Computing and Informatics, Vol. 40, 2021, 29–56, doi: 10.31577/cai_2021_1_29

BIG DATA ANALYTICS FOR ENERGY CONSUMPTION PREDICTION IN SMART GRID USING GENETIC ALGORITHM AND LONG SHORT TERM MEMORY

Sanju Kumari, Neeraj Kumar, Prashant Singh Rana

Computer Science and Engineering Department
Thapar Institute of Engineering and Technology, Patiala, India
e-mail: {skumari_phd16, neeraj.kumar, prashant.singh}@thapar.edu

Abstract. Smart Grids (SG) have smart meters and advance metering infrastuturte (AMI) which generates huge data. This data can be used for predicting energy consumption using big data analytics. A very limited work has been carried out in the literature which shows the utilization of big data in energy consumption prediction. In this paper, the proposed method is based on Genetic Algorithm – Long Short Term Memory (GA-LSTM). LSTM memorises values over an arbitrary interval that manages time series data very effictively while GA is an evolutionary process that is used for optimization. GA combines with LSTM to process hyperparameters such as hidden layers, epochs, data intervals, batchsize and activation functions. Hence, GA creates a new vector for optimum solution that provides minimum error. These methods provide the best performance when compared with existing benchmarks. Moreover, GA-LSTM is used in a multi-threaded environment which increases the speed of convergence. Here, the multi-core platform is operated for solving one-dimensional GA-based inverse scattering problems. The result shows that GA-LSTM provides better convergence as compared to random approach techniques. For validating the results, Pennsylvania-New Jersey-Maryland Interconnection (PJM) energy consumption data has been used while adopting different performance evaluation metrics.

Keywords: Big data, deep learning, energy consumption prediction, genetic algorithm, load forecasting, long short term memory, multi-threading, smart grid

1 INTRODUCTION

Smart Grid (SG) is a technologically evolved electrical grid. It incorporates information technology into the existing grid and enables two way communication between the electric utility and the end consumer. The physical infrastructure is replaced with a digital one and conventional analog technologies are replaced with improved digital and power electronics. This technology makes the existing grid more efficient and reliable by reducing the number of outages that adds to the grid a self-healing or auto restore capability. Power is immediately rerouted when an outage occurs and power is restored to the affected area. Further, it promotes the use of renewable energy resources which reduces the carbon footprint. Also, SG being technologically advanced, consists of various energy measurement devices such as smart meter and advance metering infrastructure (AMI). These appliances generate huge data which can be termed as big data [24]. The generation of huge data also depends on other equipment such as supervisory control and data acquisition (SCADA) and phasor measurement unit (PMU), which generates data in seconds [11]. Since, there is a large number of measuring devices, data generation needs to be handled in a very efficient way. Therefore, big data management becomes an important task in SG. Moreover, many other tasks can be done using this data and one amongst them is energy consumption prediction.

The demand for energy increases due to economic and population growth. This growth can lead to an increased supply and demand gap, if not predicted well, beforehand. Hence, for proper utilisation of energy, big data analytics play a large role, and energy prediction can be one of the ways to reduce the demand and supply gap. Moreover, for the stability of demand and response, load forecasting has a necessary role to play in the SG system [32]. Many researchers have tried to achieve reliable and efficient energy management through big data techniques. For getting such type of energy management system, they combined data analytics and a scalable selection procedure so that the prediction of supply and consumption of energy could be stable. Big data analytics and cloud computing have been described for managing supply and demand of energy in SG [8]. For managing data, researchers illustrated various big data techniques in SG.

Big data analysis is a critical challenging task and can be overcome by various smart tools and techniques such as support vector machine (SVM) and decision tree analysis (DTA). In a similar line, Wang et al. discussed short-term load forecasting which is based on the recurrent neural network (RNN) and long short term memory (LSTM) [31]. RNN is rather an enhancement of the artificial neural network (ANN) and it is useful for processing the output directly to the first layer. In another case, LSTM is a part of deep learning and it overcomes the drawbacks of RNN. For energy forecasting, LSTM technique plays an important role by analysing the time series data. It uses big data strategies to reduce the storage space as well as analyse the data for taking decision on different models and make several frameworks [27]. Similarly, Pasini et al. suggested encoder-predictor for short-term load forecasting as an effective energy prediction [20].

Few authors used deep neural network (DNN) for energy forecasting. Amarasinghe et al. discussed demand side management using DNN [2]. In this paper, authors tried to discover an intelligent management of energy system and smart load distribution that focused on real time pricing. In another paper, Mohammad et al. defined the energy load forecasting model, which is based on DNN [19]. Furthermore, power demand forecasting using LSTM Neural Network is discussed in [4]. Here, LSTM provides a better performance as compared to the existing work. Few authors have analysed the DNN and Genetic Algorithm (GA) and concluded that this combination provides a better performance. In addition, various authors applied optimal RNN-LSTM model for energy forecasting. In this approach, Residual Network (ResNet) and LSTM have been used to develop the forecasting approach [5]. LSTM-RNN model is largely used in energy forecasting for small datasets. Using this approach, few authors used LSTM-RNN based day ahead load forecasting [28] using smart meter data of different localities. In a similar work, Sainath et al. discussed about short term load forecasting which is based on CNN and LSTM [25]. Many authors illustrated various applications, models and challenges in predicting energy. To overcome these challenges different machine learning models have been used. In [3], authors described statistical based modeling, machine learning and deep learning based model. Further, Diamantoulakis et al. suggested a prediction model for energy which is based on dynamically demand response in SG [1]. They suggested a dynamic energy managenment so that sufficient energy can be managed and further, cost can be reduced.

1.1 Related Works

Energy consumption prediction has a great role to play in maintaining the demand and supply gap in SG. It provides better decisions for power utility. Since, energy prediction is a time series data, it is desirable to work on techniques where challenges of big data can be handled by minimising the error between actual and predicted value. In this context, Rashid used smart meter data and developed big data analytics techniques for analyzing time series data. [23]. However, the author has taken a small dataset and compared it with other techniques which are not effectively considered. A very limited work with respect to energy forecasting using big data analytics has been done using the exiting methods such as the backward propogation neural network, support vector regression (SVR), generalized radial basis function neural network and multiple linear network. In a different work, Khuri et al. described 0/1 multiple knapsac problem [14] where proposed technology works on historical data. However, the authors have not used big data analytics. Few authors work on a similar line of energy management and they tried to improve the prediction of the energy consumption using CNN and Bi-directional LSTM (Bi-LSTM) neural network [16]. They applied electrical energy consumption prediction using Bi-LSTM model for improving results. In this approach, authors used a small dataset. Other researchers described dynamic test data generation using GA in energy prediction strategy [18]. However, none of them have used large datasets.

Sulaiman et al. used smart meter data and solved big data analytics using adaptive neuro-fuzzy inference system [29]. They used this data to predict the day ahead scheduling and verified the prediction accuracy to 84.03 %. In a very close work, Teres used MapReduce algorithm and developed histogram visualisation for SG [30]. However, the research was not intended towards energy prediction. Simhan et al. discussed cloud based approach for dynamic demand response for the SG [26]. However, authors did not focus on energy forecasting. In a different approach, Kaur et al. tried to elaborate LSTM based regression approach to solve the energy management of smart homes [13]. They verified the results with the existing techniques and for validation of the results, data was taken for 112 houses. Furthermore, Couceiro et al. made a stream analytics for energy prediction [7]. They used data streaming for handling large datasets for real time applications in power systems. However, their work was not validated to a real time data stream. A short term load forecasting using LSTM-RNN has been used in the SG [15]. Here, authors validated the result for a single household to forecast the load.

