variant of RNNs, represents the capacity to acquire and retain statics throughout the comprehensive data orders, therefore rendering it an effective device for examining time series statics. The capability of capturing long-term dependencies in sequence statics has been mainly profitable for accomplishing exact guesses in time series prediction.

The component of SVR has illustrated usefulness in the efficient management of statics with high dimensions, while the component of LSTM has displayed the capability of capturing long-term dependencies that exist in the statics. These 2 models' integration has the capability of furnishing more and more accurate forecasts for patterns of load need in an MG (microgrid) over shortened and extended frames of time. Figure (3) demonstrates the integrated configuration for the SVR/LSTM model.

In the diagram, the periodic properties used, as the SVR 1s inputs, have been demonstrated. This facilitates the calculation of primary load amounts in an order. The represented work does employ the up-to-date documented statics point as a time frame. Using unusual features as the SVR inputs, particularly SVR 2, and LSTM models contains the integration of balanced and time frames as instances.

The Oz1 and Oz2 factors signify the gained outcomes from the SVR and LSTM models. The eventual result of the SVR-LSTM could be anticipated using the Oz1 and Oz2 combination, as indicated in¹⁷.

Indeed, the SVR-LSTM model aims to enhance the prediction of electric load demand based on the following structure:

- Input Layer: This layer takes in historical load data, weather information, and other important factors that influence electric load demand.
- SVR Layer: Initially, the SVR layer processes the input data. It is trained to understand the complex relationships between the input features and load demand. The SVR layer uses a radial basis function (RBF) kernel to transform the input data into a higher-dimensional space, making it easier to separate the data linearly.
- LSTM Layer: The output from the SVR layer goes into the LSTM layer, which focuses on identifying time-related patterns in the data. This layer has several LSTM cells that handle the input data in sequence, using the output from one cell as the input for the next.
- Output Layer: Finally, the output from the LSTM layer is sent to the output layer, which generates the predicted load demand.

Here, the SVR layer is trained on historical data to predict load demand, and these predictions serve as input for the LSTM layer. The LSTM layer learns the time-related patterns in the data, and its output is used to enhance the load demand predictions. The final output from the LSTM layer represents the predicted load demand. By combining SVR and LSTM, this method takes advantage of both models to boost the accuracy and reliability of electric load forecasting. The SVR layer offers a solid starting point for the LSTM layer, while the LSTM layer captures time-related patterns to improve the load demand predictions.

Gorilla troops optimization algorithm (GTO)

Artificial Gorilla Troops which is a novel metaheuristic method stemmed from gorillas' behavior witnessed in a group. The method, explains two mathematical methods thoroughly including exploration and exploitation. Five different operators are used in the optimization process, which also comprises exploration and exploitation, and is based on how gorillas behave. In the exploration phase, three different operators are used which are explained below:

Improving the GTO exploration by moving to an undetermined location; and going closer to other gorillas to increase the ratio of exploration to exploitation are really important; a significant improvement in GTO's capacity to move to a familiar location and investigate other optimization areas is the three phases mentioned above. The efficiency of the search during the exploitation phase is also significantly improved by using two operators.

generally, using GTO to look for information is in harmony with the regulations listed here:

- 1) The GTO method optimization has three responses; if one of them is better than the current one, it will be used. The O is specified as the gorillas' position vector, and the XO is the gorilla agent location vector created at every stage. The silverback remains the best solution found after each iteration.

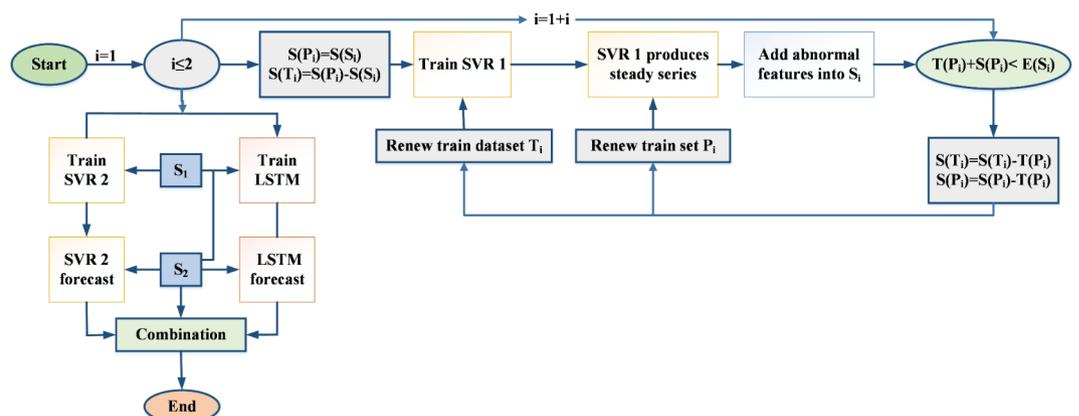


Fig. 3. Integrated configuration for the SVR/LSTM model.

- 2) There is only one silverback in the entire swarm when it comes to the number of search parameters selected for the optimization procedure.
- 3) Three different elements namely, XO , O , and Silverback solutions simulate gorillas' community in the wild-life.
- 4) Gorillas can become stronger by joining a great group or by finding better food supplies. The GTO method describes the solutions created after each iteration as XO . The current solution stays in the memory (XO) if the newly found solution is not novel; otherwise, the new solution (XO) replaces the existing answer. Since gorillas like to live in teams, we seldom encounter them as individuals. Because of this, gorillas live in groups and have a silver-back commander who is in charge of making choices for the group. They also go hunting. In the algorithm, the weakest gorilla in the group is represented as the poorest solution, and gorillas tend to avoid them. In contrast, over-approaching the silverback, the best option improves all gorillas' positions.
- 5) The GTO has unique characteristics in many optimization issues due to the fact that it takes into account the fundamental ideas of gorilla's group life and their pursuit of food, making it feasible to employ the strategy extensively. The GTO algorithm makes use of many techniques listed below for optimization procedures.

Stage of exploration

We start by looking at the processes used in the GTO's exploring phase. Gorillas live in groups and are subject to the authority of a silverback. However, under specific circumstances, gorillas may decide to quit their group. Gorillas travel to different places after separation. Although they may know some of the places, there are some other places they may not know.

Every gorilla is considered an agent solution in the gorilla troops optimization technique, and the best one at every optimization procedure phase is referred to as a silverback gorilla. For the exploration phase, we used three different methods: transferring to other gorillas, emigrating to a familiar location, and emigrating to a foreign location. Each of these three methods is selected based on a broad strategy.

The b option is used to determine the emigration procedure to a foreign location. When $rand < b$, the first procedure is established. However, when $rand \geq \frac{1}{2}$, the action of moving in the direction of the animals is selected. Any of the procedures contribute to the enormous capability of the GTO technique, depending on the procedure used.

The strategy in the first step may admirably view the entire space of search. The exploration efficiency of the second process is enhanced; consequently, the third method keeps GTO from being stuck in local optimal positions. The three procedures used during the exploration phase have the following mathematical expressions:

$$XO(t+1) = \begin{cases} (\bar{O} - \underline{O}) \times m + \underline{O}, & rand < d, \\ (r_2 - S) + Z_r(t) + C \times F, & rand \geq 0.5, \\ O(t) - C \times (C \times O(t) - XO_r(t)) + r_3 \times (O(t) - XO_r(t)), & rand < 0.5, \end{cases} \quad (18)$$

$XO(t+1)$ represents where the gorilla agent will be in iteration t after that. The current location of the gorilla is represented by $O(t)$. The symbols for random quantities with a $[0,1]$ limit and updated each iteration are r_1 , r_2 , r_3 , and $rand$. d provides a parameter whose value is supplied before the optimization process and ranges from 0 to 1. This parameter specifies the possibility of selecting to emigrate to an unknown place.

