SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V10I9P121

Volume 10 Issue 9, 221-231, September 2023


*Original Article*

# Improving Energy Demand Prediction in IoT Based Smart Grids through Hybrid CNN-LSTM Modelling with Modified Sea Lion Algorithm

S. Sivarajan[1], S. D. Sundar Singh Jebaseelan[2]

[1]*Department of Electronics and Communication Engineering, Sathyabama Institute of Science and Technology, Chennai, Tamilnadu, India.*
[2]*Department of Electrical and Electronics Engineering, Sathyabama Institute of science and Technology, Chennai, Tamilnadu, India.*

[1]*Corresponding Author : ssivarajan152@gmail.com*

Received: 28 July 2023 Revised: 30 August 2023 Accepted: 20 September 2023 Published: 30 September 2023

***Abstract*** *- Efficient energy demand forecasting is pivotal for the reliable operation of modern IoT based smart grids, ensuring optimal resource allocation and grid stability. This study introduces a novel approach that combines Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTM) networks within a hybrid framework for accurate energy demand prediction. An innovative modification of the Sea Lion Algorithm (SLA) is proposed to enhance the model's performance for optimal hyperparameter tuning. The hybrid CNN-LSTM architecture leverages the strengths of CNNs in feature extraction from sequential data and LSTM's proficiency in capturing temporal dependencies. By synergizing these capabilities, the model offers improved accuracy in predicting energy demand patterns, which is critical for effective energy management and distribution. The Modified Sea Lion Algorithm (MSLA) is employed to fine-tune the hybrid CNN-LSTM model's hyperparameters effectively. Inspired by the behaviour of sea lions in balancing exploration and exploitation during foraging, MSLA ensures an optimal configuration of model parameters, leading to enhanced forecasting accuracy. Extensive experiments use real-world energy consumption datasets to assess the proposed methodology's efficacy. Comparative analyses are conducted against conventional CNN-LSTM models with default settings, highlighting the superiority of the hybrid approach. The results demonstrate that integrating CNNs and LSTMs yields more accurate predictions, while the modified Sea Lion Algorithm provides optimal parameter values, further enhancing prediction accuracy.*

***Keywords*** *- IoT, Hybrid CNN-LSTM, Modified Sea Lion Algorithm, Energy demand prediction, Smart grid.*

## 1. Introduction

The Internet of Things (IoT) has rendered it possible to observe and operate cyber-physical structures such as smart transportation, cities and grids. Many countries' governments are developing smart city programmes to improve the use of natural resources such as energy and water [1, 2]. The expanding population necessitates an increase in energy consumption. New energy sources are being merged with traditional ones to satisfy the worldwide need for energy.

The impact of warming temperatures has necessitated a move from fossil fuels to sustainable energy [3]. The carbon footprint of electricity produced by coal-fired power plants rises. Energy generated from solar and wind is introduced to the current electrical grid to fulfil demand. This has assisted in reducing the carbon impact; thus, renewable energy gathering should be expanded [4].

The expense of the energy produced is significant; therefore, it has to be used properly. The tracking of energy use is improved by implementing smart metering technology. Smart meters can help utilities comprehend how every user uses electricity and consumers [5]. Real-time monitoring of the power grid is made possible by advances in computer technology like cloud computing, fog computing, and communications technologies like 4G and 5G. The limits of the currently conceivable are being challenged by these technologies, which are constantly developing [6, 7]. An approach known as edge computing brings analytics nearer to the system, lowering computational latency. These distributed computing approaches boost the effectiveness of monitoring and controlling globally dispersed systems [8]. Smart houses can track and manage appliances by putting in sensors and devices. Smart houses are built to use energy efficiently, lowering consumption costs. Most homes are transitioning to being consumers of power; automated metering infrastructure helps to acquire detailed information so that Time of Use (ToU) based dynamic pricing of consumers is achievable [9].

Excellent work has been done regarding security for the IoT-enabled smart grid. There are a few issues with the existing literature, still that have not been tackled. Local area network setup employing the smart grid as the network's communication foundation is described in [10]. For use in field area networks, the developed system can be expanded to include more than one sensor node for improved accuracy and in the event of node breakdown. However, using several sensor nodes to collect data causes prediction oversights. The reliability of a smart grid can be predicted using a multidirectional LSTM model, as mentioned in [11, 12].

The results of experiments demonstrate that the MLSTM technique works better than the other ML strategies. However, the context-aware model must be adopted to make smart grids more dependable. The deep convolutional neural networks for electricity theft identification for securing smart grids are described in [13]. Numerous investigations on actual datasets demonstrate that the Wide & Deep CNN model superiors other accessible approaches.

However, picking a greater epoch value may result in overfitting. Hybrid DNN is proposed in [14, 15] to identify non-technical losses in electricity smart meters. The proposed hybrid neural network architecture can surpass the accuracy of highly potent classifiers. However, the issue of accurate detection has not yet been adequately solved. An ensemble approach For PV generation power day-ahead forecasting in smart grids is given in [16]. This structure is employed to improve the efficiency of facilities' demand-side control and power comfort systems. However, the training data standard may affect the forecast outcome due to unusual weather conditions. In [17], an encrypted federated deep learning solution is given for detecting malicious information injection threats in smart grids. Compared to centralized detection approaches, it features privacy protection for data and lower connection latency. However, the capabilities of this system are only adequate for handling the detection of several cyber-attacks.

Consequently, the security issues raised by deep learning are examined in this paper on the IoT-integrated smart grid. This paper proposes a reliable and strong hybrid CNN-LSTM system based on IoT to handle the smart grid security challenge. The modified sea lion algorithm has been presented here to improve outcomes from the hybrid CNN-LSTM. This method fine-tunes the CNN-LSTM approach's hyperparameters for enhanced prediction of energy demand patterns, which is essential for efficient energy management and distribution.

The remainder of the paper is structured as follows. The proposed system is described and modelled in Section II. Section III discusses the proposed system's outcomes. Section IV offers the conclusion of the article.

