Optimal Solution for Load and Price Forecasting in Smart Grids

By

Aneeza Khalid

00000273406

Supervisor

Dr. Sohail Iqbal

Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Information Technology (MS-IT)

In

School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), Islamabad, Pakistan.

(January 2022)

Approval

It is certified that the contents and form of the thesis entitled "**Optimal Solution for Price and Load Forecasting in Smart Grids**" submitted by **Aneeza Khalid** have been found satisfactory for the requirement of the degree.

Advisor: **Dr. Sohail Iqbal**__________________

Signature: __________________

Date: __________________

Committee Member 1: **Dr. Fahad Javed**

____

Signature: __________________

Date: __________________

Committee Member 2: **Dr. Iftikhar Ahmad Rana**

Signature: __________________

Date: __________________

Committee Member 3: **Dr. Safdar Abbas Khan**

Signature: __________________

Date: __________________

Dedication

Dedicated to My family

For their Love, Kindness, and Encouragement

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's de- sign and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: **Aneeza Khalid**

Signature: ___________________

Acknowledgment

First, I would like to express my gratitude to Allah Almighty (SWT) for his abundant blessings. After that, I am sincerely grateful to my supervisor Dr. Sohail Iqbal for his priceless guidance, encouragements, devotion, and patience. Without his guidance and positive feedback, this thesis manuscript would not have seen the light of day.

I would like to thank my guidance and evaluation committee members including Dr. Fahad Javed, Dr. Iftikhar Ahmad Rana and Dr. Safdar Abbas Khan for their precious time, valuable suggestions, and invaluable constructive criticism. I would also like to thank all my friends especially Qurat ul Ain and colleagues for their suggestions, encouragement, and continuous support.

Finally, I would like to say a big thanks to my mother and my family for their endless love, kindness, and encouragement. Without their prayers and continuous support, it would not have been possible to accomplish this milestone.

Table of Contents

List of Abbreviation

Abbriviations	Full form
ABC	Artificial Bees colony
AEMO	Australian energy market
ANFIS	Adaptive Neuro-Fuzzy Interface System
ANN	Artificial Neural Network
ARIMA	Auto Regressive Integrated Moving Average
CART	Classification and regression technique
CMI	Conditional Mutual Information
CPP	Critical Peak Pricing
DALF	Day Ahead Load Forecasting
DLSTM	Deep Long Short-Term Memory
DM	Diebold-Mariano
DMD	Dynamic mode decomposition
DNN	Deep Neural Network
DT	Decision Tree
DTCWT	Dual-tree complex wavelet transforms
ECNN	Enhanced Convolutional Neural Network
EKNN	Efficient k-nearest neighbor
ELM	Extreme Learning machine
ELR	Enhanced Linear Regression
ENN	Enhanced Neural Network
ESD	Error Standard Deviation
ESSO	Enhanced Shark Smell Optimization
ESVR	Enhanced Support Vector Regression
FMSE	Forecast Mean Square Error
FT	Friedman test
FWPT	Flexible Wavelet Packet Transform
GA	Genetic Algorithm
GMI	Generalized Mutual Input
GRNN	Gated Recurrent Neural Network
GS	Grid Search

GWO	Greedy wolf optimization
HGWDE	Hybrid Greedy Wolf differential evaluation
IFA	improved fusion algorithm
IOWA	induced ordered weighted average
ISO NE	Independent System Operator New England
ISO NE CE	Independent System Operator New England Control Area
KNN	k-nearest neighbor
KPCA	Kernel Principal Component Analysis
LM	Levenberg-marquardt
LSSVM	Least square support vector machine
LSTM	Long Short-Term Memory
LTF	Long term forecasting
MAE	Mean Average Error
MAPE	Mean Average Percentage Error
MAR	Mean Average ratio
MARA	Multivariate Auto Regressive Algorithm
MDP	Markov decision process
MI	Mutual Information
MIMO	Multiple Input and Multiple output
MIT	Mutual Information Technique
MLP	Multilayer Perception
MR	Minimization-Relevancy
MSFE	multi-stage forecast engine
MTF	Medium term forecasting
NARX	Non-linear Auto Regression network
NLLSVM	non-linear least square vector machine
NMAPE	Normalized Mean Average Percentage Error
NN	Neural Network
NRSME	Normalized Root square mean error
NSA	Numerical Sensitivity Analysis
NYISO	New York Independent System Operator
OTI	Operational Time Interval
OWA	ordered weighted average

PAR	peak to average ratio
PCA	Principal Component Analysis
PSO	Particle Swarm Optimization
QOABC	Quasi-Oppositional Artificial Bee Colony
RBF	Radial Base Function
RBFNN	radial base function neural network
RELM	Recurrent Extreme Learning Machine
RF	Random Forest
RFE	Recursive Feature Elimination
RL	Reinforcement Learning
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RTP	Real Time Pricing
SARIMAX	Seasonal- Autoregressive Integrated Moving Average
SG	Smart Grid
SRWNN	Self-Recurrent Wavelet Neural Network
SSO	Shark Smell Optimization
STF	Short term forecasting
STLF	Short Term Load Forecasting
STPF	Short Term Price Forecasting
SVM	Support Vector Machine
SVR	Support Vector Regression
TV-SABC	Time-Varying Coefficients and Stumble Generation Operator
WPT	Wavelet Packet Transform

List of Figures

List of Tables

Abstract

With the advancement of technology, people need to know the things in advance so that they can make appropriate decisions for it. Similarly, when it comes to energy, we are curious to know that how much energy we are going to use in the next hour and how much it will cost them. There are many accurate price and load forecasting algorithms already working but they ignore the convergence rate. Data Science helps in evaluating the huge amount of data to predict future values. Executing any algorithm on such large amount of data needs high computational time. In the case of predicting the next hour or one-day value if the algorithm takes too much time in results formulation, then it becomes useless for us. We incorporated deep learning techniques as they process a large amount of data very fast and can predict fairly accurate results with a fast convergence rate. The proposed solution LSTM-BiGRU is formed in combination of LSTM and GRU, both are RNN variations and capable of forecasting best results. LSTM and GRU are combined in a best possible way to achieve the maximum accuracy with a fair convergence rate. The proposed solution is showing MAPE in load forecasting from 3.12% to 4.07%. The proposed solution is showing MAE in January 2019 for price forecasting from 2.35 to 3.02 and the execution of proposed solution in every scenario is recorded <1 min. So, a fair tradeoff is maintained between forecasting results and computational time. In future, the proposed method can be improved by other techniques i.e., block chains, optimization of proposed hybrid algorithms with evolutionary algorithms, and the use of GPUs and TPU can further decrease the computational time.

Chapter 1

1.1 Introduction

Energy crises are always present in this world. Many different sources are incorporating together to fulfil the energy demand. These sources are limited. So, an accurate energy forecast is very important for the producers, consumers, and the utilities. Producers and consumers are always looking for the energy prices for decision making and for making optimal bidding strategies. At the same time, accurately forecasting the load is also very important so that proper energy scheduling can be done before time.

Load and price forecasting are both important equally for utilities as they must manage the significant capital investments. The revenues are estimated from the forecasted load values. Moreover, utilities must sign contracts for energy generation with different power plants. Accurate forecasting helps the investors in deciding which technologies to invest in, i.e., classic gas turbines or renewable energy generators[1]. If price forecasting trend is showing an uplift, it is beneficial to put solar panels over the city roofs and deploy the windmills. Renewable energy production is not profitable at low prices. Similarly, inappropriate forecasting discourages investors to invest their time and efforts in energy as it not only causes loss of energy, but also investment and planning[2].

1.2 Background

Smart Grid replaces the traditional grids by providing communication between users/consumers, utilities and manufactures. It helps in managing the resources efficiently, in management of demand and supply, in enhancing reliability, in trading and in cost management. It provides bidirectional communication between power generators, transmitters, distributers, and consumers. Electrical energy once produced it can’t be stored on the large scale. It is produced according to the load demand estimation [3]. As energy is produced from many different natural sources including coal, petroleum, natural gas, hydro, nuclear, biomass and wind, a proper planning is required for its production, to save the resources and to better manage it [4]. Energy management is basically monitoring, controlling, and conserving energy. There are many techniques introduced already for managing the energy effectively. Energy management can be made better when accurate energy forecasting occurs.

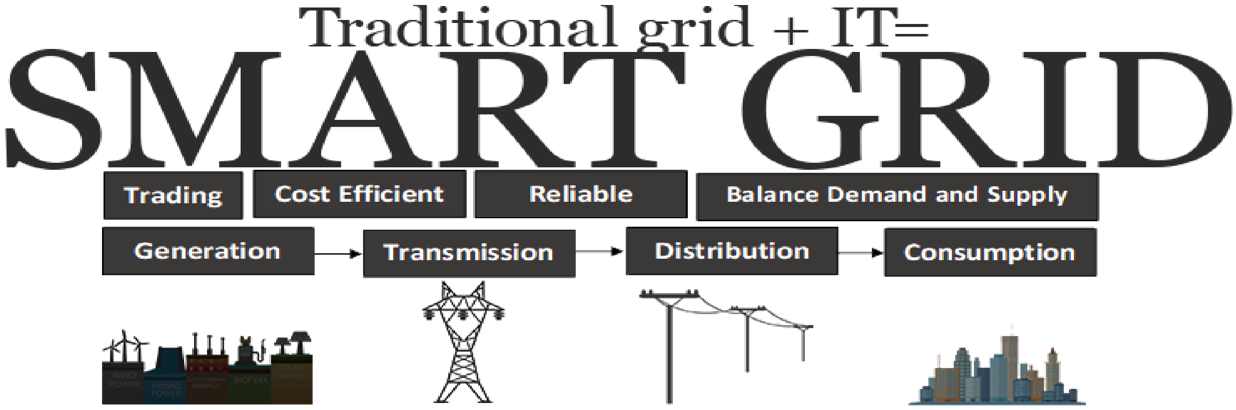

Figure 1: Conceptual Diagram of Smart Grids

Forecasting the load and price values are helpful in many ways, it helps end users to manage their usage according to their needs and budgets, it helps utilities to efficiently manage the demand and supply of electricity, load switching, network configurations and infrastructure development and electricity producers to manage producing the electricity accordingly [5]. It also helps users to shift the usage of different appliances in off peak hours and on peak hours, accordingly.

1.3 Types of Forecasting

To better manage the energy, it is necessary to better forecast it accurately. So, we generally categorize forecasting as follows:

1.3.1 Short term forecasting

Short Term Forecasting (STF) for both price and energy demand involve horizon of few hours to few days. It is generally involved where operations and decision making must take on day to day bases. Very Short-Term Forecasting (VSTF) involves forecasting from few minutes to one hour. It is generally considered and calculated separate from STF [6].

1.3.2 Medium term forecasting

Medium Term Forecasting (MTF) involves forecasting of both price and load values, its horizon is including few days to few months forecast. It is preferred for derivative pricing, balance sheet calculations and risk management. This is generally used in estimation of price and not for accurate load forecast as the forecast is made for the future time periods[7].

1.3.3 Long term Forecasting

Long Term Forecasting (LTF) involves forecasting of both price and load values for months, a year and several years. This forecasting is generally made for planning and investment profitability analysis like making contracts for fuel with different power plants[8].

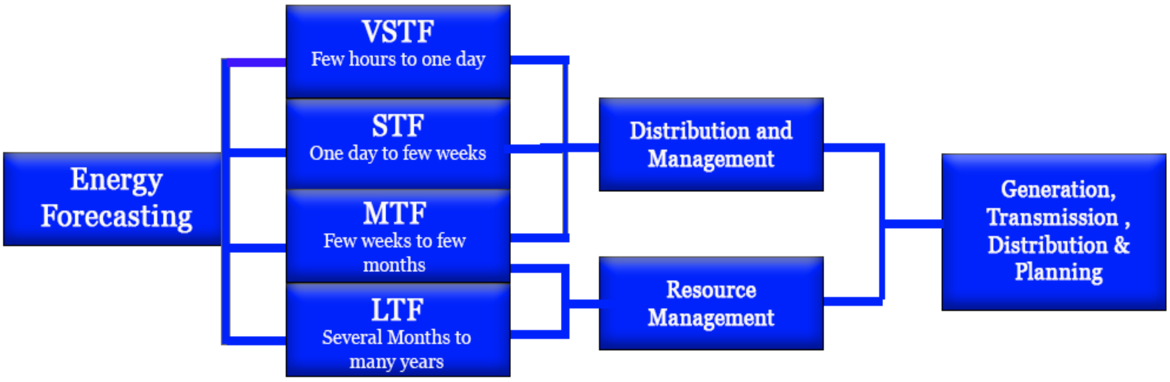

Figure 2: Types of Forecasting

The rate with which an algorithm reaches to its optimal result is called as rate of convergence. Here we are looking for fast convergent and fairly accurate result producing algorithms to get help in taking decisions. The convergence rate and accuracy are considered as inverse to each other. There are three approaches for convergence rate linear, superliner, and quadratic ordered from slowest to fastest [9].

1.4 Motivation

There are many mathematical, machine learning and deep learning algorithms are already working on load and price forecasting. The accuracy rate in predicting these values has been reached above 90%.

Beyond accuracy in results, these algorithms become successful in achieving many objectives, fast execution, decreased computational times, reliability, profit, validity, robustness, stability in models, reduced complexity, best fitting, minimizing cost and peak to average ratio, and online forecasting. But there are some limitations in each model designed, including models are not generalized they are specific for a specific type of forecasting, some of them are complex, enhanced execution and computational time, decreased convergence rate, premature convergence, and over fitting are observed.

Figure 3: Challenges in Forecasting

1.5 Problem Statement

The more the algorithm predict higher accuracy, the more time it takes in its calculations before reaching to final answer.

Figure 4: Problem Statement

We have observed two types of convergence in different papers i.e., slow convergence and fast convergence. Slow convergence occurs when the designed model is complex, the data wasn't preprocessed earlier, and the forecast algorithm is time taking. Usually, slow convergence ranges from 2 minutes and more. While fast convergence is how fast or robust an algorithm reaches to its local optimal point. It is usually a minute or less than it. There are many factors on which convergence rate depends including over fitting, complex models, models with slow training times, preprocessing requires time, 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 designing the best fit model, the model that is immune to the unseen or noise data. Unfitting may not lead us to accuracy of results, and they are procuring vague values. Best fit model is the one that is appropriate in predicting the results and don't undergo iterative or exhaust training.