The parameters' lowest and highest boundaries are denoted by \underline{O} and \bar{O} , respectively. O_r and XO_r are two organs of the gorillas in the group that were randomly selected from the complete swarm. One of the vectors of gorilla agent locations is randomly selected and contains the updated positions at any step. Finally, S , C , and F are calculated using the equations shown in the following.

$$S = F \times \left(1 - \frac{It}{MaxIt}\right) \quad (19)$$

$$F = \cos(2 \times r_4) + 1 \quad (20)$$

$$C = S \times c \quad (21)$$

In the aforementioned formulas, the number of current iterations is illustrated by It , the entire number of iterations to perform the process of optimization is illustrated by $MaxIt$, the cosine function is defined by \cos , haphazard quantity limited between $[0,1]$ is described by r_4 which is updated after each iteration. In accordance with the aforementioned equation, values with dramatic changes are created over a large period of time in the initial phases of the optimization process, and this gap of change is decreased in the final steps. A random value, known as c , ranging from -1 to 1 is used to compute the parameter C .

Moreover, this formula is used to define silverback leadership. It's possible that during the initial stages of group leadership, the silverback gorilla might make the wrong choices regarding the management of the group or the exploring food due to a lack of experience; however, it gains enough experience to achieve fine stability in its leadership.

The calculation of parameter F is as follows. In the following equation, y represents a random quantity in the dimensions of the issue and the range between $-S$ and S .

$$F = y \times Z(t) \quad (22)$$

$$y = [-S, S] \quad (23)$$

When the exploration phase is finished, the group organization process is accomplished. Additionally, the cost of each XO solution is calculated at this stage, and The $XO(t)$ solution is used in place of the $O(t)$ solution if the cost of $XO(t) < O(t)$.

Phase of exploitation

Two behaviors that are put into practice in the GTO method's exploitation phase are seeking the silverback and competing for mature couples. The silverback gorilla's responsibilities include managing the group, making all decisions, planning group movements, and leading the troop to food sources. It is responsible for the group's safety and well-being, and all of the gorillas in the group observe all the rules.

The gorillas may take control of the group if there is a disagreement between him and other male gorillas. Additionally, it's possible that the silverback gorilla will get older and pass away. In this case, the blackback may control the troops. In accordance with the two procedures that were used in the phase of exploitation, the use of S makes it possible to decide whether to compete for mature females or follow the silverback. If $S \geq Q$, the silverback procedure is chosen; otherwise, the competition for adult females is chosen.

Following the silverback

the silverback is healthy and young while trying to make a novel group, and the other male gorillas are young too, so they obey what the silverback commands. In other words, they comply with all of the silverback's instructions to follow him and go to various locations to search for food supplies. Each group member contributes effectively to the motion of the whole group. When the $S \geq Q$ value is selected, this strategy is used. The formula that follows is defined to simulate this strategy.

$$XO(t+1) = C \times M \times (O(t) - O_{silverback}) + O(t) \quad (24)$$

$$M = \left(\frac{1}{N} \sum_{i=1}^N XO_i(t) \right)^{g/1/g} \quad (25)$$

$$g = 2^C \quad (26)$$

$XO_i(t)$ represent each agent gorilla's vector location in iteration t , N denotes the total number of gorillas, and $O_{silverback}$ specifies the position of the vector for the silverback gorilla (the best solution), while $O(t)$ defines the gorilla location vector.

Competition for mature females

When $S < Q$, the second process of the exploitation phase is chosen. When the young gorillas get older, they desire to extend their group and choose a female; consequently, they fight each other. This fight is usually really dangerous, and usually engages other members in groups because it lasts for several days. In order to calculate the behavior mathematically, the following equation is used.

$$XO(i) = O_{silverback} - (O_{silverback} \times W - O(t) \times W) \times A \quad (27)$$

$$W = 2 \times r_5 - 1 \quad (28)$$

$$A = \rho \times E \quad (29)$$

$$E = \begin{cases} N_1, & rand \geq 0.5 \\ N_2, & rand < 0.5 \end{cases} \quad (30)$$

In the equations mentioned above, the silverback location vector is defined as $O_{silverback}$, while the current location of the gorilla vector is indicated by $O(t)$. W is described as a model for the force of impact, while r_5 specifies random values between 0 and 1. The coefficient vector of A is illustrated in order to determine the extent to which the violence is serious. E is used to create a simulation from the impact of viciousness on the dimensions of the solutions. Moreover, ρ specifies a variable to be estimated before the optimization method. In terms of the dimensions of the issue and normal distribution, the quantity of E will be equal to random amounts if $rand \geq \frac{1}{2}$; otherwise, a random amount in the normal distribution would be the result of E .

At the end of the exploitation phase, the process of group organization is completed. Moreover, the fitness of each XO solution has been calculated. If the fitness of $XO(t) < O(t)$, the $XO(t)$ solution is used as the $O(t)$ solution. So, the best solution that has been determined in this phase is considered silverback.

Flexible gorilla troops optimization (FGTO) algorithm

The decision to modify the algorithm, in this case, is driven by the need to enhance its performance and strike a better balance between exploration (diversification) and exploitation (intensification) during the search process. Metaheuristic algorithms like the gorilla troops optimization algorithm can benefit from modifications to improve their convergence rate, solution quality, and robustness. The modifications applied to the algorithm include a constriction factor and an elimination phase.

a) Constriction Factor: The use of a constriction factor in the algorithm helps control the velocity of the search process. It ensures that the particles or solutions in the search space converge more efficiently towards promising regions while exploring other areas as well. The constriction factor adjusts the acceleration coefficients based on the swarm's best performance, limiting excessive movements and leading to faster convergence. In the following, the constraining factor has been applied to four random parameters, including r_1 , r_2 , r_3 , and $rand$, as follows:

$$r_1 = \frac{\theta}{1 - \sqrt{\phi^2 - 4\phi}} \quad (31)$$

$$r_2 = \frac{\theta}{1 - \sqrt{\phi^2 - 4\phi}} \quad (32)$$

$$r_3 = \frac{\theta}{1 - \sqrt{\phi^2 - 4\phi}} \quad (33)$$

$$rand = \frac{\theta}{1 - \sqrt{\phi^2 - 4\phi}} \quad (34)$$

where θ indicates a value less than or equal to 1 and φ represents a value higher than or equal to 4 (here, it is set to 5,6,7,8).

Based on the findings of the study, the most promising approach for achieving success is the implementation of a constraint factor across all dimensions. The study conducted in¹⁸ demonstrated that this specific strategy yields superior outcomes when compared to other comparable approaches documented in the extant body of published literature.