In very recent research, Zhang et al. used SVR and adaptive GA to optimize the parameters to get the best load forecasting model [33]. They performed and validated their results on a specific ratio value using very small datasets. In a similar work, Eseye et al. proposed machine learning tools based on binary GA [9]. They applied feature selection process and Gaussian process regression for measuring the fitness score. A similar approach is discussed in [17], where authors used hybrid model of GA and LSTM. They used half-hourly data from the australian energy market operator. However, their testing and training datasets were verified on small datasets. In another paper, authors used GA-ANN techniques for wind forecasting [10]. In this paper, the authors used meteorological data and compared double-stage back propagation trained ANN. In a similar work, Jaidee et al. presented a method for finding optimal parameters of a deep learning model by GA [12]. They tested the results with many other techniques including LSTM. However, their validation was limited to small datasets.

1.2 Motivation

Load forecasting is a difficult task in SG due to its complex and nonlinear relationship with different datasets. Different data mining and machine learning techniques have been adopted by the researchers but very few have taken large datasets to validate their proposal. Massive use of classification and regression analysis still poses a challenge at the implemetation level when large data is considered. From the literature review, it has been observed that very little work has been done with respect to big data techniques for energy prediction. It has also been observed that when large data is involved, time series data cannot be handled using conventional machine learning tools. Further, load forecasting techniques involve large datasets and to get early convergence we need some optimization tools, along with LSTM. Few authors proposed a different algorithm to develop load forecasting with big data but none have analysed the results in terms of multi-threading approach of

GA-LSTM which increases the speed of the convergence. Further, energy prediction is one of the techniques to understand the proper utilisation of the energy resources and therefore, we need to analyse the big data and use it for load forecasting. Proper load forecasting may reduce the supply and demand gap of electrical usage.

1.3 Contributions

In this paper, a multi-layer GA-LSTM model is proposed for energy prediction. It provides a better result as compared to existing techniques. The purpose of using GA is to optimize the parameters of the LSTM. To verify the effectivness of the proposed system, different parameters of LSTM have been used for reducing the errors. The major contributions of this paper are as follows.

- Multi-threaded based GA-LSTM technique is used for improving the performance of the algorithm with overall execution time.
- After identifying the lower and upperbound of the LSTM parameter, GA is used to optimize the LSTM for better performance.
- To validate the performance of GA-LSTM approach for large data, real time data of PJM has been used to validate the results with different evaluation metrics.
- To find the interval size of the optimal data, that gives minimum mean square error.

1.4 Organisation

Section 2 explains the dataset along with the performance and evaluation parameters. Section 3 provides the methodology of the proposed work. Section 4 outlines the results and discussions. Finally, the paper is concluded in Section 5.

2 DATASET AND ITS DESCRIPTION

2.1 Data Description

The dataset is a multivariate time-series data collected from Pennsylvania-New Jersey-Maryland Interconnection (PJM) which is a regional transmission organization (RTO) in the United States of America [21]. PJM is a part of the Eastern Interconnection grid operating an electric transmission system serving all parts of Delaware, Illinois, New Jersey and North Carolina. The hourly energy consumption data comes from PJM's website and are in megawatt-hours (MWh). The dataset is a daily and weekly based time series data. The dataset is of the PJM East that consists of data from 2002–2018 for the entire eastern region where 2002 to 2015 is used for training and 2015 to 2018 is used for testing [22].

Energy consumption has unique characteristics. The regions have changed over the years, so data may only appear for certain dates per region. GA-LSTM model is applied on these large datasets. The values of variables are compared between actual and predicted values. Since, hourly based data is very complex which is not suitable for LSTM model, therefore, focus was laid on daily and weekly based data. This data is compatible for GA-LSTM model, and provided more than 90 percent of result accuracy. For validation of the proposed work, three types of datasets are used and they are hourly, daily and weekly and the sample data is mentioned in Tables 1, 2 and 3, respectively.

SN	Date	Time (hrs)	Energy (MWh)
1	01/01/2002	1.00	14 107
2	01/01/2002	2.00	14 410
3	01/01/2002	3.00	15 174
4	01/01/2002	4.00	15 261
5	01/01/2002	5.00	14 774
6	01/01/2002	6.00	14 363
7	01/01/2002	7.00	14 045
8	01/01/2002	8.00	13 478
9	01/01/2002	9.00	12 892
10	01/01/2002	10.00	14 097

Table 1. Hourly sample dataset of energy consumption

SN	Date	Energy (MWh)
1	01/01/2006	363 822
2	02/01/2006	389 012
3	03/01/2006	431 551
4	04/01/2006	439 618
5	05/01/2006	388 212
6	06/01/2006	392 685
7	07/01/2006	394 595
8	08/01/2006	393 980
9	09/01/2006	417 416
10	10/01/2006	444 514

Table 2. Daily sample dataset of energy consumption

2.2 Performance Measures Used in This Energy Forecasting

2.2.1 Mean Absolute Error (MAE)

MAE is a measurement of errors between two variables such as x and y. The observations are expressed about the same event. It is expressed as per the following

SN	Year	Week	Energy (MWh)
1	2006	1	2 799 495
2	2006	2	2 986 229
3	2006	3	2 884 968
4	2006	4	2 644 030
5	2006	5	2 614 028
6	2006	6	2 614 028
7	2006	7	2 562 487
8	2006	8	2 562 487
9	2006	9	2 356 473
10	2006	10	2 349 789

Table 3. Weekly sample dataset of energy consumption

equation:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |a_i - p_i| \tag{1}$$

where n is number of observations, a is actual energy consumption and p is the predicted energy consumption.

2.2.2 Mean Square Error (MSE)

Mean square error (MSE) is an estimator which measures the average of the **squares of errors**. Here, average square provides the difference between the predicted value and the actual value. MSE is given as follows.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (a_i - p_i)^2 \tag{2}$$

where p_i indicates predicted value and a_i indicates actual value.

2.2.3 Median Absolute Error (MDAE)

The median absolute error is very crucial due to its robust nature of tackling outliers. Here, the loss is calculated by taking the median of all absolute differences between the actual and the predicted value. In the below equation, p_i is the predicted value of the i^{th} sample and a_i is the corresponding true value. MDAE estimated over n samples is defined as follows:

$$MDAE(a, p) = median(|a_1 - p_1|, \ldots, |a_n - p_n|). \tag{3}$$

2.2.4 Correlation

Correlation describes the statistical relationships between actual and predicted values. It is defined as follows:

$$r = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(p_i - \bar{p})}{\sqrt{\sum_{i=1}^{n}(a_i - \bar{a})^2 \sum_{i=1}^{n}(p_i - \bar{p})^2}} \tag{4}$$

where r is the correlation, a is the actual value, p is the predicted value, $\bar{a}$ is the mean of all actual values, $\bar{p}$ is the mean of all predicted values and n is the number of instances. Correlation lies in the $[-1, 1]$ interval and is considered to have good correlations, if its value tends towards 1 or -1. In this paper, LSTM model is trained on 70 % of the dataset and testing is done on remaining 30 % of dataset. The trained LSTM model generates the predicted values that are compared with actual values. To understand the relationship between actual and predicted values, correlation is the best parameter. The correlation values lie between -1 and $+1$. The sign of the correlation denotes the nature of association and while the value denotes the strength of association.

2.2.5 Coefficient of Determination (R^2)

The coefficient of determination (R^2) summarizes the explanatory power of the regression model and is computed from the sums-of-squares terms and given as per the below equation:

$$R^2 = r * r \tag{5}$$

where r is the correlation as mentioned in Equation (4). R^2 lies in the $[0, 1]$ range and is considered to be good R^2, if its value tends towards 1.

3 METHODOLOGY

3.1 Proposed Work

The workflow of the complete system is shown in Figure 1. As can be seen from this figure, collected data is preprocessed and is divided into training and testing sets. Once the dataset is divided, LSTM model is trained with 70 % of the dataset and testing is done with remaining 30 % of the data. From the test data, prediction of consumed energy is obtained. Further, to improve the model, LSTM parameters are tuned with GA for calculating the evaluation points. The below subsections present modelling of LSTM, GA, GA-LSTM and multi-threading in GA-LSTM.

