IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 November 2025, accepted 18 November 2025, date of publication 27 November 2025,
date of current version 9 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3637909

RESEARCH ARTICLE

Industrial Implementation of Machine Learning Forecasting Algorithms for Large-Scale Solar Heat for Industrial Process (SHIP) Systems

ABDURRAHMAN M. MANU[1], AHMAD JAZLAN[1], MUHAMMAD NAVEED AKHTER[2], MUHAMMAD DANIAL[1], AZHAR MOHD IBRAHIM[1], (Member, IEEE), HASAN F. M. ZAKI[1], (Member, IEEE), AND VICTOR SREERAM[3], (Life Senior Member, IEEE)

[1]Department of Mechatronics Engineering, Kulliyyah of Engineering, International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia
[2]Department of Electrical Engineering, Rachna College of Engineering and Technology (A Constituent College of University of Engineering and Technology, Lahore), Gujranwala 52250, Pakistan
[3]School of Electrical and Electronics Engineering, The University of Western Australia, Perth, WA 6009, Australia

Corresponding author: Ahmad Jazlan (ahmadjazlan@iium.edu.my)

ABSTRACT Natural gas boilers are commonly used to fulfill the heated water and steam requirements needed for running specific processes inside oleochemical factories. The vast usage of natural gas boilers results in significant greenhouse gas emissions. Therefore, solar heat for industrial process (SHIP) systems are being introduced around the world as a clean alternative source of energy for supplying heated water and steam. Optimizing these SHIP industrial processes and consequent energy management requires accurate thermal energy forecasting. In this study, the thermal energy forecasting of a large-scale SHIP system was performed. This SHIP system was installed at an oleochemical factory in Johor Bahru, Malaysia. A deep learning method (LSTM) was proposed for an hour-ahead forecasting of the SHIP system output on a quarterly basis for a one-year duration. The performance of LSTM was compared with other machine learning methods, namely Support Vector Regression, Random Forest, Decision Tree, and XGBoost. Finally, three hybrid models (PSO-LSTM, GA-LSTM, SSA-LSTM) were developed using different parameter optimization techniques to tune the hyperparameters of the developed deep learning method (LSTM) in order to enhance its forecasting accuracy. The models were evaluated based on their predictive accuracy using metrics such as R^2, MAE, correlation coefficient (r), and MSE. While all models demonstrated comparable accuracy in terms of R^2, MAE and r, the PSO-LSTM model notably excelled in reducing MSE, suggesting a superior capability in managing large prediction errors and outliers. However, in terms of runtime, the GA-LSTM model significantly outperformed the others. These findings indicate that while the models share similar predictive accuracies, their practical application might be differentiated by considerations of computational efficiency.

INDEX TERMS Thermal energy, forecasting, solar heating, artificial neural networks, long short-term memory, recurrent neural networks.

I. INTRODUCTION

Thermal energy forecasting predicts heat-load demand ahead of time, enabling better system control, reduced waste, and greater integration of renewable sources. By making the necessary adjustments to heating systems, the objective is to minimize energy waste and maximize energy efficiency. Numerous techniques, such as machine learning algorithms, mathematical models, and weather-based forecasting systems, can be used to forecast thermal energy.

This study evaluates solar thermal energy produced by a Solar Heat for Industrial Processes (SHIP) system in Johor,

The associate editor coordinating the review of this manuscript and approving it for publication was Zhan-Li Sun.

Malaysia; SHIP systems utilize solar thermal collectors to generate hot water or steam for industrial processes, thereby reducing fossil fuel use. The industrial sector, accounting for nearly 50% of energy consumption emissions [1], is particularly significant in Malaysia at 48.8% of total demand [2]. The International Renewable Energy Association (IRENA) estimates SHIP systems could meet 15EJ of 173EJ industrial needs by 2030 [3]. Malaysia receives an average daily solar radiation of 4500kWh/m^2 and ample sunlight, averaging 10 hours per day, meaning it is situated in an ideal location to take advantage of solar energy to its fullest and to effectively carry out this study [4]. This is supported by Malaysia's tropical climate near the equator, featuring high humidity of between 80 to 90% and consistent temperatures ranging from 18 to 32°C, ensuring abundant solar irradiation year-round [5], [6].

This study evaluates the performance of various machine learning algorithms for thermal energy forecasting in a Solar Heat for Industrial Processes (SHIP) system in Malaysia. Algorithms including Support Vector Regression, Random Forest, Decision Tree, XGBoost, Artificial Neural Networks, and Long Short-Term Memory are implemented and compared.

Mathematical models, such as Numerical Weather Prediction (NWP), are widely used to predict and simulate weather conditions by assessing various factors. NWP can produce accurate day-ahead GHI estimates in clear conditions but performs worse under overcast skies [7]. Machine learning (ML) has advanced thermal energy forecasting, particularly for micro-grids [8], [9], where accurate predictions balance energy production and consumption more effectively than in large-scale grids. For heat-intensive systems like HVAC, which are notoriously inefficient, ML approaches like deep belief networks improve efficiency and reduce computational costs [10], [11], while short-term forecasts minimize reliance on polluting standby plants [12]. Across forecasting tasks, a few patterns emerge from comparative studies and reviews. Ensemble and tree-based methods (Random Forest, XGBoost) have shown strong performance for irradiance and solar-output prediction (often leading in R^2 and RMSE after hyperparameter tuning via Random Search Cross Validation or Optuna) [14], while kernel and feedforward approaches (SVR, MLP) can outperform others in some time-series applications when optimized (e.g., grid search), as seen in SVR and MLP outperforming XGBoost and LSTM for stock market forecasting [13]. Neural approaches, including ANNs, LSTMs, and hybrid CNN-LSTM architectures, have delivered especially promising results for short-term thermal and irradiance forecasting and for predicting collector inlet/outlet temperatures (a hybrid CNN-LSTM study reported MSE = 5.061 MWt and MAE = 2.871 MWt in Zhangbei) [17], [30]. Similar optimization techniques have also been applied to LSTM variants in other domains, such as using Tree-Structured Parzen Estimator (TPE) with Self-Attention Bidirectional LSTM for attack detection in Industrial Internet of Things (IIoT) networks, achieving high accuracy in cybersecurity intrusion detection [18]. Comparative and review studies highlight that model ranking depends strongly on forecast horizon, feature selection, preprocessing, and hyperparameter search, and that multi-criteria evaluation is often required to select the best method for each phase of deployment (e.g., favoring Multi-linear Regression for pre-processing and implementation in building thermal demand assessments) [19], [20], [29]; for example, [20] reviewed PV forecasting techniques (e.g., ARIMA, MLPNN, RNN, RBFNN, ELM, CNN, DCNN, RBM, DBN), concluding ANNs (especially CNN hybrids) excel in short-term horizons, influenced by inputs, weather, and optimizations, aligning with [21], [22], and [23] where neural networks outperformed statistical models for solar energy. A broader range of algorithms (AdaBoost, Echo State Networks, SARIMA, etc.) has also been explored for energy demand forecasting [24], [25], [26], highlighting that no single algorithm universally dominates. Furthermore, performance can degrade with longer timestamps or sudden peaks, pointing to the need for robust datasets and careful model design, as noted in declining accuracy for LSTM, CNN, and GRU models during consumption peaks in residential load forecasting [31]. In addition to forecasting accuracy, secure communication in IoT-enabled SHIP systems is crucial, such as lossless data-hiding schemes for urban sensor networks that ensure privacy during data transmission, as well as the demonstration of lossless schemes using histogram transformation for 6G IoT environments [27], [28]. Regionally, Malaysian studies found ANNs capable of accurate PV power prediction using a small set of inputs (irradiance, temperature) but recommended ensemble/regularization techniques to mitigate overfitting, while LSTM applications to SHIP parameters achieved low RMSEs in specific deployments [29], [30]. Finally, several works caution about limits, pointing out that recurrent and convolutional models can degrade when forecasting over longer timestamps or during sudden peaks, highlighting the need for large, representative datasets and careful model design for reliable operational deployment [31].

Current existing literature does extensively cover solar thermal energy forecasting, along with comparative analyses of various ML models in this application with datasets of varying sizes. Despite the presence of various regional studies using ML models to forecast solar thermal energy in the southeast Asian region, there is a need for further analysis and comparison of ML regression models using large datasets within the climate of Malaysia, as most existing literature either centered around this task do not carry out a comparative analysis of various ML regression models or utilize comparatively small datasets, leading to questions of scalability and large-scale applicability. In this paper, we address this need by building several regression models, training and testing them using a large dataset of solar thermal energy collected here in Malaysia, and ultimately evaluating their performances to determine which model is best suited to the task.

The arrangement of the sections of this paper is described as follows. The methodology, detailing the collection of data as well as the split of the data into various quarters, along with the preprocessing and partitioning of the dataset into training and test sets, is described in Section II. This section also covers the basic descriptions of the various models built, along with the techniques utilized for hyperparameter optimization. The performance evaluation metrics are also described in this section. Section III outlines the results obtained from the training and testing of the models in accordance with the performance evaluation metrics outlined in Section II, as well as the results obtained from the hyperparameter optimization. Section IV concludes the paper, highlighting which model performs the best for the given task, along with any accompanying findings.

II. METHODOLOGY

This section details the methods and procedures carried out throughout this study. Firstly, the software platform used for programming and coding of the various algorithms is described, followed briefly by the partitioning of data into training data and testing data. We also go through the performance metrics that will be used to evaluate the performance of each algorithm and model.

The software utilized to carry out the coding and evaluation for each of the algorithms is MATLAB. The dataset we are using here was acquired from a solar heat and industrial process (SHIP) system installed and commissioned at one of the oleochemical process industries in Pasir Gudang, Johor, Malaysia. Forecasting is usually divided into three categories, based on predictive duration, namely, short-term, medium-term, and long-term [32]. This paper mainly focuses on short-term energy forecasting because it considers hourly prediction using hourly energy data.

A. DATASET UTILIZED

The data collected consists of a total of 3,960 hourly generated solar heat energy reading entries from two separate heat meters, respectively designated H1 for 'useful energy' and H2 for 'delivered energy,' over a four-quarter period (Q1, Q2, Q3, Q4) in a year. Q1 consists of data collected over a three-month period, spanning January to March. The other quarters likewise span a similar time period of three months each, with Q2 being from April to June, Q3 from July to September, and Q4 from October to December. For each quarter, the data was split at a ratio of approximately 70% to 30% for training and testing, respectively, amounting to 693 entries for training and 297 entries for testing per quarter. This partitioning is applied separately in different executions of the code for each quarter, resulting in aggregated totals of 2,772 entries for training and 1,188 entries for testing across all quarters. The same per-quarter partitioning approach is used in all of the models utilized in this study. Upon training and testing, the results of the algorithms and models were evaluated based on the various performance evaluation metrics outlined in section K.

The measurements of solar heat energy from the heat meters serve as the target variable for our model, with separate modeling performed for H1 (useful energy) and H2 (delivered energy). In addition to the targets, we selected four key features for our models to train and be tested on. Those features are solar irradiation (captured via a pyranometer), supply temperature, collector inlet temperature, and collector outlet temperature, as depicted in Fig. 1. In addition to these, there were four timestamp columns in the dataset (hour, day, month, year). We excluded these from the regression because the year column is constant and therefore contains no information, while day and month showed negligible correlation with the energy targets once measured solar irradiance and temperature measurements were included.

