# Deep LSTM-BiGRU Model for Electricity Load and Price Forecasting in Smart Grids

Aneeza Khalid, Sohail Iqbal and Safdar Abbas



August 11, 2022

# Deep LSTM-BiGRU Model for Electrcity Load and Price Forecasting in Smart Grids

Aneeza Khalid
*School of Electrical Engineering and Computer Sciences (SEECS)*
*National University of Sciences and Technology* (NUST)
Islamabad, Pakistan
akhalid.msit18seecs@seecs.edu.pk

Sohail Iqbal
*School of Electrical Engineering and Computer Sciences (SEECS)*
*National University of Sciences and Technology (NUST)*
Islamabad, Pakistan
sohail.iqbal@seecs.edu.pk

Safdar Abbas
*School of Electrical Engineering and Computer Sciences (SEECS)*
*National University of Sciences and Technology (NUST)*
Islamabad, Pakistan
safdar.abbas@seecs.edu.pk

*Abstract*— **With the advancement of technology, people are curious to know how much energy they are going to use in the next hour and how much it will cost them. Many accurate prices and load forecasting algorithms are already working but ignore the convergence rate. In the case of STF when an algorithm takes too much time in results formulation, it becomes useless for end users and utilities. We incorporated deep learning techniques as they process a large amount of data quickly and can predict accurate results with a fast computational time. The proposed solution LSTM-BiGRU is formed in combination of LSTM and GRU layers, both are RNN variations and capable of forecasting the best results. LSTM and GRU are combined in the best possible way to achieve maximum accuracy with a fair computational time. The proposed solution is showing MAPE in load forecasting from 3.12% to 7.42% in different scenarios. Similarly, MAE for price forecasting is calculated between 2.35 to 3.02, and the computational time of the proposed solution in different scenarios is recorded <1 min. So, a fair tradeoff is maintained between forecasting results and computational time. In the future, the proposed method can be improved by optimization of proposed hybrid algorithms with evolutionary algorithms, and the use of GPUs and TPU can further decrease the computational time.**

***Keywords—LSTM, GRU, Load forecasting, price forecasting, short term forecasting***

## I. INTRODUCTION

Energy crises are always present in this world. Many different sources are incorporated together to fulfill the energy demand [1]. Smart Grid replaces the traditional grids by providing communication between users/consumers, utilities, and manufacturers [2]. It helps in managing the resources efficiently, in the management of demand and supply, enhancing reliability, trading, and cost management [3]. It provides bidirectional communication between power generators, transmitters, distributors, and consumers [4].

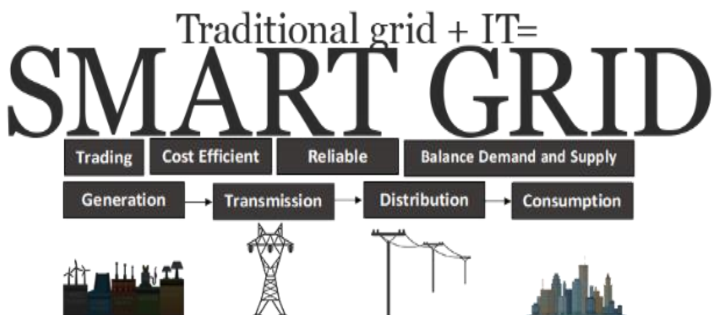

*Figure 1: Conceptual Diagram of Smart Grid*

An accurate energy forecast is very important for producers, consumers, and utilities. In this paper, we focus on both VSTF and STF. Very Short-term forecasting (VSTF) is forecasting for an hour, while short-term forecasting (STF) is for a day to a week [5,6]. For an accurate energy and price forecast, many deep learning neural networks are working very successfully. We have observed two types of convergence rates in different papers i.e., slow convergence and fast convergence [7]. Slow computations occur when the designed model is complex, the data wasn't preprocessed earlier, and the forecast algorithm is time taking [8]. Usually, slow computations range from 2 mins and more. While fast computations are how fast or robust an algorithm reaches to its local optimal point. It is usually a minute or less than it [9,10]. There are many factors on which computational time depends including overfitting, complex models, models with slow training times, data preprocessing, coding style, and optimizers [11]. Some of the algorithms work on increasing the accuracy of their results, for this, they have undergone exhaustive training of their data which is later causing overfitting. So, we need to design the best fit model.

The best fitting model produces accurate results, it will undergo fast computations and help in managing the demand side management [12]. This model helps utilities to manage the energy demand on time. This will help the end users to manage the electricity usage according to the price in that hour at the last moment.

## II. RELATED WORK

Artificial Neural Network (ANN) based Day Ahead Load Forecasting model is proposed [13] to forecast load and achieved 98.76% accurate results in 102 seconds. In [14], an ANN-based forecaster is used for forecasting electricity load and price, however, they used Q learning algorithm for finding the maximum benefit value for consumers and service providers. [15] used Deep Long Short-Term Memory (DLSTM), based forecaster, on a big dataset of ISO NE and NYISO. This DLSTM is a backpropagation of NN in which weights can be readjusted to increase the accuracy of the results. To overcome the limitations of optimizers and to keep intact the whole forecasting procedure, block models are being used. In [16], a hybrid forecast model is purposed to predict load and price accurately. This hybrid model comprises of DTCWT (Dual-Tree Complex Wavelet Transform) and Multi-Stage Forecast Engine (MSFE). The proposed solution in comparison with benchmarks Autoregressive Integrated Moving Average (ARIMA), Wavelet Transform (WT) +ARIMA, and MR-MI +NN) performed better. In [17], Enhanced Neural Networks (ENN) based forecast engine is used, and error minimization is performed by optimization of Enhanced Shark Smell Optimization (ESSO). In [18], two different NNs are implemented ridgelet and Elman NNs. With the passage of time, these neural networks are enhanced, and CNN is one of those modified and better-performance networks. A different number of the max-pooling layers can be adjusted to get accurate results. In [19], two models are implemented with CNN NN- Genetic Algorithm (GA) and NN-Particle Swarm Optimization (PSO). Results showed that NN-GA works better for STF. In [20] ECNN forecaster is used, and a very low MAPE of 0.297 is achieved for load forecasting. In [21], two different models are used ECNN and ESVR. ECNN and ESVR performed well with threshold values 0.08 and 0.15 achieving 2% and 1 % accuracy respectively. In [22], ECNN and Efficient kth neighbor neural network (EKNN) is used, while MI for feature selection. Simulation results showed the accuracy of 92% and 93% by ECNN and EKNN. In [23], Enhanced Linear Regression (ELR) and Enhanced Recurrent Extreme Learning Machine (ERELM) two different forecasters are proposed and tested on two different datasets. Results showed that ELR works well with UMASS Electric Dataset whereas ERELM works well for UCI Datasets. However, for ERELM there is a tradeoff between convergence time and accuracy. Similar to CNN, there are many RNN-modified forecasters that exist, some are discussed here. In [24], a Self-Recurrent Wavelength Neural Network (SRWNN) is used. LM trains the data to SRWNN in less than 35sec for one-day STF. Results showed that SRWNN can cope with non-smooth and volatile time series data and generate more accurate forecast results than WNN. There are three approaches designed in [25], Seasonal ARIMAX (SARIMAX), Gated Recurrent Neural Network (GRNN), and Gated Convolutional Neural Network (GCNN). The results of prediction accuracy showed that the SARIMAX model shows better results than GRNN as its accuracy decreases by 22.6% due to weather covariance. The computational accuracy of GCNN increased by 8% as compared to the SARIMAX model. While generally the performance of GRNN, and GCNN is less than that of the SARIMAX model. For time series forecasting, function approximation, and system control the NN are extended with radial base function to produce accurate results. In load and time forecasting, RBF is being used to increase the accuracy of results. In [26], two types of neural networks are fussed together i.e., Radial Base function (RBF) and Adaptive Neuro-Fuzzy Interface System (ANFIS). Results showed improvements when compared with non-hybrid models. This [27] finds out that there is no need to train the classifier recursively, but data selection for training should be selected wisely. GRNN-based forecaster is used and performed better in accuracy and computational time when compared with NN.