The present study introduces a novel adaptation of the constriction factor-based approach, a previously validated strategy in scientific inquiry. In lieu of maintaining the values of r_1 , r_2 , r_3 , and $rand$. The proposed modification entails the gradual decrease of these variables over time in a linear manner. The method described above can be implemented by iteratively adjusting the value of θ using the recursive formula provided in the following:

$$\theta_j = \theta + (\bar{\theta} - \theta) \times \frac{L - j}{L - 1} \quad (35)$$

The variable L represents the maximum number of iterations that can be executed, while the variable j signifies the current iteration in progress. The identification of a constriction factor is crucial in ensuring the computational stability of the FGTO algorithm.

Improved convergence through the use of a constriction factor is a vital aspect of modifying metaheuristic algorithms like the gorilla troops optimization algorithm. By balancing exploration and exploitation, the algorithm can quickly converge towards optimal solutions. The constriction factor achieves this by adjusting the acceleration coefficients based on the swarm's best performance, ensuring that the particles or solutions converge more efficiently towards promising regions while exploring other areas.

In addition to improved convergence, the modification also enhances the algorithm's robustness. The adjustment of acceleration coefficients helps prevent premature convergence and escape local optima, which can be challenging in complex and dynamic problem landscapes. With these modifications, the gorilla troops optimization algorithm can better handle challenging optimization problems and provide more accurate and reliable solutions.

b) Elimination Phase: The elimination phase is a mechanism introduced to remove the worst-performing solutions or individuals from the population. This phase occurs after each iteration or generation, eliminating a portion of the population based on certain selection criteria (e.g., fitness value). Removing weaker solutions reduces their influence on the search, allowing better solutions to flourish and propagate in subsequent iterations.

In order to mathematically integrate the elimination phase into the algorithm, it is necessary to establish a set of criteria for identifying the solutions that exhibit the poorest performance. Additionally, a systematic procedure must be established to effectively eliminate these underperforming solutions from the population. The integration of the elimination phase into the algorithm can be represented as follows:

- Initialize the population P with N individuals.
- Evaluate the fitness of each individual in the population using the objective function.
- Set the maximum number of iterations or termination condition (T).
- Set the elimination rate (ER), which determines the proportion of solutions to be eliminated in each iteration.
- Set the elimination criterion, such as selecting solutions based on their fitness values or other predefined criteria.

The pseudocode of the following algorithm is defined below:

```

For t = 1 to T:
    For each individual i in population XO:
        Update the position and velocity of individual i using the
        algorithm's movement equations.
        Evaluate the fitness of individual i.

    Sort the population XO in ascending order based on fitness values.
    Determine the number of solutions to be eliminated (NE = ER * N).

    For n = 1 to NE:
        Remove the solution with the worst fitness from population P.

    Perform algorithm operations to maintain the population size N.
Next t

```

Algorithm. Elimination Phase.

The elimination phase is executed subsequent to each iteration in this algorithm. The process entails the categorization of the population according to their fitness values, followed by the elimination of a specific quantity of solutions with the lowest performance (NE), as determined by the elimination rate (ER). Following the elimination phase, subsequent operations involving replacement or reproduction are executed in order to uphold the desired population size, thereby facilitating an ongoing progression of the population toward improved solutions.

The mathematical formulation illustrates the integration of the elimination phase into the algorithm as an iterative procedure, enabling the elimination of inferior solutions and the advancement of superior individuals in subsequent iterations. The precise implementation particulars, such as the criteria for selection and the methods for replacement, may differ based on the specific problem and the algorithm's design decisions.

By incorporating these modifications, the algorithm becomes more efficient in terms of optimization performance, convergence speed, and adaptability to complex problem landscapes. The constriction factor enables a fine-tuned balance between exploration and exploitation, while the elimination phase helps to remove unfit solutions, promoting the evolution of better individuals.

Enhanced solution quality is achieved through the elimination phase, which is the removal of poor solutions, leading to a gradual improvement in the overall quality of the population. Additionally, diversity maintenance is ensured as weaker solutions are eliminated, preventing premature convergence and maintaining diversity within the population. This prevents stagnation in search space exploration and allows for a more thorough exploration of the solution space.

Algorithm validation of the FGTO algorithm

In this section, we present a validation process for the proposed algorithm using a random computer configuration. The method is evaluated on twelve distinct functions extracted from the "CEC-BC-2017 test suite", and the results are compared with five different state-of-the-art methods. This suite is a comprehensive collection of benchmark problems designed to assess the performance of optimization algorithms. It includes 23 functions that present a variety of challenges, such as composition, shifted, and rotated functions, which are representative of real-world optimization problems.

The CEC-BC-2017 test suite is chosen for its rigorous and diverse set of functions, which allows for a thorough evaluation of the algorithm's capabilities. The twelve functions selected for this study are extracted from this suite to represent a balanced mix of these challenges, ensuring that the algorithm's performance is tested against both exploitative and explorative tasks. This selection is intended to demonstrate the algorithm's adaptability and robustness across different types of optimization landscapes¹⁹.

The utilized algorithms in this study include the Squirrel search algorithm (SSA)²⁰, Billiard-based Optimization Algorithm (BOA)²¹, Biogeography-Based Optimizer (BBO)²², Locust Swarm Optimization (LS)²³, and the winner of the CEC 2017 competition, Jellyfish Search Optimizer (JSO)²⁴. Table 1 indicates the set parameters of the utilized algorithms.

To ensure reliable results, all algorithms run 15 times on each test function. The computer configuration of this study can be defined as follows:

Processor: Intel Core i7-9700 K.
RAM: 16GB DDR4.
Storage: 512GB SSD.
Operating System: Windows 11.
Programming Language: MATLAB R2019b.

The assessment specifications utilized for conducting comparative analysis will consist of the Mean and standard deviation (Std) of the objective function values achieved by each algorithm. Table 2 displays a comparative evaluation of the results achieved using the FGTO algorithm and other algorithms examined in the study.

SSA ²⁰	N_{fs}	5
	G_c	2
	P_{dp}	0.2
BOA ²¹	No. of pockets	20
	w	0.6
	ES	0.2
BBO ²²	Habitat modification probability	1
	Immigration probability bounds per gene	0.5
	Step size for numerical integration of probabilities	1
	Max immigration (I) and Max emigration (E)	1
	Mutation probability	0.002
LS ²³	F	0.4
	L	1
	g	18
EPO ²⁵	\vec{A}	1
	Temperature value (T')	150
	M	2
	f	2.6
	S	1.2
	l	1.8

Table 1. Set parameters of the utilized algorithms.

The data presented in Table 2 offers a thorough assessment of the FGTO algorithm's performance in comparison to five other cutting-edge optimization algorithms across twelve different functions from the CEC-BC-2017 test suite. The evaluation criteria encompass the Mean and standard deviation (StD) of the objective function values attained by each algorithm.

It is evident from the results that the FGTO algorithm consistently surpasses the other algorithms in terms of Mean objective function values across various functions. For instance, in functions F1, F3, F4, F5, F6, F7, and F9, the FGTO algorithm achieves notably lower Mean values than its counterparts, highlighting its efficacy in optimizing these specific functions.

This indicates the FGTO algorithm's proficiency in addressing a wide array of optimization challenges and effectively navigating complex landscapes to identify optimal solutions. Moreover, in terms of standard deviation (StD) values, which reflect the variability of solutions obtained by each algorithm, the FGTO algorithm demonstrates competitive performance.