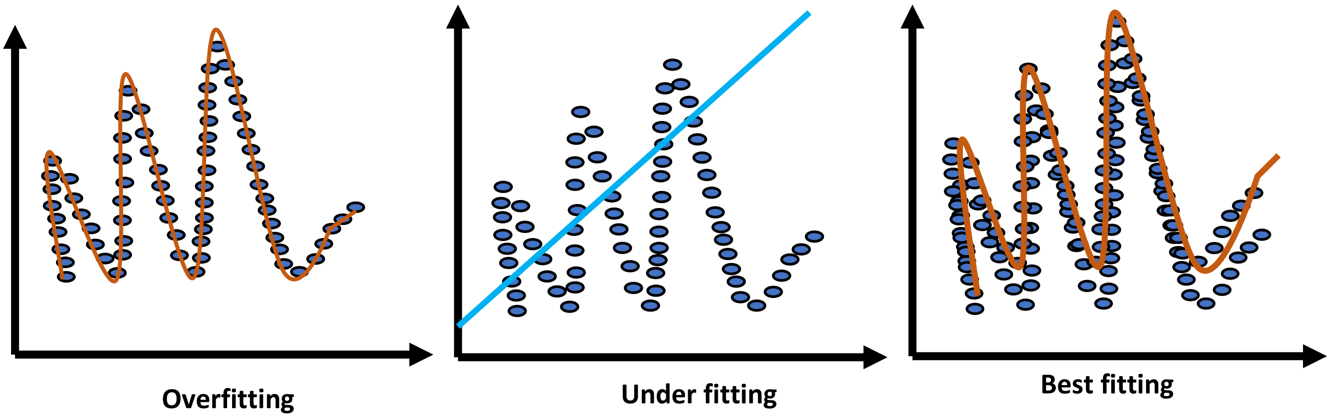

Figure 5: Types of Model Fitting

Best fitting model produces the appropriately accurate results, it will undergo fast convergence and help in managing the demand side management. This model helps utilities to manage the energy demand on time.

With the advancement of technology, people need to know the things in advance so that they can make appropriate decisions for it. Similarly, when it comes to energy, people want to know in advance how much energy they are going to use in the next hour and how much it will cost them. In this way, they can save money by minimizing the consumption. There are many accurate price and load forecasting algorithms already working but they ignore the convergence rate. Data analytic techniques evaluate huge amount of data to predict future values. Executing any algorithm on such big data needs high computational time[12]. In the case of predicting the next hour or one-day value if the algorithm takes too much time, we consider the algorithm as inefficient.

1.6 Objectives

Proposed approach presents STF of Electricity load and Price for the ISONE CA grid station. In this regard, this manuscript focuses on hourly, daily, and weekly forecasting of electricity consumption for the historical data provided by the ISO NE CA. The main contribution for our research is to enhance the accuracy of our system, and a fair computational time.

1.7 Organization of the Thesis

The objective of the thesis is to design a deep learning-based solution for load and price forecasting for smart grids, that provide results with good accuracy and a fair computation time. For this purpose, we have explained the brief introduction of SGs, types of forecasting, problems of forecasting models, our problem statement in objectives in the present chapter that is Chapter 1. In the next chapter (Chapter 2), we will be discussing the literature in this topics regard. We will brief their techniques and imitations in tabular format. In Chapter 3, we will discuss the proposed solution, each module and discuss the functioning of each in detail. In Chapter 4, We will perform simulations and present results in tabular format with discussion of each simulation. In Chapter 5, we conclude the manuscript, mentioning our objectives and how we achieve them. We will mention the future advancement with those too.

Chapter 2

2.1 Related Work

Smart grids (SG) are replacement of traditional grids. In traditional grid system, we were not sure about the energy demand and how much it will cost us. Moreover, in limited resources optimal energy management is also a challenge. To manage the resources, we should know the energy demand to make our power stations capable of meeting the demand on time. Similarly, smart grids provide us the facility of managing the load from user end. People knew about their next hour usage with the cost of that hour. The problem is how appropriate our current systems are predicting the load and price values.

To better capture the problem, we need to understand the existing techniques and their limitations. In this section, we will review different papers focusing on preprocessing of data, forecasting methods and optimization techniques. In general, we discussed some Artificial Neural Networks (ANNs) and mathematical model-based forecasting methods with different optimizers, evaluating accuracy, performance, convergence rate and execution time from them. At the end we compare all the discussed methods in tabular form for better understanding. These are discussed below:

2.1.1 Artificial Neural Network-based forecasting

ANN is trending these days; it is highly using in forecasting. It works like brain. It takes input and learn it, so that, later it can predict unseen output on unseen data. Here we reviewed the latest work on load and price forecasting based on artificial neural networks. A basic unit of Neural Network is attached below attached Figure 6:

Let's understand the basic architecture of NN with help of below mentioned literature reviews. DALF is very important as it is helping utilities to manage the load accordingly. Load values are volatile, so it is difficult to predict the load for a long period and hence forecasting is difficult. In [13], ANN based DALF model is proposed to forecast load with accuracy. Metrological and exogenous variables influence DALF. Sigmoid is used as an activation function in ANN and MARA is used for training. Due to normalization, execution time has been reduced, and due to training, non-linearities in the data set has been captured. EDE algorithm is used for optimization. Dataset has been taken from USA i.e., EKPC, FE and DAYTOWN, simulation results showed that 98.76% forecast accuracy is achieved as compared to Mutual Information (MI)+ANN based technique.

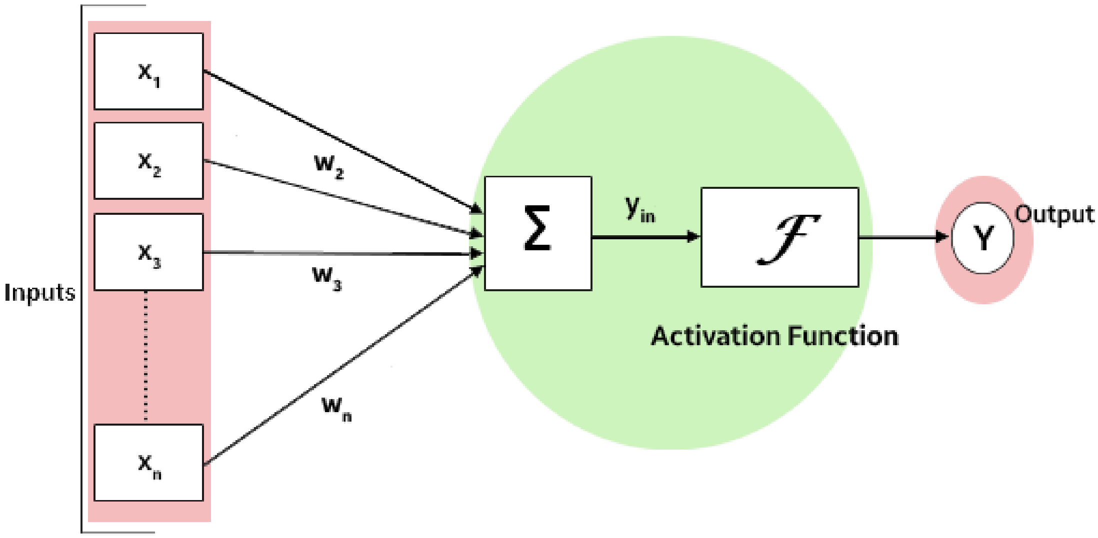

Figure 6: Architecture of Neural Network

This model increases the forecast accuracy without having any influence on convergence or execution rate. And hence, demand side management can easily be managed. Although the preprocessing of data is slow as well as the scope of this paper is limited to STF only.

To manage the energy demand and cost, an incentive-based algorithm is designed with Reinforcement Learning (RL) and Deep Neural Network (DNN).

In [14], the service provider can purchase energy from the customers to cope the energy demand. The excessive energy or reduced energy consumption by the customers help them in selling their energy to service provider on some optimal price, to enhance grid reliability and balance energy fluctuations. Separate models for customers and service providers are defined. Markov decision process (MDP) helps in deciding the price at which the service provider should buy the energy from customers. The MDP considers current environment, current states and then predict incentive rates for both service providers and customers, and when they are having long partnership so a discount factor can be multiplied with this incentive rate. Similarly, Q- learning algorithm helps in finding the maximum benefit value for both, by adjusting and updating it. Here, neural networks help in prediction of price and load values, later helps service provider to make a purchase from customers. Historical data is used as input as well as training of NN, sigmoid function is used as transfer function. Weights and biasness are adjusted by Levenberg-marquardt (LM) back propagation to get the desired results. Mean Average Error (MAE) and Mean Average Percentage Error (MAPE) is used for accuracy evaluation. Simulation results showed fluctuations in energy management are balanced and grid reliability is achieved. Both the customers and service providers got benefited. But the proposed model can cause overfitting, it is complex and increases user discomfort.

Recent research in predicting price and load, conducted on small data which reduced the accuracy, or they have overfitting problems. This [15] using big data, forecast short- and medium-term load and price accurately and separately with the involvement of DNNs. The proposed model undergoes time series forecast method with help of ANN. Long short-term memory (LSTM) is basically back propagation of NN in which weights can be readjusted to increase the accuracy of the results. The process involves preprocessing of data to normalize the values of price and load. The data used in this method is large as it uses big data techniques and it get more accurate results when data set is large. So, data taken from Independent System Operator New England (ISO NE) and New York Independent System Operator (NYISO) from 2011 to 2017 + 3 months of 2018 and NYISO 2006 to 2017 + 9 months of 2019. Deep Long short-term memory (DLSTM) network was trained on month-wise data. Data was partitioned into three parts: train, validation, and test data. Training data in form of month wise is sent to forecast engine and get validated with validation data it. Once it gets validated the next step is tuning the network and predicting the price. Every predicted value in tuning phase is the part of ANN for the next prediction. Adam is used to optimize the forecasted values. After the first initial result Normalized Root Square Mean Error (NRSME) is calculated and retuning and relearning continues until the minimum possible value of NRSME is calculated. Deep Long Short-Term Memory (DLSTM) forecasting model has lesser MAE and NRMSE as compared to Extreme Learning Machine (ELM) and Non-linear Auto Regression network (NARX). To validate the tests, Friedman test (FT) and Diebold-Mariano (DM) are performed in comparison with ELM and NARX and DLSTM shows better performance than others. But this paper lacks features selection process.

2.1.2 Block model-based forecasting

To overcome the limitations of optimizers and to keep intact the whole forecasting procedure, block models are being used these days. Some of the well-known block model-based load and price forecasting is discussed below:

Smart grid makes people to think about the increasing price and managing load problem, for this there is a need to predict the load and price accurately. Accurate predictions avoid loss of extra production of energy and enable management with less resources. In [16], a hybrid forecast model is purposed to predict load and price accurately. This hybrid model comprises of (Dual-Tree Complex Wavelet Transform) DTCWT and Multi-Stage Forecast Engine (MSFE). DTCWT is used in the new feature selection method known as MGR-MR-IG that is

(Global Redundancy Minimization-Relevancy Maximization-Interaction Gain Criteria.

Table 1: Summary of Artificial Neural Network-based forecasting Research

Related Work	Scope	Technique used Forecaster + optimizer	Target achieved	Limitations
Day ahead load forecasting [13]	VSTLF	ANN based forecaster with EDE optimizer	Reduced execution time Enhanced forecast accuracy	Limited Scope Preprocessing is slow Decreased convergence rate
Incentive based demand and supply [14]	STLF	ANN+Q-learning (reinforcement learning)	Increases profit Increased reliability	User discomfort increases Complexity increases Can cause overfitting
Price and load forecast accuracy in Smart cities [15]	STLF STPF MTLF MTPF	ANN based forecaster + Adam optimizer	Increased Accuracy	Specified feature selection is missing

First step of MSFE is preliminary forecast, which includes a block comprising of ANN, Radial base function neural network (RBFNN) and Support Vector Machine (SVM) with Shark Spell Optimization (SSO) for the direct optimized predictions so that the signal enters directly in the forecast engine block. Feed backward NN is used, with the primary function of reducing the error and entering the optimal weights and biasness in the forecast engine. RBFNN is used for avoiding the training error, in this paper, sliding window method is used for this training. SVM is used for linear constrained optimization here. Second step is Improved Fusion Algorithm (IFA) that works on Ordered Weighted Average (OWA), initially the weights are updated independently but later according to the performance of agents, hence, gathering output of MSFE. The proposed model is implemented on Australia's and New England electricity market data with short term load forecasting problem. Results showed the better accuracy when compared with benchmarks including *Autoregressive Integrated Moving Average* (ARIMA), Wavelet Transform (WT) +ARIMA, and MR-MI +NN. NRSME and NMAPE shows less error as compared to other benchmarks. But this increases the computational complexity and executional time. This model can cause overfitting as well.

The [16] focuses on accurate STLF enhancing the block forecast engine with two stage forecast engine. Input data (lagged loads) is in form of signals on which feature selection is performed with MIT to forecast the load of next hour. Then the two-step forecast engine begins where two different neural networks are implemented namely: ridgelet and Elman NNs. Where ridgelet forecast on the normalized data and ENN calculates the signal prediction error so that it can be avoided in the targeted forecast. RNN and ENN works on shark smell optimization function which undergoes chaotic and binary enhancement to predict accurately. Later, experiments are performed on Australian energy market commission, Pennsylvania-new Jersey-Maryland and North American data calculating forecasting errors including absolute forecasting errors, symmetric errors, measures based on percentage errors, scaled errors, and measures based on relative errors. Results showed the better accuracy than the other 36 compared forecasting algorithms. But the proposed method takes a long convergence time, execution time also increased. This model can cause overfitting as well

Better prediction accuracy can save a lot of energy from getting waste. In [17], best topology for ENN based different forecast engines for load and price are suggested. Input is in form of load and price signals. For feature selection and optimization, MI and greedy algorithm is applied on the signals to avoid redundant and relevant signals, the selected features than enters the Enhanced Neural Networks (ENN) based forecast engine. Here, forecast error minimization is performed by optimization of ESSO. In Enhanced Shark Smell Optimization ESSO, there is a context layer between input and hidden layer that causes sensitivity to the data entered in the hidden layer. ESSO also avoids overfitting in the training process. Numerical analysis is performed to analyze robustness of ESSO, feature selection as well as applied on real word test cases. North American and New England test cases are examined to find Short Term Load forecasting (STLF) and Short-Term Price Forecasting (STPF). Results showed that validity of the proposed strategy on different topologies. The convergence time of proposed model is high, and an expensive process.