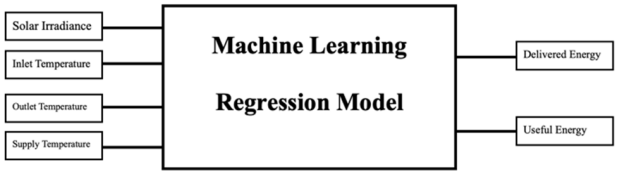

FIGURE 1. Features and targets of SHIP system data.

These variables provide essential context regarding the physical processes involved in heat capture and transfer within the system. Each feature is recorded at the same hourly intervals, enabling the model to leverage sequential temporal patterns inherent in the time-series data.

For data preprocessing, prior to model training, the dataset underwent z-score standardization, ensuring numerical stability and enhanced convergence, especially for gradient-based models like XGBoost, while preserving the original data distribution. Outlier detection was applied to each quarter's training targets for H1 (useful energy) and H2 (delivered energy) using three separate methods suited to the time-series data, namely, median absolute deviation (MAD), generalized extreme studentized deviate (GESD), and t-distributed Stochastic Neighbor Embedding (t-SNE).

For our implementation and dataset, the median absolute deviation (MAD) method, with a threshold of 3 times the MAD, and the generalized extreme studentized deviate (GESD) method, iteratively identifying up to 10% of points as outliers with a significance level of 0.05, found no anomalies across all quarters for both targets, indicating robust sensor data in the oleochemical factory setting. Exploratory t-SNE, configured with 2 dimensions and a perplexity of 30 to balance local and global structure in the 693-sample quarters, visualized multivariate data; H1 showed isolated points in Q2 (7.07%) and Q4 (3.03%), while H2 exhibited higher rates across Q1 (7.07%), Q2 (5.19%), and Q3 (4.47%), and none in Q4. This likely reflects delivery-specific variability like heat losses.

As such, no additional imputation or outlier removal was performed because the outliers reflected real operational

TABLE 1. Quarterly outlier detection results for H1 and H2.

	H1			H2		
	GESD Method	Median Method	t-SNE	GESD Method	Median Method	t-SNE
Q1	Detected 0 outliers (0% of training data)	Detected 0 outliers (0% of training data)	Detected 0 t-SNE-based outliers (0% of training data).	Detected 0 outliers (0% of training data)	Detected 0 outliers (0% of training data)	Detected 49 t-SNE-based outliers (7.0707% of training data).
Q2	Detected 0 outliers (0% of training data)	Detected 0 outliers (0% of training data)	Detected 49 t-SNE-based outliers (7.0707% of training data).	Detected 0 outliers (0% of training data)	Detected 0 outliers (0% of training data)	Detected 36 t-SNE-based outliers (5.1948% of training data).
Q3	Detected 0 outliers (0% of training data)	Detected 0 outliers (0% of training data)	Detected 0 t-SNE-based outliers (0% of training data).	Detected 0 outliers (0% of training data)	Detected 0 outliers (0% of training data)	Detected 31 t-SNE-based outliers (4.4669% of training data).
Q4	Detected 0 outliers (0% of training data)	Detected 0 outliers (0% of training data)	Detected 21 t-SNE-based outliers (3.0259% of training data).	Detected 0 outliers (0% of training data)	Detected 0 outliers (0% of training data)	Detected 0 t-SNE-based outliers (0% of training data).

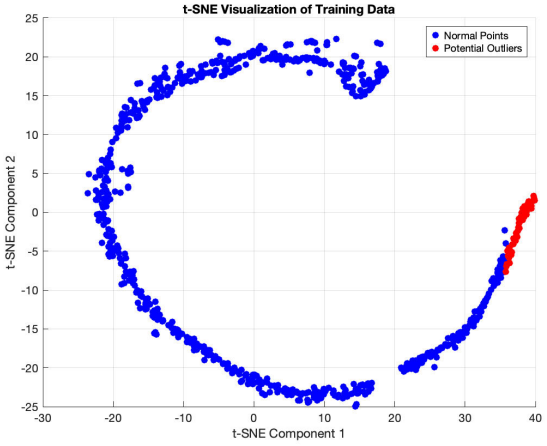

FIGURE 2. tSNE visualization of outliers for useful energy (H1) in Q2.

events relevant to forecasting performance in an industrial setting, and we wished to evaluate model robustness to such events.

Feature importance was assessed per quarter using tree-based ensembles, offering insights into solar thermal forecasting factors in Malaysia's tropical climate. The feature importance values across the four quarters presented similar patterns, and as such, we've averaged these values, as outlined in Table 2.

Random Forest's permutation importance, derived from 100 trees with out-of-bag error estimation, measured mean squared error increases upon feature shuffling; for H1, solar irradiation led (2.1779), followed by collector outlet temperature (1.0643), while H2 showed a similar trend but with a slightly lower solar irradiation average (2.1676) and elevated collector outlet influence (1.1388) due to delivery dynamics,

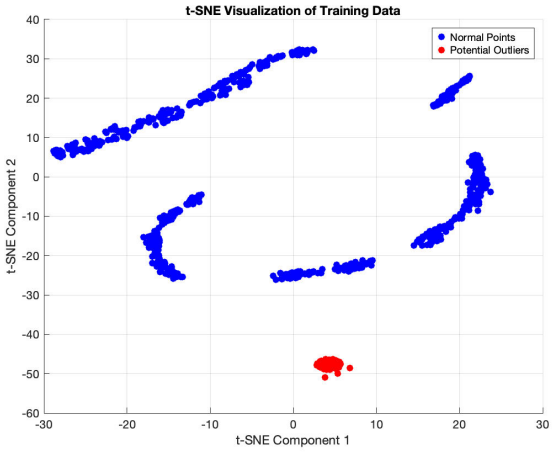

FIGURE 3. tSNE visualization of outliers for delivered energy (H2) in Q1.

TABLE 2. Averaged feature importance metrics (Q1-Q4).

Features	Avg. RF Permutation (ΔMSE)		Avg. XGBoost Split		Avg. XGBoost SHAP	
	H1	H2	H1	H2	H1	H2
Inlet Temp.	1.0383	1.0385	0.0268	0.0262	4.6624	5.1250
Supply Temp.	0.7099	0.6614	0.0175	0.0132	2.7864	2.8928
Outlet Temp.	1.0643	1.1388	0.0548	0.0589	10.1326	11.9697
Solar Irradiation	2.1779	2.1676	1.7876	1.6976	18.0417	15.8728

as can be seen in Fig. 4. XGBoost's split-based importance, based on 100 learning cycles with a maximum of 5 node splits per tree, and SHapley Additive exPlanations (SHAP) values, computed via the shapley function with training data as background and query points, reinforced this, whereby H1's solar irradiation averaged 1.7876 (split-based) and 18.0417 (SHAP), while H2's was 1.6976 (split-based) and 15.8728 (SHAP). Overall, these results illustrate solar irradiation as a dominant feature, modulated by temperature effects varying between energy capture (H1) and delivery (H2), with collector outlet temperature a relatively close second.

B. FLOW PROCESS OF SHIP SYSTEM

The Solar Heat for Industrial Process (SHIP) system consists of three arrays of ETCs, which were installed on the rooftop of the boiler section of the factory.

FIGURE 4. Side view of solar panel array.

The specific installation of the ETC array is shown in Fig. 4. The technical specifications of the specific ETC configuration are presented in Table 3.

TABLE 3. Specifications of CPC 1518 ETC panel.

Item	Unit	Values
Annual Yield	kWh/m^2	651
Dimensions	m	2.08 x 1.64 x 0.1
Gross Surface Area	m^2	3.4
Aperture Area	m^2	3.0
Collector Contents	Litre	2.4
Weight	kg	54
Maximum Working Overpressure	bar	10
Maximum Stagnation Temperature	°C	345

The schematic representation of the flow process of a SHIP system can be seen in Fig. 5. A typical SHIP system usually comprises a heat exchanger, piping, hot and cold-water storage tanks, a PLC-based control and monitoring system, and a water circulation pump with a power supply as a basic configuration. The two types of solar collectors most frequently used for low to medium temperature industrial and domestic water heating applications are evacuated tube collectors (ETC) and flat plate collectors (FPC). To allow energy to build up, these solar collectors are connected to thermal storage devices. Thus, energy delivery is made possible during the daytime hours when solar irradiation is at its peak.

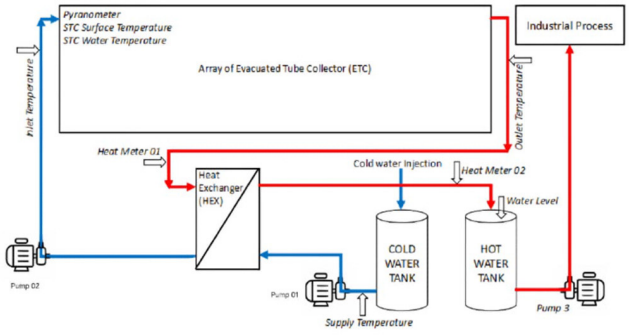

FIGURE 5. Flow process of SHIP system.

The first step of the process is carried out by supplying cold water into the cold-water tank, which has a $20m^3$ storage capacity. The first water pump will then pump the water from the aforementioned cold-water tank to the heat exchanger. A second water pump will subsequently pump the water from the heat exchanger to the array of solar panels before circulating it back heat exchanger. During this described process, the flow rate of both water pumps is fixed at 6.5 m^3/hour. In the final step, the generated hot water is channeled from the heat exchanger to the hot water tank, which has the same storage capacity as the cold-water tank. A third water pump is then used to channel the water from the hot water tank to the oleochemical factory process. The PLC's ladder logic software can only be used to control Pumps 01 and 02. When the solar irradiation is equal to or higher than 100 Wh/m^2, both of these pumps are turned on. They are deactivated when the temperature differential between the solar panel's intake and output is less than 3°C. There are heat meters installed in two different locations to monitor delivered energy (H2) and useful energy (H1). The first heat meter is located at the ETC array loop and measures the useful energy (H1). The second heat meter is located just before the hot water tank's final delivery, and measures delivered energy (H2). These heat meters can produce the energy value in kWh instantaneously and are equipped with flow and temperature sensors. An ultrasonic level sensor is mounted on top of the hot water tank to keep track of the water level within. In various places, five distinct temperature sensors are in use. Additionally, a pyranometer is placed on the rooftop close to the ETC array to track the solar irradiance.

C. MEASUREMENT AND MONITORING USING INDUSTRIAL INTERNET OF THINGS (IIoT)

Using a cloud-based application, an Industrial Internet of Things (IIoT) system was set up to remotely monitor the SHIP system's parameters. A programmable logic controller (PLC) is used for the task of data logging. It is interfaced to a range of sensors, as measurements are taken every five minutes. The measurements are then sent to the cloud via an IIoT data gateway. Table 4 displays the equipment catalogue for this SHIP system.

TABLE 4. Equipment breakdown for SHIP system.

Item	Quantity	Brand	Model
Pyranometer	1	RIKA	RK-200-03
Wind and Direction	1	RIKA	RK-120-0
Surface Temperature Sensor	1	Optris	IR-CS LT
Water Level Sensor	1	Senix	CHEM 35
Energy Meter	3	DIEHL	Sharky 775
Water Temperature Sensor	3	JUMO	PT 500
PLC	1	Siemens	S7-1200
Industrial IoT Gateway	1	Siemens	IOT2040

At various points along the SHIP system, three panels stored in weather-proof enclosures are utilized. Every one of these panels is intended to manage a certain SHIP system function. The first panel is connected to heat meter 01, pump 01, and pump 02, and is situated close to the heat exchanger. A power supply, PLC, Ethernet switch, touch HMI, IIoT gateway, relays, and magnetic contacts for the circulation pumps are all included in one panel. The second panel is situated on the rooftop and is equipped with a power supply and PLC IO module that are linked to a pyranometer, surface temperature sensor, wind speed and direction sensor, and water level sensor. In addition to this, a power supply and a PLC IO module are connected to heat meter 02. A hot water temperature sensor and a cold-water temperature sensor can also be found on the third panel, which is situated close to the

hot water tank. The SHIP System is monitored both locally and remotely via Adafruit IO. This is an IoT web platform that allows for dashboard display, logging, and general process control, as can be seen in Fig. 6.