## 2. Proposed System Description

Figure 1 depicts an Internet of Things-based structural design for a power distribution management system in which smart meters communicate with the router. Smart appliances are monitored and controlled in the event of energy metering.

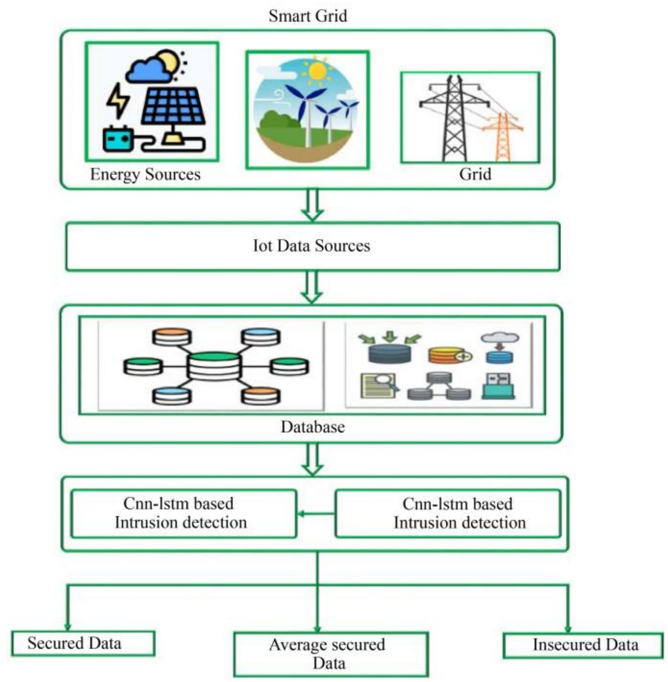


**Fig. 1 Proposed system architecture**

The receiver system gathers the data, which is subsequently authorized to protect the data from attackers. Making smart grid choices based on user consumption and preferences requires analysis of the encrypted information. The data's computing findings are sent back through the home area network for efficient utilization. A demand-side management security intrusion solution based on CNN and LSTM is applied in an IoT-based smart grid. Finally, it efficiently classifies the secured, secured and average secured data from the IoT database with the assistance of MSLO.

### *2.1. IoT Based Smart Grid*

In power systems, the widespread use of clean energy sources is becoming an increasingly significant source of uncertainty. As demand response resources, loads are becoming more involved, strongly correlated with the fluctuating regular electricity costs. The price of electricity is also associated with several factors, including the organization of the power market and current fuel price. Demand side medium, or Virtual Power Plants Sources (VPPs), is expected to be widely used shortly. In microgrids, the operator must cope with a high level of fluctuation and uncertainty as well as the grid's current limits, which in some situations may force load shedding or restriction. IoT technology can make it easier to address the problems and difficulties needed to avoid such activities and maintain the ecological power system's stability, security reliability and sustainability. Smart grids supported by IoT enable operators to have more thorough grid supervision by automatically and precisely tracking all variations and developments on both the supply and demand sides.