3.2 Long Short-Term Memory

LSTM is mainly used for time series dataset for prediction of energy. It works with the feedback connections and memorises previous information inside the network.

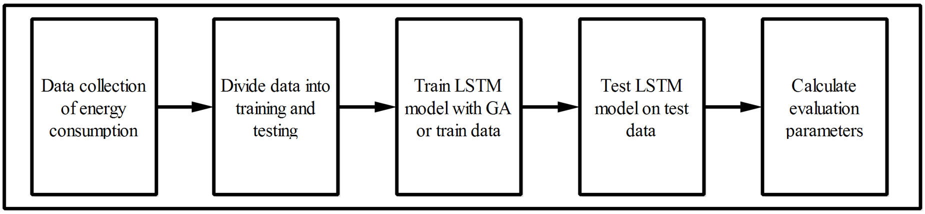
Data collection of energy consumption
Divide data into training and testing
Train LSTM model with GA or train data
Test LSTM model on test data
Calculate evaluation parameters

Figure 1. Workflow of the complete system

It has capability of solving time series and nonlinear prediction problems. The major problem of RNN is "long term dependancy", therefore, LSTM is used to overcome this problem. The cell state is the key of LSTM and it is like a conveyor belt. LSTM is capable of adding or removing the information and it is regulated by structures which are called gates. Gates are the mode where information is optionally chosen. These gates work with sigmoid activation function and a point to point multiplication operation. There are mainly three types of gates: input gate, output gate and forget gate. Tanh, sigma (σ) and Relu are the activation functions mainly used in the LSTM network. The below subsection describes the different techniques of LSTM for handling the large datasets.

3.2.1 Handling a Very Long Sequence Data with LSTM

LSTM is capable of learn and capturing of previous sequences of inputs. It can work nicely with one output, having many inputs but suffers if long input sequence exists. It is called sequence labeling or sequence classification. There are six modes of handling very long sequence data for classification problems. The starting point is to use the long sequence data as it is without any process. However, this may take long time to train. Further, attempt to back-propagate across extremely long input sequences may result in vanishing gradients, and in turn, an unlearnable model. A reasonable limit of 250–500 time steps is often used in practice with large LSTM models. Therefore, a way to handle these types of long sequence data is to simply truncate them. Here, removing a time steps from the beginning or at the end of input sequences is done. In some problem domains, it may be possible to summarize the input sequence. For example, in the case where input sequences are words, it may be possible to remove all words from input sequences that are above a specified word frequency such as and, &, the, and many more.

3.2.2 Process of LSTM

In this subsection, the step by step working of LSTM is explained. The first step in LSTM is to decide what information is to be selected from the cell state. This decision is taken by the forget gate which decides what information to keep and what information to discard. Information from the input and previous hidden states is passed through a sigmoid function which squishes the values between 0 and 1. Values closer to 1 are kept and values closer to 0 are discarded.

The second step is to obtain the current cell state from the previous cell state and input. The previous hidden state and the input are passed through the input gate, which consists of the sigmoid function which squishes the values between 0 and 1 based on their importance. Values closer to 0 are not important while values closer to 1 are. The hidden state and the input are also passed through the tanh function which creates a candidate vector between 1 and -1, this regulates the network. The output of the input gate and the candidate vector is then multiplied. Finally, the obtained value is added to the product of the previous cell state and the forget vector to obtain the current cell state.

The third step decides what the new hidden state will be. The input and previous hidden state are passed through a sigmoid function to obtain the output. Next, the current cell state is passed through a tanh function. The obtained value and the output are then multiplied to decide what information the next hidden state carries. The product of this multiplication is the hidden state which is passed to the next LSTM cell along with the current cell state. The structure is shown in Figure 2.

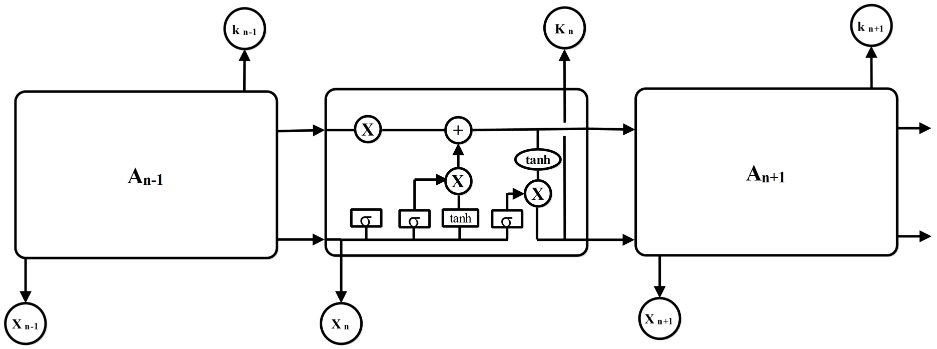

Figure 2. The operation of LSTM [6]

3.2.3 Modeling of LSTM

In this subsection, the mathematical modeling of LSTM cells is explained at every time step. LSTM cell contains several components such as forget gate F which decides what information should be thrown away or kept, a candidate layer C? which holds all the possible values to be added to the cell state, an input gate I which is used to update the cell state and output gate O which decides what the next hidden state should be. Further, we represent the hidden state by H, and the cell state is represented by C and both of these are vectors. Current LSTM cell is considered as the time step t. In the following equations '$*$' is an element-wise multiplication, '$+$' is an element-wise addition.

First, the input and previous hidden state are passed through the forget gate of the LSTM cell which has a sigmoid activation. It uses sigmoid activation because it needs to decide whether to forget information or not. The closer to 0 means to forget, and the closer to 1 means to keep.

$$F_t = \sigma(X_t * U_f + H_{t-1} * W_f) \tag{6}$$

where X_t is an input vector, U_f and W_f are the weight vectors for the forget gate and candidate gate respectively and H_{t-1} is the previous cell output or the hidden state. The new state of the LSTM is represented by follwing equation. We pass the hidden input and current input into tanh function to squish values between -1 and 1 which helps regulate the network.

$$C'_t = \tanh(X_t * U_c + H_{t-1} * W_t) \tag{7}$$

where C'_t is the current cell state at time step t, and it gets passed to next time step. H_{t-1} is the previous cell output and X_t is the input vector. The input gate is represented as per the below equation. We pass current input and previous hidden state into a sigmoid function that decides which values will be updated by transforming the values between 0 and 1. 0 means not important and 1 means important.

$$I_t = \sigma(X_t * U_i + H_{t-1} * W_i) \tag{8}$$

where I_t is an input gate at time step of t, U_i and W_i are the weight vectors for the input gate and candidate gate, respectively, whereas H_{t-1} is the previous cell output. Output gate is represented as follows. Here the input vector and the previous hidden state are passed through a sigmoid function.

$$O_t = \sigma(X_t * (U_o + H_{t-1}) * W_o) \tag{9}$$

where O_t is an output gate at time step of t, X_t is an input vector, U_o and W_o are the weight vectors for the output gate and candidate gate, respectively, whereas H_{t-1} is the previous cell output. The current time step is mentioned as below.

$$C_t = f_t * C_{t-1} + l_t * C'_t \tag{10}$$

where C_t is current cell step at time step of t, f_t is a forget gate vector, I_t is the input gate. The current cell output is mentioned in Equation (11). This uses the output gate and cell state to give us the current hidden state.

$$H_t = O_t * \tanh(C_t) \tag{11}$$

where H_t is the current cell output at time step of t and $\tanh(C_t)$ is the activation function used to find the current cell state. Now with current memory state C_t, we calculate new memory state from input state and C' layer.

$$C_t = C_t + I_t * C'_t \tag{12}$$

where C_t is the current cell state at time step t, and it gets passed to next time step and C'_t is new candidate gate. Now LSTM cell takes the previous memory state $C(t_1)$ and does element wise multiplication with forget gate F_t as per Equation (13).

$$C_t = C_{t-1} * F_t. \tag{13}$$

This output will be based on our cell state C_t but will be a filtered version. Therefore, we apply tanh to C_t and after this we make element wise multiplication with the output gate O and that will be our current hidden state H_t.

$$H_t = \tanh(C_t). \tag{14}$$

Now we pass C_t and H_t to the next time step and repeat the same process.