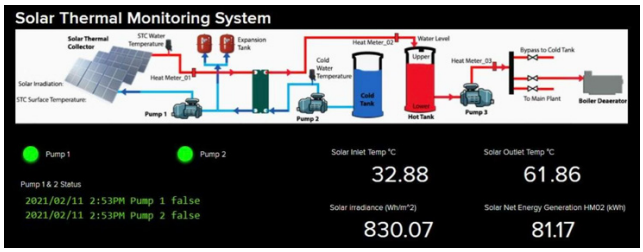

FIGURE 6. Adafruit IO for SHIP system.

D. SUPPORT VECTOR REGRESSION (SVR) MODEL FORECASTING

Support Vector Regression (SVR) extends Support Vector Machines (SVM) to regression by finding a function $f(x)$ that approximates input-output relationships within an error margin ϵ, while minimizing function complexity. This is achieved through convex optimization, minimizing

$$\frac{1}{2} \parallel w \parallel^2 + C \sum (\xi_i + \xi_i^*) \tag{1}$$

subject to

$$\mid y_i - \langle w, x_i \rangle - b \mid \leq \epsilon + \xi_i^{(*)} \tag{2}$$

and $\xi_i^{(*)} \geq 0$, where w is the weight vector, b the bias, and slack variables allow deviations. SVR excels in nonlinear time-series forecasting due to kernel flexibility.

Our SVR model was implemented using MATLAB's 'fitrsvm' function with a Gaussian kernel for non-linear mapping, and standardization enabled to scale features. No explicit hyperparameter tuning was performed, and the default values for epsilon (0.1) and box constraint (1) were used, focusing on baseline performance.

E. DECISION TREE MODEL FORECASTING

Decision Tree regression is a non-parametric algorithm for continuous prediction, splitting data into subsets based on feature values to form a tree structure. It subdivides the input space into regions $R_1, R_2, \ldots, R_n$, predicting the mean target value per region to minimize the loss function outlined in Eq. 3

$$\text{Cost}\,(T) = \sum_{j=1}^{n} \sum_{i \in R_j} \left(y_i - y_{\hat{R}_j}\right)^2 \tag{3}$$

where T is the tree, n the terminal nodes, y_i the actual value, and $\hat{y}_{R_j}$ the regional mean. Like SVR, DT handles complex nonlinear relationships in forecasting.

Our decision tree used 'fitrtree' with a minimum leaf size of 5 to control overfitting. No additional tuning was conducted, employing default settings for the split criterion (MSE) and surrogate splits.

F. RANDOM FOREST MODEL FORECASTING

Random Forest is an ensemble learning method that builds multiple decision trees and aggregates their predictions (via averaging in regression) to enhance accuracy and reduce variance.

As shown in Fig. 7, individual trees generate predictions, which are averaged for the final output. Consequently, given that it is a combination of several decision trees, it is expected to provide overall superior predictions to the decision tree algorithm.

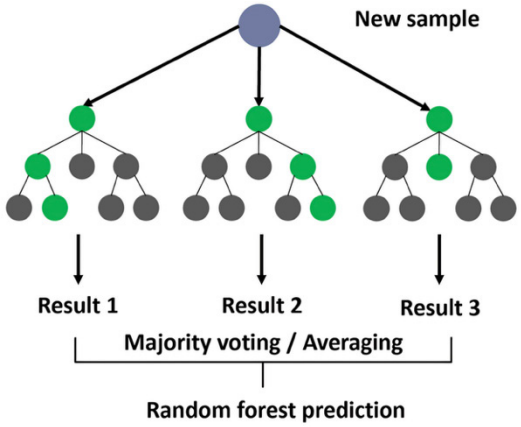

FIGURE 7. Random forest algorithm architecture.

For our model, we trained an ensemble of 100 regression trees using MATLAB's TreeBagger with out-of-bag prediction enabled. Other TreeBagger settings, like the number of predictors sampled per split, were left at their defaults. We used rng (1) to keep the ensemble deterministic for repeated experiments.

G. XGBoost MODEL FORECASTING

eXtreme Gradient Boosting (XGBoost) is a gradient boosting ensemble algorithm that iteratively adds base learners (trees) to minimize errors, combining predictions for improved accuracy. It optimizes an objective with a loss function and a regularization term outlined in Eq. 4

$$\mathcal{L}\,(\theta) = \sum_{i=1}^{n} l\left(y_i, \hat{y_i^{(t)}}\right) + \sum_{k=1}^{t} \Omega\,(f_k) \tag{4}$$

where l measures prediction error, $\hat{y}_i^{(t)}$ is the t-th iteration's output, and $\Omega(f_k)$ penalizes tree complexity. At every step, the algorithm aims to minimize the objective as outlined in Eq. 5

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t\,(x_i) + \frac{1}{2} h_i f_t\,(x_i)^2 \right] + \Omega\,(f_t) \tag{5}$$

with g_i, h_i as first/second derivatives of the prior loss, focusing new learners on residuals while regularization controls overfitting.

The usual 'XGBoost' function that is used to create this model is not readily available in MATLAB, as was the case with the other models. As such, we utilized a Least Squares Boosting (LSBoost) method to create a custom XGBoost

function. Since the goal is to minimize the error at each step by sequentially fitting models to the residuals of the previous model, LSBoost is ideal here as it works by sequentially adding trees to achieve this desired goal. The other two hyperparameters that we've configured in this model are MaxNumSplits and NumLearningCycles. The MaxNumSplits controls the maximum number of splits for each tree node, thereby influencing complexity, and is set to 5 in our model. The NumLearningCycles, on the other hand, determines the number of boosting iterations to be added to the model. This is set to 100.

H. ARTIFICIAL NEURAL NETWORK MODEL FORECASTING

In this section, a basic ANN model was built for the forecasting task, utilizing the Levenberg-Marquardt algorithm for training. As depicted in Fig. 8, the model was built with 4 input dimensions, corresponding to the 4 features in the dataset. This is accompanied by a fully connected hidden layer with 10 neurons, each of which applies a weighted sum followed by a nonlinear activation function, the output H of which is given by Eq. 6

$$H = \sigma\left(W_1 X + b_1\right) \tag{6}$$

where $W_1 \in R^{10*4}$ denotes the weights connecting the input to the hidden layer, $b_1 \in R^{10}$, the biases of the hidden layer, and σ (.), the activation function. The output Y of the output layer is given by Eq. 7

$$Y = \left(W_2 H + b_2\right) \tag{7}$$

where $W_2 \in R^{1*10}$ represents the weights connecting the hidden layer to the output layer and $b_2 \in R^1$ denotes the bias of the output layer. No look-back window was applied, as the feedforward architecture treats inputs as independent.

The other parameter of note is the learning rate, which is dynamically adjusted during training based on performance, owing to the usage of the Levenberg-Marquardt function. The batch size corresponded to the full training set of 693 samples per quarter for gradient updates. Within those 693 samples, we set the model to internally divide the data randomly into 70% training, 15% validation, and 15% internal test sets, and the validation set was used to monitor performance and enable early stopping. The maximum number of epochs was set to 1000.

I. LONG SHORT-TERM MEMORY MODEL FORECASTING

Long Short-Term Memory (LSTM) is a recurrent neural network (RNN) variant designed to address vanishing gradients in traditional RNNs by leveraging feedback connections to capture temporal dependencies in sequence data. The LSTM model for our forecasting task features a sequence input layer with 4 dimensions, corresponding to the input features, a single LSTM layer with 151 hidden units, a fully connected layer for 1 response, and a regression layer. For a time step t, the LSTM layer is described by Eq. 8.

$$\begin{aligned} i_t &= \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \\ f_t &= \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \\ C_t &= f_t \odot C_{t-1} + i_t \odot \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right) \\ o_t &= \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \\ h_t &= o_t \odot \tanh\left(C_t\right) \end{aligned} \tag{8}$$

where h_{t-1} is the prior hidden state, x_t, the current input, C_t the cell state, and i_t, f_t, o_t respectively denote the input, forget, and output gates; σ denotes sigmoid activation, *tanh* the hyperbolic tangent, and $\odot$ element-wise multiplication.

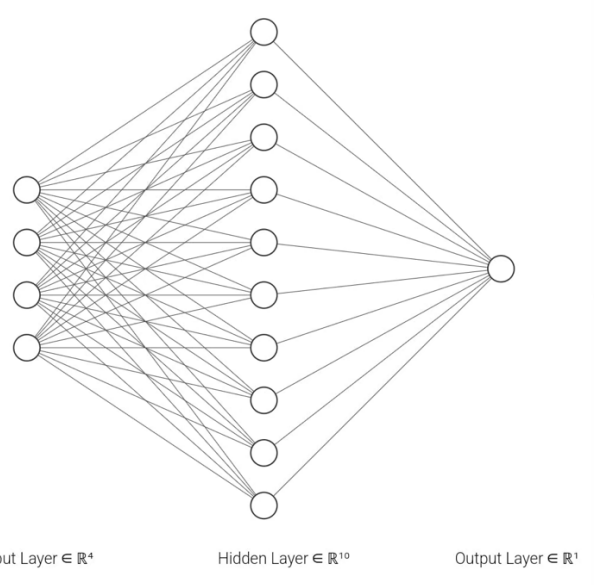

FIGURE 8. Neural network architecture for ANN.

Training employed the Adam optimizer with a 1-step look-back for hourly ahead predictions, 205 maximum epochs, a gradient threshold of 1, an initial learning rate of 0.0038, and a piecewise schedule dropping the rate every 85 epochs by a factor of 0.9613; the default mini-batch size was 128.

J. PARAMETER OPTIMIZATION

In this section, we describe the parameter optimization performed on the LSTM model at each iteration and quarter in order to maximize its performance. Three optimization techniques have been chosen for this task, namely Salp Swarm Algorithm (SSA), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO), from which the best algorithm for the task will be highlighted. The respective parameters to be tuned are the number of hidden units ($x1$), max epochs ($x2$), initial learning rate ($x3$), learning rate drop period ($x4$), and the learning rate drop factor ($x5$). Some of these hyperparameters are abbreviated in Tables 8 to 11 as Initial LR (Initial Learning Rate), Drop Period for LR (Drop Period for Learning Rate), and LR Drop Factor (Learning Rate Drop Factor). The range for determining the optimal hyperparameters was confined to the following intervals:

$$10 < x1 < 300$$

$$50 < x2 < 500$$
$$0.001 < x3 < 0.01$$
$$50 < x4 < 200$$
$$0.8 < x5 < 1.$$

1) GENETIC ALGORITHM

Genetic Algorithm (GA), as the name may indicate, is inspired by the principles of natural selection, where the fittest organisms survive and reproduce to form successive generations with improved adaptability and survivability. In this study, GA is employed to determine the ideal specifications of the previously described hyperparameters for the proposed deep learning-based method, within the previously specified constraints. The GA operates by generating an initial population of solutions randomly within the defined constraints. The algorithm proceeds through five key stages: population initialization, selection, crossover, mutation, and convergence. Each stage requires specific parameters, namely, population size, crossover rate, extra range factor for crossover, mutation rate, maximum number of generations, and number of mutants. The specific parameters used in this study are outlined in Table 5.