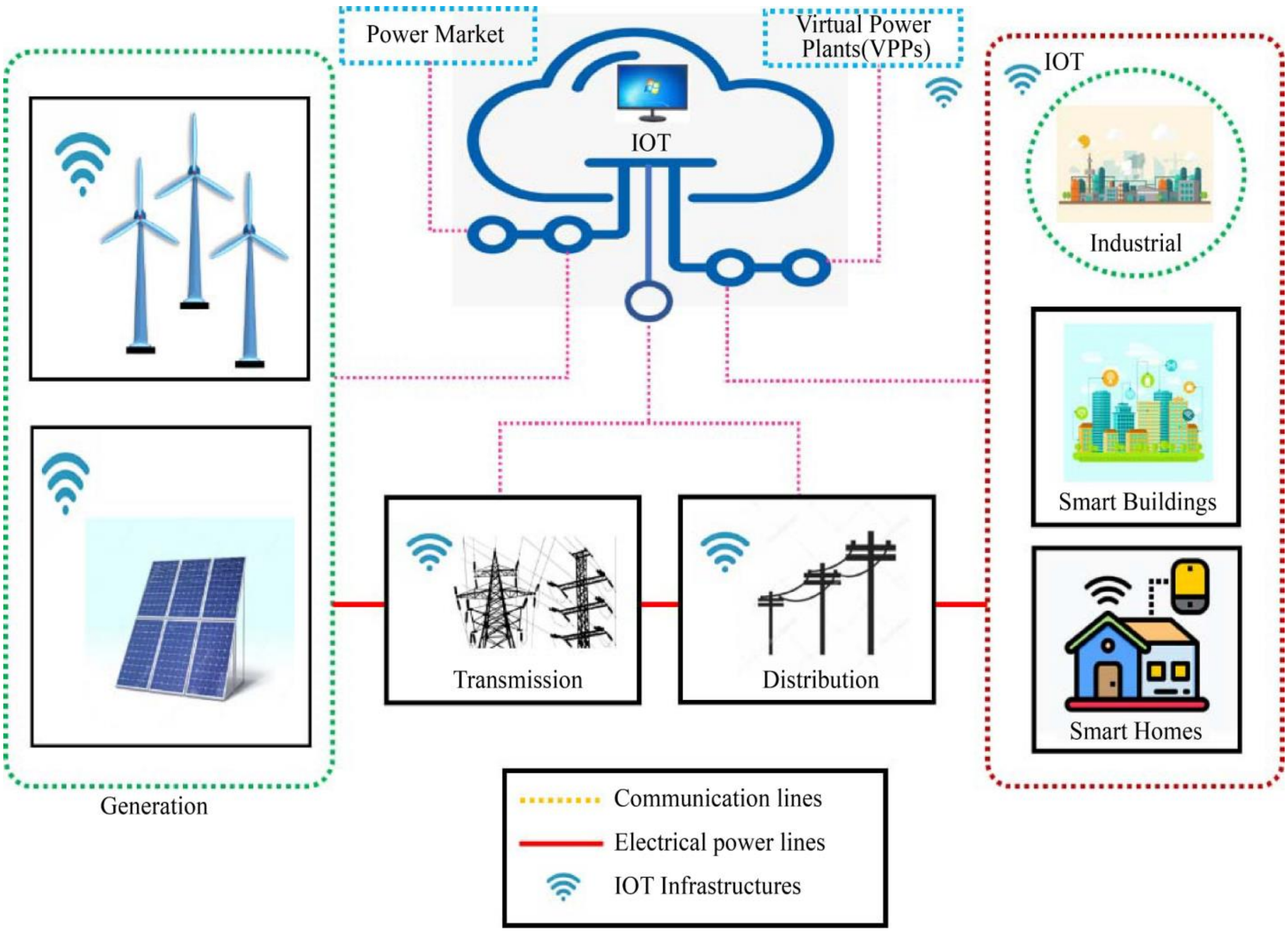


**Fig. 2 IoT based smart grid**

The combination of several energy resources, including coal, oil, gas, nuclear, and hydropower, as well as alternative forms of energy, including wind and solar, is covered by IoT technologies at the production level in a way to enhance the operation of the electricity sector as well as preserve the static and dynamic stability of the power system. Figure 2 represents the concept of IoT for the smart grid.

### *2.2. Hybrid CNN-LSTM Based Intrusion System*

This research offers a hybrid CNN and LSTM network architecture to estimate energy demand. The hybrid CNN-LSTM architecture uses both the capabilities of CNNs and LSTM to capture time-dependent relationships and extract features from sequential input. These characteristics work together to increase the model's ability to predict energy demand trends, which is essential for adequate energy utilization and distribution. The proposed CNN-LSTM intrusion system efficiently prevents the data from security cyber-attacks. A detailed explanation of the proposed CNN-LSTM is given in the below section.

#### *2.2.1. Hybrid CNN-LSTM Approach*

The CNN model can process one piece of data at a time, turning the input pixels into a matrix within the network's frame. This process uses a variety of data sets to allow an LSTM to establish a fundamental nature and alter weights using the (Backpropagation training technique) BPTT during multiple iterations of the virtual vector representations of input data. If a classifier is trained from frames, the CNN can be standardized. If the CNN is untrained, retrain it using back propagation faults from LSTM over various input data to design the CNN.

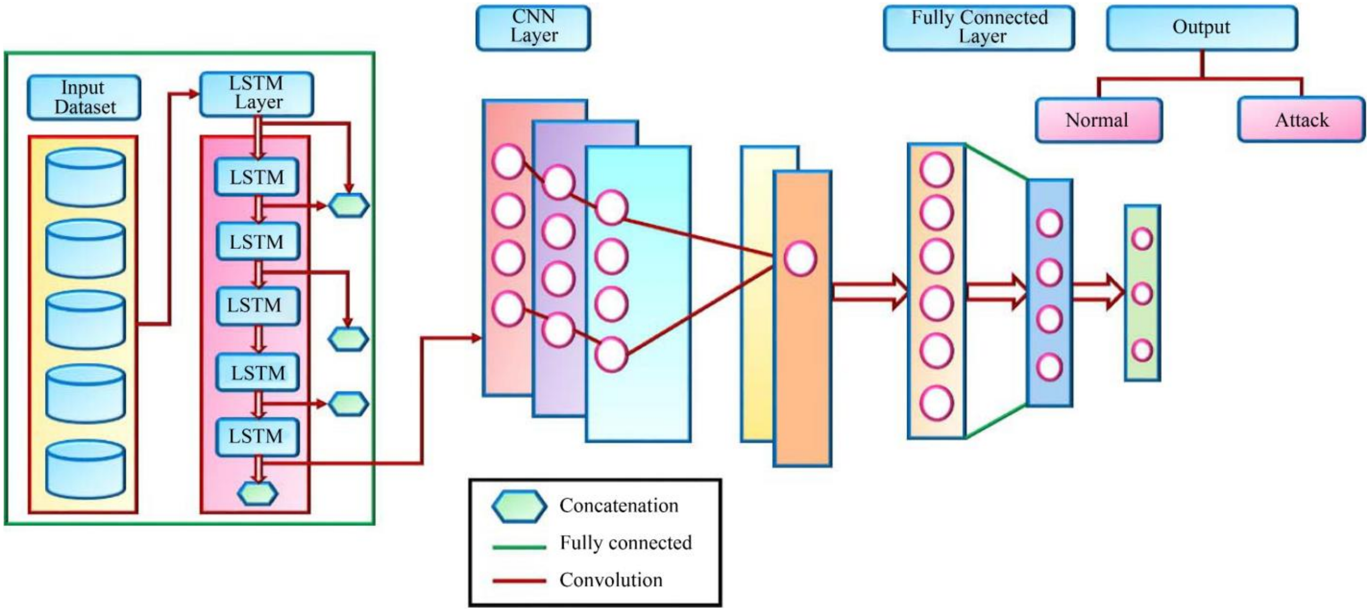


**Fig. 3 Proposed hybrid CNN-LSTM architecture**

Figure 3 depicts the CNN-LSTM model's structure of the proposed system. In the convolution layer, an enhanced attribute description is achieved. As seen in Eq. 1, the convolution process is outlined. The convolution layer as variable $i$ map is represented by the $X_i{}^a$ that is being displayed below.