Table 2: Summary of Block Model -based forecasting Research

Related Work	Scope	Technique used Forecaster + optimizer	Target achieved	Limitations
Multi-block based forecasting for price and load [18]	STLF STPF	ANN, RBFNN, SVM based forecaster +SSO	Increased accuracy Valid model	Overfitting Decreased convergence time Increased executional time
Multi-block based forecasting for price and load [17]	STLF STPF	ENN based forecaster +ESSO	Accurate Valid model Robust	Increased computational Decreased executional time
Load forecasting with two stage forecasting engine [16]	STLF	Ridgelet-NN, ENN based forecaster +SSO	Increased Accuracy Effective results	Overfitting Increased computational Increased executional time

2.1.3 Convolutional Neural Network based forecasting

With the passage of time these neural networks enhanced, CNN is one of those modified and better performance network. Different number of maxpooling layer can be adjusted to get the accurate results. Here we have discussed the latest work on CNN:

Forecasting price helps in managing the resources and price at both consumer and supplier end. Accurate load forecasting is very important as power stations must plan the production to meet the demand. In [19], an accurate load forecasting optimization technique is introduced for STF, MTF and LTF. CNN is used to process the data and give initial predictions of load. Two models are implemented with CNN NN- Genetic Algorithm (GA) and NN-Particle Swarm Optimization (PSO). Proposed model is tested through Matlab simulations, data of last 5 years are used from Pecan Street Inc., and divided into training, validation, and testing sets. Predictions are made for summer and winter season as well as for one day, one month and a year. Results showed that NN-PSO works better for MTF and LTF while NN-GA works better for STF. These predictions can be used in demand side management and supply to fulfill the future requirements.

In traditional grids, electricity once produced can't be stored, so a smart grid is required to identify the amount of need of electricity so that wastage of energy can be avoided, and cost can be reduced. For this purpose, electricity load forecasting classification technique is proposed in [20] using ECNN on one-year hourly data provided by Independent System Operator New England Control Area (ISO NE CA) market of year 2017 from January to December. Preprocessing or normalization of data is done before feature selection. Random Forest (RF) and MI are used in eliminating the features with less importance and later in feature extraction, Kernel Principal Component Analysis (KPCA) reduces dimensionality and redundant features and extract the important ones. CNN are enhanced here, by having 96 filters, 2 kernels and 32 filters, 3 kernels with pooling layer of size 2, for forecasting load. Training is done by 155 epochs in 30 batch size and rest of the data is used in testing purpose. Adam optimizer is used in this process. Simulation results showed that the proposed method is better than other benchmarks. Accuracy of the model is evaluated with loss function MAE and MAPE and a low value of 0.279 is achieved. This paper is limited to predict load only; price needs to be predicted as well.

SG manages intelligently the consumption, distribution, and generation of energy than the TG. The primary goal of SG is to keep a balance between demand and supply of electricity. Data Analytics is used it to identify patterns to forecast the load and the price of energy. In prediction of accurate load and price, complexity, overfitting, and computational time increases. In [21], two models are presented with little differences to predict load and price. For load prediction, data set is taken from ISO-NE of year 2018 and predict the load of 1st week of January. Data undergoes pre-processing i.e., features extraction and selection. For feature extraction Recursive Feature Elimination (RFE) is used and for feature selection RF. After preprocessing, the selected features are sent to ECNN to forecast load. For price prediction model, date set is taken from NYISO from November 2016 to 15 January 2017. There are two models, one is same as load forecast model using ECNN while the other used Enhanced Support Vector Regression (ESVR) method. For preprocessing of data, feature extraction is done by RFE with cross validation to maintain accuracy and followed by feature section with XGBoost and Decision Tree (DT). The most important step is tuning the parameters. This tuning by Grid Search (GS) is creating the difference in the final step. Every subset of SVM parameters undergoes classifier followed by k-fold cross validation. Now it is passed to ESVR to predict the final output. Results are compared with 27 Benchmarks and performance matrices i.e., MAPE, MSE, RMSE and MAE, results are compared. Both these methods performed well, and ECNN and ESVR with threshold values 0.08 and 0.15 achieve 2% and 1 % accuracy respectively.

The [22] paper focuses on load prediction accurately and efficiently by considering ECNN and Efficient kth neighbor neural network (EKNN). Data of year 2016-2017 was taken from NYISO on hourly rate that is used as an input, preprocessing is performed on it. 80% of data is being utilized in training purpose while rest is being utilized in testing purpose. Feature selection involves Mutual Information (MI) which removes less important features. Feature extraction involves RFE to remove recurrent features. For forecasting load, KNN and CNN is enhanced so that they can work efficiently. Efficiency of CNN increased by 2 max-pooling and 3 convolutional layers, for KNN, the tuning of the classifier is performed by hyper-parameters. EKNN and ECNN is used to forecast load. For optimization, hyper parameters are used. Simulation results showed the accuracy of 92% and 93% by ECNN and EKNN. Their performance is later evaluated by MAPE, MAE, MSE and RMSE, which showed the better performance than other benchmarks.

To forecast the load and price accurately in [23], two methods are proposed. First method includes normalization of data, feature selection, extraction and forecasting utilizing Classification and regression technique (CART), RFE, Relief-F and Enhanced Linear Regression (ELR). Whereas the second method includes preprocessing of data, feature selection by (Recurrent Extreme Learning Machine) RELM, forecasting (RELM enhanced by GWO: ERELM) and cross validation (Monte Carlo & k-fold). UMASS Electric Dataset (multi-variate dataset) and UCI (uni-variate dataset) both models are applied to both datasets. Accuracy of both the models can be checked on bases of MAPE, MAE, MSE and RMSE on one day, one weak and one-year time. 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.

Table 3: Summary of Convolutional Neural Network Based Forecasting Research

Related Work	Scope	Technique used Forecaster + optimizer	Target achieved	Limitations
Hourly Load forecast [20]	VSTLF	ECNN based forecaster + Adam optimizer	Increased Accuracy	Scope is very limited.
Price and load forecast accuracy in Smart grids [21]	STLF STPF	ECNN, ESVR based forecaster + Adam optimizer	Reduced computations Reduced execution time Stable model Accurate results	Overfitting Complex Limited Scope
Efficient load forecasting [22]	STLF	ECNN+EKNN based forecaster	Better prediction accuracy Less complex Efficient Reduce overfitting	Results need optimization Decreased convergence time Slow speed
Forecasting energy demand [19]	STLF MTLF LTLF	CNN based foreca0ster and NN-GA, NN-PSO	Increased accuracy Minimize cost	Large data set required Computational complexity Slow convergence
Short term load and price forecasting [23]	STLF STPF	RELM based forecaster +GWO	Forecast accuracy increased	Increased computational time Slow convergence Increased waiting time Expensive

2.1.4 Recurrent Neural Networks

It is the type of neural networks that works on LSTM. It undergoes proper training of neurons from the data set and thus to produce more accurate forecasting results. Some of the latest papers in this area are discussed below, consider the attached figure7 to better understand the described literature below:

Micro-grid manages the demand and supply at micro-level. Load time series have some volatile and non-volatile characteristic, due to which accurate STLF is an issue. This [24] proposed SRWNN as forecasting engine for addressing the nonvolatile nature of load time series, as it have feedback loops to deal with it. One-year hourly data set from Columbia and California is used. MI is used to avoid less important features during feature selection. LM train 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 accurate forecast results than WNN. However, results need optimization.

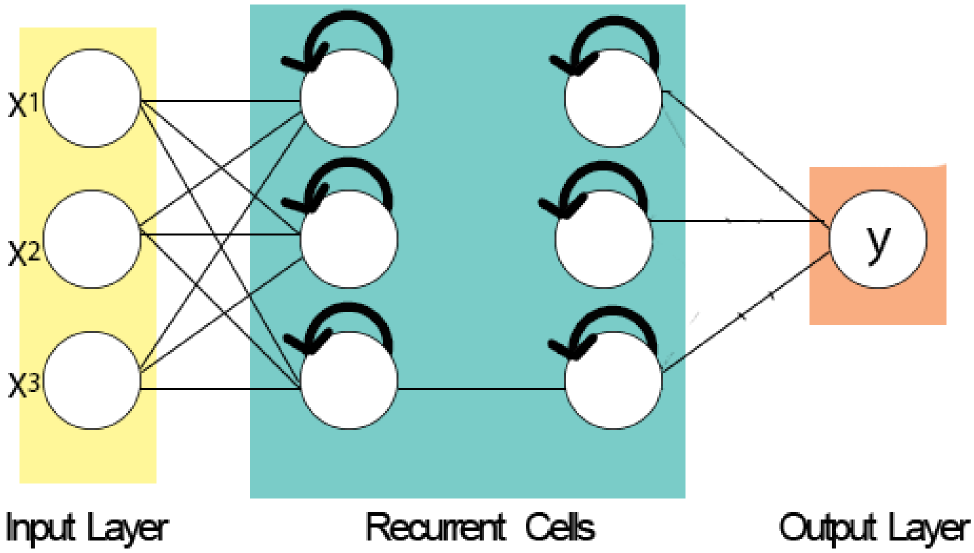

Figure 7: Basic Architecture of Recurrent Neural Network

To predict the load for a building in [25], three-time series approaches are designed. Seasonal ARIMAX (SARIMAX) is basically a non-DL technique which computes that part of temperature that is in difference from the outdoor temperature. Gated Recurrent Neural Network (GRNN) keep in memory the weights for a long time and GCNN considers input variables in synchronous series and concatenated weather predictions. Datasets of three buildings (academics, primary school, and a grocery store) are considered which includes outdoor features including temperature, humidity, air pressure and wind speed. On this raw data, data cleaning (removing missing values), data segmentation (separating the training, validation, and test data), dividing the data according to time series and normalizing it. Then fed to respective models for output. The performance is analyzed on direct multi-step (one day) and recursive multistep (24 times) forecasting manners. The results of prediction accuracy showed that SARIMAX model shows better results than GRNN24 as its accuracy decreases by 22.6% due to weather covariance. The computational accuracy of GCNN24 increased by 8% as compared to SARIMAX model. While generally the performance of GRNN1, GCNN1 and GCNN24 is less than that of the SARIMAX model. In DL scope, GCNN24 model performs better than all other Deep Learning (DL) models and have better accuracy as compared to GCNN1. However, there is need to extend these models for predicting energy consumption.

2.1.5 Radial Base Function-Neural Network

For time series forecasting, function approximation and system control the NN are extended with radial base function to produce the accurate results. In load and time forecasting, RBF is being used to increase the accuracy of results.

To fulfil the demand and supply, appropriate forecasting is crucial. The [26] considers seven different influential features and energy sources that can vary with time and conditions. Like feature 'rain' can be ignored in summers. In order to avoid these instabilities and forecasting issues two types of neural networks are fussed together i.e. Radial Base function (RBF) and Adaptive Neuro-Fuzzy Interface System (ANFIS). Multilayer Perception (MLP) involves in training phase. Cuckoo optimizer is used in forecasting phase. Later the output is sent to IOWA operator for aggregation where MSE is calculated. The results showed improvements as compared to other non-hybrid techniques.

Some variations in RBF-NN makes it Generalized-NN(GRNN). To forecast load and price with accuracy and efficiently, GRNN are used in comparison with ANNs. This [27] finds out that there is no need to train the classifier recursively, but data selection for training should be selected wisely. For this, data assembled in ordered way (first six months in training and rest six in testing, random way, mixed weeks (first four weeks in training and rest one for testing) and adaptive way (splitting data in half). GRNN smoothing parameters are applied to both selection and final forecast model, while in ANN Levenberg-Marquardt algorithm for training purpose is used. To evaluate the performance of both the models, they are compared with MAPE. Due to ordered and adaptive data sampling for GRNN, it is considered better than ANN, although both have same accuracy, but the difference is in the computational time, GRNN takes less computational time than ANN.

Table 4: Summary of Radial Base Function Neural Network Research

Related Work	Scope	Technique used Forecaster + optimizer	Target achieved	Limitations
Day ahead load forecast for a building [25]	STLF	RNN, CNN based forecaster + Adam optimizer	Better accuracy Better computation efficiency Robust generalizable technique	Need to extend the scope
Short term load forecasting for a building [24]	STLF	SRWNN based forecaster	Efficient Good forecasting performance Less computational time	Need optimization of results Accuracy can be increased
Efficient load forecasting [26]	STLF MTLF LTLF	RBFNN, ANFIS based forecaster + cuckoo optimizer	Accurate and precise forecast results	Slow convergence Slow computational speed
Fast computations for price and load [27]	LTLF	GRNN based forecaster	Decreased computational time Better performance	Accuracy can be increased Optimization is required

2.1.6 Least square support vector machine-based forecasting

For identifying patterns and analyzing data Least Square Support Vector Machine (LSSVM) algorithms are used. And to identifying the patterns for load and price so that we can understand the usage and hence devise methods to get the maximum benefit from it. Some of the papers that uses LSSVM are discussed below:

Smart grids enable the users to manage the load according to the change in the price. The load and price values are highly correlated. Previous all methods help in forecasting the load and price values, but they allow the change in use of energy pattern with the change in price. This

[28]. Multiple input Multiple Output (MIMO) model helps in calculating the correlation values of price and load. First step is preprocessing, in which Wavelet Packet Transform (WPT) and Generalized Mutual Input (GMI) helps in calculating the best features i.e. by detecting the change of seasons, searching similar days and tagging days. The second step is MIMO-LSSVM that gives predictions on hourly bases. The last step is optimization that is performed by Quasi-Oppositional Artificial Bee Colony (QOABC) algorithm to increase the forecast accuracy. Proposed model is tested on popular electricity markets including NYISO, NSW and PJM electricity market. Simulation results showed that great performance when performance evaluated with Rastrigin and Ackley benchmark functions. For testing the accuracy of proposed algorithm, the MAPE, FMSE and ESD are calculated on daily and weekly bases, showing better accuracy of the proposed method than ABC. The proposed method doesn't need a-priori data to forecast load and price on the forecast day and thus this method can be adapted to real markets.