TABLE 5. Genetic algorithm parameters.

Parameter	Value
Population Size	30
Crossover Rate	System-defined
Extra Range Factor for Crossover	Unused
Mutation Rate	System-defined
Maximum Number of Generations	50
Number of Mutations	System-determined

The GA, by combining and mutating the selected chromosomes, is able to evolve the population iteratively. By minimizing the value of the following fitness function, outlined in Eq. 9

$$F = \sum_{\lambda_1} (1 - A_{TM})^2 + \sum_{\lambda_2} (A_{TM})^2 \tag{9}$$

where, F represents the fitness function, λ_1 and λ_2, the wavelengths, and A_{TM}, the absorption value, this procedure seeks to determine the ideal hyperparameter configuration, which should enhance the overall forecast performance for the solar thermal parameters [33], [34], [35].

2) PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO), much like GA, takes inspiration from nature and utilizes a population-based algorithm inspired by the social behavior of some groups of animals, like a flock of birds or schools of fish. It tackles optimization problems by initializing a swarm of particles, each representing a potential solution to the problem or a point in the search space. Each particle has two notable properties: the position, which represents the current solution it embodies, and the velocity, which determines how it moves through the search space. Each particle then proceeds to update its position based on its own best-known position and the best-known position of the entire swarm [36], [37], [38]. Based on their individual experiences as well as the collective experience of the swarm, particles change their velocity at each iteration. This can be mathematically represented as follows in Eq. 10

$$\begin{aligned} X_2 &= X_1 + V_2 \\ V_2 &= \omega V_1 + c_1 r_1 (P_{best} - X_1) + c_2 r_2 (g_{best} - X_1) \end{aligned} \tag{10}$$

where X_1, X_2, V_1, V_2, c_1, c_2, ω and r respectively represent the current position, new position, current velocity, new velocity, positive acceleration, constant acceleration, inertia weight, and random values, the specific values of which are outlined in Table 6.

TABLE 6. Particle swarm optimization parameters.

Parameter	Value
Population Size	30
Maximum Number of Iterations	50
Inertia Weight (ω)	0.5
Positive Acceleration (c_1)	1.5
Constant Acceleration (c_2)	1.5
Maximum Number of Generations	50
Number of Mutants	Random Between 0 and 1

The PSO algorithm continuously improves solutions by iteratively updating the positions and velocities of the particles. The same fitness function that was described in the section on GA is used to assess each particle's fitness. This iterative process will continue until a termination criterion, such as reaching the maximum number of iterations or attaining an acceptable solution quality, is met.

3) SALP SWARM ALGORITHM

The Salp Swarm Algorithm (SSA), similar to the GA and PSO, is inspired by the behavior of organisms found in nature. It specifically mimics the collective behavior of salps, marine organisms that form chains in the ocean while navigating to improve their movement and navigation. A SSA consists of a population of salps, where each salp represents a potential solution in the search space. This population is usually divided into two types of agents, namely the leader and the follower. The leader, representing the first salp in the chain, explores the search space while the followers, representing the rest of the salps in the chain, adjust their positions based on the leader's movements. The algorithm operates iteratively, where the leader's position is updated based on a food source, which is represented by the fitness function. The algorithm defines the food position as the optimal solution

in the search space, which is determined by evaluating the fitness of each salp, and the positions of both aforementioned agents are updated thereafter. This dynamic allows the swarm to explore and exploit the search space effectively, ultimately converging towards the best solution identified so far [39], [40], [41]. The specification of the parameters that influence the SSA's performance is outlined in Table 7.

TABLE 7. Salp swarm algorithm parameters.

Parameter	Value
Population Size	20
Maximum Iterations	50
Leader Behavior	Fixed (From Existing Data)
Follower Behavior	Stochastic (if $r < 0.5$)
Boundary Handling	Clamping Bound

The optimized hyperparameters that were obtained using each of the algorithms (GA, SSA, PSO), across the four quarters, will vary, and are displayed in Tables 8 through to Table 11. The forecasting accuracy of the model after applying the optimized parameters is outlined in Tables 25 to 28.

TABLE 8. Optimized hyperparameters for Q1.

Methods	Heat Meter	Hidden Units	Epochs	Drop Period for LR	Initial LR	LR Drop Factor
GA-LSTM	H1	72	191	180	0.0082	0.9895
	H2	162	191	182	0.0068	0.9865
SSA-LSTM	H1	170	300	139	0.0010	0.9854
	H2	200	300	200	0.0010	0.8445
PSO-LSTM	H1	65	333	33	0.0071	0.6769
	H2	38	429	19	0.0099	0.9134

TABLE 9. Optimized hyperparameters for Q2.

Methods	Heat Meter	Hidden Units	Epochs	Drop Period for LR	Initial LR	LR Drop Factor
GA-LSTM	H1	72	64	16	0.0021	0.9328
	H2	184	89	130	0.0035	0.9773
SSA-LSTM	H1	200	75	151	0.0022	0.9481
	H2	200	50	131	0.0054	1
PSO-LSTM	H1	199	50	10	0.0020	0.99
	H2	135	117	41	0.0063	0.5360

K. PERFORMANCE EVALUATION METRICS

The first performance metric used to evaluate the performance and accuracy of each model is the Mean Squared Error (MSE), which is described in Eq. 11:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{11}$$

TABLE 10. Optimized hyperparameters for Q3.

Methods	Heat Meter	Hidden Units	Epochs	Drop Period for LR	Initial LR	LR Drop Factor
GA-LSTM	H1	55	134	113	0.0075	0.9855
	H2	95	166	70	0.0082	0.9749
SSA-LSTM	H1	97	300	65	0.0035	0.8421
	H2	10	300	197	0.01	0.8
PSO-LSTM	H1	125	209	73	0.0047	0.8773
	H2	11	301	69	0.0036	0.5042

TABLE 11. Optimized hyperparameters for Q4.

Methods	Heat Meter	Hidden Units	Epochs	Drop Period for LR	Initial LR	LR Drop Factor
GA-LSTM	H1	27	196	42	0.0072	0.9256
	H2	27	414	53	0.0056	0.9648
SSA-LSTM	H1	10	300	200	0.01	0.9695
	H2	10	211	141	0.01	0.9676
PSO-LSTM	H1	82	371	79	0.0050	0.5492
	H2	10	448	14	0.0079	0.9015

Mean Squared Error (MSE) is a performance metric that calculates a quantitative difference between the estimated predicted value ($\hat{y}_i$) and the actual true value (y_i). In doing so, it effectively measures how close a regression line is to a set of data points. Root Mean Square Error (RMSE), the square root of MSE, provides an interpretable scale in the same units as the target variable, emphasizing larger errors more than MAE and facilitating comparison across datasets.

The other performance metric is the Coefficient of Determination (R^2). It estimates how well the model fits the data by assessing the proportion of variance in the dependent variable that can be explained by the independent variables. It is mathematically outlined in Eq. 12

$$R^2 = 1 - \frac{\sum\left(y_i - \hat{y}_i\right)^2}{\sum\left(y_i - \bar{y}\right)^2} \tag{12}$$

where, much like the MSE, $\hat{y}_i$ denotes the predicted value, y_i, the actual value, and $\bar{y}$, the mean of the actual values.

Another performance metric we've used is the Pearson Correlation Coefficient (r), which would further help us quantify the linear correlation between the predicted value and the actual value. It is denoted by Eq. 13

$$r = \frac{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)\left(\hat{y}_i - \bar{\hat{y}}\right)}{\sqrt{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2}\sqrt{\sum_{i=1}^{n}\left(\hat{y}_i - \bar{\hat{y}}\right)^2}} \tag{13}$$

where y_i and $\hat{y}_i$ represent the actual and predicted values, and y_i and $\hat{y}_i$, the respective means of the actual and predicted values. Here, a value of $+1$ expresses a positive linear relationship, -1 signifies a negative linear relationship, and 0 indicates that there is no linear correlation. As such,

the closer r is to either +1 or -1, the stronger the linear relationship.

The final performance metric used is the Mean Absolute Error (MAE), which evaluates the average absolute difference between the actual value and the predicted value. It is less sensitive to outliers in the dataset when compared to the MSE performance metric, owing to the fact that it does not differentiate between overestimations and underestimations. It is expressed as follows in Eq. 14

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{14}$$

whereby, much like MSE, y_i and $\hat{y}_i$ respectively denote the actual and predicted values.

III. RESULTS & DISCUSSION

The experimental outcome, along with the significance of the performance of each algorithm and model, is discussed in this section.

The performance in forecasting of useful energy and delivered energy in Q1 was very good in terms of R^2 and MSE for each and every model, with every single model yielding an R^2 of above 0.96 and with no model exceeding a MSE of 36. Across the board (H1 and H2), the models that performed the best in terms of both MSE and R^2 were SVR, ANN, and LSTM, with each having an R^2 of above 0.98 and a MSE of, at most, 16.8, as can be seen in Table 12 and Table 13.

TABLE 12. Forecasting results of useful energy (H1) for Q1.

Method	R^2	MSE	MAE	RMSE
SVR	0.9844	15.6802	3.3289	3.9598
ANN	0.9835	16.1435	3.4911	4.0179
DT	0.9644	35.8282	4.7035	5.9857
RF	0.9754	24.7444	4.1842	4.9744
LSTM	0.9736	24.0955	3.2673	4.9087
XGBoost	0.9678	32.4061	4.6243	5.6926

TABLE 13. Forecasting results of delivered energy (H2) for Q1.

Method	R^2	MSE	MAE	RMSE
SVR	0.9852	14.4587	3.1976	3.8025
ANN	0.9901	9.3942	2.4701	3.0650
DT	0.9642	34.8972	4.6070	5.9075
RF	0.9726	26.7320	4.1865	5.1703
LSTM	0.9801	16.7994	3.1096	4.0987
XGBoost	0.9773	22.1258	3.7051	4.7038

The Decision Tree model, while good, performed the lowest out of the bunch, as outlined in the tables above.

In the second quarter (Q2), the task of forecasting both the useful energy (H1) and the delivered energy (H2) proved to be much more challenging for the models, with some degree of reduction in performance observed for almost every model that was trained and tested.

TABLE 14. Forecasting results of useful energy (H1) for Q2.

Method	R^2	MSE	MAE	RMSE
SVR	0.6145	326.6419	11.4284	18.0732
ANN	0.9543	42.7206	4.6194	6.5361
DT	0.7969	172.0667	9.6525	13.1174
RF	0.6581	289.7506	12.4152	17.0221
LSTM	0.9276	63.9250	6.2680	7.9953
XGBoost	0.6476	298.5949	13.0309	17.2799

TABLE 15. Forecasting results of delivered energy (H2) for Q2.

Method	R^2	MSE	MAE	RMSE
SVR	0.6010	334.2292	11.5049	18.2819
ANN	0.9692	26.8932	3.7816	5.1859
DT	0.7027	249.0110	11.3667	15.7801
RF	0.5872	345.7720	12.9099	18.5949
LSTM	0.9698	26.3430	3.8505	5.1325
XGBoost	0.6362	304.7357	12.8109	17.4567

Despite this, however, the ANN and LSTM models were able to give comparatively good results, showing the lowest drop-off in R^2 values, as well as the smallest increase in MSE. This can be seen in Table 14 and Table 15. Both models performed with similar results. Similar to the first quarter, all of the models trained and tested on the data from the third quarter performed relatively well, but still not quite as good as Q1, with notably lower MSE results, and with no model giving a coefficient of determination lower than 0.89.