Table 1: Performance Analysis of Different Forecasters

| Forecasters Main Component | Accuracy | Convergence Rate | Remarks |
|---|---|---|---|
| ANN Based forecasters [13-15] | High | Slow to Moderate | Convergence rate need improvement |
| Block Model Based [16-18] | High | Slow | Convergence rate need improvement |
| CNN based forecasters [19-23] | High | Slow | Overfitting can be avoided |
| RNN based forecasters [24-25] | Moderate | Slow to Fast | Accuracy can be increased further |
| RBF NN based forecasters [26-27] | High | Slow | Overfitting can be avoided |
| Least square support vector machine-based forecasters [28-29] | High | Very Slow | High algorithmic complexity causing very slow convergence |
| Hybrid Greedy Wolf and Differential Evolution based forecaster [30] | High | Slow | Convergence rate need improvement |
| Dynamic Mode Decomposition based forecaster [31] | Low | Fast | Early convergence. Accuracy can be increased |
| Rf-based forecaster [32] | High | Fast | Very limited scope of predictions. |

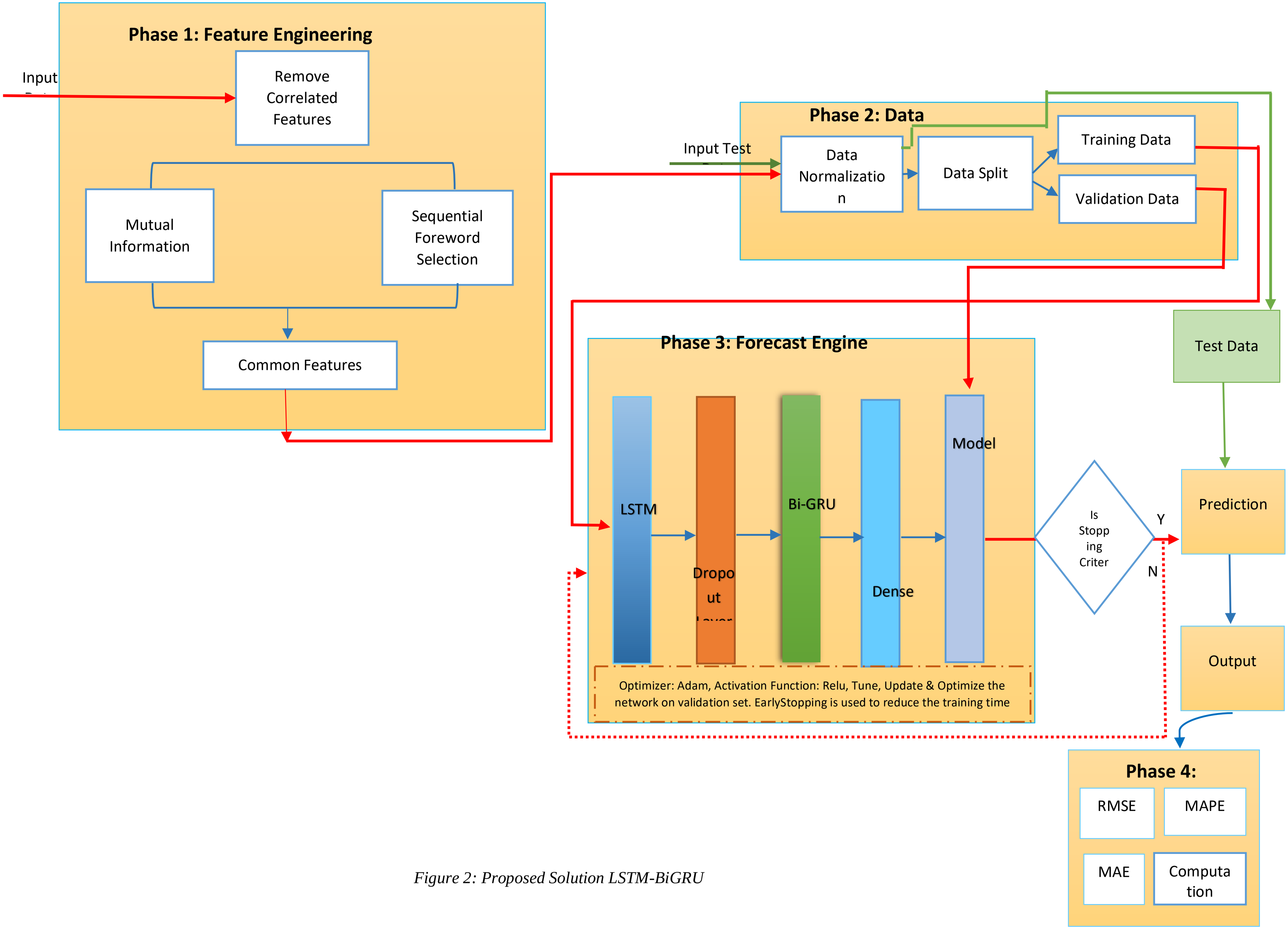


*Figure 2: Proposed Solution LSTM-BiGRU*

Table [1] compares different algorithms with reference to their main forecaster, some of them are hybrid while some have data preprocessing section. We discuss comparing a list on the base of their cumulative accuracy and convergence rate.

## III. PROPOSED SOLUTION

The proposed solution considers all the factors discussed above. It is divided into 4 phases.

1. *Phase 1: Data Preprocessing*
2. *Phase 2: Data Determination*
3. *Phase 3: Forecast Engine*
4. *Phase 4: Evaluation*

Consider the block diagram of the proposed multilayer solution to understand all these phases.

The dataset is input in the data preprocessing phase. Here we removed the correlated features and performed feature selection techniques. We performed MI and SFS algorithms, to select the common features from them. The selected features are used as input in the Data Determination phase, here we undergo data scaling and splitting in train-test-validate datasets. We used a separate dataset for testing. The split data enters the multilayered LSTM-BiGRU forecasting engine. Here model training and validation are done. Now we test the trained model on test data and calculate the error matrices. Let's discuss each phase in detail

### 1) *Phase 1: Data Preprocessing*

Correlated features are the features that measure strength between multiple features. We removed the correlated features because the features that are showing strong correlation between them are the features that are providing almost the same information for the forecast [33]. So having the same information repeatedly may not always increase the accuracy of forecasting but can decrease it. We utilized eq (1) for this.

$$corr = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \tag{1}$$

$$MI(x,y) = \sum_x \sum_y p(x,y) log_2 \left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{2}$$

In correlation, we measure the linear relationships between variables while in Mutual Information (MI) eq (2) we measure the nonlinear relationships between variables. MI considers dependencies between variables that are not detected by covariance [34]. Sequential Feature Selection (SFS) belongs to the family of greedy search algorithms. SFS reduces the initial d-dimensional features to k<d. SFS helps in automatically selecting the best features from the pool of features. By removing the irrelevant features or noise, SFS helps in the reduction of generalization error. Also, it is computational active [35]. We select the common features which came out as output from these two mentioned algorithms, these were 3 in number. We have designed the whole data preprocessing technique to get the best features. If we have the best features selected only then we can make accurate forecasting results.