To get the maximum benefits, there is a need to make the market cost effective. Load and price are correlated terms; these should be predicted with great accuracy to make better bidding polices for the real market. The [29] proposed a novel method MIMO- Non-linear Least Square Support Vector Machine (NLLSVM) and ARIMA. The first step is preprocessing in which input data in form of signals are sent to FWPT so that it can decompose the wavelet into different frequencies and noisy data can be removed. Later, feature selection is performed by Conditional Mutual Information (CMI) to select the best feature from the input data. Second step is forecasting, MIMO-NLLSVM and ARIMA are used to correlate or forecast between linear and non-linear load and price values. The proposed strategy works better when tested, it can predict the peak values when the data is volatile and have price spikes. Third step is optimization by modified ABC, that is Time-Varying Coefficients and Stumble Generation Operator (TV-SABC) which considers load factors and cost function to eliminate the cost high price peak values. This model is tested in real word markets (NYISO, NSW and PJM) and have better accuracy than the benchmarks, but the computational complexity and time is high.

2.1.7 Hybrid Greedy Wolf and Differential Evolution based forecasting

Besides NN, there are many other methods that can facilitates in forecasting accurately are discussed below:

For establishing balance between electricity demand and supply, DSM, and DR help in decreasing the cost of energy for end users and helps in stabling the operations of grids. HGWDE is proposed in [30] to reduce cost and PAR for RTP and CPP. 17 different appliances are considered from one house and categorized as shift-able (washing machine, dishwasher etc.), non-shift-able (TV, telephone, refrigerator etc.) and controllable appliances (lights, heating system, air condition etc.), they are considered under different pricing schemes. In the hybrid model, EDE and EWO works in coordination. First step is initializing the parameters, this generated the population in second step. In third step EDE and GWO works in coordination, EDE is generating the population and selecting the best mutants from it and checking its fitness in comparison of α, β, δ of GWO until the termination criterion is reached. Simulation results shows that HGWDE performed better than EDE and GWO, both cost and PAR have low values than the others but waiting time has been increased. RTP reduced to 53.02%, 29.02% and 26.55%, and the electricity bill is reduced to 12.81%, 12.012% and 12.95%, for 15, 30 and 60 min OTI. PAR using the CPP tariff and electricity bill are reduced to 47.27%, 22.91%, 22% and 13.04%, 12%, 11.11%. However, this paper was only limited to few numbers of devices and one house only. Other optimization algorithms need to be explored in this regard.

Table 5: Summary of Evolutionary Algorithms based Research

Related Work	Scope	Technique used Forecaster + optimizer	Target achieved	Limitations
Simultaneous load and price forecast [28]	STLF	MIMO-LSSVM based forecaster + QOABC	High Accuracy Robust Fast Convergence rate Highly flexible	High algorithmic complexity Excess memory required Overfitting Early Convergence
Price and load forecast for demand side management [29]	STLF	MIMO-NLSSVM based forecaster + TV-SABC	High Accuracy Robust Fast Convergence Highly Flexible	Excess Memory required Complex Premature convergence Overfitting

Power scheduling with critical and real time pricing scheme [30]	STPF	HGWDE based forecaster	PAR minimizes cost minimizes Accuracy increased	Increased waiting time Slow convergence
Data driven strategy for load forecasting [31]	STLF	DMD	Satisfied accuracy Comprehensive ability Less computational complexity Less execution speed	Early convergence Accuracy can be increased

2.1.8 Dynamic Mode Decomposition based forecasting

Load prediction is a difficult task as it has fluctuations, non-linearity, and variations because of different seasons. Accurate STLF is necessary to predict load for next one or two day's at-least. Different model-based methodologies are being designed for this however there is a need of data-driven method exits. Data driven model have the capability of deriving the meaningful and relevant features from the system whose physical models are unknown. DMD is proposed in [31], that is a data driven technique. Data of four days is selected i.e., data from two last days, the same day in the last week and last day in the last week, so that the load can be predicted for the targeted day. Data preparation involves the input data in normalized dynamic load series, which undergoes hankelization of it. In next step, DMD undergoes Eigen-decomposition that reduces the dimensionality of it and dynamic mode estimation is made. Foresting is performed by de-normalization of these Eigen vectors followed by rearrangement and averaging of these values. For testing the half an hour data (48-observations) of North American electric utility and Australian Energy Market (AEMO) are considered and experiments are performed in different regions. Later the performance is compared on RMSE, MAPE, MAE, and running time with benchmarks (ARMAX, SVR, ANN etc). Computational time of DMD is 0.125 seconds which make its performance satisfactory, and it can be used as an efficient tool in real time STLF. However, real time price calculation is still needed to explore by this data driven method.

2.1.9 Random forest-based forecasting

To fulfil the electricity demand, service provider must plan. Day ahead load forecasting with accuracy needs to be predicted. In [32], load for next one day is forecasted, on hourly bases, considering the precious day load, weather predictions, general holidays and non-geographical holidays. RF is used in making predictions while expert selection algorithm is used to select best features. The basic architecture of RF is in figure 8.

This forecasting process is online as it has the load variations. Random forest is immune to parameters change, measure importance of variables and perform internal validations. It generated 24 outputs for 24 hrs. This method is tested on Tunisian's data. The inputs are

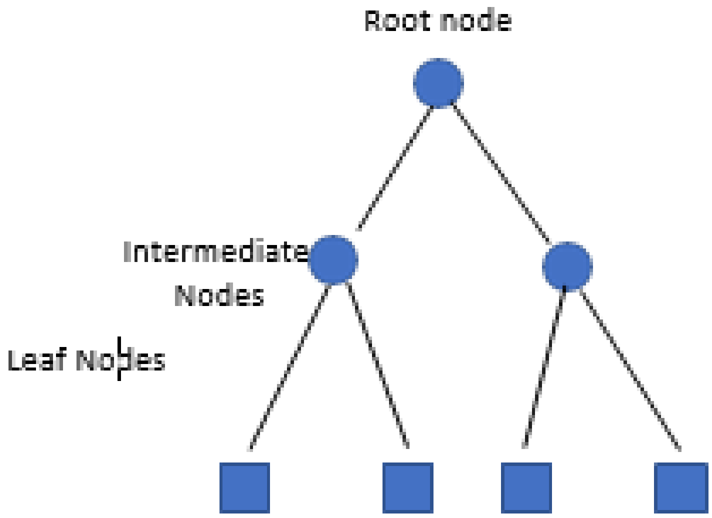

Figure 8: Basic Architecture of Random Forest

selected on if-then rules. Simulation results showed accuracy in results.

We have reviewed multiple forecasting models on base of different optimizers, evaluating accuracy, performance, and execution time. We have discussed the objectives, approaches and their limitations. So far, all the approaches are concerned with increasing accuracy of the proposed model and ignoring the convergence rate. We need to consider this aspect as well as higher convergence rate help us in taking decisions on time. And this will be beneficial for both utilities and end users.

Chapter 3

Proposed Methodology

As discussed in the literature review, many papers showed difference accuracy results for short term load forecasting and a very few of them work on convergence rate. We are considering the convergence rate with fair accuracy so that decision making can be done before time. A very generic methodology for forecasting is as follows:

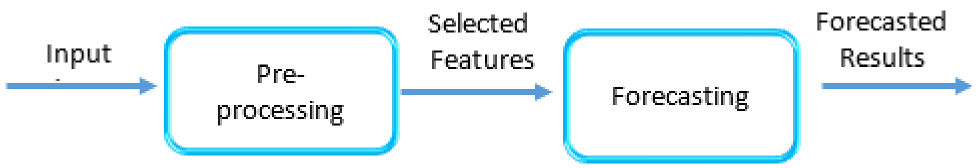

Figure 9: Overview of Proposed Module

Considering figure 9, we have designed a similar forecast module. Our proposed module is made up of two RNN variations. The explanation is below:

3.1 Proposed Methodology

As discussed in chapter 2, there are many mathematical, machine learning and deep neural networks are already working to predict the load and price values for the short-term memory with good accuracy, we are here to recommend a robust, fast, and fairly accurate model to provide us the best possible results with suitable convergence rate.

So, we have used a combination of statistical and deep learning algorithms. Our methodology is designed in 4 phases. These 4 phases are below:

1. Preprocessing
2. Data Determination
3. Forecast Engine
4. Evaluation

The detailed figure of our proposed solution is as follows, the above mentioned 2 modules are further divided into 4 modules. The detail of every step is mentioned on the figure 10 below. The details that we miss mentioning on the diagram is explained in the description below it.

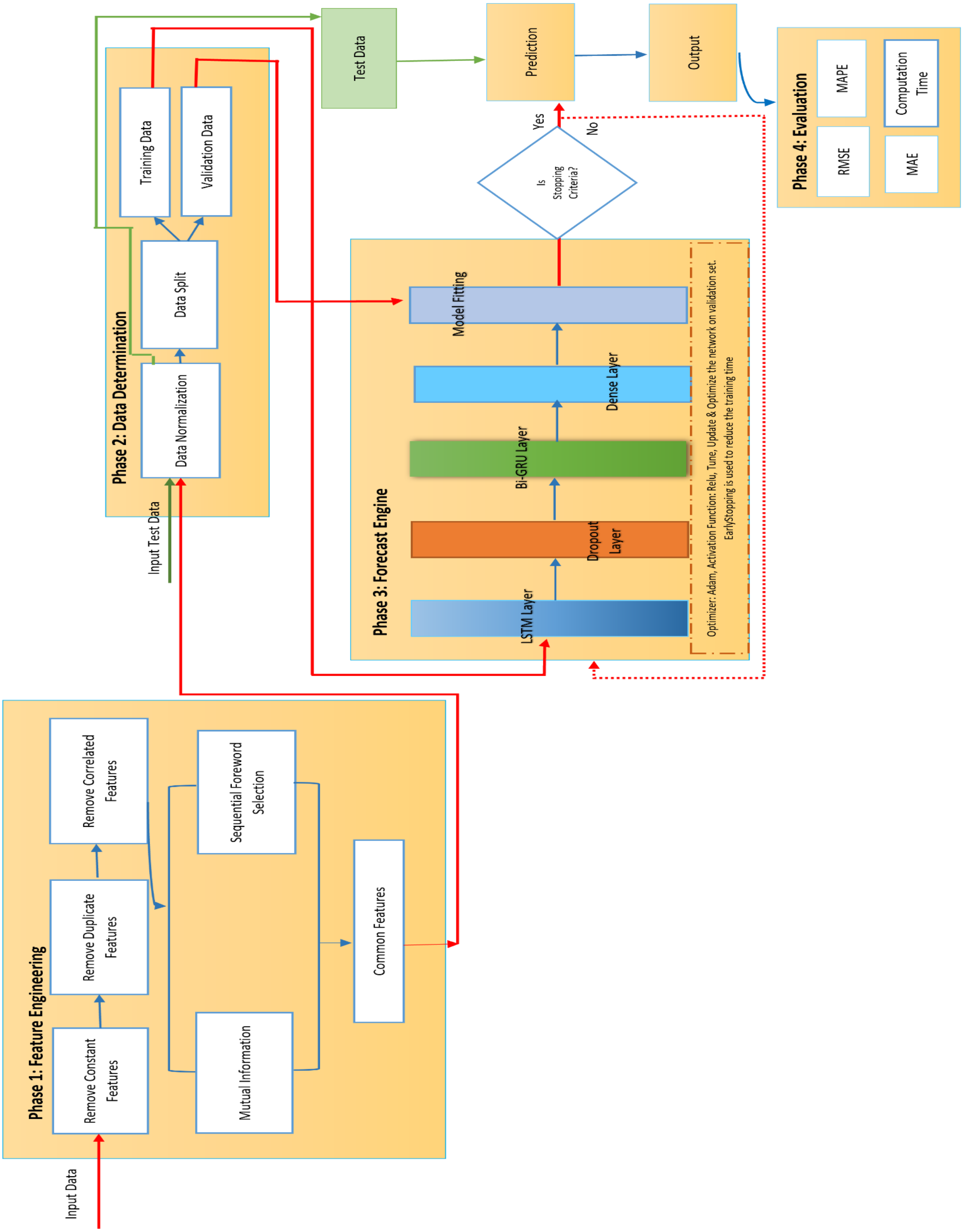

Figure 10 Proposed Solution LSTM BiGRU

Let us discuss each phase in detail.

3.1.1 Feature Engineering

3.1.1.1 Remove Constant Features

Constant features are the features that are constant, hence not having major impacts on output or target values. These constant features can be redundant in the dataset. As their presence is not adding any value to the target so it is better to remove them. Eliminating constant and redundant features and keeping only useful features comes under the filter method for Feature Selection Methods [33] .

3.1.1.2 Remove Duplicate Features

In large datasets there may exist some repeating features, having similar features in repetitive manner is not adding value to the final output. So, it is always advisable to remove these [33].

3.1.1.3 Remove Correlated Features

Correlated features are the features that measure strength between multiple features. Correlation can be positive, negative and no correlation. Two features out of which if one's strength increase causes an increase in other's strength, this kind of linear relationship is known as positive correlation. Similarly, if strength of one variable decreases with the increase

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

or other and vice versa, this is known as negative correlation. And the features that have no relationship and have no impact with increase or decrease of others is the state of no correlation [33].

means positive correlation. 0.9 and 1 means strong correlation, -1 means strong negative correlation.

Here we are removing 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. So having the same information repeatedly may not always increase the accuracy of forecasting but can decrease it. That is why, we have removed the features with high correlation values and selected 3 features out of 18.

3.1.1.4 Mutual Information

Mutual Information (MI) measures non-linear relationships between two variables. MI value represents how much information can be obtained from one variable by observing the other.

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

In correlation, we measure linear relationship between variables while in MI we measure nonlinear relationship between variables. The benefit of using features based on MI is that it considers dependencies between variables that are not detected by covariance. This means MI is 0 when two random variables are independent [34].