TABLE 16. Forecasting results of useful energy (H1) for Q3.

Method	R^2	MSE	MAE	RMSE
SVR	0.9088	101.0797	7.1697	10.0538
ANN	0.9668	35.0321	4.1415	5.9188
DT	0.8961	115.0509	7.6003	10.7262
RF	0.9205	88.0356	6.9045	9.3827
LSTM	0.9551	48.3399	5.3373	6.9527
XGBoost	0.9393	67.2273	6.0863	8.1992

We likewise see a similar trend to Q1 in Q3, where the best-performing models were the ANN and the LSTM, once again outperforming the traditional non-neural network-based algorithms. Out of these, the best performing was XGBoost, while the worst performing was the Decision Tree.

TABLE 17. Forecasting results of delivered energy (H2) for Q3.

Method	R^2	MSE	MAE	RMSE
SVR	0.9226	83.4224	5.6959	9.1336
ANN	0.9718	30.9009	4.3275	5.5589
DT	0.9110	95.8976	6.7077	9.7927
RF	0.9376	67.1942	5.7035	8.1972
LSTM	0.9604	44.1546	5.0346	6.6449
XGBoost	0.9570	46.3327	4.8654	6.8068

TABLE 18. Forecasting results of useful energy (H1) for Q4.

Method	R^2	MSE	MAE	RMSE
SVR	0.6563	282.2581	10.2721	16.8005
ANN	0.8863	96.3119	6.2756	9.8139
DT	0.5985	329.7300	10.7454	18.1585
RF	0.7496	205.6225	8.6208	14.3395
LSTM	0.8883	83.1271	5.8804	9.1174
XGBoost	0.7294	222.2274	8.9687	14.9073

TABLE 19. Forecasting results of delivered energy (H2) for Q4.

Method	R^2	MSE	MAE	RMSE
SVR	0.6711	245.0989	9.5384	15.6556
ANN	0.9338	46.1877	4.1820	6.7962
DT	0.6178	284.7999	9.9451	16.8760
RF	0.7898	156.6370	7.3273	12.5155
LSTM	0.8965	66.3002	5.3783	8.1425
XGBoost	0.7305	200.8542	8.9125	14.1723

In the final quarter (Q4), all of the models struggled with accuracy in performance yet again, with the best performing models, ANN and LSTM, being the only ones to return MSEs of below 100, and R^2 of above 0.87, as outlined in Table 18 and Table 19.

The traditional regression algorithms all showed similar performance in accuracy to Q2. In this instance, the best performing model was the ANN model, with the LSTM model performing to a similar but lesser degree in terms of accuracy.

Additionally, we observe that neural network-based models, specifically, while often yielding better sequence prediction accuracy, are substantially more computationally intensive to train than tree-based models, which can be an important consideration for industrial deployments with limited compute resources. This is especially the case for the LSTM model, which had the highest average runtime of all the models, as can be seen in Table 20.

TABLE 20. Average runtime of models (Q1-Q4).

Method	H1	H2
	Time (seconds)	Time (seconds)
SVR	0.96	0.62
ANN	6.56	5.99
DT	0.75	0.60
RF	1.28	1.22
LSTM	25.47	24.51
XGBoost	1.13	1.12

As previously mentioned, the training and testing for each of the models across the four quarters were done in a 70% (training) to 30% (testing) split. Due to the inconsistency in the performance levels and the accuracy displayed by the various models within this configuration over the four quarters, we applied the K-Fold Cross Validation technique to each of the models to provide a more reliable and consistent estimate of performance and accuracy for each of the models. As recommended by [29], this technique should help reduce overfitting and increase the generalization of the model. K-Fold Cross Validation works by dividing the dataset into **k** equally sized folds and using each fold as a validation set while the remaining $\mathbf{k-1}$ folds are used for training [42]. In our application, **k** was designated as 5. As such, the previously described process was repeated a total of 5 times. Following this, the MSE is calculated to get a mean value, reflecting the model's overall accuracy. This technique is very useful as it ensures that every data point is used for both training and validation. The technique was in every model for forecasting both useful energy (H1) and delivered energy (H2), and the coefficient of determination, as well as the MSE obtained, are outlined in Table 21 and Table 22.

TABLE 21. Forecasting results of useful energy (H1) using K-Fold cross validation.

Method	R^2	MSE	MAE	RMSE
SVR	0.9627	35.4304	3.7840	5.9523
ANN	0.9605	36.2670	3.8027	6.0222
DT	0.9331	63.9425	5.0998	7.9964
RF	0.9551	42.6850	4.0288	6.5334
LSTM	0.9500	47.9264	4.6803	6.9229
XGBoost	0.9461	51.6671	4.8862	7.1880

Using K-Fold Cross Validation, we can see a marked increase in the accuracy and predictive capability of each model. Every single model performed to a satisfactory level, with even the lowest accuracy model, being the Decision Tree, having an MSE of 61.6 and 48.3, and an $\boldsymbol{R^2}$ of 0.93 and 0.94 for H1 and H2, respectively. Interestingly enough, the

TABLE 22. Forecasting results of delivered energy (H2) using K-Fold cross validation.

Method	R^2	MSE	MAE	RMSE
SVR	0.9729	24.3641	2.8472	4.9360
ANN	0.9719	24.7642	2.8563	4.9764
DT	0.9489	46.3873	4.1267	6.8108
RF	0.9661	31.0442	3.1899	5.5717
LSTM	0.9591	37.4168	3.7425	6.1169
XGBoost	0.9576	38.4321	3.8922	6.1994

TABLE 23. Average forecasting results of useful energy (H1).

Method	R^2	$\overline{MSE}$	$\overline{MAE}$	$\overline{RMSE}$
SVR	0.7910	181.4150	8.0498	13.4690
ANN	0.9477	47.5834	4.6319	6.8981
DT	0.8140	163.1690	8.1754	12.7738
RF	0.8259	152.0383	8.0312	12.3304
LSTM	0.9378	53.5902	5.1183	7.3205
XGBoost	0.8210	155.1139	8.1776	12.4545

neural network models, ANN and LSTM, that have consistently outperformed every other model prior to K-Fold Cross Validation, did not display the highest accuracy of the bunch, with that being the SVR model, across both H1 and H2. The ANN was only the second-best-performing model, and LSTM, the fourth-best.

In addition to the results obtained from the performance evaluation metrics, the validation of the LSTM model can also be substantiated through an examination of the model's stability. Furthermore, for the optimized hybrid models, an analysis of the computational runtime can also be employed to this end.

In evaluating the stability of the models, we propose calculating the mean values of various performance metrics across the different quarters. This approach will provide a comprehensive view of each model's performance. Specifically, it will allow for the identification of models exhibiting the least variability in the key statistical measures of R^2, MSE, RMSE, MAE, and the correlation coefficient r. By doing so, we will obtain average scores for these metrics for each model, pertaining to both the delivered and useful energy, and gain a clear picture of which models are the most stable over various quarters. These findings are detailed in Table 23 and Table 24.

The results point to the ANN model demonstrating the lowest amount of variance in terms of performance across the four quarters, thereby making it the most stable in this application. The LSTM model displays the second-best performance in terms of stability, with results very similar to those of the ANN model, both of which marginally outperform all other models.

TABLE 24. Average forecasting results of delivered energy (H2).

Method	R^2	$\overline{MSE}$	$\overline{MAE}$	$\overline{RMSE}$
SVR	0.7950	169.3023	7.4842	13.0116
ANN	0.9664	28.1846	3.6903	5.3089
DT	0.7989	166.1514	8.1566	12.8900
RF	0.8218	149.0838	7.5318	12.2100
LSTM	0.9517	38.3993	4.3433	6.1967
XGBoost	0.8253	143.5121	7.5735	11.9797

TABLE 25. Forecasting results of Q1 with optimized parameters.

Method	Heat Meter 1				Heat Meter 2			
	R^2	MSE	r	MAE	R^2	MSE	r	MAE
GA-LSTM	0.9792	20.998	0.9924	4.5824	0.9702	69.2825	0.9289	8.3232
SSA-LSTM	0.9785	21.62	0.9934	3.8316	0.9839	16.199	0.9931	3.1182
PSO-LSTM	0.9673	5.5379	0.9902	4.5572	0.9565	6.2015	0.9957	5.2874

TABLE 26. Forecasting results of Q2 with optimized parameters.

Method	Heat Meter 1				Heat Meter 2			
	R^2	MSE	r	MAE	R^2	MSE	r	MAE
GA-LSTM	0.8619	117.0694	0.9603	10.8199	0.9717	23.6544	0.9881	4.8636
SSA-LSTM	0.9668	28.1621	0.9869	4.3706	0.9719	24.7259	0.9892	3.8191
PSO-LSTM	0.9622	5.9618	0.9822	4.9385	0.9601	6.1087	0.9825	4.6036

TABLE 27. Forecasting results of Q3 with optimized parameters.

Method	Heat Meter 1				Heat Meter 2			
	R^2	MSE	r	MAE	R^2	MSE	r	MAE
GA-LSTM	0.9580	46.549	0.9837	6.8227	0.9741	27.8665	0.9870	5.2789
SSA-LSTM	0.9658	37.8974	0.9828	4.5607	0.9757	27.5685	0.9889	3.9364
PSO-LSTM	0.9641	6.1802	0.9827	4.6893	0.9775	5.0672	0.9890	3.7512

TABLE 28. Forecasting results of Q4 with optimized parameters.

Method	Heat Meter 1				Heat Meter 2			
	R^2	MSE	r	MAE	R^2	MSE	r	MAE
GA-LSTM	0.9382	50.6533	0.9768	7.1171	0.9157	62.726	0.9583	7.92
SSA-LSTM	0.8548	119.1115	0.9251	6.2937	0.9335	52.1058	0.9702	4.9324
PSO-LSTM	0.9169	7.6333	0.9659	5.4971	0.9224	7.3346	0.9703	5.1308

Using the previously described performance metrics, the predictive accuracy results of the LSTM model with

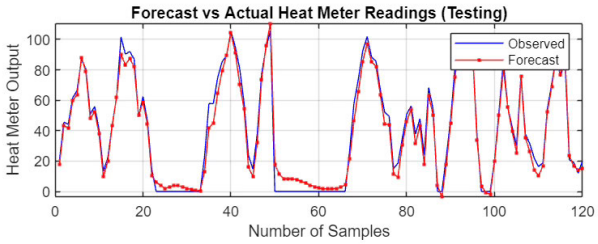

FIGURE 9. Forecast curve of PSO-LSTM of useful energy (H1) for Q1.

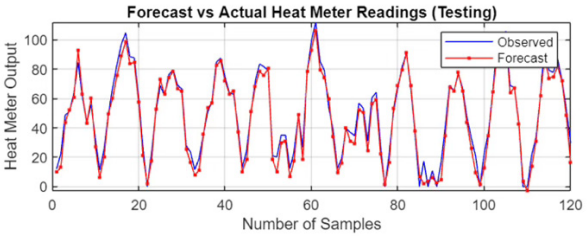

FIGURE 12. Forecast curve of PSO-LSTM of useful energy (H1) for Q2.