### 2) *Phase 2: Data* Determination

Data determination is basically data preparation. In this phase, we prepare the data to enter the next phase (forecast engine) as deep neural networks require data in specified format and shape [36]. For this, we scale the data using MinMax scaling and bounds the values in a certain range 0 and 1 [37]. We performed MinMax Scaling as follows:

$$y = \frac{(x - \min (x))}{(\max (x) - \min (x))} \tag{3}$$

We split the dataset in a 7:3 ratio. We used 70% data in training and 30% in validation of that training model. Whereas we used separate data for testing purposes.

### 3) *Phase 3: Forecast Engine*

The proposed forecast engine is designed in Python using Keras. It's a multilayered model utilizing different LSTM and GRU layers. Let's discuss LSTM and GRU briefly:
Both LSTM and GRU are variations of RNN. RNN is the state-of-the-art algorithm for sequential or time series data. They were created in the 1980s [38]. But they have vanishing gradients or long short-term memory problems. To solve this problem, LSTM, GRU, and many other RNN variations were seen.

#### *a) Long Short-Term Memory (LSTM)*

LSTM is a refined variation of RNN, addressing the problem of vanishing gradient. It was introduced by Hochreiter and Schmidhuber in 1997. It works on the backpropagation principle as it must calculate gradients for the process optimization. It changes weight according to the error rate it calculates at each cell level. LSTM is capable enough to learn long-term dependencies for a long time using its memory unit [39].
The key component of the LSTM is the cell state. It runs straight down the entire time steps with only minor but important interactions. LSTM can add or remove information from the cell state using several gates. Each gate is made of a sigmoid neural network layer. These sigmoid layers produce output numbers between 0 and 1, which represents how much information each component should be let through. 0 means nothing through the layers whereas 1 represents letting everything through 3 layers out of the four are used to control the cell state tanh.
Consider the following diagram to understand the LSTM architecture. LSTM consists of three functions of gate controllers.

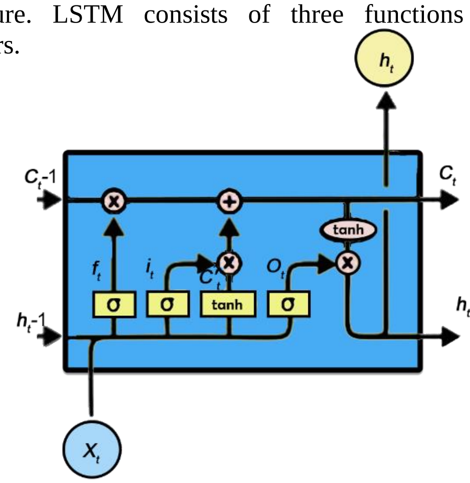


*Figure 3: Long Short-Term Memory (LSTM)*

•Forget gate $f_t$ decides which part of long-term state $C_t$ should be omitted.
• Input gate $i_t$ controls which part of $C_t$ should be added to long-term state *ct*
• Output gate $O_t$ determines which part of $C_t$ should be read and Outputs to $h_t$ and $O_t$.

In the above figure, $x_t$ is the input into the LSTM cell, ht-1 is the output of the previous cell and ct-1 is the cell state that is received by the current cell. It helps in the prediction of the current cell. First gate is the forget gate, the equation is below:

$$f_t = \sigma\left(W_f.\ [h_{t-1}, x_t] +\ b_f\right) \tag{4}$$

The sigmoid of the multiplication of the input added with the bias value happened here. This layer helps in returning 0 and 1, whether we need this information in prediction or not.

The next gate is input gate, consider the below mentioned equation.

$$i_t =\ \sigma(W_i.\ [h_{t-1}, x_t] +\ b_i) \tag{5}$$
$$\tilde{C}_t =\ tanh(W_C.\ [h_{t-1}, x_t] +\ b_C) \tag{6}$$
$$C_t =\ f_t * C_{t-1} +\ i_t *\ \tilde{C}_t \tag{7}$$

The first part of the input layer equation is like the equation of forget gate, except by the weight and bias. It undergoes a sigmoid function. In the next two equations, it controls which part to add as cell state using tanh function.

The mechanism of LSTM can be broken down into 3 stages. First of which is the decision of what information is to be extracted from the cell state. This is done by the sigmoid layer also known as the $f_t$ forget gate. It observes $h_t$-1 and $x_t$ from the last step performed, to produce an output range between 0 and 1. The next stage of this process is to what information will be stored in the cell state. The sigmoid layer named as output gate $O_t$ determines the values that needs updating. Afterwards, a new vector of the proposed values is created by tanh layer. These values are termed as $C_t$ and are added to cell state. Then old cell state $C_t$-1 needs to be updated into the new cell state Ct. The last stage is to determine what values are system is going to provide as the output. The output depends on the cell state yet a sifted edition of it. First the sigmoid layer chooses what parts of the cell state will be introduced as output. Then, at that point the cell state is put through the tanh function to change over the qualities between - 1 and 1, the resultant of which is them multiplied with sigmoid layers output to get the result [39]. The mathematical equations for this stage are:

$$o_t =\ \sigma(W_o.\ [h_{t-1}, x_t] +\ b_o) \tag{8}$$
$$h_t = o_t * \tanh(C_t) \tag{9}$$

So that's how LSTM works, as discussed above, we use GRU in combination with LSTM. Let's discuss GRU in detail:

### *b) Gated Recurrent Unit (GRU)*

Gated Recurrent Units become the most promising algorithm and were introduced in 2014 by Cho et al. It solves the problem of vanishing gradient. GRU is considered as the variation of LSTM. Both these algorithms provide the best results in certain scenarios.

To understand the architecture of GRU [40], consider the following figure. It consists of three sigmoid layers, namely: update gate, reset gate, and tanh layer. Consider the attached diagram to better understand the equations. GRU uses the update gate and reset gate for vanishing gradient problems and these help in deciding the output as well. Let us discuss each gate below:

The initial point of this algorithm is update gate. First, the following formula calculates the update gate $z_t$ at time interval t:

$$z_t = \sigma\left(W^{(z)}x_t +\ h_{t-1}\right) \tag{10}$$

Where $x_t$ is added to product h(t-1) and its weight. Afterwards, a sigmoid function normalizes the resultant between 0 and 1. This determines the required amount of past information to pass along for the future time step with the help of update gate.