The function of activation is represented by the $b_j^a$ . The input layer of the attribute set $(a-1)$ is called. $k_i$. The $i$ of the attribute convolution layer $a$ and $j$ of the attribute layer $(a-1)$ are connected by the connection weight. $w_{ji}^a$. The divergence in the relevant layer is called. $b_j^a$.

$$X_i{}^a = \emptyset\left[\sum_{i \epsilon k_i} x_j^{a-1} * w_{ji}^a + b_j^a\right] \tag{1}$$

The pooling layer follows the convolution layer. This layer aims to minimize the attribute map size. Through this process, essential characteristics are identified, data complexity is decreased, and the network's resistance to environmental changes is increased. As seen in Equation 2, the pooling layer is given by.

$$x_i^a = \emptyset[\beta_i^a(x_i^{a-1} + b_i^a)] \tag{2}$$

The sub-sampling function is represented by $c$ in this example, while the weighting matrix is defined by $\beta$. After the convolution and pooling layers, the fully connected layer performs the classification function. The resultant process of the ultimately linked layer is shown in Equation 3.

$$y^m = \emptyset[w^m x^{m-1} + b^m] \tag{3}$$

Here, the symbol $m$ stands for the layer index, whereas the symbols $y^m$ and $x^{m-1}$ represent the input and output of a fully connected layer, variance and weighting factor, respectively. The recurrent networks' hidden units are replaced by LSTM cells, which also feature recurrent interconnections. In the LSTM block, the variables $x^t$ and $h_{t-1}$ stand for the vector input time step $t$, time step hidden state $(t-1)$ and time step memory cell state $(t-1)$, respectively. These make up the block's inputs. Gates for input, forget, and output are included in the LSTM.

The mathematical formulas below outline the computations for the LSTM's cell state, input, forget and output gates. As a result, an activation function of sigmoid $(\sigma)$ in Equation 4 selects which data is possible to obtain via or not based on the forget gate $f_i^{(t)}$ for cell $i$ at time step $t$. For the forget gates $b^f$, $Z^f$ and $D^f$ Stands for deviation, input weight, and recurring weights correspondingly.

$$f_i^{(t)} = \sigma(b_i^f + \sum jz_{i,j}^f\, x_j^{(t)} + \sum jD_{i,j}^f h_j^{(t-1)}) \tag{4}$$

While $b, Z,$ and $D$ stand for deviation, input weight, and recurrent weights reaching the LSTM cell, correspondingly, Equation 5 shows the modification in the cell state. $n_i^{(t)}$. The computation of the cell input gate $p_i^{(t)}$is shown in Equation 6, and this analysis is carried out similarly to how the forget gate is calculated.

$$n_i^{(t)} = f_i^{(t)} n_i^{(t-1)} + p_i^{(t)} \sigma(b_i + \sum jz_{i,j} x_j^{(t)} + \sum jD_{i,j} h_t^{(t-1)} \tag{5}$$

$$p_i^{(t)} = \sigma(b_i^p + \sum jz_{i,j}^p x_j^{(t)} + \sum jD_{i,j} h_j^{(t-1)}) \tag{6}$$

$$h_i^t = \tanh(n_i^{(t)}) s_i^{(t)} \tag{7}$$

$$s_i^{(t)} = \sigma(b_i^0 + \sum jz_{i,j}^0 x_j^{(t)} + \sum jD_{i,j}^0 h_j^{t-1}) \tag{8}$$

**Algorithm 1:** Hybrid CNN-LSTM
1: $Input: Train\_X, Train\_Y$
2: $Hyper\ Parameters: optimizer, rate, feature\_layers, poolsize, batchsize$
3: $Initialize()$
4: $Normalization(Train\_X, Train\_Y)$
5: $Convolution\_1 = Sequential\ ((Convolution\ 2D(optimizer, dro$
$pout, name = "Conv2D\_1"), MaxPooling2D(poolsize), dropout\ (rate))$
6. $Convolution\_1.compile(Train\_X, Train\_Y, epochs, batchsize)$
7. $Convolution\_1.fit(Train\_X, Train\_Y, epochs, batchsize)$
8. $Convolution\_1\_feature = Model(inputs, convolution\_1("Con\ volution2D: ).output)$
9. $Convolution\_1\_feature.predict(Trai\ n\_X)$
10. $Lstmmodel$
$= Sequential(Lstm\ (units, activation, recurrent\_activation), flatten\ (units, activation)$
11. $Lstmmodel.compile(lossfunction, optimizer))$
12. $Lstmmodel.fit(Convolution\_1\_feature, Train\_Y, batchsize, epochs))$

In Equation 7, $h_i^t$ stands for the hidden state and $s_i^{(t)}$ for the output gate. Eq. 8 illustrates the result of the gate equation, where $b^0$, $Z^0$ and $D^0$ stand for input weight, deviation and recurrent weights, correspondingly. Algorithm 1 comprises the CNN + LSTM models' pseudocode.

Hyperparameters are particular weights or variables that regulate an algorithm's learning process. Generally, CNN provides an extensive range of hyperparameters. Modifying CNN's hyperparameters makes obtaining the best value out of its functionality achievable. There are going to be numerous significant hyperparameters and additional choices for design. These established variables are manually added to the algorithm throughout the training process. The Modified Sea Lion Optimisation Algorithm (MSLO) has been employed in the present research to tune the hyperparameters. The optimal parameters are more accurately selected with the help of this MSLO, which is explained in the following portion.

### *2.3. Modified Sea Lion Optimization Algorithm (MSLO)*

SLO has been created to address large-scale optimization. It imitates sea lions' hunting techniques, such as circling and grabbing prey or employing their tail and whiskers. SLO can produce outcomes comparable with other recognized particle swarm optimization algorithms when applied to various benchmark functions. First, utilizing an identical random distribution in searching space, SLO creates N (the population's size) D-dimensional solutions Equation 9 as follows. Prey is regarded as the best existing solution or the one that comes near the ideal solution. These actions are shown in Equation 10.

$$X_{i,j}^{init} = X_{i,j}^{min} + rand_{i,j}(X_{i,j}^{max} - X_{i,j}^{min}) \tag{9}$$

where $i = 1, 2, N$, and j=1, 2, D, $X_{i,j}^{init}$ is the beginning location vector of the $ith$ solution; $X_{i,j}^{min}$ and $X_{i,j}^{max}$ are the minimum and maximum values, respectively, for the $jth$ dimension of the $ith$ solution, and $rand$ is a uniform random number in the range [0, 1].