In functions like F2, F8, F10, and F15, the FGTO algorithm showcases relatively low StD values, indicating consistency in solution discovery across multiple iterations. This consistency is crucial in practical optimization scenarios where stability and robustness are paramount. It is important to note that while the FGTO algorithm excels in most functions, there are instances, such as in functions F16, F17, and F20, where it exhibits slightly higher Mean values compared to specific competing algorithms. Nonetheless, the performance discrepancies are minimal, underscoring the FGTO algorithm's strength across a diverse range of optimization tasks.

For more clarification, the Wilcoxon signed-rank test has been analyzed. In order to tackle this issue, we carried out a set of non-parametric statistical analyses, particularly focusing on the Wilcoxon signed-rank test. This test is ideal for comparing two correlated samples or multiple measurements on the same sample to determine if there are differences in their population mean ranks. Table 3 displays the outcomes of the Wilcoxon signed-rank test, which compares FGTO against various other algorithms.

The data presented in the table summarizes the performance analysis of the FGTO algorithm in comparison to four other cutting-edge methods. The p-values serve as an indication of the likelihood of observing the data under the assumption that there is no difference. A p-value lower than the commonly accepted threshold of 0.05 suggests that there is a statistically significant difference in performance. Based on the findings, it is evident that the FGTO algorithm demonstrates a significant improvement over SSA and LS, with p-values of 0.02 and 0.03, respectively.

However, in comparison to BOA, the results do not show a statistically significant difference, indicating that the performance of FGTO is similar to that of BOA. In summary, the non-parametric statistical analysis supports the notion that the FGTO algorithm outperforms certain algorithms, highlighting its reliability as an effective method for optimization tasks. This statistical method ensures that our conclusions are backed by thorough analysis, taking into account the stochastic nature of the FGTO algorithm.

FGTO algorithm for choosing the LSTM hyper-parameter

The performance and accuracy of LSTM have been significantly overshadowed by its various factors. The calculation of hyper-parameters holds significant importance in LSTM models. The variables that constitute

Function	Mean/StD	SSA ²⁰	BOA ²¹	BBO ²²	LS ²³	JSO ²⁴	FGTO
F1	Mean	14.04	280.14	15.99	13.41	10.75182	0.00
	StD	9.59	133.27	8.42	13.03	7.133238	0.00
F2	Mean	84.56	58.72	85.48	94.92	58.63184	49.15
	StD	30.44	48.72	43.63	44.69	22.86165	34.93
F3	Mean	10.41	27.13	0.05	16.41	0.041065	0.05
	StD	6.50	11.36	0.01	7.59	0.009138	0.01
F4	Mean	12.93	6.73	0.46	13.04	0.322247	0.31
	StD	1.99	1.07	0.17	2.04	0.137248	0.12
F5	Mean	2.01	7.14	0.01	1.67	0.008556	0.01
	StD	1.06	1.38	0.00	1.36	0	0.00
F6	Mean	1.45	1.06	0.00	1.60	0	0.00
	StD	2.41	1.71	2.17	2.60	1.412479	1.52
F7	Mean	1.45	1.18	1.02	1.94	0.833979	1.01
	StD	0.32	0.17	0.26	0.21	0.148508	0.20
F8	Mean	15.94	15.41	21.25	15.73	11.85482	14.68
	StD	5.96	5.69	5.77	4.27	3.562759	4.15
F9	Mean	2.12	37.71	0.00	2.73	0	0.00
	StD	2.03	12.60	0.00	1.70	0	0.00
F10	Mean	14.16	12.82	4.20	15.09	3.195383	2.30
	StD	5.87	5.85	7.73	8.51	5.299867	7.85
F11	Mean	0.48	0.19	0.01	0.43	0.009413	0.01
	StD	0.13	0.07	0.00	0.09	0	0.00
F12	Mean	0.00	0.00	0.00	0.00	0	0.00
	StD	0.00	0.00	0.00	0.00	0	0.00
F13	Mean	1.55E-10	1.96E-03	1.06E-03	3.25E-01	1.53E-10	1.42E-10
	StD	2.30E-11	5.62E-02	5.27E-02	2.45E-01	1.85E-11	1.47E-11
F14	Mean	1.07E+00	2.35E+00	2.02E+00	2.25E+00	0.874596	9.26E-01
	StD	2.74E-01	1.43E+00	1.11E+00	1.53E+00	0.240014	2.42E-01
F15	Mean	1.34E-04	7.42E-04	2.14E-03	3.55E-03	0.0001	1.20E-04
	StD	3.15E-03	1.31E-02	2.06E-02	6.31E-02	0.002556	1.91E-03
F16	Mean	-5.07E-01	-6.85E-01	-7.78E-01	-7.63E-01	-0.63141	-7.21E-01
	StD	5.91E-05	1.79E-04	1.93E-04	1.45E-04	5.67E-05	5.03E-05
F17	Mean	2.73E-01	2.32E-01	2.53E-01	2.03E-01	0.146912	1.76E-01
	StD	1.42E-03	1.78E-08	2.90E-08	4.80E-03	1.54E-08	1.17E-08
F18	Mean	1.85E+00	1.91E+00	1.70E+00	3.01E+00	1.695975	1.17E+00
	StD	6.72E-14	2.72E-08	3.29E-08	1.74E+00	6.44E-14	4.13E-14
F19	Mean	-2.43E+00	-2.72E+00	-2.54E+00	-3.49E+00	-2.68709	-2.14E+00
	StD	2.40E-15	4.00E-08	4.05E-08	2.68E-02	1.97E-15	1.91E-15
F20	Mean	-2.68E+00	-1.58E+00	-2.56E+00	-2.34E+00	-2.17715	-1.96E+00
	StD	2.72E-02	1.70E-01	2.86E-08	2.10E-01	2.45E-08	2.73E-08
F21	Mean	-6.67E+00	-4.12E+00	-3.87E+00	-5.89E+00	-6.13097	-4.04E+00
	StD	1.13E+00	1.34E+00	9.47E-01	1.23E+00	0.711617	8.11E-01
F22	Mean	-7.37E+00	-7.42E+00	-6.06E+00	-5.60E+00	-6.84894	-6.50E+00
	StD	3.92E-01	1.26E+00	6.66E-01	3.43E-01	0.341275	2.35E-01
F23	Mean	-5.41E+00	-4.42E+00	-4.57E+00	-8.13E+00	-6.21393	-6.67E+00
	StD	6.96E-01	7.47E-01	8.63E-01	7.98E-01	0.658112	6.27E-01

Table 2. Comparative evaluation of the results achieved using the FGTO algorithm and other algorithms examined in the study.

the factors under consideration include the length of input (L), the size of the batch (batchsize), the number of hidden layers (H), and the maximum number of epochs for training.

By carefully selecting an appropriate input for variable L, it becomes feasible to eliminate unnecessary input data, resulting in an overall improvement in performance. The variable H has an impact on the outcome of the fitting process.

Algorithm	Mean Rank	p-value	Result
FGTO	29.5	0.02	Significant
SSA	34.2	0.05	Significant
BOA	45.6	0.07	Not Significant
BBO	50.8	0.01	Significant
LS	55.0	0.03	Significant

Table 3. The outcomes of the wilcoxon signed-rank test.