The following equation computes the reset gate rt, at time step t:

$$r_t = \sigma\left(W^{(r)}x_t +\ h_{t-1}\right) \tag{11}$$

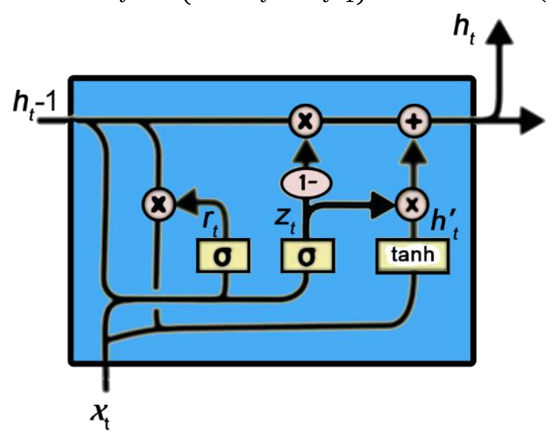

*Figure 4: Gated Recurrent Unit (GRU)*

Calculation starts when $x_t$ is added to product h(t-1) and its weight. Then, at that point a sigmoid function is utilized to change over the output between the worth 0 and 1. Reset gate assists the model with deciding the amount of the past data should be neglected.

This is engaged with the reset gate. This begins with presenting another memory content that will utilize the reset gate and store the important data from an earlier time. The numerical condition is as per the following:

$$h'_t = \tanh\,(Wx_t + r_t \odot h_{t-1}) \tag{12}$$

The estimation begins with the augmentation of the information xt with its weight. Then the element-wise multiplication is done to the reset gate rt and the preview output ht-1. This permits us to just pass the significant past data. Then, at that point both determined outcomes are added together, and a tanh function is applied.

Lastly, the unit needs to figure the $h_t$ vector which holds data for the current unit, and it will pass the data further down to the network. The update gate zt assumes a critical part in this. The numerical equation for this is:

$$h_t = z_t \odot\ h_{t-1} + (1 - z_t) \odot h'_t \tag{13}$$

From the computation, if the vector $z_t$ is near 0, a major piece of the current substance will be disregarded since it is unimportant for the forecast. Simultaneously, since $z_t$ will be near 0 right now step, 1-zt will be near 1, permitting most of the past data to be kept [40].

*c) LSTM-BiGRU Architecture*

Consider the model from left to right. The model consists of layers of LSTM and GRU considering feed-forward and bidirectional layers to better train and accurate forecasting the electrical load and price values. Input normalized features feed to the first layer is the LSTM layer, and its training input features in a feed-forward manner. GRU layer applied in the bidirectional layer. First, it undergoes training in a feed-forward manner than in a feed backward manner. The hybrid layer containing layers of both LSTM and GRU in a specified manner is enhancing the accuracy of predictions of the forecasting model, respectively. The LSTM Layer is followed by a dropout layer to avoid overfitting, followed by a bidirectional GRU layer. The dense layer at the end is receiving the output from all the input neurons and is connected deeply.

The model training is evaluated on the MSE calculated at each iteration. This decrease in MSE is stopped after a certain number of iterations and then it becomes constant, with no further decrease. This is the point when our model is fully trained with the training dataset. To stop further iterations to occur, we have used the EarlyStopping criterion [41]. EarlyStopping is basically used to stop iterations when the MSE error further stops decreasing. So instead of executing a fixed number of iterations model usually stops training after a certain number of iterations or when the error stops decreasing. This helps in decreasing the training time, and the composition of the hybrid model helps in better training and thus more accurate predictions.

Accuracy and convergence rate are inversely proportional to each other. In order of increasing accuracy, we usually observe the convergence rate very slow. But in predicting electricity load and price value the convergence rate and computational time are very crucial. We must maintain a balance between accuracy and convergence rate, so we proposed this method. The multiple layered and directional training and early stopping both are satisfying to solve the problem statement.

After forecasting the load and price values, the next step is to calculate the error between the actual and predicted values. This error calculation will help us in validating the model.

*4) Phase 4: Evaluation*

For evaluation of the proposed solution, we calculated RMSE, MAE, MAPE and computational time. All these evaluation metrices [42] are calculated by using the below mentioned equations.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{14}$$

$$MAPE = \frac{100\%}{N} \sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{15}$$

$$MAE = \sum_{i=1}^{N} \frac{(\hat{y}_i - y_i)^2}{n} \tag{16}$$

$$start = timer.start\,() \tag{17}$$

$$end = timer.end\,() \tag{18}$$

$$Time\ Elapsed(s) = end - start \tag{19}$$

For computing the computational time, we used python library. We imported default_timer from timeit library

## IV. Results and Discussion

The dataset is taken from ISO-NE CA [43]. We have considered 2016-2018 data for training the model and tested it on the 2019 dataset. The dataset contains both the target variable i.e Load and Price values along with many other features.

We have selected first 5 features with highest MI and SFS values and then take a common of those. They were 3 in number. These are as follows

'DA_LMP', 'System_Load', 'Dew_Point'

The proposed model LSTM-BiGRU is trained with below mentioned parameters.

*Table 2: List of Optimized Hyper Parameters Selected during model building*

| | |
|---|---|
| Training Batch Size | 70% (24,544 rows) |
| Validation Batch Size | 30% (10,519 cols) |
| Batch size | 1024 |
| Activation Function | ReLU |
| Optimizer | Adam |
| Loss | 0.18 to 0.01 |
| Dropout | 0.2 |
| Number of LSTM Units | 100 |
| Number of Bidirectional GRU | 50 |

The attached training loss graph is based on MSE. It can be seen in the graph that the training MSE has decreased from 0.20 to below 0.04 and then it became stable. Similarly, on validation set the MSE value decreases from 0.11 to below 0.03.

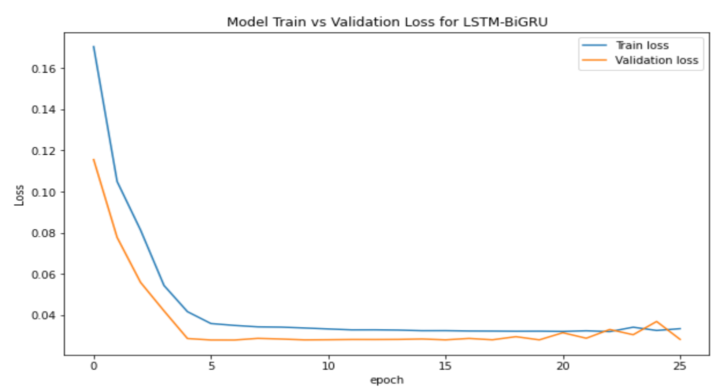


*Figure 5:Training Loss graph for Load Forecasting*

Both the cures are seen very near to each other that means error values are least at that time. The use of EarlyStopping criterion that makes the model to stop further training after 25 iterations. This helps us in decreasing the training time and composition of the model helps in robust and more precise forecasting.

Consider the following graph, solid black graph representing the actual forecast values while green is representing the proposed LSTM-BiGRU model forecast. We have compared our proposed solution with benchmark solutions including LSTM, GRU and SVR. The LSTM ad GRU composition is multilayered and similar to the proposed solution. In both the graphs it is clearly seen that the proposed LSTM-BiGRU is performing better fig (6) than the comparison benchmark solution.

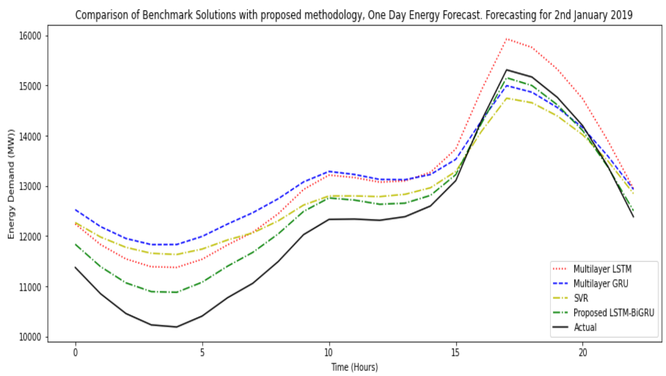


*Figure 6: Comparison graph for Load Forecast 2nd January,2019. LSTM-BiGRU is performing better than all the comparison models*

The results of proposed solution have an accuracy of 96.8% in January graph for 1 day forecasting Table 3, respectively.