The objective function is used to assess the fitness of solutions.

$$X^{g+1} = X_{best} - C|2rX_{best} - X^g| \tag{10}$$

$$C = 2(1 - \frac{g}{gmax}) \tag{11}$$

Where, $X_{best}^g$ is the optimal solution's location vector; $X^g$ is the SLA in iteration $g$, $g$ is the present generation, and $gmax$ is the maximum amount of generations which may occur. $r$ is a random number between [0, 1], which is divided by 2 to widen the scope of the searching operation; $X^{g+1}$ represents the updated search agent's updated position; During the iteration, the variable $C'$s values declined linearly from 2 to 0, representing the sea lions' encircling behaviour as they approached and surrounded their prey.

The sea lion is the group's leader, whose actions will direct the group's movements and determine its behaviour. These behaviour are represented mathematically in Equations 12, 13 and 14.

$$SP_{leader} = |(V_1(1 + V_2)/V_2| \tag{12}$$

$$V_1 = \sin(\theta) \tag{13}$$

$$V_2 = \sin(\emptyset) \tag{14}$$

Where $\theta$ is a reflection of voice angle in the water; $\emptyset$ is refraction voice of angle in the water; $SP_{leader}$ is illustrating the choice of leader followed by other sea lions in the group;

In Equation 15, where m is a random value between [-1, 1], sea lions hunt the bait ball of fish and begin their hunt at the edges.

$$X^{g+1} = X_{best} + \cos(2\pi m)\,|X_{best} - X^g| \tag{15}$$

Equation 16 illustrates the procedure for choosing an unknown agent and the circumstance that permits the exploitation phase to occur when the value of $C$ exceeds 1.

$$X^{g+1} = X^{g}_{rand} - C|2rX^{g}_{rand} - X^{g}| \tag{16}$$

Where the sea lion chosen at random from the present population is $X^{g}_{rand}$. The value of $r$ is chosen randomly from [0, 1]. However, conventional SLO has noticeable issues with nature-inspired algorithms, such as delayed convergence and being stuck in local optima. In contrast to the original SLO, the exploitation and exploration phases for ISLO are modified in this research.

$$dif_1 = (2r_1X^{g}_{best} - X^{g}) \tag{17}$$

$$dif_2 = (2r_1X^{g}_{local} - X^{g}) \tag{18}$$

$$X^{g+1} = X^{g} + C.dif_1 + C.dif_2 \tag{19}$$

Where, $X^{g}_{local}$ local represents the user's optimal location up to iteration $g$; $r1, r2$ represent random values between 0 and 1;

This case limits the algorithm's ability to exploit multidimensional space, where the actual global best solution can be hidden in the opposite direction of the present global best answer. Eq. 10 is modified to solve this issue by removing the minus sign and absolute function. The local best represents the $X_{best}$ point so far discovered, while the red star is the global optimum point, according to Eq. 20. This enhances exploitation at first.

$$X^{g+1} = X_{best} + C\ N(0,1)(2r_3X_{best} - X^{g}) \tag{20}$$

Construct an opposite solution by using $x_{best}$ to determine $X^{g+1}$ oppositional position, you may calculate $X^{g+1}_{oppo}$.

$$X^{g+1}_{oppo} = LB + UB - x_{best} + r_4(X_{best} - X^{g+1}) \tag{21}$$

**Algorithm 2** Modified Sea Lion Optimization (MSLO)
**Input:** Population size $N$, the maximum number of generations $g_{max}$
**Output:** The best solution $X_{best}$

```
Initialize the Sea Lion population X_i (i = 1,2, … . , n) randomly
Sort the population by its fitness value and find the global solution X_best
g = 1
while g < g_max do
   Calculate the value of C by Eq.11
   Calculate SP_leader using Eq.10
    for i<N do
         if SP_leader < 1.0  then
             if |C|>1 then
                   Calculate dif_1 and dif_2 using Eq.17 and Eq.18
                    Update the location of the current search agent using Eq: 19
                 else
                   Create a new solution using Eq.20
                   Create its opposite solution X_oppo^{g+1} using Eq.21.
                   Calculate fitness of both solution
                   Compare and keep the location of the better one as the new position for current individual.
          end if
   else
     if rand() < 0.5 then
         Update the location of the current search agent by Eq: 23
     else
        Update the location of the current search agent
        by Levy-flight Eq.23
     end if
   end if
   Check the bound and calculate the fitness of the new solution.
   Replace the old solution by the new one if it has a better fitness value
 end for
 Sort the population by its fitness values and
   Update the global best solution X_best
g = g + 1
end while
Return: X_best
```

By that justification, an extra operation employing the Levy-Flight Trajectory (LFT) is suggested for MSL during the sea lions' circling phase. It contributes to the population's increased diversity and capacity for local exploitation. The Levy step size is typically represented as:

$$Levy(s) \sim |s|^{-1-\beta} \quad with \quad 0 < \beta \leq 2 \qquad (22)$$

The goal of employing the Levy-flight method for SLO is to improve local exploitation variety and the ability to locate global optima by its intricate trajectory. As a result, the following is the preferred Levy-flight updated equation.

$$X^{g+1} = X_{best} + ss\ Levy(s) \otimes (X_{beest} - X^{g}) \qquad (23)$$

The hybrid CNN-LSTM model's hyperparameters are effectively tuned using MSLO. By ensuring that model parameters are configured optimally, MSLA improves the precision of forecasting.

## 3. Results and Discussion

This research offers a dependable and robust hybrid CNN-LSTM system based on IoT to tackle the smart grid security dilemma. The MSLO algorithm is proposed here to enhance the hybrid CNN-LSTM performance. This approach optimizes the CNN-LSTM approach's hyperparameters for improved estimation of energy demand patterns, which is critical for successful energy administration and distribution. The obtained results are presented in the below section.