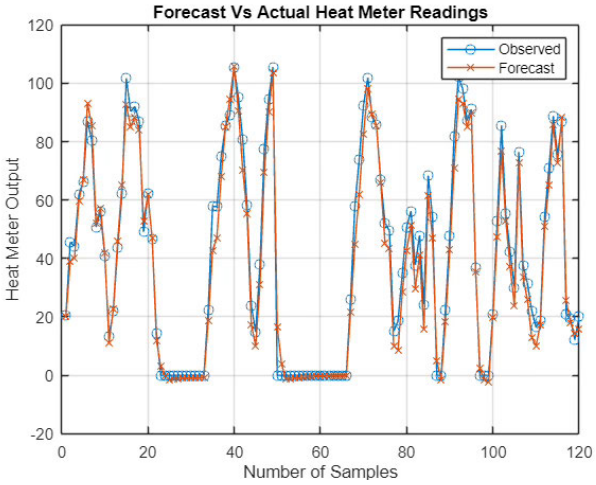

FIGURE 10. Forecast curve of GA-LSTM of useful energy (H1) for Q1.

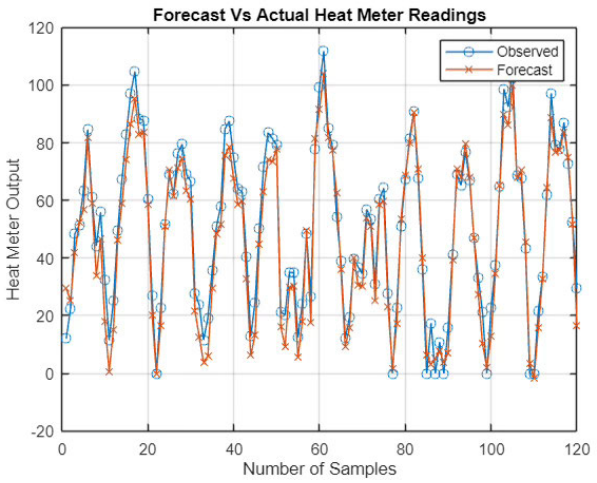

FIGURE 13. Forecast curve of GA-LSTM of useful energy (H1) for Q2.

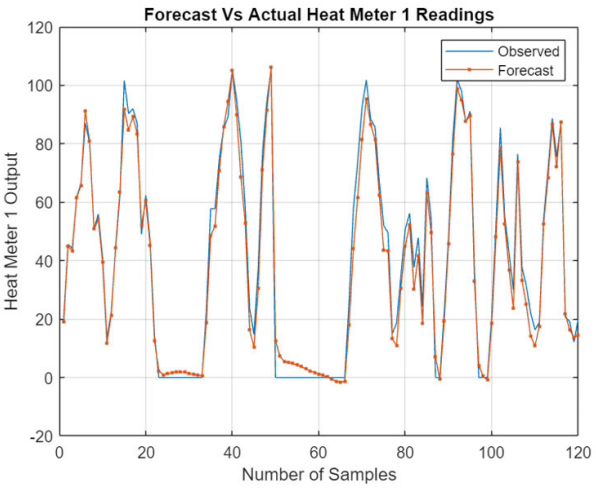

FIGURE 11. Forecast curve of SSA-LSTM of useful energy (H1) for Q1.

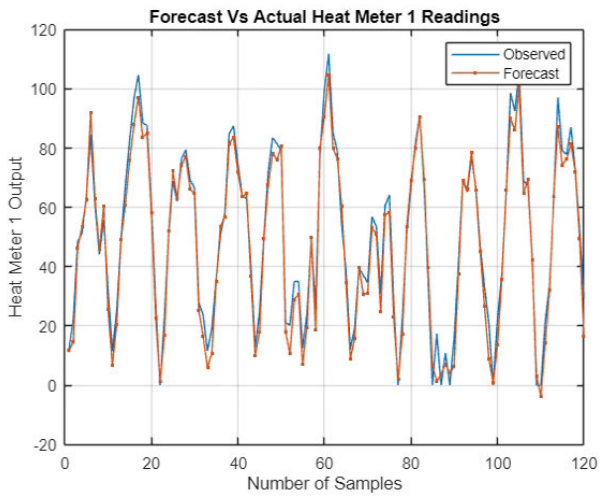

FIGURE 14. Forecast curve of SSA-LSTM of useful energy (H1) for Q2.

optimized parameters are depicted in Table 25 through Table 28, with the forecast curves displayed in Fig. 9 to Fig. 20. Varying levels of performance were observed from the three different parameter optimization algorithms that were used. All three optimization algorithms displayed satisfactory levels of performance and predictive accuracy, greatly improving on the performance of the LSTM model prior to optimization across the four quarters.

To validate the superiority of any one hybrid model over the other, two additional criteria were employed being the analysis of stability and an analysis of the runtime of each hybrid model. The same method that was utilized to evaluate the stability of the regular models is going to be used to evaluate the stability of the hybrid optimized models. The results of this are outlined in Table 29.

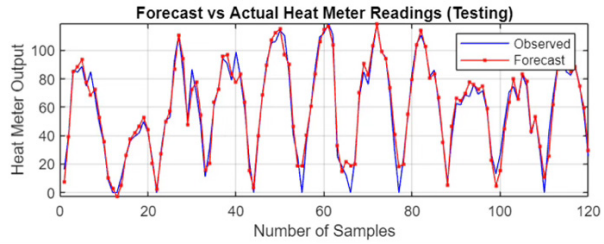

FIGURE 15. Forecast curve of PSO-LSTM of useful energy (H1) for Q3.

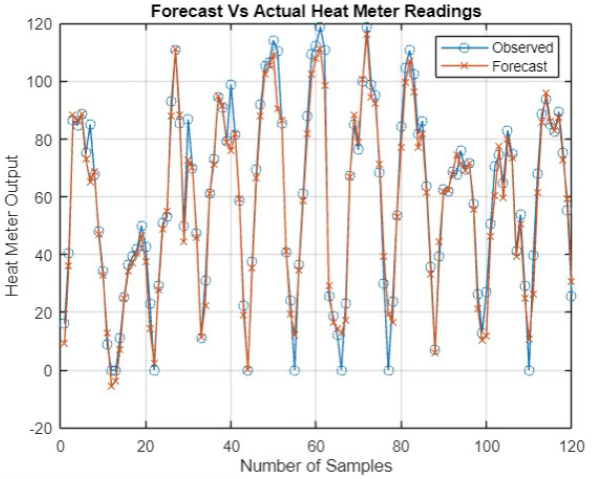

FIGURE 16. Forecast curve of GA-LSTM of useful energy (H1) for Q3.

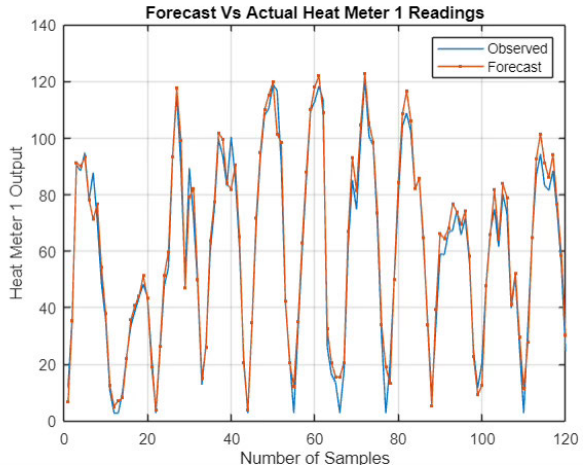

FIGURE 17. Forecast curve of SSA-LSTM of useful energy (H1) for Q3.

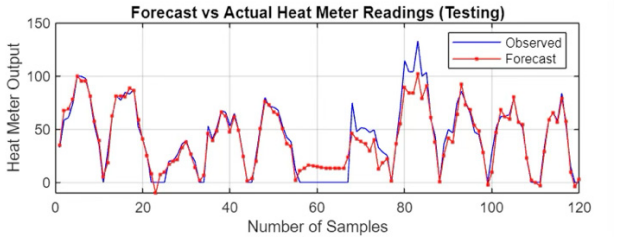

FIGURE 18. Forecast curve of PSO-LSTM of useful energy (H1) for Q4.

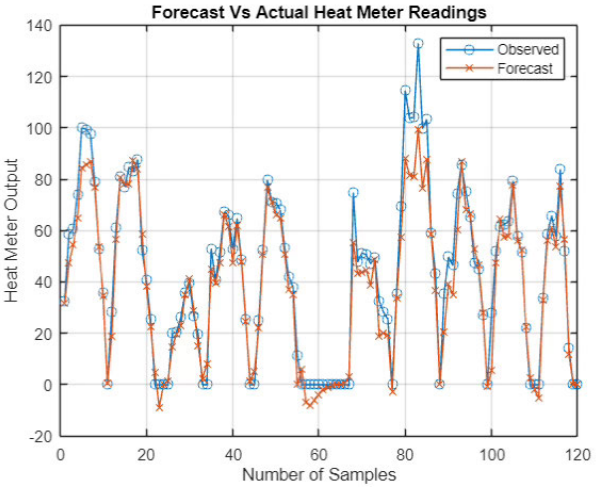

FIGURE 19. Forecast curve of GA-LSTM of useful energy (H1) for Q4.

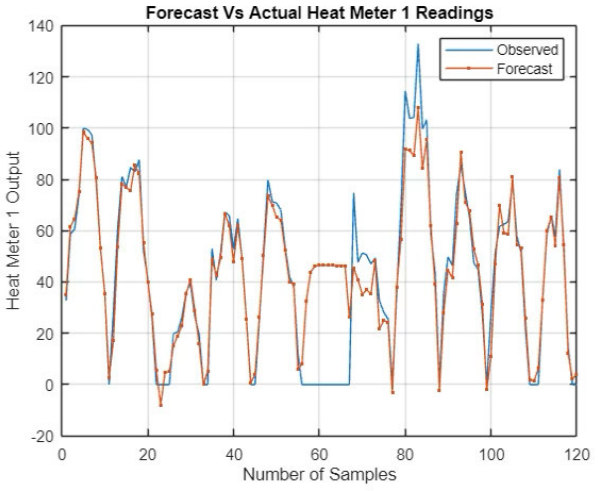

FIGURE 20. Forecast curve of SSA-LSTM of useful energy (H1) for Q4.

Overall, the performance of the three hybrid models was notably consistent, with only negligible variations observed in terms of R^2, MAE and correlation coefficient (r). The only marginal difference of statistical significance was in the MSE, where the PSO-LSTM model markedly outperformed the others by yielding significantly lower values. This suggests that the PSO-LSTM model might be more effective in minimizing large prediction errors, potentially indicating a better capability for handling outliers compared to the GA-LSTM and SSA-LSTM models. However, considering the similarity in MAE across all models, the advantage of the PSO-LSTM in MSE might be less impactful, as it implies that the average error magnitude remains comparable. Thus, the difference in MSE could be attributed to a few isolated instances of larger errors. Consequently, in terms of forecasting accuracy, the distinctions among these models appear to be marginal, rendering them nearly indistinguishable in practical application.

TABLE 29. Average forecasting results of optimized hybrid LSTM models.

Method	Heat Meter 1				Heat Meter 2			
	R^2	MSE	r	MAE	R^2	MSE	r	MAE
GA-LSTM	0.9343	58.8174	0.9761	7.3355	0.9579	45.8824	0.9656	6.5964
SSA-LSTM	0.9415	51.6978	0.9721	4.7642	0.9663	30.1498	0.9854	3.9515
PSO-LSTM	0.9526	6.3283	0.9803	4.9205	0.9541	6.1780	0.9844	4.6933

TABLE 30. Runtime of optimized hybrid models for Q1.

Method	Heat Meter	Time	Time (seconds)
GA-LSTM	H1	1 Hours 22 Minutes 16 Seconds	4936
	H2	2 Hours 57 Minutes 53 Seconds	10673
SSA-LSTM	H1	4 Hours 58 Minutes 35 Seconds	17915
	H2	3 Hours 57 Minutes 31 Seconds	14251
PSO-LSTM	H1	2 Hours 52 Minutes 22 Seconds	10342
	H2	3 Hours 35 Minutes 10 Seconds	12910

TABLE 31. Runtime of optimized hybrid models for Q2.