3.3 Genetic Algorithm (GA)

GA is based on the survival of the fittest, which was proposed by Darwin. Mainly five steps are involved in GA: initial population, selection operator, fitness function, crossover and mutation. The fitness function has great role in GA. Based on the requirements of LSTM, seven sets of chromosome samples are taken and they are data interval size, number of epochs, batchsize, number of hidden layers, dropout rate and number of units in each layer. The selected dimensions are used for processing GA-LSTM model. The results depend on fitness score which provides better result after comparing the value between predicted and actual value. Moreover, mutation and crossover have important role in this algorithm. Here, chromosomes work as a potential solution of target problem. It behaves as a binary string in a chromosome for processing the model. The chromosomes are generated randomly and the one which provides the best performance is selected. The basic process of the flow chart of a GA is shown in Figure 3.

3.4 Optimization in LSTM Network with GA

The operation of LSTM cell is shown in Figure 2 where three gates perform in coordination with each other. In these operation, LSTM is allowed to keep or forget information according to the requirements. This proposed work is divided into two stages. First stage is experimental part, where appropriate network parameters of the LSTM are designed. In the LSTM design, sequential input layer works on five hidden layers. By applying GA, optimal number of hidden neurons are found in each layer. GA searches the optimized hidden layers in LSTM model. In this model, tangent hyperbolic function is used for input nodes and hidden nodes. The range of tanh is (-1 to 1). The activation function of output node is designed as

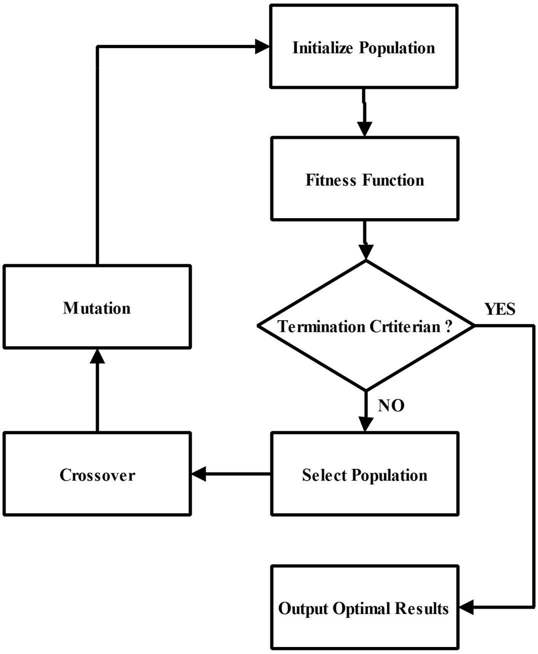

Figure 3. The flow chart of genetic algorithm

a non-linear function which works with the regression method. The objective of this model is to predict the energy consumption for the next year. The random values are set by the initial weight of the network.

In second stage, GA is combined with LSTM model, where fitness function is the main feature. GA is the evolutionary algorithm where initial population is selected on the basis of fitness function. At initial stage, population is generated randomly. After reproduction, best pairs of fitness score are selected. The experimental results depend on fitness score. Here, seven dimensions in one chromosome sample are created. Performance is measured through benchmark and GA-LSTM. This approach has an advantage in prediction of energy consumption with large dataset. The experimental result is compared with Mean Absolute Error (MAE), Mean Square Error (MSE), Median Absolute Error (MDAE), correlation, coefficient of determination and accuracy. GA-LSTM provides the optimal solution for large dimension data. Here, chromosomes are represented by strings of arrays and to obtain fitness value, MSE of the prediction model is used. The detailed algorithm is mentioned in Algorithm 1. This algorithm describes the use of GA to optimize the LSTM parameters. It uses crossover, mutation and selection of best chromosome that gives the best accuracy as fitness value. In step 1, GA parameters are initialized and in step 2 LSTM parameters are initialized. Similarly, Algorithm 2 describes random approach for LSTM parameters optimization. The fitness function (F) is defined as

Algorithm 1 Genetic Algorithm with LSTM

```
1. Initialize the GA parameters
   • cr = 0.9;
   • mr = 0.1;
   • iterations = 20;
   • popSize = 20;
2. Initialize LSTM parameters
   • d = 7;
   • dataInterval = $[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$;
   • nEpochs = $[50, 100, 150, 200, 250, 300, 350, 400, 450, 500]$;
   • batchSize = $[8, 16, 32, 64]$;
   • nHiddenLayer = $[2, 3, 4, 5]$;
   • dropoutRate = $[0.1, 0.2, 0.3, 0.4]$;
   • nUnits = $[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$;
   • nActivationFunction = ['relu', 'sigmoid', 'tanh']
3. t = 1
4. InitPop[$P(t)$]; Initializes the population
5. EvalPop[$P(t)$] = LSTM (chromosome); Evaluates the population
while stopping condition do
  Crossover()
  Mutation()
  MemoriseGlobalBest()
end while
6. Return the individual with the best fitness as the solution;
```

per the below equation.

$$F = \min(MSE(LSTM(x))) \tag{15}$$

where x is a vector of parameter and the sample chromosomes is like $x = [3, 30, 200, 32, \text{Relu}, 0.1, 30]$ which can be verified from Table 4. It returns the MSE between actual and predicted value of the testing dataset.

3.5 Multi-Threading in GA-LSTM

Multi-threading uses the CPU cache, translation lookaside buffer (TLB) cache and single core or multiple cores to carry out a wide range of tasks concurrently. It is a process in which the CPU provides multiple threads simultaneously for the execution of a task in a less amount of time. The CPU cache reduces the average data

Algorithm 2 Random approach with LSTM

```
1. Parameter initialization
   • iterations = 20;
   • bestchromosome = []
   • bestAccuracy = 0
2. Initialize LSTM parameters
   • d = 7;
   • datainterval = $[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$;
   • nEpochs = $[50, 100, 150, 200, 250, 300, 350, 400, 450, 500]$;
   • batchSize = $[8, 16, 32, 64]$;
   • nHiddenLayer = $[2, 3, 4, 5]$;
   • dropoutRate = $[0.1, 0.2, 0.3, 0.4]$;
   • nUnits = $[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$;
   • nActivationFunction = ['Relu', 'sigmoid', 'tanh'];
3. t = 1
4. While $t \leq$ iterations;
chromosome = Generate random set of LSTM parameters;
Evaluate accuracy = LSTM (chromosome);
if accuracy > bestAccuracy;
bestAccuracy = accuracy;
bestChromosome = chromosome;
$t = t + 1$
5. Return the bestAccuracy and bestChromosome as solutions.
```

access time from the main memory while TLB reduces the average time for memory allocation in the main memory. In GA-LSTM, data is loaded and the model is trained thereafter. These processes go step by step and the user needs to wait for their execution. But through multi-threading these tasks can be performed in parallel by running a number of threads which get queued and operate at a high speed without getting blocked. There are many benefits of multi-threading. Firstly, it eliminates the multiple processor subsystem and the hardware completely. Secondly, a single server can perform a number of tasks simultaneously by dispatching multiple threads at a time. This reduces the number of servers required while loading the large data. Thirdly, the applications run one after the other and wait for the former to get over. The latter applications do not get blocked, instead of that, they get queued and increase responsiveness to the operation. Finally, the memory to be allocated to processes is quite high if multi-threading is not used.