3.1.1.5 Sequential Feature Selection

Greedy search algorithms are the algorithms that are simple and intuitive in nature. These algorithms select optimal solutions at every point when solving a problem. Hence overall providing the simplest optimal solution. 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 best features from the pool of features. By removing the irrelevant features or noise, SFS helps in reduction of generalization error. Also, it is computational active [35].

Algorithm 1 Pseudocode of Sequential Foreword Selection

Algorithm 1 Sequential Foreword Selection

Input: **Y**= {y_1, y_2,, y_n}[36]

Output: x_k= {x_j|j=1, 2…, k; x_jϵY} where k<Y

1.Initialization x_0=∅, k=0

2. Step 1 Inclusion

x^+= argmax $J(x_k+x)$, where x ϵ Y-x_k

X_k+1=x_k+x^+

K=k+1

Go to step 1

3.Termination k=p

3.1.1.6 Common Features

We have set a threshold of selecting 5 features from MI and SFS. Then we select the common features which came out as output from these two mentioned algorithms, So the common features came out were 3 in number. We have designed the whole data preprocessing techniques to get the best features. If we have the best features selected only then we can make accurate forecasting results.

3.1.2 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. For this, we scale the data and split it into training, validation, and testing phases, respectively.

3.1.2.1 Data Normalization

Data normalization is basically scaling of data in a certain limit of values. The selected features are numeric in nature, and they appeared in 1 digit to 5-digit values. It is not a good practice to use values with large differences, in training the network. So, we perform MinMax scaling and scale the selected features between 0 and 1.

3.1.2.2 MinMax Scaling

MinMax Scaling is the approach in which we scale the data in a specific range that is usually between 0 and 1. The purpose of using MinMax scaling is bounding the values in a certain range that will end up in smaller standard deviations and thus mitigate the effect of outliers [12, 37]. We performed MinMax Scaling as follows:

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

Where x is the value in each row of each column, min x is the minimum value of that column and max x is the maximum value of that column.

3.1.2.3 Data Split

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. In predicting time series, we are required to test data of length we want to predict. It means we need one day's data to test prediction of the next day's data.

3.1.3 Forecast Engine

3.1.3.1 An overview of LSTM and GRU

The proposed model consists of variations of recurrent neural network (RNN) i.e., LSTM and GRU. The hybrid model consists of layers of LSTM and GRU arranged in specified sequence to maximize performance. LSTM and GRU are very effective and popular among researchers, especially for time series analysis. First, we explain each component and then we will explain the composition of these layers.

3.1.3.2 Recurrent Neural Network (RNN)

Recurrent neural networks are the state-of-the-art algorithm for sequential or time series data. They are very powerful and robust designed with internal memory. They were created in the 1980s. They have high computational power with massive data. The true potential was brought into light and the invention of LSTM brought RNN to foreground in the 1990s [38].

RNN came with their internal memory. They consider the prior inputs, as they influence the current input and output. They are not like traditional DNNs, considering the inputs and outputs are independent of each other. Instead, output of RNNs depends on the previous sequence of inputs.

To demonstrate the working of RNN, let us consider the following figure. xt are the inputs in sequence form, ht are the output, where A is the RNN black box. From the figure, it is obvious that an amount of information is being passed to another cell. The cell utilized the information it received in forecasting the sequence. This is how it works.

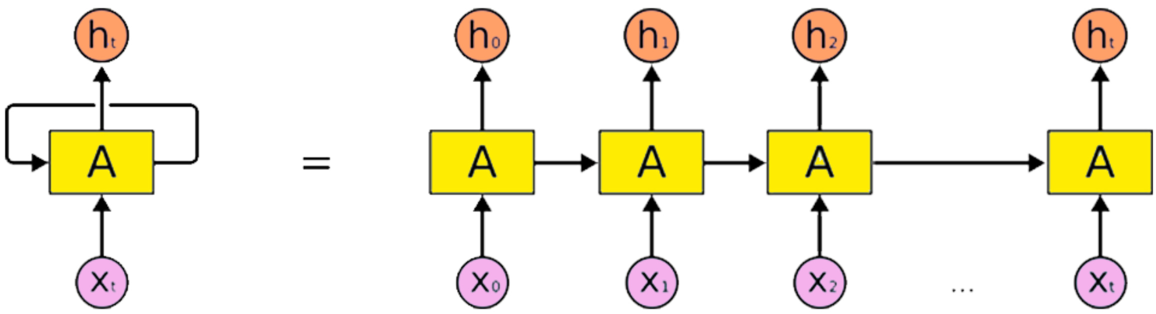

Figure 11: Recurrent Neural Network (RNN)

Not like all DNNs, RNN considers the information coming from the previous cell that increases the prediction accuracy but at the same time there was a problem with this structure. The problem is known as vanishing gradient or long short-term memory problem [38].
Now consider, if x2 is the input and it utilizes the information received from its previous cell. But in case, like word prediction, what happens if the x2 input requires information a few blocks behind it (not exactly behind it). In this case, the cell forgets that information that is a few cells behind that. This is called the vanishing gradient problem. It means the RNN structure could not remember the weight and biases value it utilized in the previous cells and that weight or bias vanishes. The similar condition is known as long term dependency problem, where the cell could not remember the long term stored information/weight/biases.
To solve this problem, many versions of RNNs were developed. To solve our problem statement, we utilized two of them. Namely, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). Let us discuss their structure in detail.

3.1.3.3 LSTM (Long Short-Term Memory)

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 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 LSTMs architecture. LSTM consists of three functions of gate controllers.

•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.

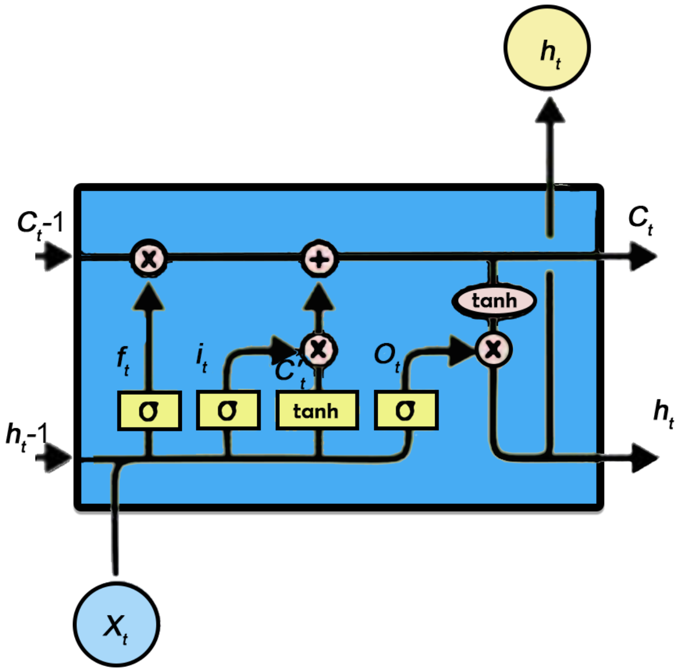

Figure 12: Basic Architecture of LSTM

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 ranges 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:

3.1.3.4 GRU (Gated Recurrent Unit)

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:

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

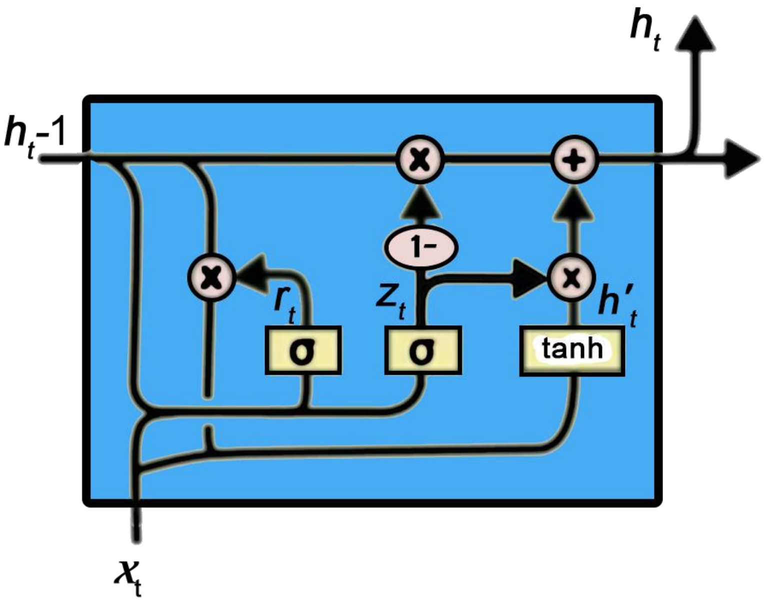

Figure 13: Basic Architecture of GRU

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:

$$h_t' = \tanh\left(Wx_t + r_t \odot h_{t-1}\right) \tag{12}$$

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].

3.1.3.5 Working of Proposed Hybrid Model LSTM-BiGRU

We designed the hybrid model by utilizing different LSTM and GRU layers provided by Keras

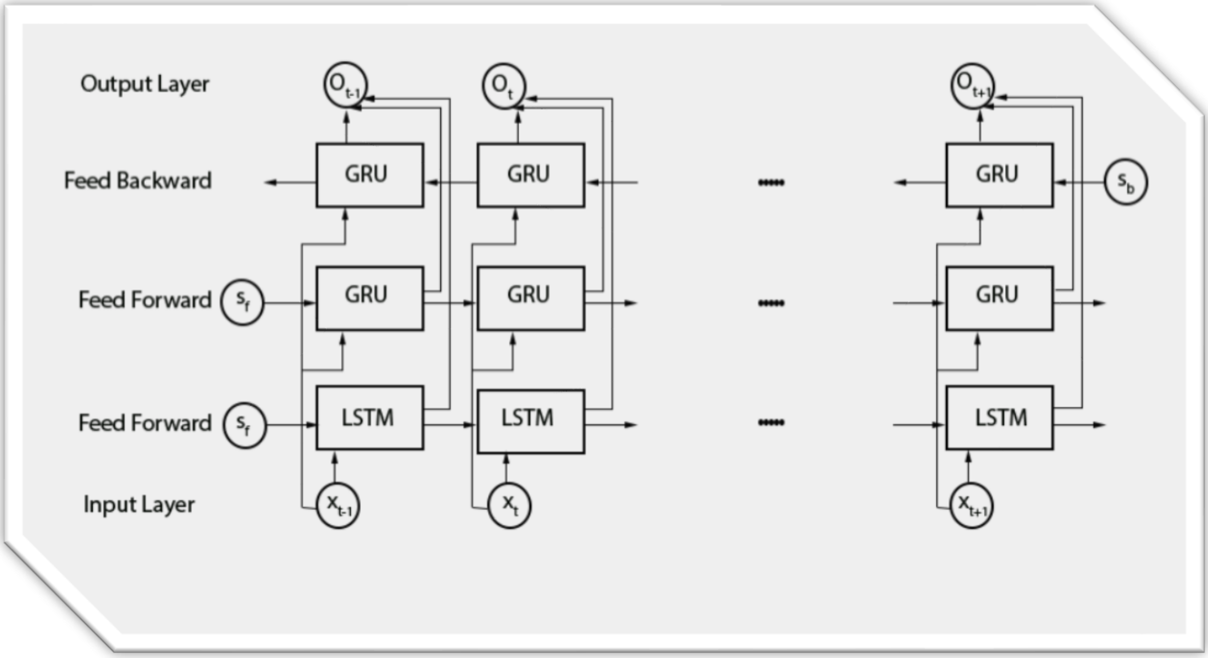

Figure 14:Architecture of proposed Model LSTM-BiGRU

library introduced by Python. The organization of the model is as follows: Consider the model from bottom to top. 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. The input layer consists of normalized feature set. The input features are {DA_LMP', 'Dew_Point', 'System_Load'}. Input features feed to first layer that is LSTM layer training input features in feed forward manner. GRU layer applied in bidirectional layer. First it undergoes training in feed forward manner then in feed backward manner. The hybrid layer containing layers of both LSTM and GRU in specified manner is enhancing the accuracy of predictions of the forecasting model, respectively. The LSTM Layer contains 256 units in first layer followed by a dropout layer to avoid over fitting. Similarly, the bidirectional GRU layer contains 20 neurons. 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. With each iteration the MSE is observed to be decreasing. This decrease in MSE is stopped after certain number of iterations and then it becomes constant, no further decreasing. This is the point when our model is fully trained with the training dataset. To stop further iterations to occur, we have used

EarlyStopping, EarlyStopping is basically used to stop iterations when the MSE error further stops decreasing. So instead of executing 100 complete 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 hybrid model helps in better training 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 time are very crucial. We must maintain a balance between accuracy and convergence rate, so 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.

Algorithm 2 Proposed Methodology: Electricity Load and Price Forecasting

---

Algorithm 2 Electricity Load and Price Forecasting;

Input: Electricity Load/Price Data 3 years X={x_1,x_2,x_3,… x_n};
Output: Forecasted Load/Price Values Y={y};
Feature Selection: F={f_1,f_2,f_3,… f_n**}**
    input: X={x_1,x_2,x_3,… x_n}
    target: y={y}
    x_train, x_test, y_train, y_test = train_test_split(X, y, split_ratio=0.3);
    X1=**Remove** constant_features(X)
    X2=**Remove** duplicate_features(X1)
    X3**=Remove** correlated_features(X2)
    Calculate mutual_info_regression (X3)
    SelectKBest(mutual_info_regression, k=5) (1)
    Calculate SFS (X3)
    SelectKBest(SFS, k=5) (2)

```
14                 F= Select common features from (1) and (2)
15   Date Determination train_x, test_x, train_y, test_y
16                 F1=Scaling   MinMax (F1)
17                 (train_x, test_x, train_y, test_y) =Splitting Training Set (F, 70%)
                                                             Validation Set (F, 30%)
18   Forecast Engine: Forecast Y={y}
19                 model=Sequential ()
20                 model.add(LSTM (train_x))
21                 model.add(Dropout (0.2))
22                 model.add(Bidirectional(GRU))
23                 model.add(Dense (1, activation= Relu))
24                 model.compile(optimizer=Adam, matrices=mse)
25                 LSTM-BiGRU model = model.fit (train_x, train_y,
                                        validation_data=(test_X, test_y),
                                        callbacks=[EarlyStopping]

26                 Y= LSTM-BiGRU model.predict(test_x)

27   Evaluation RMSE, MAE, MAPE, TIME
28                    Invert_Scaling (Y, test_y)
29                    RMSE= Calculate RMSE (Y, test_y)
30                    MAE=Calculate MAE (Y, test_y)
31                    MAPE=Calculate MAPE (Y, test_y)
32                    TIME= Calculate Execution time
```

3.1.4 Evaluation

3.1.4.1 Root Mean Square Error (RMSE)

The standard deviation of the prediction errors is known as RMSE. The prediction errors are generally consideration of prediction value that how far it is from the regression line [41]. It is calculated by below mentioned formula:

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

Where, i is the number of observations, yi is actual value and $\hat{y}_i$ is forecasted value.