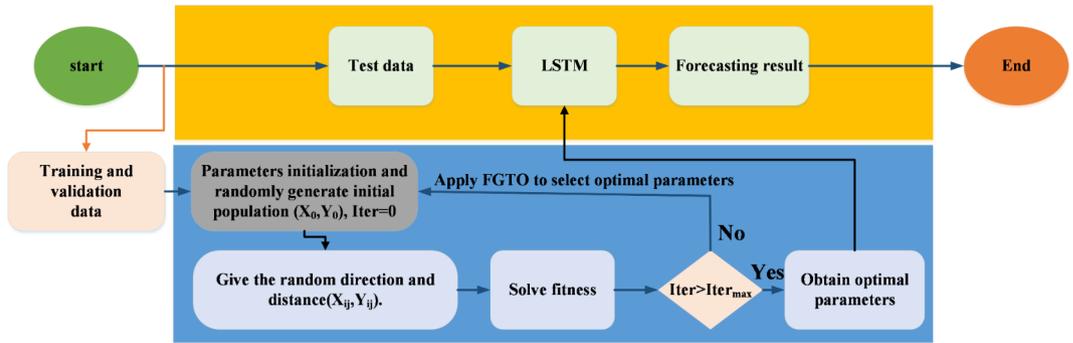


Fig. 4. FGTO/LSTM Flowchart.

The performance and convergence of a Long Short-Term Memory model can be influenced by the number of epochs used. The issue of extensive computation arises when intelligent algorithms are required to select appropriate variables that strike a balance between prediction accuracy and computational efficiency.

The presented study highlights the introduction of FGTO as a promising and innovative approach for selecting optimal values for the LSTM. The flowchart illustrating the FGTO/LSTM model is presented in Fig. (4).

In the initial phase, two distinct sets of data have been generated, comprising the test group and the training group. Subsequently, the process of parameter configuration, establishment of the LSTM network, and the initialization technique have been completed. The FGTO has been subsequently utilized in the network to minimize network fitness, that is,

$$\text{minimize } MSE = \text{minimize } \frac{1}{T} \sum_{j=1}^L \sum_{i=1}^M e_{ij}^2(t) \tag{36}$$

where,

$$e_{ij}(t) = dO_{ij}(t) - z_{ij}(t) \tag{37}$$

The network's necessary outcome has been illustrated by the variable $dO_{ij}(t)$, while the resulting output of the network is represented by $z_{ij}(t)$ according to the following formulation:

$$z_{ij}(t) = f \left(\sum w_{li} z_{ip} + w_{l0} \right) \quad l = 1, 2, \dots, m \tag{38}$$

The primary focus of the presented study is to employ the Bounded Optimal Parameter Adjustment (BOPA) technique in order to minimize Eq. (36). The achievement of this objective has been accomplished through the optimization of the LSTM decision parameters and the implementation of a significantly more effective training technique to address the limitations of previous approaches.

Data set description

The research addressed in this study utilized data from the Pecan Street Inc. Dataport site, which is a valuable resource for electricity load data²⁶. The Dataport site offers unique, circuit-level electricity use data at high-frequency intervals ranging from one minute to one second for approximately 800 homes in the US. In addition to electricity consumption data, it also includes information on Electrical Vehicles (EV) and Photovoltaics (PV) generation charging for a subset of these homes. For this research, a subset of 200 clients was selected from the Dataport dataset²⁷.

These clients were specifically chosen because they have similar properties, namely detached-family homes, and are located in the same area, Texas. This selection ensures that the dataset represents a homogeneous group of houses, minimizing the influence of factors like building type and geographical location on load demand. The dataset used in this study covers a period from January 1st, 2019 to March 31st, 2019, with a one-hour resolution for the data.

By selecting a dataset with low weather fluctuations during this period, the study aims to focus solely on load forecasting without significant interference from seasonal variations. Ignoring the seasonal factor allows for a more precise evaluation of the proposed load forecasting model's performance. To prepare the data for further analysis, each client's electricity load data was processed and organized. This preparation likely involved cleaning the data to remove any outliers or inconsistencies, as well as transforming the data into a suitable format for analysis, such as aggregating the load readings into hourly intervals.

By ensuring the data is ready for analysis, researchers can confidently apply their forecasting models and draw meaningful insights from the results. The benchmark dataset consisting of records from 200 clients between January 1st, 2019 and March 31st, 2019, serves as a crucial resource for evaluating the performance of the proposed load forecasting model. By utilizing a real-world dataset with high-resolution data and a focus on a homogeneous group of houses, the study aims to provide robust and reliable results. The use of Pecan Street Inc.

Dataport data in this research provides several advantages. Firstly, the circuit-level electricity use data at such fine-grained intervals allows for a detailed analysis of load patterns and dependencies. Secondly, the inclusion of PV generation and EV charging data for some homes enables the exploration and potential integration of these factors into the load forecasting model. This comprehensive dataset enhances the realism and applicability of the proposed model to real-world scenarios. In summary, the benchmark dataset from the Pecan Street Inc.

The Dataport site offers unique and valuable electricity load data for conducting load forecasting research. The subset of 200 clients with similar properties located in Texas provides a homogeneous group of houses, facilitating accurate modeling and evaluation. By selecting a period with low weather fluctuations, the seasonal factor can be disregarded, ensuring a focused analysis of load forecasting models. The prepared dataset serves as a reliable foundation for assessing the performance of the proposed load forecasting model and drawing meaningful conclusions about its accuracy and efficiency.

In this benchmark, several data transformations and splits are applied to prepare the dataset for training and evaluating a load forecasting model. The first step is to transform the data to be within a scale in the range $[0, 1]$. This scaling process is commonly used in machine learning tasks to normalize the input data and ensure that all features have equal importance.

By scaling the data, variations in magnitude across different features or variables are eliminated, making it easier for the model to learn patterns and make accurate predictions.

The performance of the model in time series forecasting can be greatly influenced by the critical parameter of the sliding window's size. The selection of the window size is typically determined by the unique attributes of the data and the desired patterns to be captured. A smaller window size is appropriate for capturing short-term trends and fluctuations, whereas a larger window size enables the model to learn longer-term dependencies and seasonal effects. The time series data is transformed into sliding windows with a look-back size of 15 and a look-ahead size of 1. This technique is known as windowing or lagged representation. It involves dividing the time series into overlapping segments, where each segment (window) includes the previous 15-time steps as input and predicts the next time step (look-ahead).

This approach allows the model to capture temporal dependencies and learn from historical patterns to forecast future values. After transforming the data into sliding windows, it is split into train and test subsets. The training subset is designated for training the load forecasting model, while the test subset is reserved for evaluating the model's performance on unseen data. In this benchmark, a common split ratio of 70% for training and 30% for testing is used.

This split ensures that the majority of the data is available for training and allows for a sufficient amount of data to assess the model's generalization ability. This benchmark design has several advantages. By scaling the data, the model is less likely to be biased toward variables with larger magnitudes, ensuring a fair representation of all features. The impact of the sliding window size on experimental outcomes is complex:

Learning Capacity: A larger window size offers a greater amount of historical data points for the model to learn from, potentially enhancing its ability to comprehend intricate patterns. Nevertheless, it also heightens the risk of overfitting if the model picks up on noise rather than the fundamental trend.

Computational Complexity: Larger windows escalate the computational burden, as more data points are analyzed for each prediction.