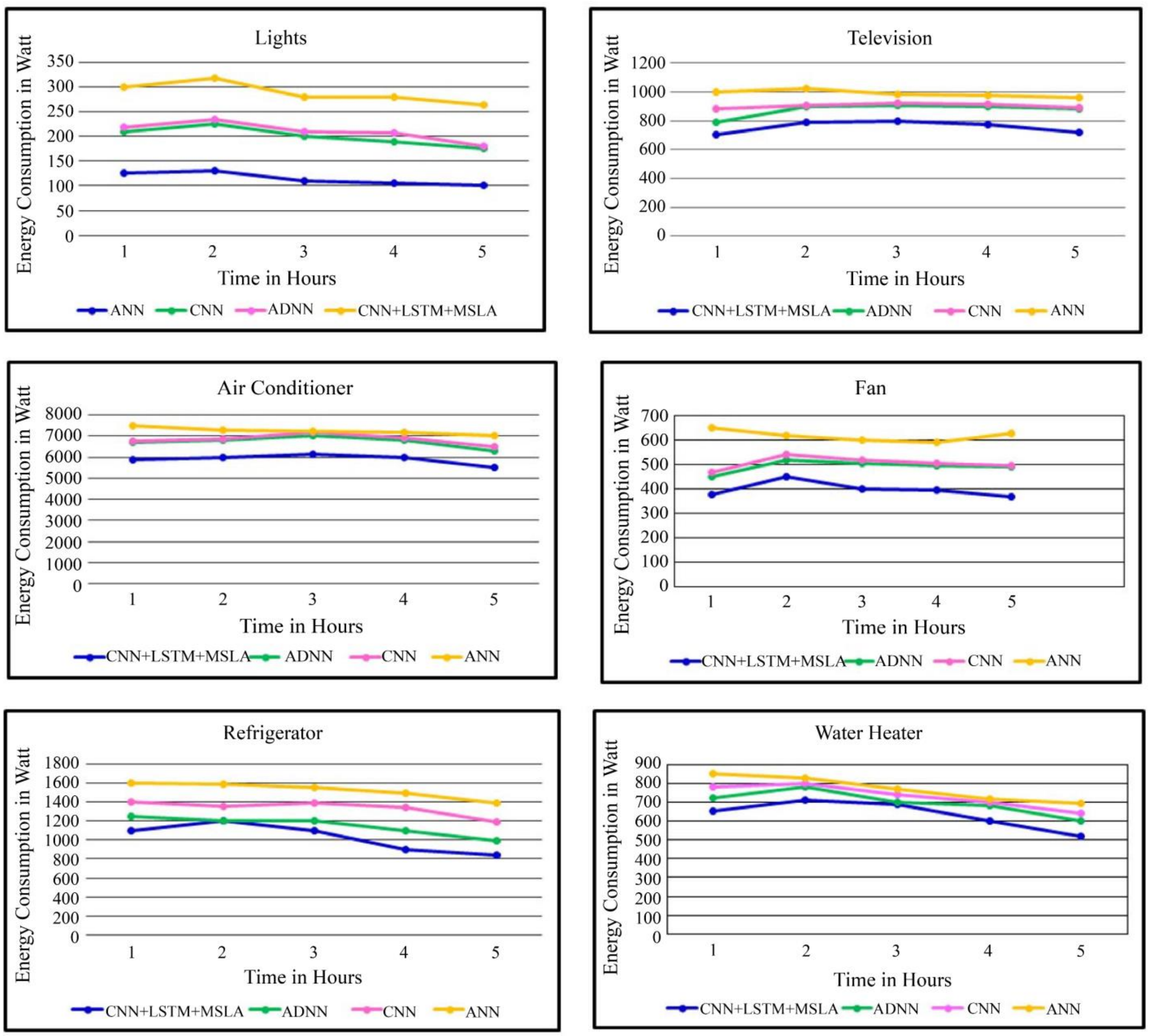


**Fig. 4 Energy consumption comparison**

Figure 4 depicts the proposed approach, as well as existing methods such as ANN (Artificial Neural Network), CNN, and ADNN (Adaptive Deep Neural Network), used to analyze the energy usage of IoT based devices, televisions, lights, air conditioners, refrigerator, fan and water heater. The essential factors for consideration are the consumer's desires and needs. The proposed system turns electronic devices on and off based on consumer needs and demand, reducing energy waste.

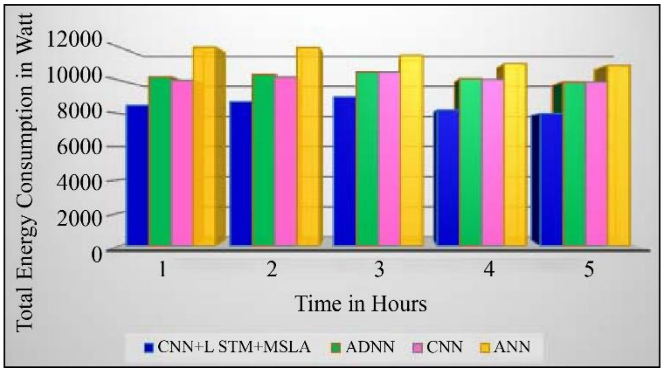


**Fig. 5 Total energy consumption comparison**

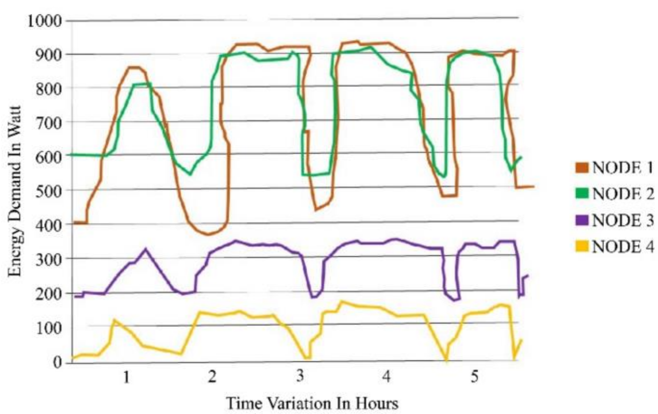


**Fig. 6 Energy demand at nodes**

The overall energy usage of all home appliances with ADNN, CNN, and ANN is depicted in Figure 5. Compared to other approaches; the proposed MSLA-optimized CNN-LSTM attains less energy consumption in smart home appliances. The proposed system minimizes energy usage because the Automated Connection structure maintains effective energy utilization. Furthermore, Figure 6 displays the energy requirement of various nodes at different periods.