3.1.4.2 Mean Absolute Percentage Error (MAPE)

To calculate accuracy of any forecast system, we calculate the MAPE value. Accuracy is measured in percentage. It is calculated by below mentioned formula:

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

Where, N is the number of observations**,** yi is actual value and $\hat{y_i}$ is forecasted value. The more it is close to zero, the more accurate results are [41, 42] . We have observed the limitations of MAPE when calculating it for Price forecasting. Fortunately, we have load values greater than 0, so we calculated the MAPE accurately. But we have 0 and negative values in price forecasting and thus we get a huge value of MAPE even when there happened a small deviation but in a negative or nearly zero value. For this purpose, we are calculating Mean Absolute Error (MAE) for Price forecasting especially and for load values as well. Let's discuss this below:

3.1.4.3 Mean Absolute Error (MAE)

MAE measures the distance between the actual data and the predicted data. It provides us with the absolute average between the actual and forecasted data points [42]. The formula is as follows:

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

Where $\hat{y_i}$ is predicted value, y_i is actual value and n is the number of observations. The more it is close to zero, the more accurate results are.

3.1.4.4 Convergence Rate (sec)

We have defined convergence rate as time elapsed. Time taken by algorithm to predict its first output is the convergence rate. We calculate it by using following python commands [13]:

start = timer.start () (16)

end = timer.end () (17)

Time Elapsed(s)=end – start (18)

So, that is how the whole proposed model is build. During building the model the dependency of every algorithm is checked and adjusted accordingly. Now it is time to build the proposed model and perform simulations on it. In the next Chapter we will explain the results of each phase possible for us. During this model, we have used the generic terms i.e., forecasting, we have not used the specified term Load Forecasting as we are going to perform simulations for Price Forecasting as well. Let us discuss the details of Load and Price Forecasting in the next chapter in detail.

Chapter 4

Results and Discussion

4.1 Forecasting with LSTM-BiGRU based hybrid Solution.

This chapter discusses the simulations and results while implementing our improved methodology for time series electricity load and price forecasting. Moreover, data analysis and behavior of load curves on yearly, daily, and hourly basis is also done for the data provided by the ISO NE. ISO is International System Operators for New England. It is responsible for reliably operating 32,000 MWH bulk electricity generation and transmission, and to provide tariffs for the prices of the energy supply in New England. In this work, three evaluation cases are presented to predict load and price values and performance of all methods is compared and evaluated through well-known forecasting errors. Simulations are performed utilizing Google Collaboratory.

4.2 Exploring Dataset

To better understand the dataset, we need to explore it first. The dataset is taken from ISONE. The dataset is available on the below mentioned link: https://www.iso-ne.com/isoexpress/web/reports/pricing/-/tree/zone-info . ISONE dataset is in excel sheet and it contains hourly data file including load, day-ahead and real-time prices for the ISO New England Control Area (ISO-NE CA) and the eight load wholesale zones. These zones include Massachusetts, Connecticut, Vermont, New Hampshire, Maine, and Rhode Island. The dataset included column as below:

Table 6: Exploring Features in Dataset

Column Name	***Explanation***
Date	*MM/DD/YYYY format.*
Hour	*Hour ending value with Hour 1 equal to the hour e nding at 1:00 am.*
DA_DEMD	*Day-ahead demand consisting of fixed and price sensitive demand bids plus decrement bids and increment offers.*
RT_DEMD	*RT_DEMAND is the Non-PTF Demand for ISO-NE CA, and the load zones as determined*

	by metering. Non-PTF Demand is the load
DA_LMP	*Day ahead locational marginal price.*
DA_EC	*Energy component of the day ahead price.*
DA_CC	*Congestion component of the day ahead price.*
DA_MLC	*Marginal loss component of the day ahead price.*
RT_LMP	*Real time locational marginal price.*
RT_EC	*Energy component of the real time price.*
RT_CC	*Congestion component of the real time price.*
RT_MLC	*Marginal loss component of the real time price.*
DryBulb	*Dry bulb temperature in degrees Fahrenheit for the corresponding weather station.*
DewPt	*Dew point temperature in degrees Fahrenheit for the corresponding weather station.*
SYSLoad (for ISO-NE CA only)	Actual system load in MW as determined by metering. The system load is used for day-ahead.
RegCP (on the ISO-NE CA Worksheet only)	Regulation Clearing Price in $/MWh.

Prices in the ISO-NE CA worksheet are the hub prices, and all prices are reported as $/MWh. All the model training and testing are performed on ISO-NE CA worksheets. We have considered 2016-2018 data for training the model and tested it on the 2019 dataset, respectively. Let us have some insights from the training dataset.

Dataset is numeric. Dataset is going to be used for supervised learning. Load and Price forecasting is a regression problem, we are applying a regression-based hybrid model on it.

4.3 Observing Electricity Consumption in Dataset

When it comes to observing a dataset, we observe different load patterns in the dataset. The load consumption varies around the day, on weekends, public holidays and due to weather situations. We have comprehended the dataset exploration below with a few load observations. As our dataset is hourly organized, we plot hours on x-axis and Load consumption values in

MWh on y-axis. We have mentioned titles of every graph to avoid confusion. The legends are marked below x-axis, respectively.

Load curves of 1st January of the years 2016, 2017 and 2018 as follows. In this graph, we observe the general load pattern in a single day and how it varies round the day. We are observing the load consumption curve for 3 years simultaneously, so that we get insights from it. The load value lies between 10,000 MWh and 20,000 MWH. As we are considering a dataset of ISO NE Control Area that is why the values of load consumption are very high.

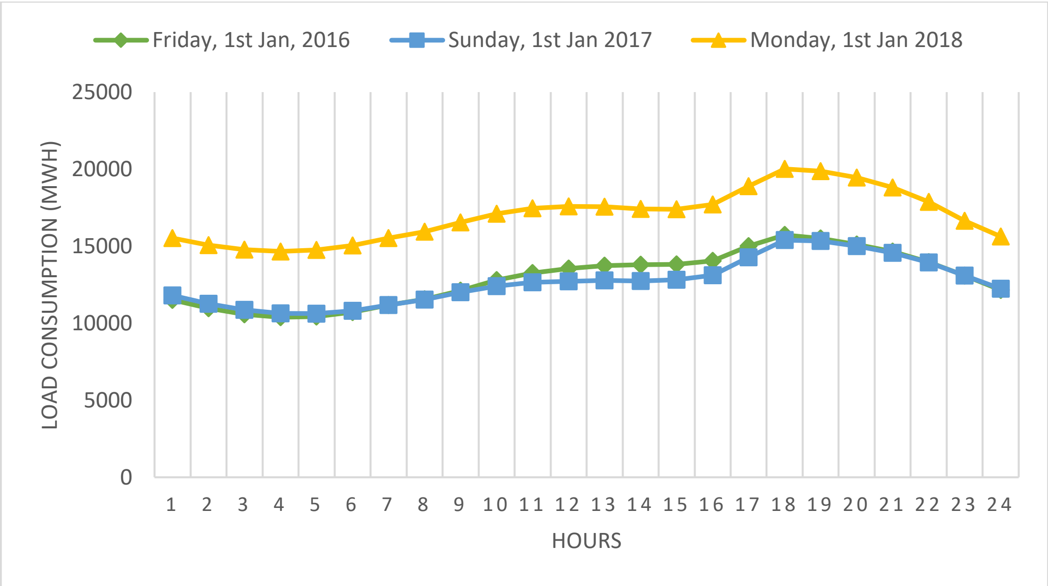

Figure 15: 1-Day Load Comparison Graph for 2016, 2017, and 2018

The load consumption around the day slightly rises in hours 9 to 15 and it decreases a bit and then shows its maximum curve of the day in hours between 17 and 20. And, then normalizes again. These are load curves of 1st January of the years 2016, 2017 and 2018, respectively.

Let us observe load curves of 1st June of the years 2016, 2017 and 2018, respectively. We made this observation considering it a summer day.

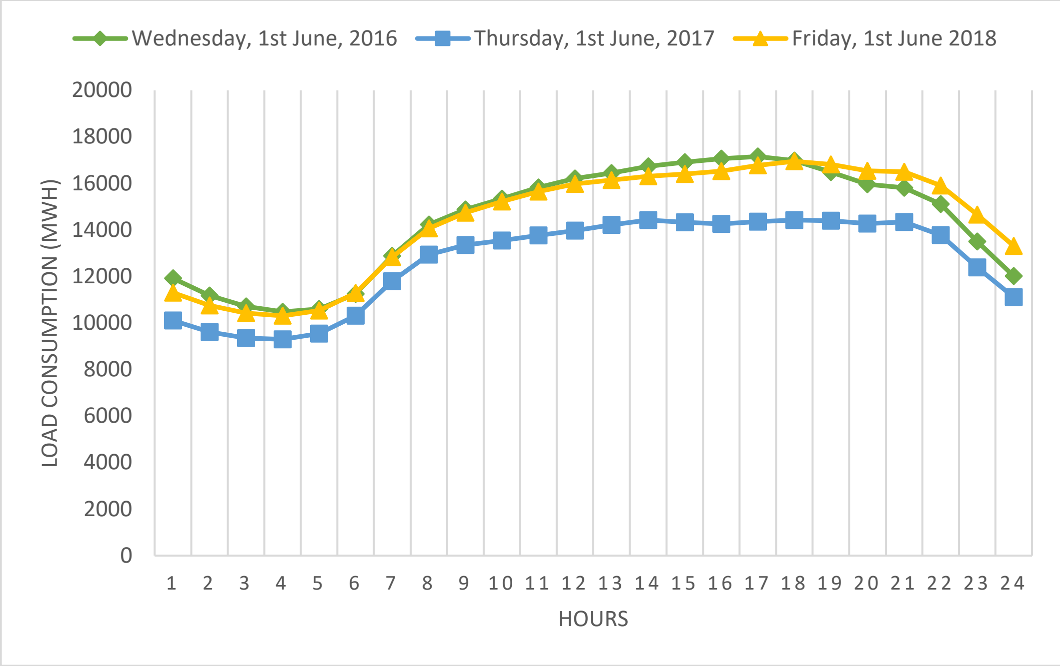

Figure 16: 1-Day Load Comparison Graph for 2016, 2017, and 2018

Here we observe that the pattern is very much different than a winter day. The load consumption increases from the 6th hour of the day and keeps on rising till 17^{th} and then it starts decreasing. Along with this observation, the load consumption of all the three years is almost similar. Here load consumption of 2018 is not more than the other two. This shows that in summers the energy consumption has not been increased to an entirely different load consumption range as in summers.

So, we conclude that the winter season of 2018 showed expanded energy consumption than the other years which means it will increase or remain approximately equal to it.

Load curves of the first week of January of years 2016, 2017 and 2018, respectively. In the below attached graph the 7 curves have been observed and each represents one day. Here we again compared the one-week load of winter season.

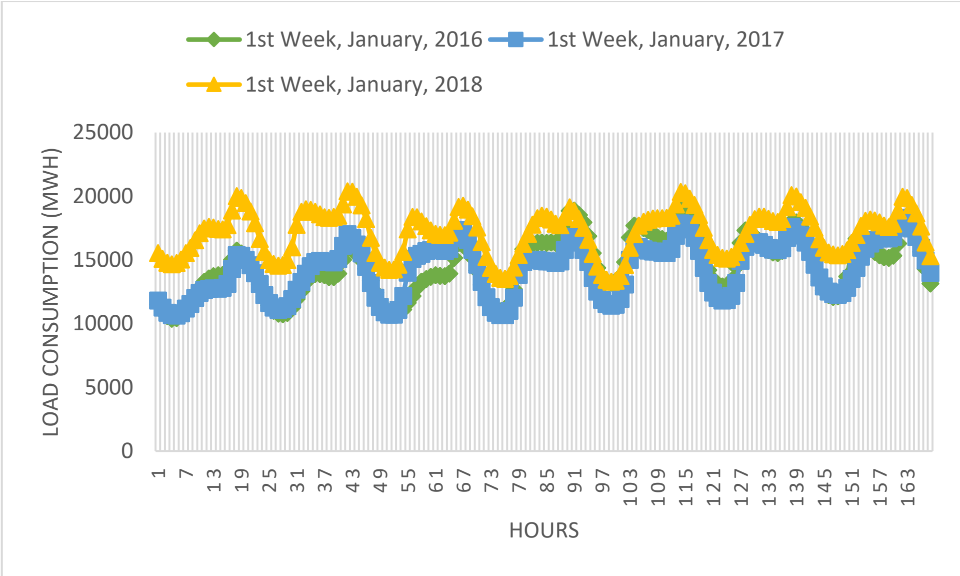

Figure 17: 1 Week Load Comparison for 2016,2017, and 2018

All the seven days have different load curves. Considering each day of three years we find similarities in the curve style, although load consumption fluctuations are more prominent in 2018, respectively.

We made a similar observation from the summer season. We observed load curves of the first week of June of 2016 2017 and 2018, respectively. And to our surprise the load curve of 2016 in the last 4 days of the week is higher than the other 2 years. The load consumption of 2017 is like 2018. The load curve of 2018 in last 4 days of the week is decreasing as compared to the trend in previous 2 days.