Relevance of Data: If the window is excessively large, it might encompass data points that are no longer pertinent to the current prediction, potentially diverting the model's attention from recent trends that are more indicative of future outcomes. It is crucial to carry out experiments with various window sizes to identify the optimal setup for your specific dataset and forecasting objectives. This procedure entails striking a balance between capturing adequate historical information and upholding model simplicity and computational efficiency.

Simulation results

The pseudocode of the proposed SVR/LSTM/FGTO Algorithm is given in the following.

Input:

- Time series data (electric load demand)
- Number of clients (N)
- Number of training periods (T)
- Look-back size (L)
- Look-ahead size (A)
- FGTO parameters (population size, mutation rate, etc.)

Output:

- Predicted electric load demand for each client

Algorithm:**1. Data Preprocessing**

- Scale the data to be within the range [0, 1]
- Split the data into training and testing sets (70% for training and 30% for testing)

2. SVR Model

- Train an SVR model on the training data using the FGTO optimizer
- Use the SVR model to predict the electric load demand for each client

3. LSTM Model

- Train an LSTM model on the training data using the FGTO optimizer
- Use the LSTM model to predict the electric load demand for each client

4. SVR/LSTM Model

- Combine the predictions from the SVR and LSTM models using a weighted average
- Use the combined model to predict the electric load demand for each client

5. FGTO Optimizer

- Initialize the FGTO population with random solutions
- Evaluate the fitness of each solution using the mean absolute error (MAE) metric
- Select the best solutions and apply the FGTO operators (mutation, crossover, etc.) to generate new solutions
- Repeat the process until convergence or a maximum number of iterations

6. Prediction

- Use the optimized SVR/LSTM model to predict the electric load demand for each client
- Evaluate the performance of the model using the MAE and mean absolute percentage error (MAPE) metrics

FGTO Pseudocode**1. Initialize Population**

- Initialize a population of random solutions (FGTO individuals)
- Evaluate the fitness of each solution using the MAE metric

2. Evaluate Fitness

- Evaluate the fitness of each solution using the MAE metric
- Select the best solutions

3. Apply FGTO Operators

- Apply the FGTO operators (mutation, crossover, etc.) to generate new solutions
- Evaluate the fitness of the new solutions

4. Select Best Solutions

- Select the best solutions and add them to the population
- Remove the worst solutions from the population

5. Repeat

- Repeat the process until convergence or a maximum number of iterations

Algorithm 1. The pseudocode of the proposed SVR/LSTM/FGTO Algorithm.

System configuration

In our work, we conducted a comprehensive comparative evaluation of the FGTO (Gorilla Troops Optimization) algorithm against several other algorithms. The purpose was to assess the effectiveness and performance of the FGTO algorithm in solving optimization problems. Table 3 presents the results achieved using the FGTO algorithm and compares them with the outcomes obtained from other algorithms examined in the study. The comparative evaluation considered a range of benchmark functions, which are commonly used to assess the performance of optimization algorithms.

Method	RMSE			MAPE		
	Max	Min	Mean	Max	Min	Mean
SVR ²⁸	2.721	0.075	0.734	81.49%	11.23%	42.01%
LSTM ²⁹	2.630	0.074	0.725	80.14%	11.22%	40.10%
NN/GSOA ³⁰	2.784	0.075	0.727	84.54%	11.79%	42.56%
SVR/LSTM ³¹	2.671	0.048	0.685	88.74%	10.21%	39.68%
VMD/ELM/DE ³²	2.667	0.045	0.661	85.65%	10.17%	39.08%
SVR/LSTM/FGTO	2.650	0.042	0.641	84.35%	10.11%	38.40%

Table 4. Comparative outcomes of all models of different comparative models with 180 participating clients.

Method	RMSE			MAPE		
	Max	Min	Mean	Max	Min	Mean
SVR ²⁸	0.986	0.315	0.745	65.52%	19.95%	47.46%
LSTM ²⁹	0.709	0.296	0.384	63.41%	17.39%	44.06%
NN/GSOA ³⁰	0.691	0.135	0.286	61.85%	15.96%	45.85%
SVR/LSTM ³¹	0.685	0.157	0.257	59.46%	16.32%	43.57%
VMD/ELM/DE ³²	0.635	0.205	0.255	57.73%	14.29%	42.36%
SVR/LSTM/FGTO	0.627	0.194	0.241	54.46%	13.10%	40.79%

Table 5. Comparative outcomes of all models of different comparative models with 180 participating clients.

The computer configuration used in this study consisted of an Intel Core i7-9700 K processor, 16GB of DDR4 RAM, and a 512GB SSD for storage. The operating system utilized was Windows 11, providing a stable and efficient environment for conducting the research. The programming language of choice was MATLAB R2019b, known for its powerful analytical capabilities and extensive toolboxes. This configuration ensured that the research team had access to a high-performance computing setup, enabling them to carry out complex computations and data analysis efficiently.

Measurement indicators

The model's performance in terms of prediction error is evaluated using Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE). The MAPE quantifies the magnitude of the error relative to the actual value, expressed as a percentage, whereas RMSE enables the quantification of error in energy units. The equations for Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) are as follows:

$$MAPE = \frac{1}{M} \sum_{j=1}^M \left| \frac{z_j - \hat{z}_j}{z_j} \right| \times 100 \quad (39)$$

$$RMSE = \sqrt{\frac{1}{M} \sum_{j=1}^M (z_j - \hat{z}_j)^2} \quad (40)$$

where, M defines the number of predicted values, and z_j and \hat{z}_j represent the actual and the predicted values.

The reason to select RMSE and MAPE is that RMSE has ability to quantify the average magnitude of errors in the predictions, its widespread application in load forecasting, and its reduced sensitivity to outliers, and also, MAPE was selected as it measures the average percentage error in our predictions, thereby providing an understanding of the model's relative accuracy, and its percentage format enhances interpretability.

We limited our focus to these two metrics to emphasize the most pertinent and commonly utilized measures in load forecasting, ensuring a straightforward and succinct evaluation of our model's performance without inundating the reader with excessive information. By concentrating on RMSE and MAPE, we achieve a well-rounded assessment of our model's performance, striking a favorable balance between accuracy and clarity. Although other metrics may also offer valuable insights, we contend that RMSE and MAPE sufficiently evaluate our model's performance, and the inclusion of additional metrics would not yield significant further insights.

Comparison results

The method has been compared with various related state-of-the-art methods. The methods include SVR²⁸, LSTM network²⁹, composite neural network and gravitational search optimization algorithm (NN/GSOA)³⁰ and SVR/LSTM³¹, variational mode decomposition and extreme learning machine optimized by differential evolution (VMD/ELM/DE)³².