4 RESULTS

PJM dataset from 2002 to 2018 has been used to validate the results, wherein, dataset from 2002 to 2015 has been taken for training and dataset from 2015 to 2018 is taken for testing. GA-LSTM model is trained and tested and the validation of the proposed work is analysed. In the initial stage, the number of LSTM unit is formed into vectors of hidden layers, epochs, batch size, interval size and activation functions. In the proposed work, parameters of LSTM are optimized and verified for its effectiveness of the GA-LSTM model. The performance of the GA-LSTM network is measured using MAE, MSE, MDAE, correlation, coefficient of determination and accuracy. Comparsion of actual and predicted results was done and it was found that error is reduced using the proposed model. For validation of the proposed work, three types of datasets have been used hourly, daily and weekly whose sample data is mentioned in Section 3. Since, hourly based dataset is very complex and not suitable for GA-LSTM model, we have used daily and weekly based dataset in our work.

4.1 Experimental Setup and Simulation Parameters

The proposed algorithm uses Xeon Processor with 64 GB RAM (20 cores) and a 1 TB SSD. To increase the speed of the simulation, multi-threaded GA-LSTM algorithm is used. LSTM parmeters have been shown in Table 4. For better validation of the results, maximum of 5 hidden layers are used. It is seen from this table that as the data interval size increases, epoch is also increased. Further, three types of activation functions tanh, sigmoid and Relu are used. The purpose of using three types of activation functions is to verify the proposed methodology for large dataset. Further, these activation functions will give better choice while making crossover and mutation in GA. It can be seen in the table that dropout rate varies between 0.1 to 0.5 and the number of units are taken between 10 to 100 at an interval of 10.

SN	Name Parameters	Values
1	Number of hidden layers	$[2, 3, 4, 5]$
2	Data interval size	$[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$
3	Epochs	[50 to 500 with an interval of 50]
4	Batch size	$[8, 16, 32, 64]$
5	Activation Function	[Tanh, Sigmoid, Relu]
6	Dropout rate	$[0.1, 0.2, 0.3, 0.4, 0.5]$
7	Number of units	$[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$

Table 4. LSTM hyperparameters

GA parameters such as crossover, mutation, population size and number of iterations are mentioned in Table 5. It is seen from the table that the single point crossover is used. Another parameter is mutation where rate at single point is taken as 0.1. The size of the initial population is 100. Further, the maximum number of iterations are taken as 20. The selection criteria used is roulette wheel. After optimizing the parameters of LSTM, the best parameters are found which are mentioned in Table 6. This table provides the best parameters for daily and weekly energy prediction and it can be observed that Relu activation function provides the best perfomance. Similarly, it can be seen that the optimized batch size is 16 for both daily and weekly energy prediction. Epochs are found to be 450 for both the cases. Optimized data interval size is 60 for both the cases. The details of other parameters of LSTM and Random are mentioned in Table 6.

SN	**Name of Parameters**	**Values**
1	Crossover rate	0.9 (Single point crossover)
2	Mutation rate	0.1 (Single point mutation)
3	Population size	100
4	Iteration	20

Table 5. GA parameters

Approach		LSTM		Random	
SN	**Parameters**	**Daily Energy Prediction**	**Weekly Energy Prediction**	**Daily Energy Prediction**	**Weekly Energy Prediction**
1	Number of hidden layers	4	3	3	2
2	Data interval size	60	60	50	40
3	Epochs	450	450	400	450
4	Batch size	16	16	8	16
5	Activation function	Relu	Relu	Relu	Relu
6	Dropout rate	0.3	0.2	0.3	0.2
7	Number of units	60	40	50	50

Table 6. Optimized parameters of LSTM and Random for daily and weekly energy consumption prediction using GA

Table 7 shows the experimental values of results with GA and Random approach. The performance metrics have been calculated for daily and weekly dataset with respect to MAE, MSE, MDAE, correlation, coefficient of determination (R^2) and accuracy. It can be seen from this table that accuracy of GA is 82.42 as compared to the random approach which is 51.26 for the daily dataset. Similarly for weekly dataset, accuracy of GA is 80.27 and random approach is 48.22. Further it can be observed that correlation and R^2 is better in GA as compared to random approach. The other evaluations parameters such as MAE, MSE and MDAE are also shown and it gives the best performance for the daily as well as weekly dataset in GA as compared to random approach. The acceptable error is mentioned as 10^3 and 10^4 for the daily and weekly dataset, respectively.

	Daily Energy Consumption		**Weekly Energy Consumption**	
Evaluation Parameters	**GA**	**Random**	GA	**Random**
Mean Absolute Error	5.34×10^2	1.27×10^3	1.35×10^3	2.23×10^4
Mean Squared Error	1.27×10^3	7.66×10^4	2.90×10^4	6.45×10^6
Median Absolute Error	7.46×10^1	9.80×10^2	4.20×10^2	8.31×10^3
Correlation	0.931	0.551	0.892	0.496
R^2	0.868	0.304	0.792	0.246
Accuracy	82.42*	51.26*	80.27@	48.22@

* with acceptable error of 10^3; @ with acceptable error of 10^4

Table 7. Performance and evaluation paramenters for daily and weekly dataset for Random and GA

4.2 Energy Predication on a Daily Dataset

GA-LSTM predication can be used for large datasets where GA is used to optimize the parameters of the LSTM. Figure 4 shows the energy consumption of actual versus predicted daily energy. Daily energy curve is given in MWh since PJM covers larger area of the USA. It is seen that predicted energy of PJM is very close to the actual energy. This prediction will help to schedule the generating units of PJM. LSTM uses 70 % of the data for training and 30 % for testing. This property of LSTM reduces the testing data. With GA approach, the energy consumption prediction on daily dataset gives much less error.

Figure 5 shows the convergence of daily energy prediction by using random approach and GA-LSTM approach. The convergence refers to different system moving towards performing the same task. Random sets approach is heuristic by nature hence, it is very helpful in discovering things themselves. In this graph, it is observed that random approach convergence takes larger iterations while GA approach takes 15 iterations to converge. The convergence graph is taken with MSE and it shows lower value as compared to random approach. This proves that GA provides optimized result with a fewer number of iterations and it converges at low iterations.

Figure 6 shows parameter sensitivity for different data interval size for daily energy consumption prediction. Parameter sensitivity analysis shows uncertainty in the output of a model. It can be used to validate with sources of uncertainty in the model input. Further, this is a method for finding or establishing the response of a model which changes when parameters are varied. This graph shows MSE versus data interval size. It is observed from the figure that MSE is low for data interval of size 60. Similarly, other optimized parameters can be seen from Table 6.