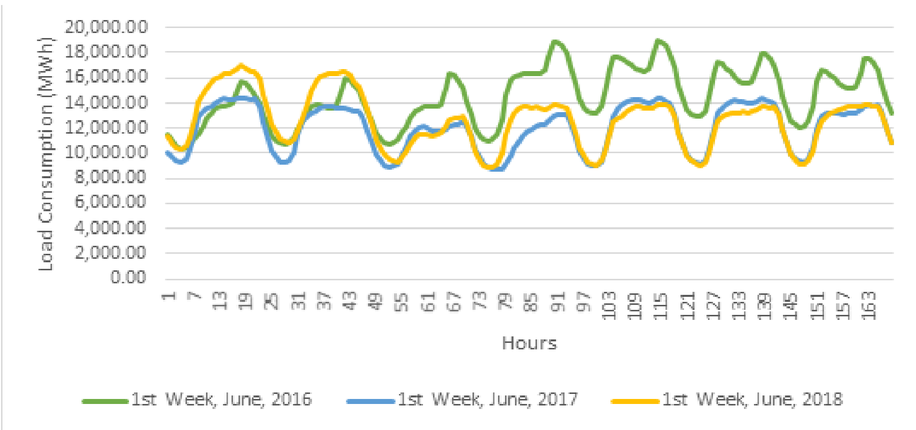

Figure 18: 1 Day Load Comparison for 2016, 2017 and 2018

So, we conclude that in winters the load consumption of 2018 is more than the other two which means the load curve is going to raise or remain the same to 2018 in the year 2019, respectively. But in the summer, the load of 2016 showed unexpected behavior. The 2018 curve does not show any major fluctuations, so we expect that in the year 2019 predictions we have more or less the same curves.

The observation of two weeks of load consumption is daily and weekly but when we observe the yearly load consumption only then we get an idea of the whole year. Consider the load curve of years 2016, 2017 and 2018.

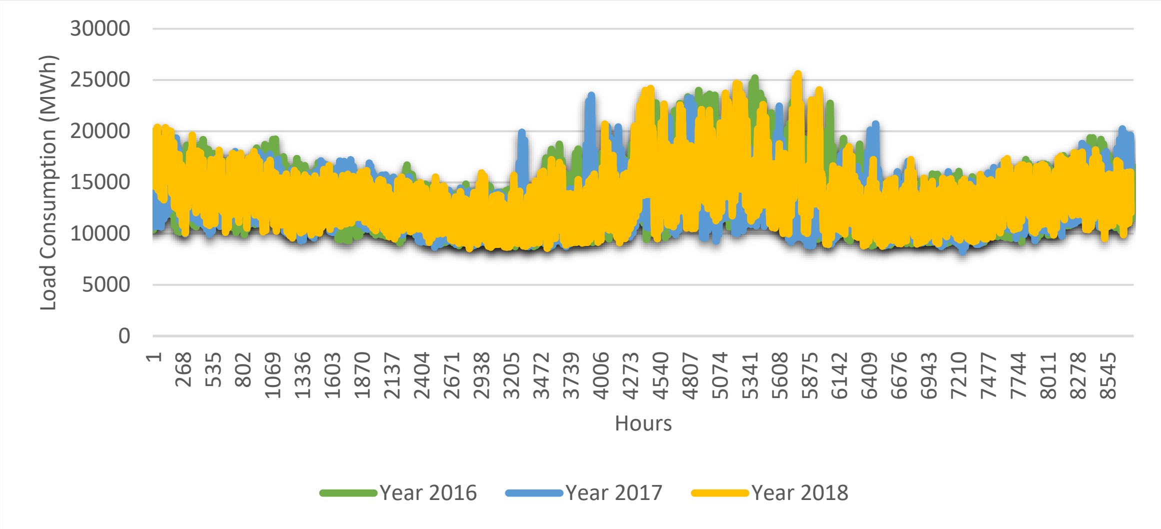

Figure 19: Yearly Comparison of Load Consumption

From the above graph we conclude that there is a rise in load consumption in the winter season and similarly in the summers too. So, the data exploration has been completed here. Let us perform simulations and come to the results section. Before this, let us observe feature selection.

4.4 Load Forecasting

4.4.1 Feature Engineering for Load Forecasting

First phase is feature engineering in which we undergo feature selection. Feature Selection undergoes multiple steps as explained in Chapter 3. Here we explain the results of those applied techniques. After removal of constant, duplicate and correlated features, we performed MI and SFS side by side and then took common features from them. For feature selection, we used the dataset of 2018 only. Let us discuss the feature selection below:

4.4.1.1 Mutual Information

We calculated MI of each feature with load consumption. We plot the following graph with the obtained value.

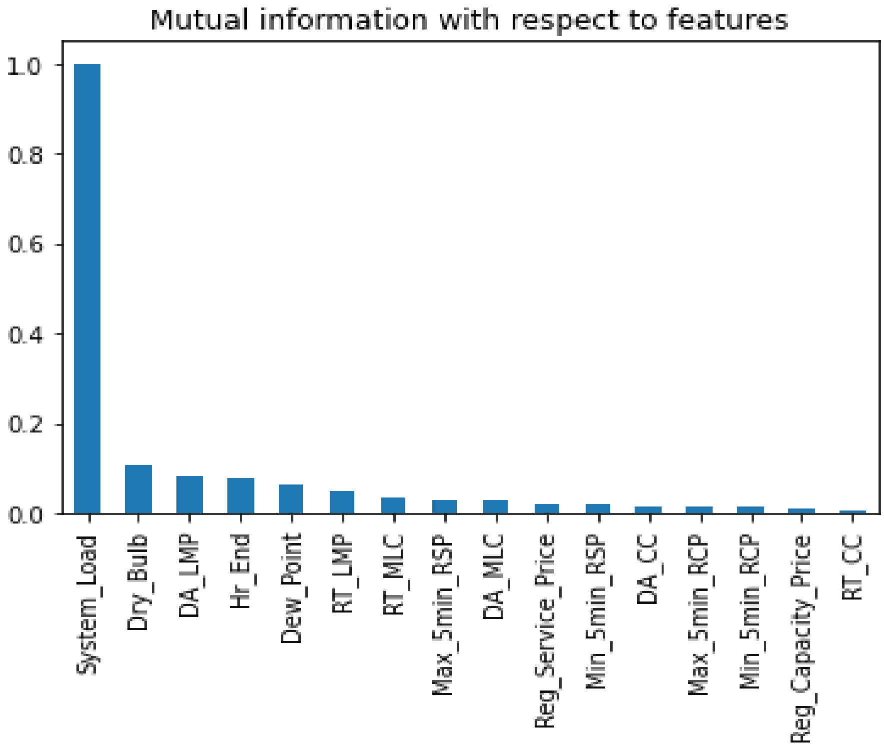

Figure 20: Mutual Information for Feature Selection (Load Forecasting)

It is obvious from the graph that the most relevant features of load demand are System_Load, Dry_Bulb , DA_LMP and continue. The least related feature is RT_CC. So, we considered the first 5 most relevant features for now.

Next, we performed SFS, the insights are explained below:

4.4.1.2 Sequential Forward Selection

SFS is a greedy way of selecting features. Here we generate a subset of features and perform regression as a learning algorithm on it. Then evaluate the performance and select the best features from them.

As described in the methodology section, here we must select 5 features out of all. Here in the attached graph below, blue lines are the calculated performance values against each feature while light below region shows the standard deviation

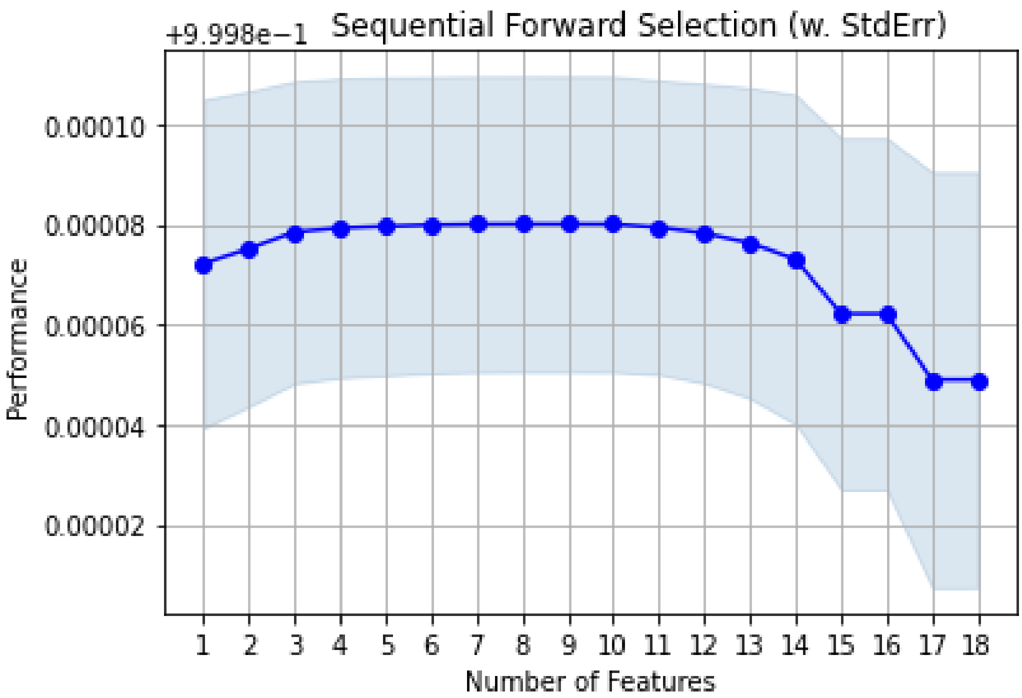

Figure 21: Sequential Feature Selection for Feature Selection (Load Forecasting)

The algorithm begins with 0 features, and it keeps on adding a feature and calculating the importance. Here in the attached graph, we have importance of each feature marked with its standard deviation value. We have fixed a threshold of 5 features. So, the 5 selected features came out are

'DA_LMP', 'Dew_Point', 'System_Load', 'Min_5min_RSP', and 'Min_5min_RCP'.

4.4.1.3 Common Features

We select features from both section methods described above. As we are working on short term load forecasting, we must make everything to select with best possible precision. We select common features from the feature selecting methods. They were 3 in number. These are as follows

'DA_LMP', 'System_Load', 'Dew_Point'

Now, we use these features in the next data determination step and forecasting steps.

We have used a dataset from ISO NE CA. The dataset is hourly based. We used a dataset of three years 2016, 2017 and 2018 for training and evaluating the model and performing testing in 2019’s dataset, respectively.

4.4.2 Model Training Loss Graph

We are using 90% data in training and 10% in validation. We will be using separate data from 2019 for testing models. Considering the scope of our project we will test the model for one day, and one week forecasting. Some other hyper parameters are as follows:

Table 7: List of Optimized Hyper Parameters Selected during model building

Input Sequence Size	24 or 168
Output Sequence Size	24 or 168
Training Batch Size	70% (24,544 rows)
Validation Batch Size	30% (10,519 cols)
Testing Batch Size	Separate time series of hrs (24 or 168)
Batch size	1024
Activation Function	ReLU
Optimizer	Adam
Learning Rate	0.18 to 0.01
Dropout	0.2
Number of LSTM Units	100
Number of Bidirectional GRU Units	50

The composition of the proposed hybrid model has been explained in Chapter 3. The graph attached below is the graph for calculating training loss during the model training phase. The error is measured in MSE.

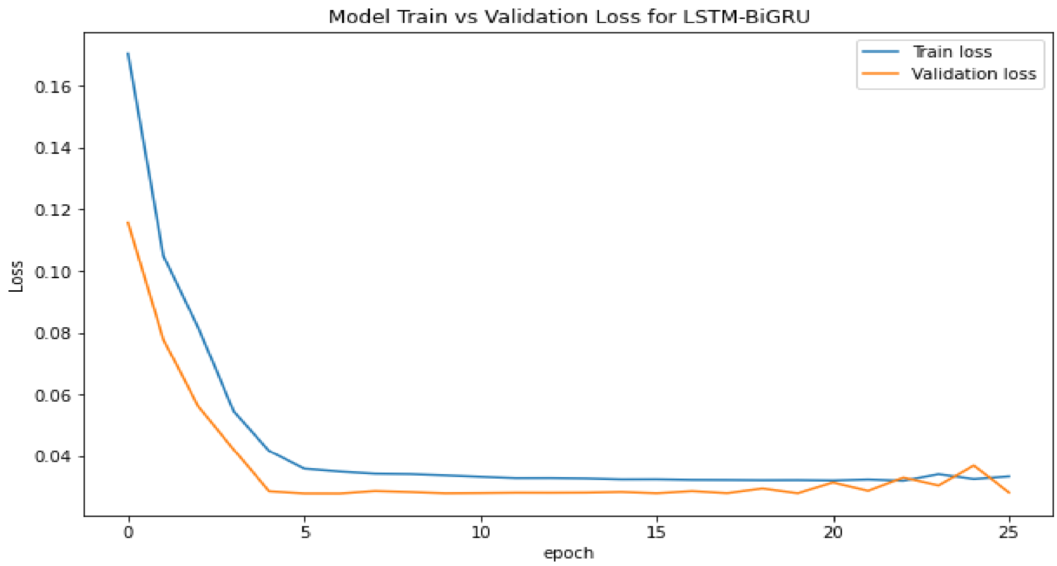

Figure 22: Training Loss graph for Load Forecasting

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. Both the cures are seen very near to each other that means error values are least at that time. It means the model got trained very well. The minor fluctuations are okay as validation loss is fluctuating while training loss is very stable. The second observation is 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.

4.4.2.2 LSTM-BiGRU based Forecasting

After training the model, now it is time for forecasting. So, here we are going to present results of 1 day and 1-week graphs of load and price forecasting. We train the model once and then on base of that training we predict the time series. The predictions of one day and one week are predicted at the same time, and we just draw graphs to display the accuracy of the proposed solution.