Experiments have been conducted in this part to examine 4 typical performance metrics for time series forecasting, namely MSE, RMSE, MAE, and MAPE, in addition to the processing time of the experimental approaches. The initially chosen metric is MSE, which calculates the mean of the error squares. In simple terms, it is the average squared variance among expected and observed values. The MSE equation is as follows.

$$MSE = \frac{1}{n}\sum_1^n (y - \hat{y})^2 \tag{24}$$

Furthermore, the standard deviation of errors in forecasting is denoted by RMSE. First, consider residuals that indicate the distance the data points are from the regression line. As an outcome, RMSE is a measurement of the way dispersed these residuals appear.

This metric is frequently employed in environmental science, forecasting, and regression analysis to validate experimental models and is calculated as follows.

$$RMSE = \sqrt{\frac{1}{n}\sum_1^n (y - \hat{y})^2} \tag{25}$$

Meanwhile, MAE computes the average size of the prediction errors while ignoring their directions. It is the mean of the absolute differences between the actual and predicted outcomes for every scenario in the testing set. It must be noted that this evaluation gives equal weight to all individual variances. The following equation is used to calculate MAE.

$$MAE = \frac{1}{n}\sum_1^n |y - \hat{y}| \tag{26}$$

The final metric, MAPE, measures the precision of a predicting approach, including time-series forecasting. The subsequent equation represents precision in percentage for this metric:

$$MAPE = \frac{100\%}{n}\sum_1^n \left|y - \frac{\hat{y}}{y}\right| \tag{27}$$

To demonstrate the efficacy of the proposed CNN+LSTM+MSLA approach, this part analyses the RMSE, MSE, MAE, and MAPE calculated using Equations (24)-(24) of LSTM [18], Linear Regression [18] and CNN-LSTM [18] and the proposed model for minutely, hourly, daily, and weekly observations.

**Table 1. Experimental approach effectiveness for minute datasets**

| No | Model | MSE | RMSE | MAE | MAPE | Predicting Time (s) | Training Time (s) |
|---|---|---|---|---|---|---|---|
| 1 | CNN+LSTM+MSLA | 0.034 | 0.200 | 0.084 | 12.67 | 40.85 | 3867 |
| 2 | CNN-LSTM [18] | 0.374 | 0.611 | 0.349 | 34.84 | 62.99 | 2070 |
| 3 | LSTM [18] | 0.748 | 0.865 | 0.628 | 51.45 | 114.26 | 6880 |
| 4 | Linear Regression [18] | 0.405 | 0.636 | 0.418 | 74.52 | 37.48 | 1028 |

**Table 2. Experimental technique effectiveness for hourly dataset**

| No | Model | MSE | RMSE | MAE | MAPE | Predicting Time (s) | Training Time (s) |
|---|---|---|---|---|---|---|---|
| 1 | CNN+LSTM+MSLA | 0.198 | 0.438 | 0.289 | 28.05 | 1.50 | 1345.33 |
| 2 | CNN-LSTM [18] | 0.355 | 0.596 | 0.349 | 32.83 | 2.31 | 820.70 |
| 3 | LSTM [18] | 0.515 | 0.717 | 0.526 | 44.37 | 5.95 | 2281.50 |
| 4 | Linear Regression [18] | 0.425 | 0.652 | 0.502 | 83.74 | 2.88 | 692.12 |

**Table 3. Experimental technique effectiveness for daily dataset**

| No | Model | MSE | RMSE | MAE | MAPE | Predicting Time (s) | Training Time (s) |
|---|---|---|---|---|---|---|---|
| 1 | CNN+LSTM+MSLA | 0.050 | 0.167 | 0.146 | 20.15 | 0.54 | 55.16 |
| 2 | CNN-LSTM [18] | 0.104 | 0.322 | 0.257 | 31.83 | 1.91 | 42.35 |
| 3 | LSTM [18] | 0.241 | 0.491 | 0.413 | 38.72 | 2.97 | 106.06 |
| 4 | Linear Regression [18] | 0.253 | 0.503 | 0.392 | 52.69 | 1.32 | 27.83 |

**Table 4. Experimental technique effectiveness for weekly dataset**

| No | Model | MSE | RMSE | MAE | MAPE | Predicting Time (s) | Training Time (s) |
|---|---|---|---|---|---|---|---|
| 1 | CNN+LSTM+MSLA | 0.035 | 0.203 | 0.105 | 27.28 | 0.7 | 19.7 |
| 2 | CNN-LSTM [18] | 0.095 | 0.309 | 0.238 | 31.83 | 2.06 | 14.12 |
| 3 | LSTM [18] | 0.105 | 0.324 | 0.244 | 35.78 | 3.66 | 24.42 |
| 4 | Linear Regression [18] | 0.148 | 0.385 | 0.320 | 41.33 | 1.48 | 11.23 |

Table 1 shows the outcomes of experimental procedures for a minute dataset. The proposed framework achieves the highest possible MSE, RMSE, MAE, and MAPE values of 0.034, 0.200, 0.084, and 12.67. Furthermore, other methodologies failed to produce acceptable outcomes for the data set. The differences in predicting performance between this method and others are rather considerable. As a result, the proposed approach outperformed other methods, such as Linear Regression, LSTM, and CNN-LSTM for minute datasets.

Table 2 shows the achievement findings for the hourly dataset. According to the table results, the proposed strategy CNN+LSTM+MLSA is the most favourable regarding MSE and RMSE, whereas CNN-LSTM retained the highest scores in terms of MAE and MAPE for the hourly dataset. Concerning processing time, when contrasted to the CNN-LSTM model, the proposed strategy boosts training time and decreases prediction time for hourly datasets.

Table 3 shows that the proposed strategy produces the best results for daily datasets. As a result, the proposed model is the most effective strategy for predicting electricity consumption over intermediate periods. Furthermore, the proposed model takes 55.36 seconds to train on a daily dataset, whereas CNN-LSTM takes 42.35 seconds. Comparable to the preceding dataset, the proposed method takes just 0.54 s for predicting time, which is 37% less than the CNN-LSTM model's forecasting time.