Method	Heat Meter	Time	Time (seconds)
GA-LSTM	H1	1 Hours 58 Minutes 15 Seconds	7095
	H2	1 Hours 30 minutes 7 Seconds	3607
SSA-LSTM	H1	3 Hours 20 Minutes 15 Seconds	12015
	H2	2 Hours 59 Minutes 8 Seconds	10748
PSO-LSTM	H1	2 Hours 32 Minutes 12 Seconds	9132
	H2	3 Hours 20 Minutes 1 Seconds	12001

TABLE 32. Runtime of optimized hybrid models for Q3.

Method	Heat Meter	Time	Time (seconds)
GA-LSTM	H1	1 Hours 45 Minutes 7 Seconds	6307
	H2	2 Hours 12 Minutes 16 Seconds	7936
SSA-LSTM	H1	4 Hours 13 Minutes 32 Seconds	15212
	H2	3 Hours 56 Minutes 0 Seconds	14160
PSO-LSTM	H1	2 Hours 48 Minutes 15 Seconds	10095
	H2	3 Hours 23 Minutes 1 Seconds	12181

An analysis of the runtime of each model may serve as a critical differentiator between the three hybrid models.

TABLE 33. Runtime of optimized hybrid models for Q4.

Method	Heat Meter	Time	Time (seconds)
GA-LSTM	H1	2 Hours 30 Minutes 27 Seconds	9027
	H2	4 Hours 51 Minutes 9 Seconds	17469
SSA-LSTM	H1	4 Hours 7 Minutes 34 Seconds	14854
	H2	4 Hours 28 Minutes 50 Seconds	16130
PSO-LSTM	H1	4 Hours 53 Minutes 15 Seconds	17595
	H2	5 Hours 12 Minutes 40 Seconds	18760

TABLE 34. Average runtime of optimized hybrid LSTM models.

Method	Average Time (seconds)
GA-LSTM	8381.25
SSA-LSTM	14410.625
PSO-LSTM	12877

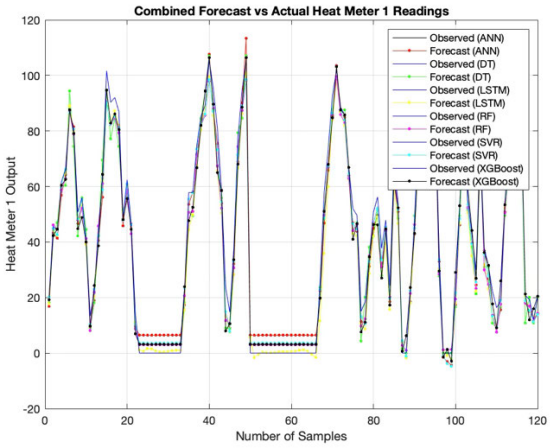

FIGURE 21. Combined forecast curve of useful energy (H1) for Q1.

Looking at the results outlined in Tables 30 through 33, we can see that the GA-LSTM model consistently took the lowest amount of time to run in each of the four quarters. Conversely, both the PSO-LSTM and SSA-LSTM models demonstrated fluctuating performance in terms of runtime.

An analysis of the mean runtime for each model corroborates the initial findings, revealing that the GA-LSTM model possesses the lowest runtime, executing in nearly half the time required by the SSA-LSTM model, which records the highest runtime. The PSO-LSTM model falls in between the two in this regard.

IV. CONCLUSION

The empirical results of this study showed that neural network models (ANN and LSTM) outperformed others in training, testing, and average performance, while the Decision Tree model consistently exhibited the lowest accuracy. SVR and XGBoost performed best but with quarterly inconsistencies, and Random Forest surpassed Decision Tree as hypothesized. K-Fold Cross Validation elevated underperforming models to levels comparable with ANN and LSTM, though the Decision Tree remained the weakest. Forecast visualizations for H1 in Q1 are shown in Fig. 21. To boost LSTM's accuracy, hybrid models (PSO-LSTM, SSA-LSTM, GA-LSTM) were developed via optimization; all showed similar basic metrics, but PSO-LSTM's lower MSE suits outlier-sensitive scenarios, GA-LSTM's faster runtime favors resource-constrained environments, and SSA-LSTM is competitive without standout advantages in efficiency or error handling. Additionally, we note that the LSTM model, while often achieving higher forecast accuracy, is noticeably more computationally intensive to train than other models, which is an important consideration for large-scale industrial deployment. To balance cost and accuracy, we recommend simple mitigations such as reducing LSTM layer size, lowering sequence length, early stopping, and using ensemble surrogates for real-time inference.

This study was performed on measurements from a single SHIP system installation in Johor Bahru, Malaysia, so results could vary across different climates, plant scales, collector configurations, or energy demand and generation patterns. Future work could extend the analysis to multi-site datasets or integrate adaptive models for varying climates in order to enhance robustness and broader applicability in SHIP systems.

ACKNOWLEDGMENT

During the preparation of this work, the author(s) used ChatGPT and Perplexity AI in order to assist with coding. After using this tool/service, they reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

SUPPLEMENTARY MATERIALS

All of the codes and data used in this study are available at the GitHub Repo URL, https://github.com/Axscel/SHIP-System-Solar-Forecasting, ensuring reproducibility and transparency of our results.

REFERENCES

[1] (Oct. 2024). *World Energy Outlook 2024*. IEA, Paris, France. [Online]. Available: https://www.iea.org/reports/world-energy-outlook-2024

[2] S. N. Abdul Latif, M. S. Chiong, S. Rajoo, A. Takada, Y.-Y. Chun, K. Tahara, and Y. Ikegami, "The trend and status of energy resources and greenhouse gas emissions in the Malaysia power generation mix," *Energies*, vol. 14, no. 8, p. 2200, Apr. 2021, doi: 10.3390/en14082200.

[3] C. A. Schoeneberger, C. A. McMillan, P. Kurup, S. Akar, R. Margolis, and E. Masanet, "Solar for industrial process heat: A review of technologies, analysis approaches, and potential applications in the United States," *Energy*, vol. 206, Sep. 2020, Art. no. 118083, doi: 10.1016/j.energy.2020.118083.

[4] H. S. Min, A. Lomi, E. Okoroigwe, and L. Rodríguez-Urrego, "Investigation of solar energy: The case study in Malaysia, Indonesia, Colombia and Nigeria," *Int. J. Renew. Energy Res.*, vol. 9, no. 1, pp. 86–95, 2019, doi: 10.20508/ijrer.v9i1.8699.g7620.

[5] S. Mekhilef, A. Safari, W. E. S. Mustaffa, R. Saidur, R. Omar, and M. A. A. Younis, "Solar energy in Malaysia: Current state and prospects," *Renew. Sustain. Energy Rev.*, vol. 16, no. 1, pp. 386–396, Jan. 2012, doi: 10.1016/j.rser.2011.08.003.

[6] S. T. Mohammad, H. H. Al-Kayiem, M. A. Aurybi, and A. K. Khlief, "Measurement of global and direct normal solar energy radiation in seri iskandar and comparison with other cities of Malaysia," *Case Stud. Thermal Eng.*, vol. 18, Apr. 2020, Art. no. 100591, doi: 10.1016/j.csite.2020.100591.

[7] W. Dou, K. Wang, S. Shan, C. Li, J. Wen, K. Zhang, H. Wei, and V. Sreeram, "Evaluation of performance for day-ahead solar irradiance forecast using numerical weather prediction," *J. Renew. Sustain. Energy*, vol. 16, no. 4, Jul. 2024, Art. no. 043703, doi: 10.1063/5.0216528.

[8] M. Cenek, R. Haro, B. Sayers, and J. Peng, "Climate change and power security: Power load prediction for rural electrical microgrids using long short term memory and artificial neural networks," *Appl. Sci.*, vol. 8, no. 5, p. 749, May 2018, doi: 10.3390/app8050749.

[9] F. Otieno, N. Williams, and P. McSharry, "Forecasting energy demand for microgrids over multiple horizons," in *Proc. IEEE PES/IAS PowerAfrica*, Jun. 2018, pp. 457–462, doi: 10.1109/POWERAFRICA.2018.8521063.

[10] Z. Afroz, G. Shafiullah, T. Urmee, and G. Higgins, "Modeling techniques used in building HVAC control systems: A review," *Renew. Sustain. Energy Rev.*, vol. 83, pp. 64–84, Mar. 2018, doi: 10.1016/j.rser.2017.10.044.

[11] G. Fu, "Deep belief network based ensemble approach for cooling load forecasting of air-conditioning system," *Energy*, vol. 148, pp. 269–282, Apr. 2018, doi: 10.1016/j.energy.2018.01.180.

[12] D. Rolnick et al., "Tackling climate change with machine learning," *ACM Comput. Surv.*, vol. 55, no. 2, pp. 1–96, Feb. 2022, doi: 10.1145/3485128.

[13] H. Oukhouya and K. E. Himdi, "Comparing machine learning methods—SVR, XGBoost, LSTM, and MLP—For forecasting the Moroccan stock market," *Comput. Sci. Math. Forum*, vol. 7, no. 1, p. 39, Apr. 2023, doi: 10.3390/iocma2023-14409.

[14] D. P. Mishra, S. Jena, R. Senapati, A. Panigrahi, and S. R. Salkuti, "Global solar radiation forecast using an ensemble learning approach," *Int. J. Power Electron. Drive Syst. (IJPEDS)*, vol. 14, no. 1, p. 496, Mar. 2023, doi: 10.11591/ijpeds.v14.i1.pp496-505.

[15] P. Benalcazar and J. Kamiński, "Short-term heat load forecasting in district heating systems using artificial neural networks," in *Proc. IOP Conf., Earth Environ. Sci.*, vol. 214, Jan. 2019, Art. no. 012023, doi: 10.1088/1755-1315/214/1/012023.

[16] E. Saloux and J. A. Candanedo, "Forecasting district heating demand using machine learning algorithms," *Energy Proc.*, vol. 149, pp. 59–68, Sep. 2018, doi: 10.1016/j.egypro.2018.08.169.

[17] J. Wang, L. Guo, C. Zhang, L. Song, J. Duan, and L. Duan, "Thermal power forecasting of solar power tower system by combining mechanism modeling and deep learning method," *Energy*, vol. 208, Oct. 2020, Art. no. 118403, doi: 10.1016/j.energy.2020.118403.

[18] S. Dontu, S. R. Addula, P. Kumar Pareek, R. Vallabhaneni, and M. M. Adnan, "Attack detection from Internet of Things using TPE based self-attention based bidirectional long-short term memory," in *Proc. Int. Conf. Intell. Algorithms Comput. Intell. Syst. (IACIS)*, Aug. 2024, pp. 1–6, doi: 10.1109/iacis61494.2024.10722002.

[19] A. D'Amico, G. Ciulla, L. Tupenaite, and A. Kaklauskas, "Multiple criteria assessment of methods for forecasting building thermal energy demand," *Energy Buildings*, vol. 224, Oct. 2020, Art. no. 110220, doi: 10.1016/j.enbuild.2020.110220.

[20] R. Ahmed, V. Sreeram, Y. Mishra, and M. D. Arif, "A review and evaluation of the state-of-the-art in PV solar power forecasting: Techniques and optimization," *Renew. Sustain. Energy Rev.*, vol. 124, May 2020, Art. no. 109792, doi: 10.1016/j.rser.2020.109792.