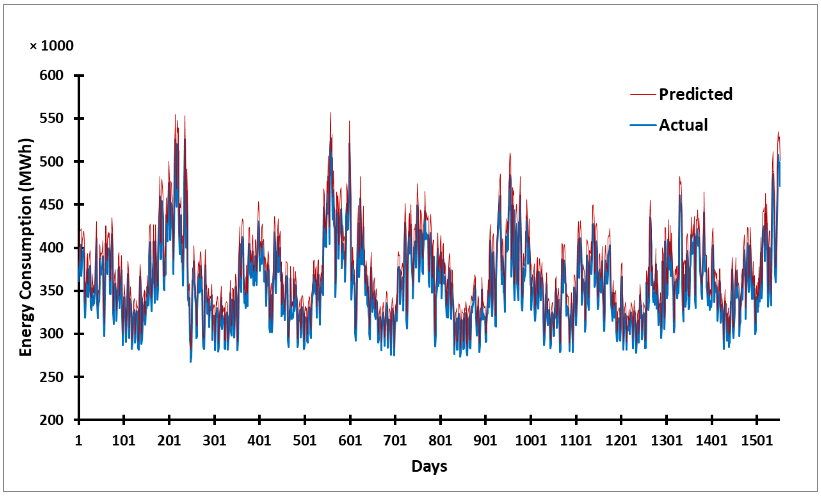

Figure 4. Actual vs. prediction of daily energy consumption

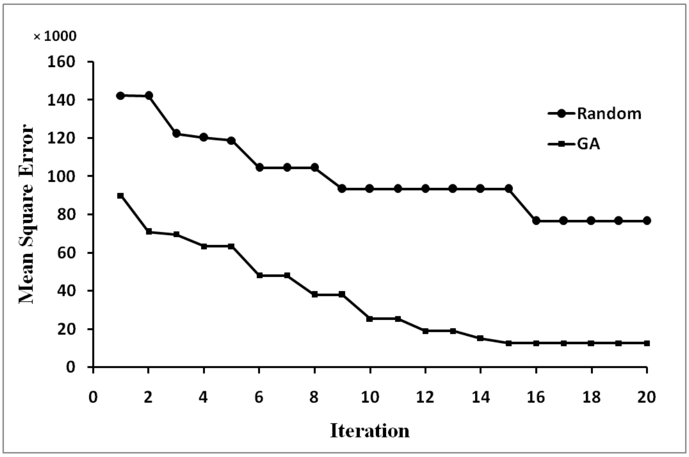

Figure 5. Convergence of daily energy consumption prediction

4.3 Energy Prediction on Weekly Dataset

Figure 7 represents weekly actual energy consumption versus predicted energy consumption. Weekly prediction is mostly done by the utility for week ahead scheduling of the generating units and it is one of the most widely used short term load forecasting in the SG. Since data is large, it can be seen from this figure that the energy is in 1 000 of MWh. It is also observed that actual versus predicted energy consumption are very close to each other with a very small error. Further, this energy

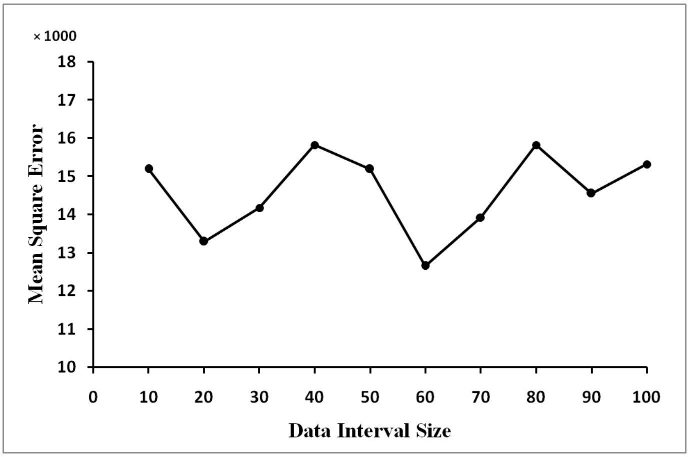

Figure 6. Parameter sensitivity for data interval size of daily energy consumption prediction

consumption prediction can be used to maintain the balance between demand and supply of the PJM. This prediction will save a large amount of money for utility and better utilization of the generating plants.

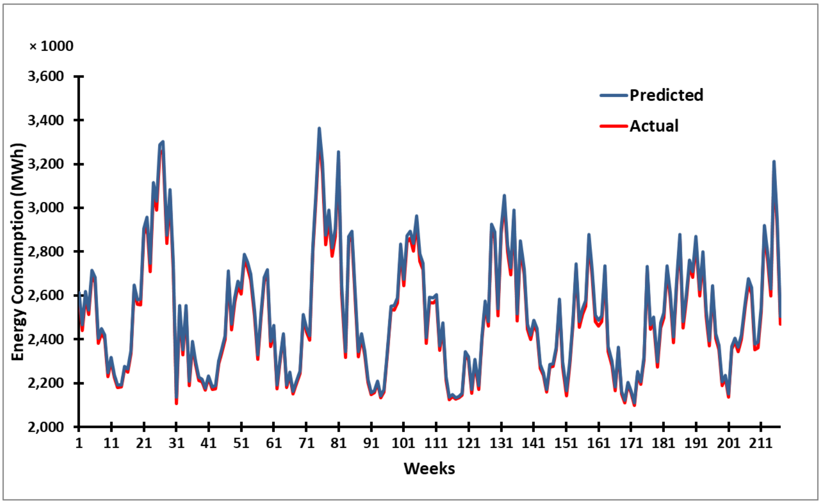

Figure 7. Actual vs. prediction of weekly energy consumption

The validation of the weekly energy forecasting is achieved through convergence graph. Figure 8 represents the convergence of weekly energy consumption prediction with GA and a random approach. This convergence graph is shown till 20th iteration and found that the random approach has slower convergence rate as compared to GA. Random approach converges at 15th iteration while there is no certainity of convergence through the random approach. It is also observed that MSE has a higher value of the random approach as compared to GA.

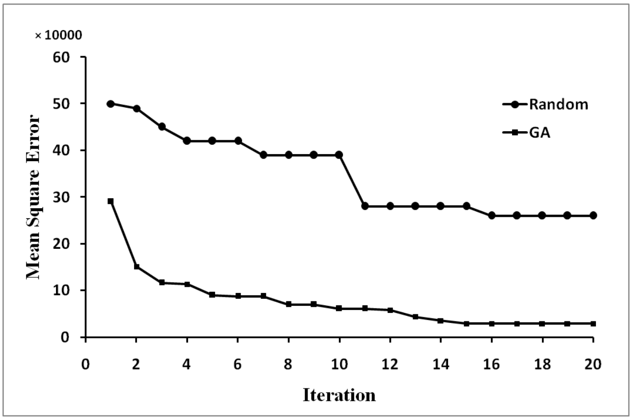

Figure 8. Convergence of weekly energy consumption prediction

Parameter sensitivity is a method for finding or establishing how responses of a model change when parameters are varied. There is great role of parameter sensitivity in optimization problem. Figure 9 shows parameter sensitivity with respect to MSE versus data interval size. It has been observed from the figure that MSE has a lower value at data interval of size 60. This proves that data interval size optimization is a very accurate method which will give a better convergence at lower iterations. Similarly, other optimized parameters can be seen from Table 6.

4.4 Multi-Threading

The competitive performance of multiple threads is shown in Table 8. Here, the program is run on a machine having 4 cores. Different number of threads are run which is starting from 1 to 8. As the number of threads are increasing from 1 to 4, the total execution time decreases, but as we increase the number of threads from 5 to 8, the total execution time increases and this is evident from Figure 10. Therefore, the optimal number of threads must be 4 to run the LSTM-GA program in a 4-core machine.

4.5 Variability of MSE with GA and Random Approach

To validate the performance of GA and random approach, box plot is shown in this subsection. Boxplot is a standardized way of displaying the variation of any quantity which emphasizes on stablity of the system. Further, we need to have

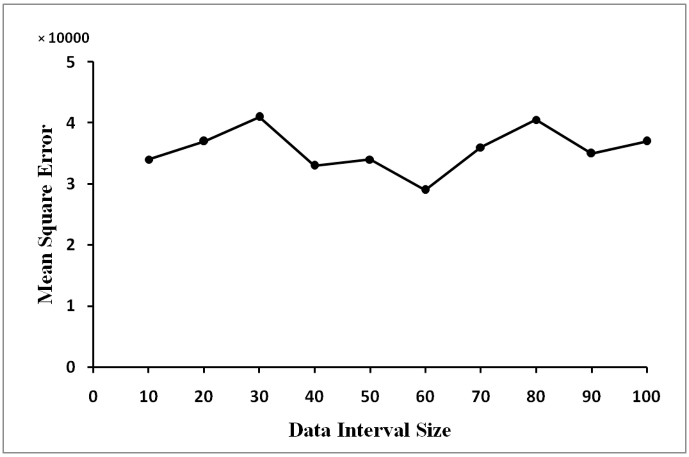

Figure 9. Parameter sensitivity for data interval size of weekly energy consumption prediction

Number of Threads	Time Taken in (sec)
1	822
2	545
3	476
4	364
5	390
6	490
7	600
8	900

Table 8. Time taken by GA-LSTM with different number of threads

information on the variability or dispersion of the data. A boxplot is a graph that gives a good indication of how the values in the data are spread out and also identify outliers. The variation of MSE with random approach and GA is shown to prove the efficiency of the proposed GA-LSTM algorithm. A total of 10 simulations are performed for daily and weekly energy consumption prediction. Figure 10 and Figure 11 present the box-plot for energy consumption for daily and weekly energy consumption prediction. Here, the variation of MSE in random approach is more than GA. It is found that GA variation is very less as compared to random approach which proves stability of MSE. Since there is less variation of MSE for the energy prediction using GA, it outperforms the random approach.