Finally, Table 4 displays the experimental techniques' performance on a weekly dataset. As a result, the proposed CNN+LSTM+MSLA technique is the most appropriate method for predicting electric energy usage over long periods. The proposed method and CNN-LSTM have training times of 19.7 and 14.12 seconds, correspondingly.

To summarize, Figure 7 depicts approximate percentages of experimental procedures across four datasets. In terms of four popular performance measures, including RMSE, MSE, MAE and MAPE, the proposed methodology, namely modified sea lion optimized CNN-LSTM, is the most effective method for a majority of the datasets mentioned above, comprising minutely, daily, hourly and weekly datasets.

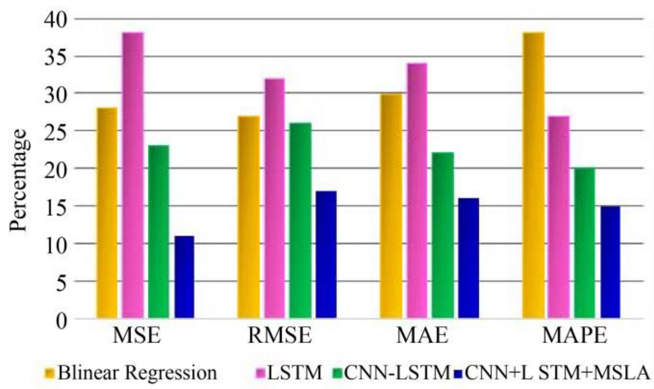


**Fig. 7 Percentages of tests performed across four datasets.**

## 4. Conclusion

This paper presents a unique approach for predicting energy demand that integrates CNN and LSTM networks in a hybrid architecture. A novel adaptation of the SLA for excellent hyperparameter adjustment is provided to improve the model's efficiency substantially. The hybrid CNN-LSTM architecture takes advantage of CNNs' strengths in feature extraction from sequential data and LSTM's expertise in preserving temporal dependencies. The MSLA is used to optimize the hybrid CNN-LSTM model's hyperparameters successfully. MSLA provides an appropriate construction of the model's variables, resulting in improved reliability of forecasts, and is motivated by the behaviour of sea lions in balancing exploration and exploitation during the hunt for food. Numerous evaluations are carried out utilizing real-world energy usage datasets to evaluate the effectiveness of the proposed approach. The findings show that combining CNNs with LSTMs produces better predictions and that the improved Sea Lion Algorithm delivers optimal parameter values, further improving prediction accuracy.

## References

[1] Weixian Li et al., "A Novel Smart Energy Theft System (SETS) for IoT-Based Smart Home," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5531-5539, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[2] Shuoyao Wang, Suzhi Bi, and Ying-Jun Angela Zhang, "Locational Detection of the False Data Injection Attack in a Smart Grid: A Multilabel Classification Approach," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8218-8227, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[3] Mohammed J. Abdulaal et al., "Real-Time Detection of False Readings in Smart Grid AMI Using Deep and Ensemble Learning," *IEEE Access*, vol. 10, pp. 47541-47556, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[4] Mahmoud M. Badr et al., "Detection of False-Reading Attacks in Smart Grid Net-Metering System," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1386-1401, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Ilias Siniosoglou et al., "A Unified Deep Learning Anomaly Detection and Classification Approach for Smart Grid Environments," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1137-1151, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[6] Michael A. Devlin, and Barry P. Hayes, "Non-Intrusive Load Monitoring and Classification of Activities of Daily Living Using Residential Smart Meter Data," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 3, pp. 339-348, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[7] Kun Wang et al, "Green Energy Scheduling for Demand Side Management in the Smart Grid," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 2, pp. 596-611, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[8] Zhenyu Zhou et al., "Energy-Efficient Industrial Internet of UAVs for Power Line Inspection in Smart Grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2705-2714, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[9] Amrit Paudel et al., "Peer-to-Peer Energy Trading in Smart Grid Considering Power Losses and Network Fees," *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 4727-4737, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[10] Lipi Chhaya et al., "IoT-Based Implementation of Field Area Network Using Smart Grid Communication Infrastructure," *Smart Cities*, vol. 1, no. 1, pp. 176-189, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[11] Mamoun Alazab et al., "A Multidirectional LSTM Model for Predicting the Stability of a Smart Grid," *IEEE Access*, vol. 8, pp. 85454-85463, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[12] Fangyu Li et al., "Detection and Diagnosis of Data Integrity Attacks in Solar Farms Based on Multilayer Long Short-Term Memory Network," *IEEE Transactions on Power Electronics*, vol. 36, no. 3, pp. 2495-2498, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[13] Zibin Zheng et al., "Wide and Deep Convolutional Neural Networks for Electricity-Theft Detection to Secure Smart Grids," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1606-1615, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[14] Zhou Su et al., "Secure and Efficient Federated Learning for Smart Grid with Edge-Cloud Collaboration," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1333-1344, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Madalina-Mihaela Buzau et al., "Hybrid Deep Neural Networks for Detection of Non-Technical Losses in Electricity Smart Meters," *IEEE Transactions on Power Systems*, vol. 35, no. 2, pp. 1254-1263, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[16] Muhammad Qamar Raza et al., "An Ensemble Framework for Day-Ahead Forecast of PV Output Power in Smart Grids," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 8, pp. 4624-4634, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[17] Yang Li et al., "Detection of False Data Injection Attacks in Smart Grid: A Secure Federated Deep Learning Approach," *IEEE Transactions on Smart Grid*, vol. 13, no. 6, pp. 4862-4872, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Tuong Le et al., "Improving Electric Energy Consumption Prediction Using CNN and Bi-LSTM," *Applied Sciences*, vol. 9, no. 20, pp. 1-12, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[19] S. Balaji, and S. Karthik, "Energy Prediction in IoT Systems Using Machine Learning Models," *CMC-Computers Materials and Continua*, vol. 75, no. 1, pp. 443-459, 2023. [CrossRef] [Google Scholar] [Publisher Link]