[21] I. Jebli, F.-Z. Belouadha, M. I. Kabbaj, and A. Tilioua, "Prediction of solar energy guided by Pearson correlation using machine learning," *Energy*, vol. 224, Jun. 2021, Art. no. 120109, doi: 10.1016/j.energy.2021.120109.

[22] H. Sharadga, S. Hajimirza, and R. S. Balog, "Time series forecasting of solar power generation for large-scale photovoltaic plants," *Renew. Energy*, vol. 150, pp. 797–807, May 2020, doi: 10.1016/j.renene.2019.12.131.

[23] D. Markovics and M. J. Mayer, "Comparison of machine learning methods for photovoltaic power forecasting based on numerical weather prediction," *Renew. Sustain. Energy Rev.*, vol. 161, Jun. 2022, Art. no. 112364, doi: 10.1016/j.rser.2022.112364.

[24] W. Sun and Q. Gao, "Exploration of energy saving potential in China power industry based on AdaBoost back propagation neural network," *J. Cleaner Prod.*, vol. 217, pp. 257–266, Apr. 2019, doi: 10.1016/j.jclepro.2019.01.205.

[25] L. Liu, J. Liu, Z. Chen, Z. Jiang, M. Pang, and Y. Miao, "Research on predictive control of energy saving for central heating based on echo state network," *Energy Rep.*, vol. 9, pp. 171–181, Jun. 2023, doi: 10.1016/j.egyr.2023.02.065.

[26] S. Chaturvedi, E. Rajasekar, S. Natarajan, and N. McCullen, "A comparative assessment of SARIMA, LSTM RNN and fb prophet models to forecast total and peak monthly energy demand for India," *Energy Policy*, vol. 168, Sep. 2022, Art. no. 113097, doi: 10.1016/j.enpol.2022.113097.

[27] R. Abbasi, A. K. Bashir, A. Mateen, F. Amin, Y. Ge, and M. Omar, "Efficient security and privacy of lossless secure communication for sensor-based urban cities," *IEEE Sensors J.*, vol. 24, no. 5, pp. 5549–5560, Mar. 2024, doi: 10.1109/JSEN.2023.3305716.

[28] R. Abbasi, A. K. Bashir, A. O. Almagrabi, M. B. B. Heyat, and G. Yuan, "Efficient lossless based secure communication in 6G Internet-of-Things environments," *Sustain. Energy Technol. Assessments*, vol. 57, Jun. 2023, Art. no. 103218, doi: 10.1016/j.seta.2023.103218.

[29] M. Aiqal Iskandar, M. Azfar Shamil Abd Aziz, S. S. Sivaraju, N. Borhan, W. Abd Al-Qadr Imad Wan Mohtar, and N. Ahmad, "Long-term solar power generation forecasting in the eastern coast region of Malaysia using artificial neural network (ANN) method," *J. Adv. Res. Fluid Mech. Thermal Sci.*, vol. 117, no. 2, pp. 60–70, Jun. 2024, doi: 10.37934/arfmts.117.2.6070.

[30] M. Ali Akbar, A. Jazlan, M. Mahbubur Rashid, H. F. Mohd Zaki, M. Naveed Akhter, and A. H. Embong, "Solar thermal process parameters forecasting for evacuated tubes collector (ETC) based on RNN-LSTM," *IIUM Eng. J.*, vol. 24, no. 1, pp. 256–268, Jan. 2023, doi: 10.31436/iiumej.v24i1.2374.

[31] M. Frikha, K. Taouil, A. Fakhfakh, and F. Derbel, "Limitation of deep-learning algorithm for prediction of power consumption," *Eng. Proc.*, vol. 18, no. 1, p. 26, Jun. 2022, doi: 10.3390/engproc2022018026.

[32] P. P. Phyo, C. Jeenanunta, and K. Hashimoto, "Electricity load forecasting in Thailand using deep learning models," *Int. J. Electr. Electron. Eng. Telecommun.*, vol. 8, no. 4, pp. 221–225, Jan. 2019, doi: 10.18178/ijeetc.8.4.221-225.

[33] K. Wang, Y. Hua, L. Huang, X. Guo, X. Liu, Z. Ma, R. Ma, and X. Jiang, "A novel GA-LSTM-based prediction method of ship energy usage based on the characteristics analysis of operational data," *Energy*, vol. 282, Nov. 2023, Art. no. 128910, doi: 10.1016/j.energy.2023.128910.

[34] W. Lu, H. Rui, C. Liang, L. Jiang, S. Zhao, and K. Li, "A method based on GA-CNN-LSTM for daily tourist flow prediction at scenic spots," *Entropy*, vol. 22, no. 3, p. 261, Feb. 2020, doi: 10.3390/e22030261.

[35] H. Zhen, D. Niu, K. Wang, Y. Shi, Z. Ji, and X. Xu, "Photovoltaic power forecasting based on GA improved bi-LSTM in microgrid without meteorological information," *Energy*, vol. 231, Sep. 2021, Art. no. 120908, doi: 10.1016/j.energy.2021.120908.

[36] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle swarm optimization: A comprehensive survey," *IEEE Access*, vol. 10, pp. 10031–10061, 2022, doi: 10.1109/ACCESS.2022.3142859.

[37] J. Nayak, H. Swapnarekha, B. Naik, G. Dhiman, and S. Vimal, "25 years of particle swarm optimization: Flourishing voyage of two decades," *Arch. Comput. Methods Eng.*, vol. 30, no. 3, pp. 1663–1725, Dec. 2022, doi: 10.1007/s11831-022-09849-x.

[38] X. Ren, S. Liu, X. Yu, and X. Dong, "A method for state-of-charge estimation of lithium-ion batteries based on PSO-LSTM," *Energy*, vol. 234, Nov. 2021, Art. no. 121236, doi: 10.1016/j.energy.2021.121236.

[39] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017, doi: 10.1016/j.advengsoft.2017.07.002.

[40] L. Abualigah, M. Shehab, M. Alshinwan, and H. Alabool, "Salp swarm algorithm: A comprehensive survey," *Neural Comput. Appl.*, vol. 32, no. 15, pp. 11195–11215, Nov. 2019, doi: 10.1007/s00521-019-04629-4.

[41] L. Jovanovic, D. Jovanovic, N. Bacanin, A. Jovancai Stakic, M. Antonijevic, H. Magd, R. Thirumalaisamy, and M. Zivkovic, "Multi-step crude oil price prediction based on LSTM approach tuned by salp swarm algorithm with disputation operator," *Sustainability*, vol. 14, no. 21, p. 14616, Nov. 2022, doi: 10.3390/su142114616.

[42] O. Chamorro-Atalaya, J. Arévalo-Tuesta, D. Balarezo-Mares, A. Gonzáles-Pacheco, O. Mendoza-León, M. Quipuscoa-Silvestre, G. Tomás-Quispe, and R. Suarez-Bazalar, "K-fold cross-validation through identification of the opinion classification algorithm for the satisfaction of university students," *Int. J. Online Biomed. Eng. (iJOE)*, vol. 19, no. 11, pp. 1–19, Aug. 2023, doi: 10.3991/ijoe.v19i11.39887.

ABDURRAHMAN M. MANU received the Bachelor of Technology degree in engineering with a specialization in mechatronics from Sharda University, India. He is currently pursuing the Master of Science degree in mechatronics with International Islamic University Malaysia (IIUM). His research interests and expertise include machine learning and artificial intelligence, renewable energy systems, and the integration of intelligent systems for forecasting.

AHMAD JAZLAN received the Ph.D. degree in control systems engineering from The University of Western Australia. He is currently an Associate Professor at the Mechatronics Engineering Department of the Faculty of Engineering, International Islamic University Malaysia (IIUM). His expertise extends to an industrial attachment with Texas Instruments Malaysia, where he focuses on enhancing traceability in legacy machines and forecasting die bond daily output. His research interests include smart materials, renewable energy, biomedical instrumentation, signal processing, control theory, and machine learning. He is a member of The Institution of Engineers, Malaysia (IEM). He is a registered Professional Engineer of the Board of Engineers Malaysia (BEM).

MUHAMMAD NAVEED AKHTER received the B.Sc. degree in electrical engineering from UET Taxila, in 2005, the M.Sc. degree from UET Lahore, in 2012, and the Ph.D. degree from the University of Malaya, in 2021. He has been an Associate Professor with the Electrical Engineering Department, Rachna College of Engineering and Technology (A Constituent College of University of Engineering and Technology, Lahore), Gujranwala. He has expertise in the performance of PV systems, forecasting of solar power output, neural networks, deep learning, and machine learning techniques.

MUHAMMAD DANIAL is currently pursuing the Bachelor of Engineering degree (Hons.) in mechatronics with International Islamic University Malaysia (IIUM). His research interests and expertise include mechatronics, control systems, and energy forecasting.

AZHAR MOHD IBRAHIM (Member, IEEE) received the Ph.D. degree from the School of Computer Sciences, Universiti Sains Malaysia (USM), under the research theme of collective intelligence. He is currently an Associate Professor at International Islamic University Malaysia (IIUM). His research focuses on computational intelligence, reinforcement learning, collective intelligence, and remote sensing. His research aims to develop models for intelligent agents and robots and to analyze the coordination between them, with a specific focus on addressing the challenges of building emergency response and navigation systems. He has published articles in reputed high-impact journals, executed several research grants, and supervised Ph.D. scholars.

HASAN F. M. ZAKI (Member, IEEE) received the Ph.D. degree in computer science from The University of Western Australia. He is currently an Associate Professor with the Department of Mechatronics Engineering, International Islamic University Malaysia (IIUM), and the Director of the Centre for Unmanned Technologies (CUTe). He is also a Certified Professional Technologist. His research interests include robotic vision, RGB-D object and scene recognition, machine learning, 3D face analysis, action recognition, and embedded AI systems, with applications on autonomous robotics, intelligent surveillance, and UAV-based sensing systems. His contributions drive advancements in human–robot interaction, AI-driven perception, and real-time decision-making for smart automation and autonomous platforms. Beyond academia, he actively collaborates with industries and government agencies as a Consultant and a Subject Matter Expert, bridging the gap between cutting-edge research and real-world applications.

VICTOR SREERAM (Life Senior Member, IEEE) received the bachelor's degree from Bangalore University, India, in 1981, the master's degree from Madras University, India, in 1983, and the Ph.D. degree from the University of Victoria, BC, Canada, in 1990, all in electrical engineering. He was a Project Engineer with Indian Space Research Organization, from 1983 to 1985. He joined the Department of Electrical, Electronic, and Computer Engineering, The University of Western Australia, in 1990, as a Lecturer. He was a Graduate Research Coordinator, from 2008 to 2017. He is currently a Professor and the Head of Department. He has authored over 250 publications, including over 125 journal publications. He has successfully supervised over 25 Ph.D. students and has examined over 50 Ph.D. theses in his career. His teaching and research interests include control systems, signal processing, communications, power engineering, smart grid, and renewable energy. He is a fellow of the Institution of Engineers, Australia. He was a co-recipient of the 2012 Premium Award for Best Paper in *IET Power Electronics* and the 2014 Premium Award for Best Paper in *IET Generation, Transmission and Distribution*. He was also a recipient of the 2013 Outstanding Reviewer for IEEE Transactions on Automatic Control Award. He has been the Chairman of Australia New Zealand Control Conference Steering Committee Association Incorporated, since 2017. He is an Associate Editor of *IET Control Theory and Applications* and *Asian Journal of Control*. He was the Vice President of the Asian Control Association for Technical Activities.