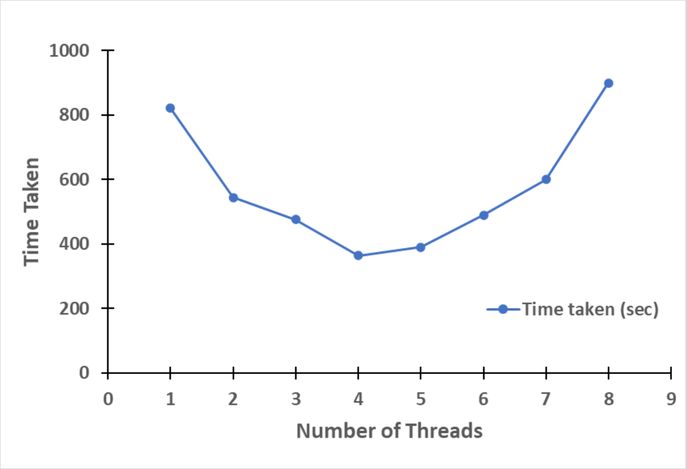

Figure 10. Multi-threading – number of threads vs. execution time

5 CONCLUSION

This paper introduces the prediction of energy consumption on real time large dataset obtained from PJM. It uses big data analytics using machine learning to predict the energy consumption for large dataset. To achieve this, mainly two mod-

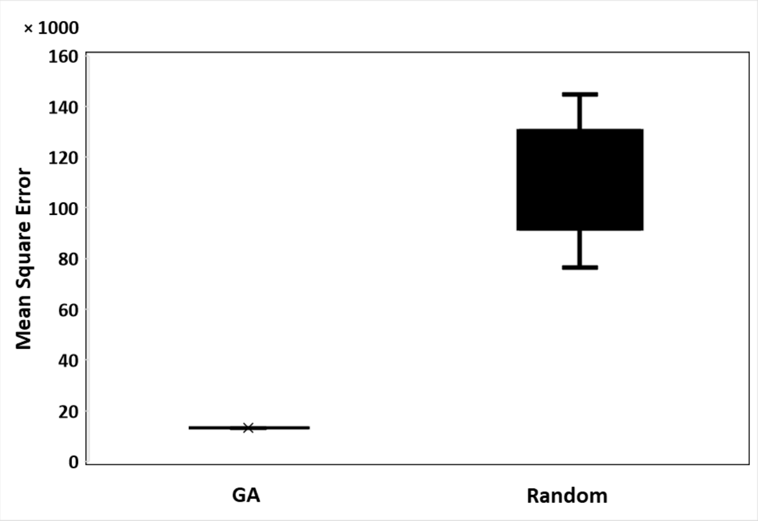

Figure 11. Comparison of genetic algorithm and random approach for mean absolute error – daily basis

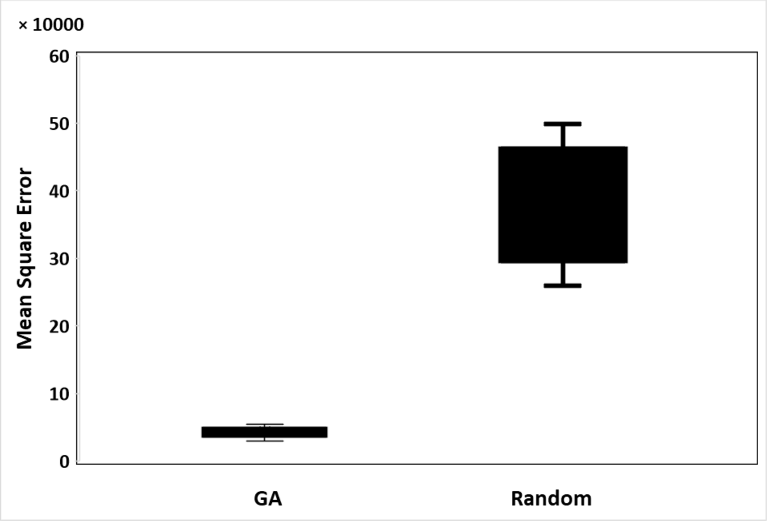

Figure 12. Comparison of genetic algorithm and random approach for mean absolute error – weekly basis

els – random and GA approach – are applied. On comparing the performance of both models, it was found that GA-LSTM outperforms the LSTM. Further, multi-threaded GA-LSTM is used to increase the speed of convergence. It has been observed that GA has higher accuracy as compared to random approach. The comparison is conducted experimentally for real datasets of PJM for daily and weekly energy consumption. It has been proved that GA-LSTM model provides optimized effective performance. The novelty of the paper lies in the multi-threaded based GA-LSTM technique used for improving the performance of the algorithm with overall low execution time. Further, after identifying the lower and upperbound of the LSTM parameter, GA is used to optimize LSTM for better performance. The results of the proposed work are verified with the variability of MSE and it was found that the proposed algorithm passes all the evaluation parameter checks.

REFERENCES

[1] Aman, S.—Frincu, M.—Chelmis, C.—Noor, M.—Simmhan, Y.—Prasanna, V. K.: Prediction Models for Dynamic Demand Response: Requirements, Challenges, and Insights. 2015 IEEE International Conference on Smart Grid Communications (SmartGridComm), 2015, pp. 338–343, doi: 10.1109/smartgridcomm.2015.7436323.

[2] Amarasinghe, K.—Marino, D. L.—Manic, M.: Deep Neural Networks for Energy Load Forecasting. 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), 2017, pp. 1483–1488, doi: 10.1109/ISIE.2017.8001465.

[3] Goodfellow, I.—Bengio, Y.—Courville, A.: Deep Learning. The MIT Press, 2017.

[4] Cheng, Y.—Xu, C.—Mashima, D.—Thing, V. L. L.—Wu, Y.: PowerLSTM: Power Demand Forecasting Using Long Short-Term Memory Neural Network. In: Cong, G., Peng, W. C., Zhang, W., Li, C., Sun, A. (Eds.): Advanced Data Mining and Applications (ADMA 2017). Springer, Cham, Lecture Notes in Computer Science, Vol. 10604, 2017, pp. 727–740, doi: 10.1007/978-3-319-69179-4_51.

[5] Choi, H.—Ryu, S.—Kim, H.: Short-Term Load Forecasting Based on ResNet and LSTM. 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), 2018, pp. 1–6, doi: 10.1109/smartgridcomm.2018.8587554.

[6] Understanding LSTM Networks. Colah's Tutorial on LSTM. `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`, 2017.

[7] Couceiro, M.—Ferrando, R.—Manzano, D.—Lafuente, L.: Stream Analytics for Utilities. Predicting Power Supply and Demand in a Smart Grid. 2012 3rd International Workshop on Cognitive Information Processing (CIP), 2012, pp. 1–6, doi: 10.1109/cip.2012.6232904.

[8] Diamantoulakis, P. D.—Kapinas, V. M.—Karagiannidis, G. K.: Big Data Analytics for Dynamic Energy Management in Smart Grids. Big Data Research, Vol. 2, 2015, No. 3, pp. 94–101, doi: 10.1016/j.bdr.2015.03.003.

[9] Eseye, A. T.—Lehtonen, M.—Tukia, T.—Uimonen, S.—John, R. J.: Machine Learning Based Integrated Feature Selection Approach for Improved Electricity Demand Forecasting in Decentralized Energy Systems. IEEE Access, Vol. 7, 2019, pp. 91463–91475, doi: 10.1109/access.2019.2924685.