*Table 3: 2nd January 2019 Load Forecast Error Metrics*

| 2nd January 2019 Load Forecast Error Metrics | | | | |
|---|---|---|---|---|
| Model | Hours | MAPE | MAE($/MWH) | RMSE ($/MWH) |
| LSTM | 1-24 | 6.74% | 739.48 | 829.52 |
| GRU | 1-24 | 7.53% | 853.47 | 1003.32 |
| SVR | 1-24 | 5.94% | 677.01 | 795.90 |
| LSTM-BiGRU | 1-24 | 3.12% | 355.38 | 419.49 |

Similarly, for 1 day January forecast the MAE and RMSE calculated 355.38 and 419.49 that's lowest of comparison models.

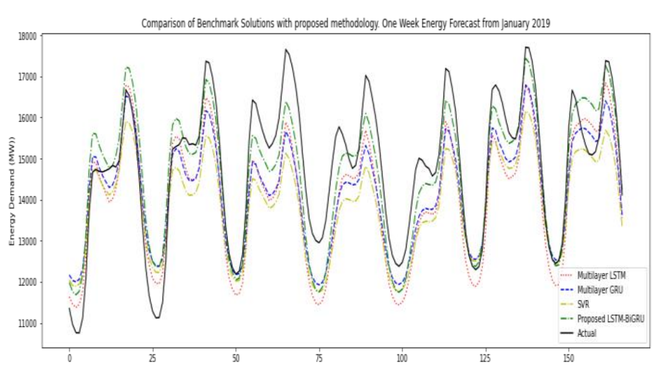


*Figure 7: Comparison graph for Load Forecast 1st Week January 2019. At some hours comparison algorithms seems working good, however LSTM BiGRU achieve overall better results.*

model. Figure 7 demonstrating 1 week energy forecast, and it is clearly seen that the proposed solution is performing better than the comparison models. Consider Table 4 for evaluation metrics:

*Table 4:1st Week January 2019 Load Forecast Error Metrics*

| 1st Week January 2019 Load Forecast Error Metrics | | | | |
|---|---|---|---|---|
| Model | Hours | MAPE | MAE($/MWH) | RMSE ($/MWH) |
| LSTM | 1-168 | 6.15% | 910.65 | 1029.53 |
| GRU | 1-168 | 5.58% | 829.72 | 983.23 |
| SVR | 1-168 | 7.04% | 1062.81 | 1238.55 |
| LSTM-BiGRU | 1-168 | 4.07% | 585.57 | 711.422 |

The second scenario is summers forecast. For this we forecasted July one day and one week energy. Please consider Figure 8 for one day July forecast.

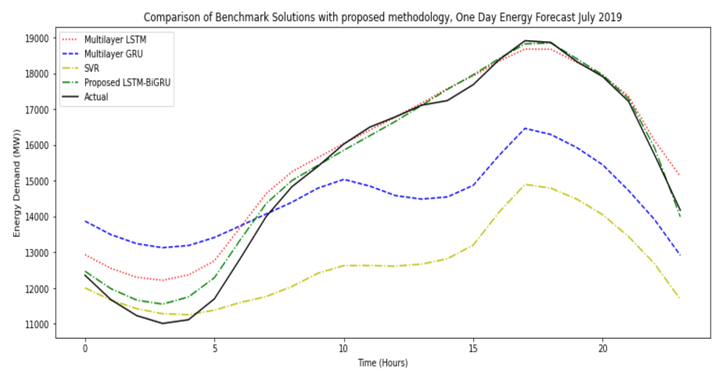


*Figure 8: Comparison graph for Load Forecast 2nd July 2019 LSTM-BiGRU is performing better than all the comparison models*

The proposed solution showing 98.2% accurate results. Consider Table 5 for comparison values. Proposed solution's MAE and RMSE achieved lowest of all models, i.e., 231.72 and 299.23, respectively.

*Table 5: 2nd July 2019 Load Forecast Error Metrics*

| 2nd July 2019 Load Forecast Error Metrics | | | | |
|---|---|---|---|---|
| Model | Hours | MAPE | MAE($/MWH) | RMSE ($/MWH) |
| LSTM | 1-24 | 3.58% | 456.93 | 617.64 |
| GRU | 1-24 | 12.15% | 1848.52 | 2000.17 |
| SVR | 1-24 | 16.19% | 2694.62 | 3143.06 |
| LSTM-BiGRU | 1-24 | 1.76% | 231.72 | 299.26 |

For summers weekly forecast considers figure 9. The error matrices calculated below:

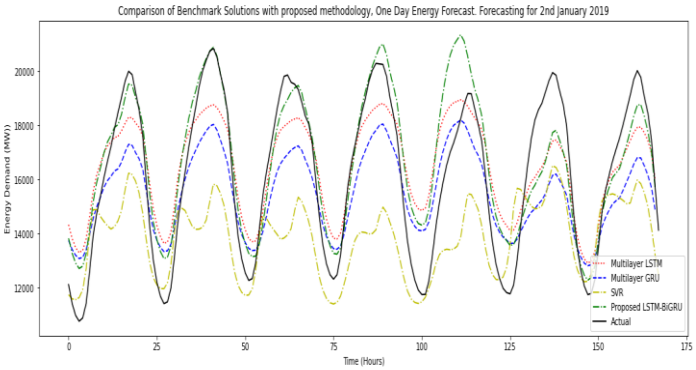


*Figure 9: Comparison graph for Load Forecast 1st Week July 2019. LSTM BiGRU achieves better results in accuracy in almost 6 days of the week in comparison to the other models*

The results for July weekly forecasts 92.5% accurate results are recorded. MAE and RMSE values are lowest of all the comparison models as seen in the table 6.

*Table 6: 1st Week July 2019 Load Forecast Error Metrics*

| 1st Week July 2019 Load Forecast Error Metrics | | | | |
|---|---|---|---|---|
| Model | Hours | MAPE | MAE ($/MWH) | RMSE($/MWH) |
| LSTM | 1-168 | 9.02% | 1353.5 | 1574.06 |
| GRU | 1-168 | 9.85% | 1605.21 | 1843.82 |
| SVR | 1-168 | 14.73% | 2573.35 | 3067.18 |
| LSTM-BiGRU | 1-168 | 7.42% | 1110.89 | 1400.11 |

Considering figure 10 in both the scenarios i.e., January and July forecast for both one day and one week the computational values recorded are 40.7 and 52.2 secs. The recorded computational time is lesser than the comparison graphs.