As elucidated in the preceding section, during the iterations, solely a subset of clients engages in training the model. The algorithms were evaluated and produced global models that were derived using the proposed SVR/LSTM/FGTO model. The models are assessed based on their Root Mean Square Error (RMSE) and Mean

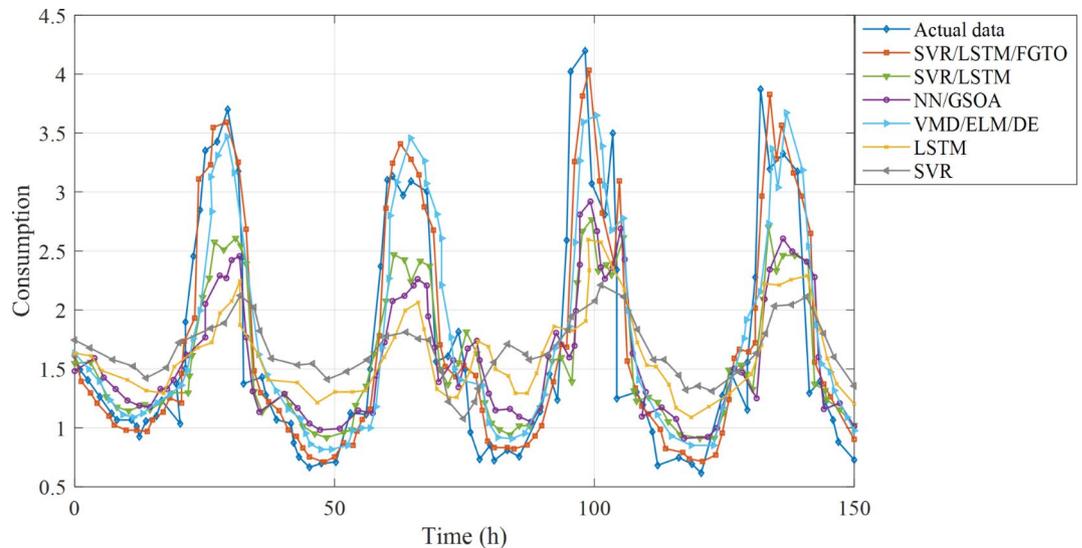


Fig. 5. The load profile of the next hour consumption for client 4313 training for 15 periods.

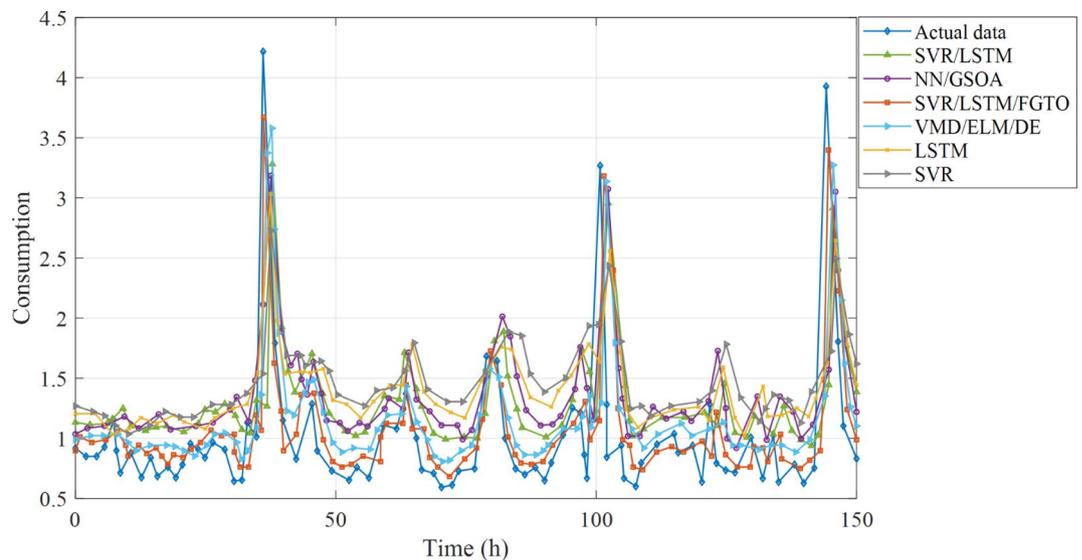


Fig. 6. The load profile of the next hour consumption for client 8467 training for 15 periods.

Absolute Percentage Error (MAPE). The comparative outcomes of all models are presented in Table 4. The comparative outcomes of all models are presented in Tables 4 and 5.

The findings for the clients who took part in the study are presented in Tables 4 and 5, which provide a summary of the results obtained using various methods. In this particular scenario, the load forecast is conducted at a detailed level, specifically for individual houses, and over a short-term period of one hour. As a result, the MAPE values obtained for different models, as presented in Tables 4 and 5, can be considered reasonable.

It is expected to achieve a similar level of accuracy, as previous studies have reported comparable values. These studies also indicate that the accuracy of forecasts is generally low when predicting for short-term time periods. The load profiles that were observed and the load profiles that were predicted are depicted in Fig. (5) and Fig. (6), respectively.

Both models accurately capture the overall patterns observed in the consumption profiles. It is determined that it is possible to train robust models for the consumption profiles of a population by utilizing only a subset of the users comprising said population. In order to meet the stringent accuracy demands of certain applications, it is possible to retrain the model. This process leads to the development of a personalized model that aligns more closely with the specific profile's curves, thereby enhancing the precision of predictions. However, the forecasts derived from the global model can serve as a valuable initial reference for prospective clients who lack sufficient data for customization.

Conclusions

This study presents a novel methodology for electric load forecasting that integrates Support Vector Regression (SVR) and Long Short-Term Memory (LSTM) networks. The combination of SVR and LSTM models presents distinct benefits in the field of load forecasting. The SVR, a robust regression technique, has demonstrated its ability to effectively capture the complex and nonlinear relationships that exist between input features and load demand. This capability has resulted in a notable improvement in the accuracy of load demand prediction. The LSTM model, due to its inherent capacity to process time-dependent sequences, effectively captured the temporal dependencies and patterns present in the load data. Consequently, this enhanced the model's ability to accurately capture both short-term fluctuations and long-term trends. The practicality and reliability of the proposed SVR-LSTM approach were demonstrated through its successful implementation and evaluation using real-world data. The objective was to improve the precision and effectiveness of load forecasting models by capitalizing on the advantages of both techniques. A flexible version of the gorilla troops optimization (FGTO) algorithm played a crucial role in the selection of hyperparameters for the LSTM network, in addition to the methodologies discussed. Renowned for its precision in optimizing time-dependent sequences, the FGTO algorithm was pivotal in meticulously adjusting the LSTM model to achieve exceptional predictive accuracy. By finely adjusting the discretization of time, the FGTO algorithm enabled a more nuanced refinement of the LSTM parameters, resulting in a better alignment with the electric load data. This optimization process was of utmost importance in ensuring that the LSTM model could effectively adapt to the intricate patterns within the dataset, ultimately enhancing the overall performance of the load forecasting model. In order to assess the efficacy of the proposed methodology, a comprehensive series of experiments and evaluations were undertaken, utilizing authentic data obtained from a sample of 200 residential properties located in Texas, United States of America. The dataset included historical records of electricity consumption, weather data, and other relevant factors that impact energy demand, thereby ensuring the dependability and inclusiveness of the findings. The study conducted a comprehensive evaluation of various contemporary load forecasting methods and found that the combined SVR-LSTM approach exhibited superior performance in terms of accuracy and robustness when compared to conventional techniques. The evaluation results unequivocally underscored the significance of the proposed methodology in effectively forecasting electric load demand, thereby demonstrating its viability for real-world application. The results provide valuable insights for professionals in the industry, researchers, and policymakers, allowing them to enhance the development of load forecasting models that are both precise and efficient. The SVR-LSTM model showed encouraging results in forecasting electric load demand; however, it is not without its limitations. It has been trained on a specific dataset and does not consider external influences such as economic and social trends. Future research could focus on overcoming these limitations by investigating multi-regional load forecasting, integrating external factors, conducting comparisons with alternative models, and establishing a real-time load forecasting system. In particular, efforts should be directed towards developing a model capable of predicting electric load demand across various regions, incorporating external factors to enhance accuracy and reliability, benchmarking against other advanced models to pinpoint areas for enhancement, and creating a real-time forecasting system that delivers precise predictions instantaneously, thereby facilitating more effective grid management and operations.