[10] Eseye, A. T.—Zhang, J.—Zheng, D.—Ma, H.—Gan, J.: Short-Term Wind Power Forecasting Using a Double-Stage Hierarchical Hybrid GA-ANN Approach. 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), 2017, pp. 552–556, doi: 10.1109/ICBDA.2017.8078695.

[11] Hu, J.—Vasilakos, A. V.: Energy Big Data Analytics and Security: Challenges and Opportunities. IEEE Transactions on Smart Grid, Vol. 7, 2016, No. 5, pp. 2423–2436, doi: 10.1109/tsg.2016.2563461.

[12] Jaidee, S.—Pora, W.: Very Short-Term Solar Power Forecasting Using Genetic Algorithm Based Deep Neural Network. 2019 4th International Conference on Information Technology (InCIT), 2019, pp. 184–189, doi: 10.1109/incit.2019.8912097.

[13] Kaur, D.—Kumar, R.—Kumar, N.—Guizani, M.: Smart Grid Energy Management Using RNN-LSTM: A Deep Learning-Based Approach. 2019 IEEE Global Communications Conference (GLOBECOM), 2019, pp. 1–6, doi: 10.1109/globecom38437.2019.9013850.

[14] Khuri, S.—Bäck, T.—Heitkötter, J.: The Zero/One Multiple Knapsack Problem and Genetic Algorithms. Proceedings of the 1994 ACM Symposium on Applied Computing (SAC '94), 1994, pp. 188–193, doi: 10.1145/326619.326694.

[15] Kong, W.—Dong, Z. Y.—Jia, Y.—Hill, D. Y.—Xu, Y.—Zhang, Y.: Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network. IEEE Transactions on Smart Grid, Vol. 10, 2019, No. 1, pp. 841–851, doi: 10.1109/tsg.2017.2753802.

[16] Le, T.—Vo, M. T.—Vo, B.—Hwang, E.—Rho, S.—Baik, S. W.: Improving Electric Energy Consumption Prediction Using CNN and Bi-LSTM. Applied Sciences, Vol. 9, 2019, No. 20, Art. No. 4237, doi: 10.3390/app9204237.

[17] Mamun, A.—Hoq, M.—Hossain, E.—Bayindir, R.: A Hybrid Deep Learning Model with Evolutionary Algorithm for Short-Term Load Forecasting. 2019 8th International Conference on Renewable Energy Research and Applications (ICRERA), 2019, pp. 886–891, doi: 10.1109/icrera47325.2019.8996550.

[18] Michael, C. C.—McGraw, G. E.—Schatz, M. A.—Walton, C. C.: Genetic Algorithms for Dynamic Test Data Generation. Proceedings of 12th IEEE International Conference Automated Software Engineering, 1997, pp. 307–308, doi: 10.1109/ASE.1997.632858.

[19] Mohammad, F.—Kim, Y. C.: Energy Load Forecasting Model Based on Deep Neural Networks for Smart Grids. International Journal of System Assurance Engineering and Management, Vol. 11, 2020, pp. 824–834, doi: 10.1007/s13198-019-00884-9.

[20] Pasini, K.—Khouadjia, M.—Samé, A.—Ganansia, F.—Oukhellou, L.: LSTM Encoder-Predictor for Short-Term Train Load Forecasting. In: Brefeld, U., Fromont, E., Hotho, A., Knobbe, A., Maathuis, M., Robardet, C. (Eds.): Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2019). Springer, Cham, Lecture Notes in Computer Science, Vol. 11908, 2019, pp. 535–551, doi: 10.1007/978-3-030-46133-1_32.

[21] PJM. Pennsylvania New Jersey Maryland Interconnection. `https://www.pjm.com`, 2020.

[22] PJM Dataset. PJM Time Series Analysis and Forecasting Data. `https://www.kaggle.com/brahimmebrek/pjm-east-eda-and-forecasting`, 2020.

[23] Rashid, M. H.: AMI Smart Meter Big Data Analytics for Time Series of Electricity Consumption. 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), 2018, pp. 1771–1776, doi: 10.1109/trustcom/bigdatase.2018.00267.

[24] Sagiroglu, S.—Terzi, R.—Canbay, Y.—Colak, I.: Big Data Issues in Smart Grid Systems. 2016 IEEE International Conference on Renewable Energy Research and Applications (ICRERA), 2016, pp. 1007–1012, doi: 10.1109/icrera.2016.7884486.

[25] Sainath, T. N.—Vinyals, O.—Senior, A.—Sak, H.: Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 4580–4584, doi: 10.1109/icassp.2015.7178838.

[26] Simmhan, Y.—Aman, S.—Kumbhare, A.—Liu, R.—Stevens, S.—Zhou, Q.—Prasanna, V.: Cloud-Based Software Platform for Big Data Analytics in Smart Grids. Computing in Science and Engineering, Vol. 15, 2013, No. 4, pp. 38–47, doi: 10.1109/mcse.2013.39.

[27] Stimmel, C.: Big Data Analytics Strategies for the Smart Grid. CRC Press, 2014.

[28] Sulaiman, S. M.—Jeyanthy, P. A.—Devaraj, D.: Artificial Neural Network Based Day Ahead Load Forecasting Using Smart Meter Data. 2016 Biennial International Conference on Power and Energy Systems: Towards Sustainable Energy (PESTSE), 2016, IEEE, pp. 1–6, doi: 10.1109/pestse.2016.7516422.

[29] Sulaiman, S. M.—Jeyanthy, P. A.—Devaraj, D.: Big Data Analytics of Smart Meter Data Using Adaptive Neuro Fuzzy Inference System (ANFIS). 2016 International Conference on Emerging Technological Trends (ICETT), 2016, pp. 1–5, doi: 10.1109/icett.2016.7873732.

[30] Teres, A. D.: Histogram Visualization of Smart Grid Data Using MapReduce Algorithm. 2019 2$^{\text{nd}}$ International Conference on Power and Embedded Drive Control (ICPEDC), 2019, pp. 307–312, doi: 10.1109/icpedc47771.2019.9036693.

[31] Wang, L.—Mao, S.—Wilamowski, B.: Short-Term Load Forecasting with LSTM Based Ensemble Learning. 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2019, pp. 793–800, doi: 10.1109/ithings/greencom/cpscom/smartdata.2019.00145.

[32] Wang, Y.—Chen, Q.—Kang, C.—Zhang, M.—Wang, K.—Zhao, Y.: Load Profiling and Its Application to Demand Response: A Review. Tsinghua Science and Technology, Vol. 20, 2015, No. 2, pp. 117–129, doi: 10.1109/tst.2015.7085625.

[33] Zhang, G.—Guo, J.: A Novel Method for Hourly Electricity Demand Forecasting. IEEE Transactions on Power Systems, Vol. 35, 2020, No. 2, pp. 1351–1363, doi: 10.1109/tpwrs.2019.2941277.

Sanju Kumari received her B.E. degree in computer science engineering from the Solapur University, Solapur, India in 2010 and M.E. degree in information technology from the Rajiv Gandhi Prodhiki Viswavidalaya, Bhopal, India in 2014. Presently, she is doing Ph.D. from the Thapar Institute of Engineering and Technology, Patiala, India. Her research interest is machine learning, deep learning, big data and smart grid.

Neeraj Kumar received his Ph.D. in CSE from the Shri Mata Vaishno Devi University, Katra (J & K), India, and was a postdoctoral research fellow in the Coventry University, Coventry, UK. He is working as Professor in the Department of Computer Science and Engineering, Thapar University, Patiala, India. He has published more than 100 technical research papers in leading journals and conferences from IEEE, Elsevier, Springer, John Wiley etc. He has guided many research scholars leading to Ph.D. and M.E. His research is supported by funding from TCS, DST and UGC.

Prashant Singh Rana is Assistant Professor in the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, India. He also worked as a Project Scientist at IIT Delhi, India. He received his Ph.D. from ABV Indian Institute of Information Technology and Management, Gwalior, India and his areas of research are machine learning, soft computing, combinatorial problems, and computational biology.