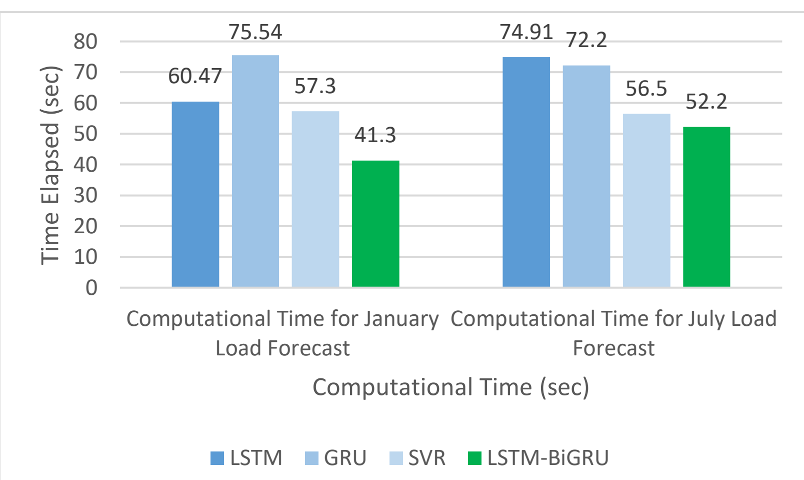


*Figure 10: Model Computational Time for Load Forecasting*

all the four phases discussed above, for all scenarios and comparison models.

So, for load forecasting, we performed forecasting on two scenarios i.e. winters and summers forecasting for daily and weekly. And the proposed solution achieved best MAPE 3.12%, 4.07%, 1.76% and 7.42% in only 40.7 and 52.2 seconds.

For price forecasting, the dataset showed a lot of noise and fluctuations. For price forecasting, we performed the feature engineering similar to load forecasting.
We select features from both section methods described above. We select common features from the feature selecting methods. These are as follows

"DA_LMP, RT_EC, RT_MLC"

Where DA_LMP is day ahead locational marginal price, RT_EC is energy component of the real time price and RT_MLC is Marginal loss component of the real time price. We have trained the proposed model with same hyper parameters. The training loss is calculated on MSE and that decreased from 0.025 to below 0.004.

We performed price forecasting for the same scenarios as for load forecasting. We performed price forecasting for winters and summers. For winters, we used 2nd January and 1st week of January for forecast and for summers we selected 2nd July and 1st week of July from 2019 or test dataset as discussed. Consider the figure 11, the black line represents the actual price values and green one represents the forecasted values. The proposed solution works better than the comparison model with too much fluctuating data.

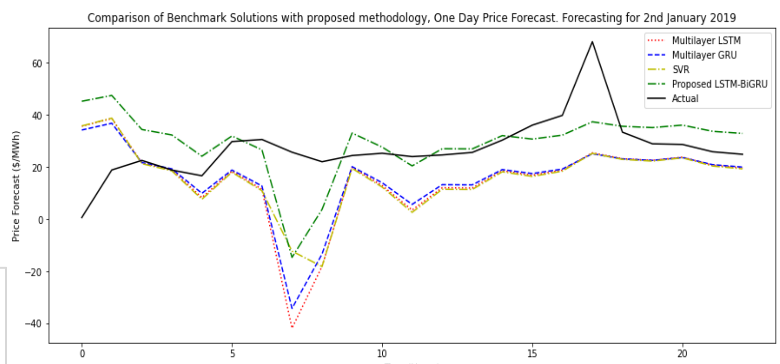


*Figure 11: Comparison graph for Price Forecast 2nd January,2019. LSTM-BiGRU is performing better than all the comparison models*

The error metrics calculated in Table 7 as follows:

*Table 7: 2nd January 2019 Price Forecast Error Metrics*

| 2nd January 2019 Price Forecast Error Metrics | | | |
|---|---|---|---|
| Model | Hours | MAE($/MWH) | RMSE ($/MWH) |
| LSTM | 1-24 | 17.07 | 23.12 |
| GRU | 1-24 | 15.91 | 21.38 |
| SVR | 1-24 | 16.14 | 20.2 |
| LSTM-BiGRU | 1-24 | 2.35 | 16.85 |

The
proposed solution for 1 week January forecast is represented

(Figure 12) with green line and achieves the better accuracy as compared the other comparison models.

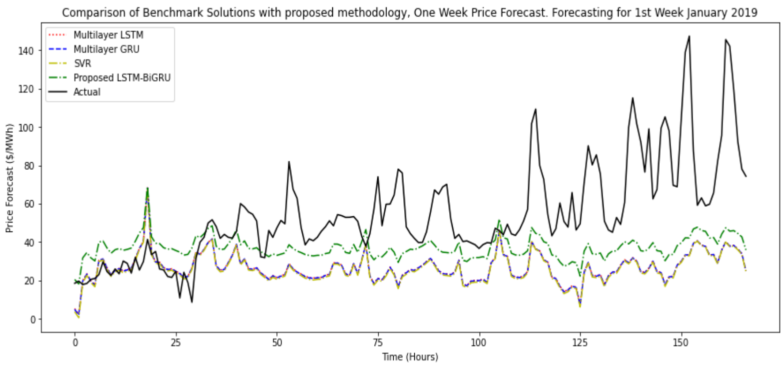


*Figure 12: Comparison graph for Price Forecast 1st Week January 2019. LSTM-BiGRU is performing better than all the comparison models. All other comparison models forecasted very close values*

The performance metrics Table 8 is below:

*Table 8: 1st Week January 2019 Price Forecast Error Metrics*

| 1st Week January 2019 Price Forecast Error Metrics | | | |
|---|---|---|---|
| Model | Hours | MAE($/MWH) | RMSE ($/MWH) |
| LSTM | 1-168 | 29.47 | 37.68 |
| GRU | 1-168 | 29.58 | 37.78 |
| SVR | 1-168 | 30.04 | 38.20 |
| LSTM-BiGRU | 1-168 | 22.63 | 30.79 |

So here we have calculated MAE and RMSE only. And it is seen that the calculated values for MAE and RMSE is lowest than the comparison model 22.63 and 30.76. These are 168hr price forecast in winter season. LSTM-BiGRU calculated lowest MAE and RMSE value, respectively.

Let's consider the final price forecast graphs for summers season. One day summer price forecast is as follows: Multilayered GRU produced the closest forecast to the LSTM-BiGRU forecast.

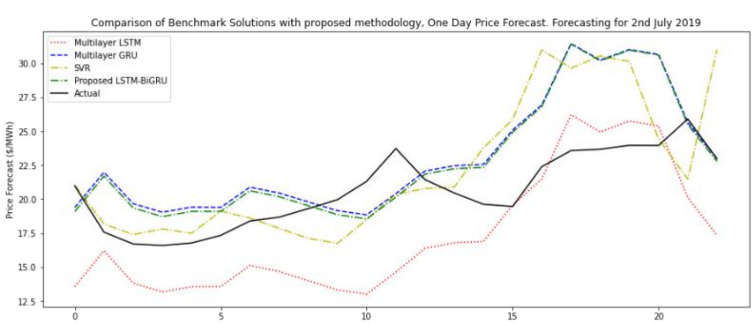


*Figure 13: Comparison graph for Price Forecast 2nd July 2019. Multilayered GRU and SVR forecast are very close to the actual values. However, LSTM-BiGRU is performing better in accuracy slightly than the comparison models*

However, the proposed solution shows slight accuracy and lesser computational time. MAE and RMSE calculated 3.02 which is very less than the rest.

| 2nd July 2019 Price Forecast Error Metrics | | | |
|---|---|---|---|
| Model | Hours | MAE ($/MWH) | RMSE ($/MWH) |
| LSTM | 1-24 | 3.89 | 4.55 |
| GRU | 1-24 | 3.11 | 3.84 |
| SVR | 1-24 | 3.04 | 4.06 |
| LSTM-BiGRU | 1-24 | 3.02 | 3.02 |