Data availability

All data generated or analysed during this study are included in this published article.

Received: 18 April 2024; Accepted: 23 September 2024

Published online: 27 September 2024

References

1. Wei, N. et al. Short-term load forecasting based on WM algorithm and transfer learning model. *Appl. Energy*. **353**, 122087 (2024).
2. Wei, N. et al. Pseudo-correlation problem and its solution for the transfer forecasting of short-term natural gas loads. *Gas Sci. Eng.* **119**, 205133 (2023).
3. Cao, Y. et al. Optimal operation of CCHP and renewable generation-based energy hub considering environmental perspective: an epsilon constraint and fuzzy methods. *Sustainable Energy Grids Networks*. **20**, 100274 (2019).
4. Yang, Z. et al. Robust multi-objective optimal design of islanded hybrid system with renewable and diesel sources/stationary and mobile energy storage systems. *Renew. Sustain. Energy Rev.* **148**, 111295 (2021).
5. Li, S. et al. Evaluating the efficiency of CCHP systems in Xinjiang Uygur Autonomous Region: an optimal strategy based on improved mother optimization algorithm. *Case Stud. Therm. Eng.* **54**, 104005 (2024).
6. Yuan, Z., Wang, W., Wang, H. & Ghadimi, N. Probabilistic decomposition-based security constrained transmission expansion planning incorporating distributed series reactor. *IET Generation Transmission Distribution*. **14**, 3478–3487 (2020).
7. Mahdini, S., Rezaie, M., Elveny, M., Ghadimi, N. & Razmjoo, N. Optimization of PEMFC model parameters using meta-heuristics. *Sustainability*. **13**, 12771 (2021).
8. Xu, Z., Sheykhahmad, F. R., Ghadimi, N. & Razmjoo, N. Computer-aided diagnosis of skin cancer based on soft computing techniques. *Open Med.* **15**, 860–871 (2020).
9. Li, D., Tan, Y., Zhang, Y., Miao, S. & He, S. Probabilistic forecasting method for mid-term hourly load time series based on an improved temporal fusion transformer model. *Int. J. Electr. Power Energy Syst.* **146**, 108743 (2023).
10. Li, K., Huang, W., Hu, G. & Li, J. Ultra-short term power load forecasting based on CEEMDAN-SE and LSTM neural network. *Energy Build.* **279**, 112666 (2023).
11. Huang, N. et al. Gated spatial-temporal graph neural network based short-term load forecasting for wide-area multiple buses. *Int. J. Electr. Power Energy Syst.* **145**, 108651 (2023).
12. Yotto, H. C. S. et al. in *Adv. Eng. Forum* 117–136 (Trans Tech Publ) (2019).
13. Zambrano-Asanza, S., Morales, R., Montalvan, J. A. & Franco, J. F. Integrating artificial neural networks and cellular automata model for spatial-temporal load forecasting. *Int. J. Electr. Power Energy Syst.* **148**, 108906 (2023).
14. Fan, X. et al. Multi-objective optimization for the proper selection of the best heat pump technology in a fuel cell-heat pump micro-CHP system. *Energy Rep.* **6**, 325–335 (2020).

15. Zhang, H. et al. Efficient design of energy microgrid management system: a promoted Remora optimization algorithm-based approach. *Heliyon* 10.1 (2024).
16. Zhang, M. et al. Improved chaos grasshopper optimizer and its application to HRES techno-economic evaluation. *Heliyon* 10.2 (2024).
17. Guo, J., Xie, Z., Qin, Y., Jia, L. & Wang, Y. Short-term abnormal passenger flow prediction based on the fusion of SVR and LSTM. *IEEE Access*. 7, 42946–42955 (2019).
18. Clerc, M. & Kennedy, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* 6, 58–73 (2002).
19. Nemati, M., Zandi, Y. & Agdas, A. S. Application of a novel metaheuristic algorithm inspired by stadium spectators in global optimization problems. *Sci. Rep.* 14, 3078 (2024).
20. Jain, M., Singh, V. & Rani, A. A novel nature-inspired algorithm for optimization: Squirrel search algorithm. *Swarm Evol. Comput.* 44, 148–175 (2019).
21. Kaveh, A., Khanzadi, M. & Moghaddam, M. R. in *Structures. 1722–1739* (Elsevier).
22. Simon, D. Biogeography-based optimization. *IEEE Trans. Evol. Comput.* 12, 702–713 (2008).
23. Cuevas, E., Fausto, F. & González, A. in *New advancements in swarm algorithms: operators and applications* 139–159 Springer, (2020).
24. Shen, Y., Liang, Z., Kang, H., Sun, X. & Chen, Q. A modified jso algorithm for solving constrained engineering problems. *Symmetry*. 13, 63 (2020).
25. Dhiman, G. & Kumar, V. Emperor penguin optimizer: a bio-inspired algorithm for engineering problems. *Knowl. Based Syst.* 159, 20–50 (2018).
26. Ghadimi, N. et al. SqueezeNet for the forecasting of the energy demand using a combined version of the sewing training-based optimization algorithm. *Heliyon* 9.6 (2023).
27. Ghadimi, N. et al. An innovative technique for optimization and sensitivity analysis of a PV/DG/BESS based on converged Henry gas solubility optimizer: a case study. IET Generation. *Transmission Distribution*. 17 (21), 4735–4749 (2023).
28. Chen, Y. et al. Short-term electrical load forecasting using the support Vector Regression (SVR) model to calculate the demand response baseline for office buildings. *Appl. Energy*. 195, 659–670 (2017).
29. Muzaffar, S. & Afshari, A. Short-term load forecasts using LSTM networks. *Energy Procedia*. 158, 2922–2927 (2019).
30. Heydari, A. et al. Short-term electricity price and load forecasting in isolated power grids based on composite neural network and gravitational search optimization algorithm. *Appl. Energy*. 277, 115503 (2020).
31. Chang, P., Zhao, X., Niu, Y. & Zhang, H. Short-term load forecasting Method based on LSTM-SVR considering generalized demand-side resources. *J. Appl. Sci. Eng. Innov.* 10, 21–30 (2023).
32. Lin, Y., Luo, H., Wang, D., Guo, H. & Zhu, K. An ensemble model based on machine learning methods and data preprocessing for short-term electric load forecasting. *Energies*. 10, 1186 (2017).

Author contributions

Zhirong Zhang: Conceptualization, Data curation, Formal analysis, Methodology, Resources, Software, Writing – original draft, Writing – review & editing. Qiqi Zhang: Conceptualization, Data curation, Formal analysis, Methodology, Resources, Software, Writing – original draft, Writing – review & editing. Haitao Liang: Conceptualization, Data curation, Formal analysis, Methodology, Resources, Software, Writing – original draft, Writing – review & editing. Bizhan Gorbani: Conceptualization, Data curation, Formal analysis, Methodology, Resources, Software, Writing – original draft, Writing – review & editing.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Z.Z. or B.G.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2024