*Table 9: 2nd July 2019 Price Forecast Error Metrics*

Price forecasting for 1 week in July graph is attached figure 14. The simulation results showed better accuracy of the proposed model. The calculated evaluation metrices showed that the proposed solution has MAE and RMSE is 9.60 and 15.40 and that is better performing than the comparison model

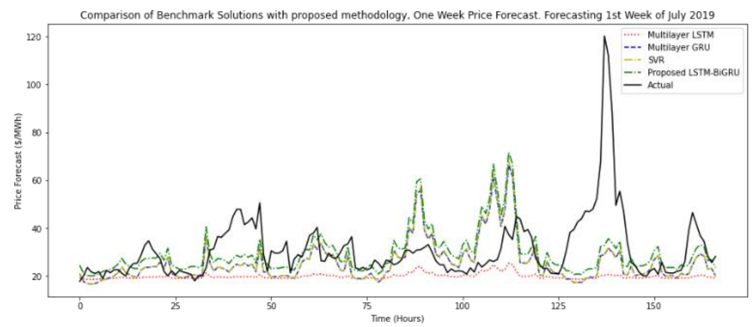


*Figure 14: Comparison graph for Price Forecast 1st Week July 2019. Multilayered GRU and SVR forecast are very close to the actual values. However, LSTM-BiGRU is performing better in accuracy slightly than the comparison models*

The performance matrices Table 10 is below

*Table 10: 1st Week July 2019 Price Forecast Error Metrics*

| 1st Week July 2019 Price Forecast Error Metrics | | | |
|---|---|---|---|
| Model | Hours | MAE ($/MWH) | RMSE ($/MWH) |
| LSTM | 1-168 | 11.14 | 17.54 |
| GRU | 1-168 | 10.13 | 16.04 |
| SVR | 1-168 | 10.22 | 16.12 |
| LSTM-BiGRU | 1-168 | 9.60 | 15.40 |

The computational time graph for price forecasting is attached figure 15. Due to multilayered model composition and Stopping criterion, we managed to present the best accurate results in minimum possible time.

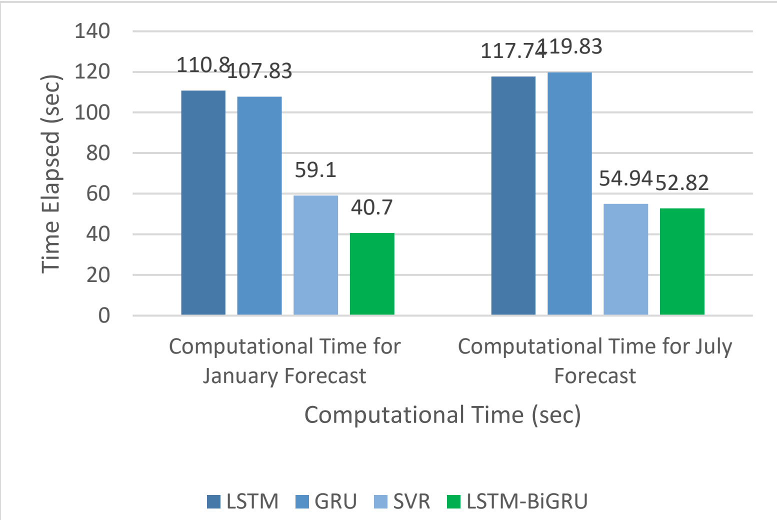


*Figure15: Model Computational Time for Price Forecasting*

As seen in the graph attached, the proposed solution took very little time i.e., less than a minute.

The proposed solution outperformed all the comparison model, without creating a tradeoff in accuracy and computational time. The green bar represents the proposed solution predicting accurate results in both scenarios of summer and winter forecast in 407 and 52.8secs only.

## V. Conclusion and Future Work

In SGs, STLF and STPF is very important as they have direct impact on the planning schedules of utilities. These forecasting have strong effect on the energy market.

In this work, the importance of short-term load forecasting is discussed and analyzed for maintaining the stability between generation, transmission, and consumer end. As discussed in Chapter 5, due to high volatility in the historical load curves, STLF/STLP in SGs become more challenging when it comes to forecast for a longer time duration/ time series. Only electricity load and price values are not sufficient in accurate forecasting, we must consider other features too. We have discussed the details of all features in Chapter 5. We have designed the Feature selection module separately so that we can only get the best feature out of the pool, as all features are not adding any significant part in forecasting and sometimes, they only decrease the accuracy and enhances the computation time. Considering limitations of LSTM, including LSTMs require more memory to train, easy to over fit, and LSTMs are sensitive to different random weight initializations. We consider all these when implementing our hybrid model. For memory issues, we have not jumped on the larger dataset, we have used a small to medium sized dataset and from adjusting weights in the dropout layer we made LSTM to not undergo over fitting in discussed cases. We do consider limitations of GRU, their slow convergence and low learning efficiency. We mitigate the limitation of low learning efficiency using its bidirectional layer in combination with LSTM, and slow convergence using EarlyStopping criterion. The proposed model significantly reduced the execution time and enhanced the forecast accuracy as discussed. Moreover, ReLU activation function enable the forecast strategy to capture non-linearity's in the time series. Tests are conducted on ISO NE CA dataset that contains hourly load and price values besides other 18 features. Results show that the proposed model achieves relatively better forecast accuracy (96.9%) in comparison to other models i.e., LSTM, GRU and SVR. Moreover, improvement in forecast accuracy is achieved while not paying the cost of slow convergence rate [13]. Thus, the trade-off between convergence rate and forecast is not created. Finally, from application perspective, the proposed model can be used by utilities to launch better offers in the electricity market. The proposed solution is showing MAPE in January 2019 load forecasting from 3.12% to 4.07%. The MAE is 355.28 that is very less than the comparison models. Similarly, in July the MAPE error calculated in load forecasting is 1.76% and the MAE is 231.72, and these are again the best results achieved than the comparison solution. This means that the utilities can save significant amount of money due to better adjustment of their generation and demand schedules simply because of high accuracy of the proposed model. The proposed solution is showing MAE in January 2019 price forecasting from 2.28 to 2.35. The MAE calculated is very less than the comparison models. Similarly, in July the MAE calculated is 0.87 to 1.10, and these are again the best results achieved than the comparison solution. The objective of this research is to predict the future short term electricity demand and price values on hourly basis. We achieved this goal using historical data set of 3 years. This forecasted values not only helps power companies but on the other hand help users to use electricity according to the hourly price predicted and thus can manage their high load consumption activities accordingly.

In future the proposed method can be improved by other techniques i.e., block chains or more powerful neural networks. Optimization of proposed hybrid algorithms can help in better results. There are many evolutionary algorithms that can predict better results. We can increase the scope from STLF/STPF to at least MTLF/MTPF. With the use of GPUs and TPUs we can decrease the computational time or by designing a simple network can also help in reduction of computational time further.

## References


[1] Hafeez, G., et al., *A novel hybrid load forecasting framework with intelligent feature engineering and optimization algorithm in smart grid.* Applied Energy, 2021. **299**: p. 117178.

[2] Rai, S. and M. De, *Analysis of classical and machine learning based short-term and mid-term load forecasting for smart grid.* International Journal of Sustainable Energy, 2021: p. 1-19.

[3] Zhang, W., et al., *A novel asynchronous deep reinforcement learning model with adaptive early forecasting method and reward incentive mechanism for short-term load forecasting.* Energy, 2021: p. 121492.

[4] Moradzadeh, A., et al., Deep Learning-Assisted Short-Term Load Forecasting for Sustainable Management of Energy in Microgrid. Inventions, 2021. **6**(1): p. 15.

[5] Guo, W., et al., *Machine-Learning based methods in short-term load forecasting.* The Electricity Journal, 2021. **34**(1): p. 106884.

[6] Eskandari, H., M. Imani, and M.P. Moghaddam, *Convolutional and recurrent neural network based model for short-term load forecasting.* Electric Power Systems Research, 2021. **195**: p. 107173.

[7] Shirzadi, N., et al., Medium-term regional electricity load forecasting through machine learning and deep learning. Designs, 2021. **5**(2): p. 27.

[8] Farrag, T.A. and E.E. Elattar, *Optimized Deep Stacked Long Short-Term Memory Network for Long-Term Load Forecasting.* IEEE Access, 2021. **9**: p. 68511-68522.

[9] Zhang, J., et al. *A Novel Wavelet Neural Network Load Forecasting Algorithm with Adaptive Momentum Factor*. in *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. 2021. IEEE.

[10] Hafeez, G., et al., *A novel accurate and fast converging deep learning-based model for electrical energy consumption forecasting in a smart grid.* Energies, 2020. **13**(9): p. 2244.

[11] Hafeez, G., K.S. Alimgeer, and I. Khan, Electric load forecasting based on deep learning and optimized by heuristic algorithm in smart grid. Applied Energy, 2020. **269**: p. 114915.

[12] Fekri, M.N., et al., Deep learning for load forecasting with smart meter data: Online adaptive recurrent neural network. Applied Energy, 2021. **282**: p. 116177.

[13] hmad, A., et al., *Short-term load forecasting in smart grids: an intelligent modular approach.* Energies, 2019. **12**(1): p. 164.

[14] Lu, R. and S.H. Hong, *Incentive-based demand response for smart grid with reinforcement learning and deep neural network.* Applied energy, 2019. **236**: p. 937-949.

[15] Mujeeb, S., et al., Deep long short-term memory: A new price and load forecasting scheme for big data in smart cities. Sustainability, 2019. **11**(4): p. 987.

[16] Ghadimi, N., et al., *Two stage forecast engine with feature selection technique and improved meta-heuristic algorithm for electricity load forecasting.* Energy, 2018. **161**: p. 130-142.

[17] Gao, W., et al., *Different states of multi-block based forecast engine for price and load prediction.* International Journal of Electrical Power & Energy Systems, 2019. **104**: p. 423-435.

[18] Ghadimi, N., et al., *A new prediction model based on multi-block forecast engine in smart grid.* Journal of Ambient Intelligence and Humanized Computing, 2018. **9**(6): p. 1873-1888.

[19] Muralitharan, K., R. Sakthivel, and R. Vishnuvarthan, *Neural network based optimization approach for energy demand prediction in smart grid.* Neurocomputing, 2018. **273**: p. 199-208.

[20] Khan, A.B.M., et al. *Hourly Electricity Load Forecasting in Smart Grid Using Deep Learning Techniques*. in *International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. 2019. Springer.

[21] Zahid, M., et al., *Electricity price and load forecasting using enhanced convolutional neural network and enhanced support vector regression in smart grids.* Electronics, 2019. **8**(2): p. 122.

[22] Aimal, S., et al. An Efficient CNN and KNN Data Analytics for Electricity Load Forecasting in the Smart Grid. in Workshops of the International Conference on Advanced Information Networking and Applications. 2019. Springer.

[23] Naz, A., et al., *Short-term electric load and price forecasting using enhanced extreme learning machine optimization in smart grids.* Energies, 2019. **12**(5): p. 866.

[24] Chitsaz, H., et al., Short-term electricity load forecasting of buildings in microgrids. Energy and Buildings, 2015. **99**: p. 50-60.

[25] Cai, M., M. Pipattanasomporn, and S. Rahman, *Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques.* Applied energy, 2019. **236**: p. 1078-1088.

[26] Lajevardy, P., et al., *A hybrid method for load forecasting in smart grid based on neural networks and cuckoo search optimization approach.* International Journal of Renewable Energy Resources, 2017. **5**(1): p. 13-20.

[27] Paulos, J.P. and J.N. Fidalgo. Load and electricity prices forecasting using Generalized Regression Neural Networks. in 2018 International Conference on Smart Energy Systems and Technologies (SEST). 2018. IEEE.

[28] Shayeghi, H., et al., Simultaneous day-ahead forecasting of electricity price and load in smart grids. Energy conversion and management, 2015. **95**: p. 371-384.

[29] Ghasemi, A., et al., A novel hybrid algorithm for electricity price and load forecasting in smart grids with demand-side management. Applied energy, 2016. **177**: p. 40-59.

[30] Naz, M., et al., Efficient power scheduling in smart homes using hybrid grey wolf differential evolution optimization technique with real time and critical peak pricing schemes. Energies, 2018. **11**(2): p. 384.

[31] Mohan, N., K. Soman, and S.S. Kumar, *A data-driven strategy for short-term electric load forecasting using dynamic mode decomposition model.* Applied energy, 2018. **232**: p. 229-244.

[32] Lahouar, A. and J.B.H. Slama, *Day-ahead load forecast using random forest and expert input selection.* Energy Conversion and Management, 2015. **103**: p. 1040-1051.

[33] Kong, X., et al., *Short-term electrical load forecasting based on error correction using dynamic mode decomposition.* Applied Energy, 2020. **261**: p. 114368.

[34] Samuel, O., et al., *Towards modified entropy mutual information feature selection to forecast medium-term load using a deep learning model in smart homes.* Entropy, 2020. **22**(1): p. 68.

[35] Wahab, A., et al., A Novel Technique for Short-Term Load Forecasting using Sequential Models and Feature Engineering. IEEE Access, 2021.

[36] Massaoudi, M., et al., *A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for Short-Term Load Forecasting.* Energy, 2021. **214**: p. 118874.

[37] Ioannis, A.A., ShortTerm Load Forecasting Using Artificial Neural Networks. 2021.

[38] Jagait, R.K., et al., *Load Forecasting Under Concept Drift: Online Ensemble Learning with Recurrent Neural Network and ARIMA.* IEEE Access, 2021.

[39] Dudek, G., P. Pełka, and S. Smyl, A hybrid residual dilated LSTM and exponential smoothing model for midterm electric load forecasting. IEEE Transactions on Neural Networks and Learning Systems, 2021.

[40] Jung, S., et al., An Attention-Based Multilayer GRU Model for Multistep-Ahead Short-Term Load Forecasting. Sensors, 2021. **21**(5): p. 1639.

[41] Aguilar Madrid, E. and N. Antonio, *Short-term electricity load forecasting with machine learning.* Information, 2021. **12**(2): p. 50.

[42] Singh, S. and A. Mohapatra, *Data driven day-ahead electrical load forecasting through repeated wavelet transform assisted SVM model.* Applied Soft Computing, 2021: p. 107730.

[43] Dataset from ISONE CA https://www.iso-ne.com/isoexpress/web/reports/pricing/-/tree/zone-info