4.4.2.2.1 One Day Load Forecast 2nd January 2019

The attached graph below is the graph for one day forecast for the month of January date 2nd

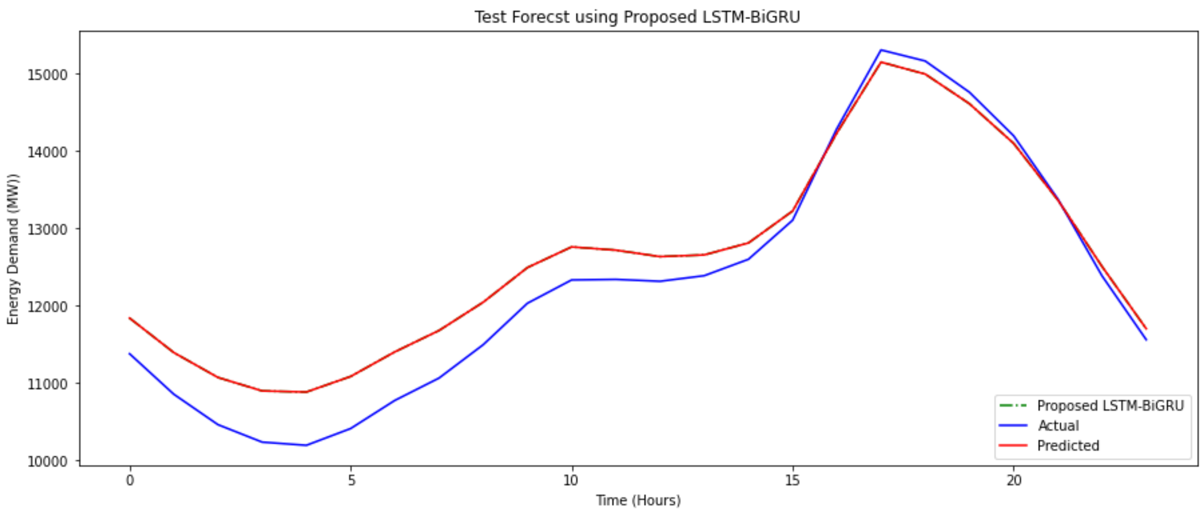

Figure 23: One Day Load Forecast (January 2019)

It is clearly seen that the predicted values follow the pattern as in the actual data graph. The MAPE error calculated is 3.12% and the deviations from the actual values in terms of RMSE is calculated and attached in the graph below.

4.4.2.2.2 One Week Load Forecast January 2019

The attached graph below is the graph for one week forecast for the month of January.

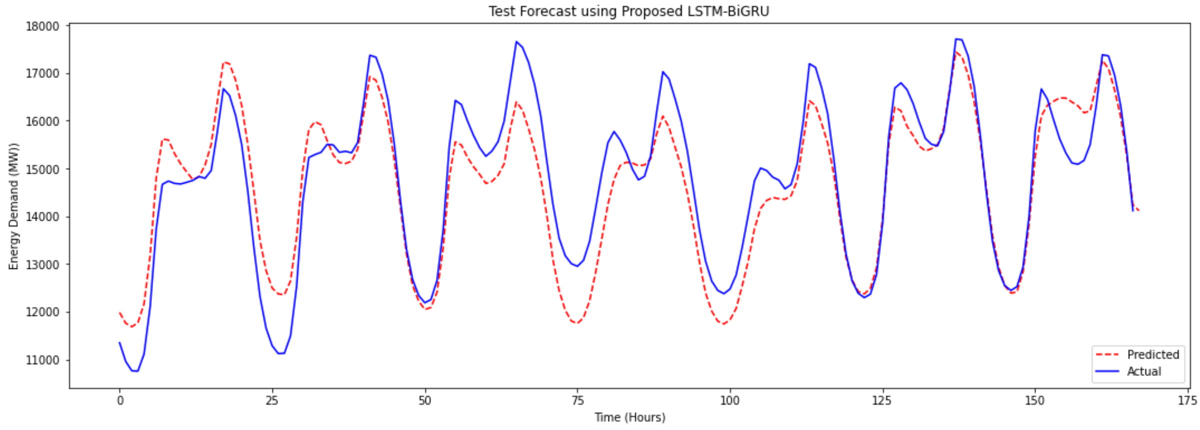

Figure 24: One Week Load Forecast (January 2019)

It is clearly seen that the predicted values follow the pattern as in the actual data graph. The MAPE error calculated is 4.07% and the deviations from the actual values in terms of RMSE is calculated and attached in the graph below.

As we have forecasted energy demand for the month of June, similarly we forecasted energy for the month of July as well. And for forecasting energy for July, we train our model with time series of 3.5 years and then used that specific time series in forecasting, respectively

4.4.2.2.3 One Day Load Forecast July 2nd, 2019

The attached graph below is the graph for one day forecast for the month of July date 2nd

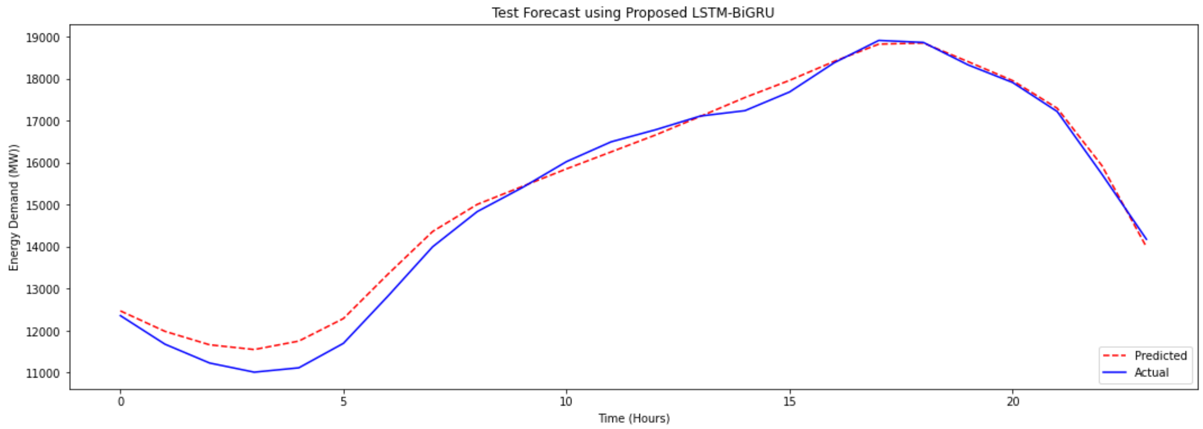

Figure 25: One Day Load Forecast (July 2019)

It is clearly seen that the predicted values follow the pattern as in the actual data graph. The MAPE error calculated is 1.76% and the deviations from the actual values in terms of RMSE is calculated and attached in the graph below.

4.4.2.2.4 One Week Load Forecast July 2019

The attached graph below is the graph for one week forecast for the month of January.

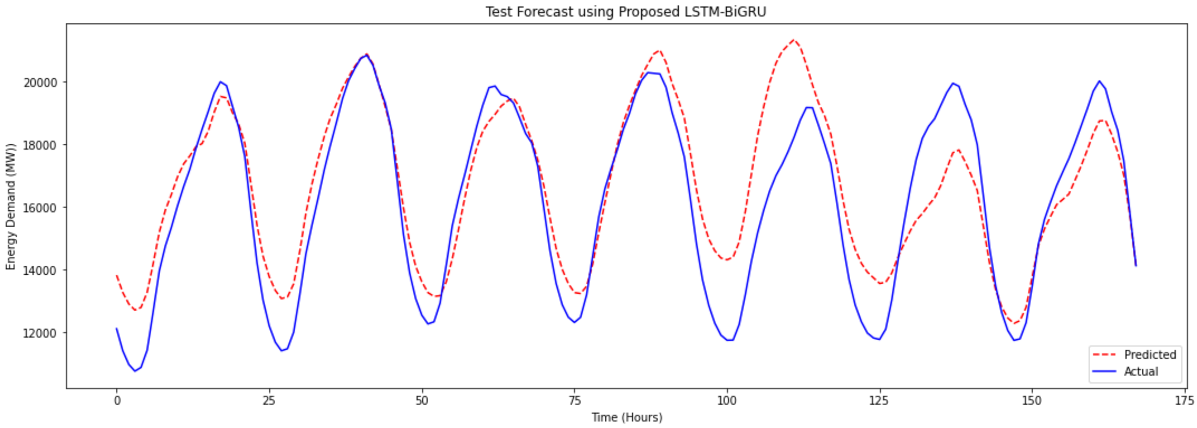

Figure 26: One Week Load Forecast (July 2019)

It is clearly seen that the predicted values follow the pattern as in the actual data graph. The MAPE error calculated is 7.42% and the deviations from the actual values in terms of RMSE is calculated and attached in the graph below.

4.4.2.3 Comparison graphs

4.4.2.3.1 One Day Load Forecast 2nd January 2019

Consider the following graph, solid black graph representing the actual forecast values while green is representing the proposed LSTM-BiGRU model forecast.

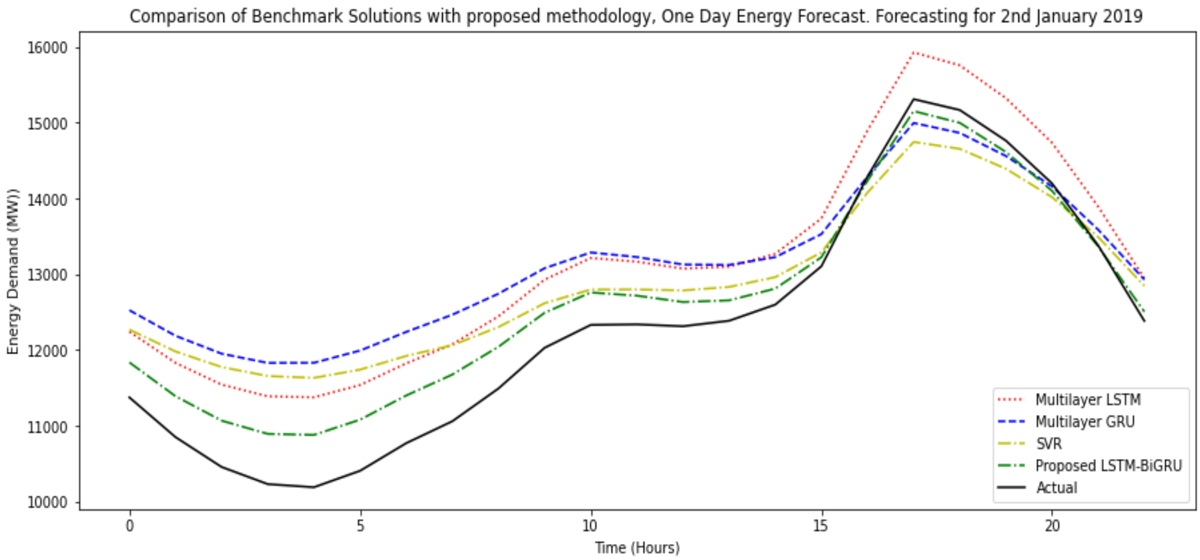

Figure 27: Comparison graph for Load Forecast 2nd January,2019

Considering the above graph, it is clearly seen that the proposed solution performs better than the other benchmark solution with the decreased computational time. The computational graph is attached at the end of the section. Please consider the below table for MAPE error in the forecasting

Consider the MAPE calculated in the above-mentioned graph.

Table 8: MAPE on 2nd January 2019 Load Forecasting

Hr	LSTM	GRU	SVR	LSTM-BiGRU
1-24	6.74%	7.53%	5.94%	3.12%

It is clearly seen that the MAPE predicted in one day forecasting is only 3.12% that is the best result among the presented comparison result.

Consider the MAE calculated as follows:

Table 9: MAE on 2nd January 2019 Load Forecasting

Hr	LSTM	GRU	SVR	LSTM-BiGRU
1-24	739.48	853.47	677.01	355.38

It is clearly seen that the MAE predicted in one day forecasting is only 355.38that is the best result among the presented comparison result.

For considering the deviation at hourly level, we have calculated RMSE, please consider the following graph

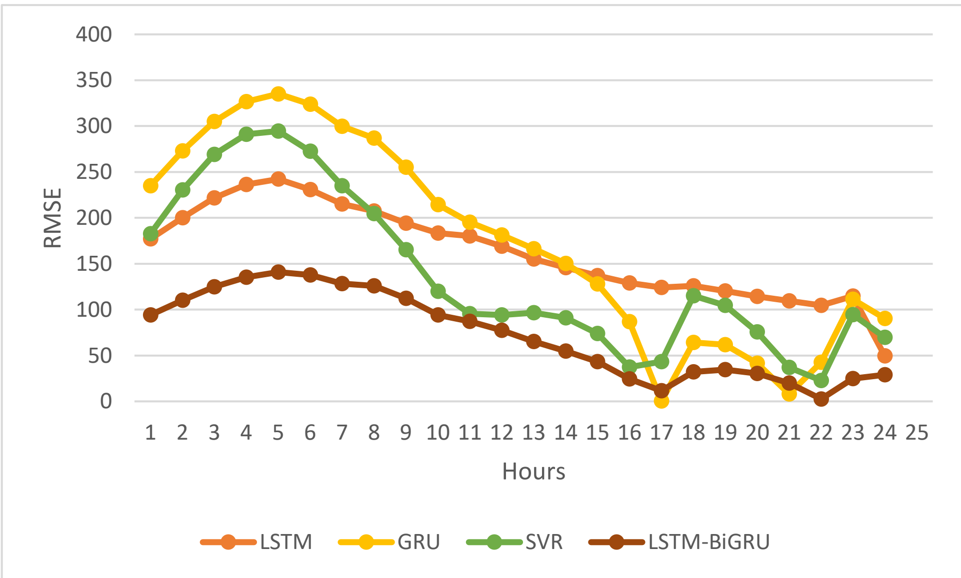

Figure 28: Hourly RMSE deviation on 2nd January 2019 Load Forecasting

However, we have also calculated the RMSE of the whole 24hrs together to get a clear picture. Please consider the table below:

Table 10: RMSE on 2nd January 2019 Load Forecasting

Hr	**LSTM**	**GRU**	**SVR**	**LSTM-BiGRU**
1-24	829.52	1003.32	795.90	419.49

It is obvious from the above table that RMSE calculated for BI-GRU at each hour is less than the comparison models. There may be some hours where RMSE of the proposed solution be higher than the comparison models. We consider it alright as we are comparing them with the best benchmark solutions available and our aim of writing this thesis is to make a fairly accurate rate between forecasting and computational time. On average or considering load forecasting for the day it is alright of the proposed solution missed accuracy in a very few hours. In rest of the cases, the accuracy of forecasting is very close to original values.

4.4.2.3.2 One Week Load Forecast January 2019

Consider the following graph, solid black graph representing the actual forecast values while green is representing the proposed LSTM-BiGRU model forecast.

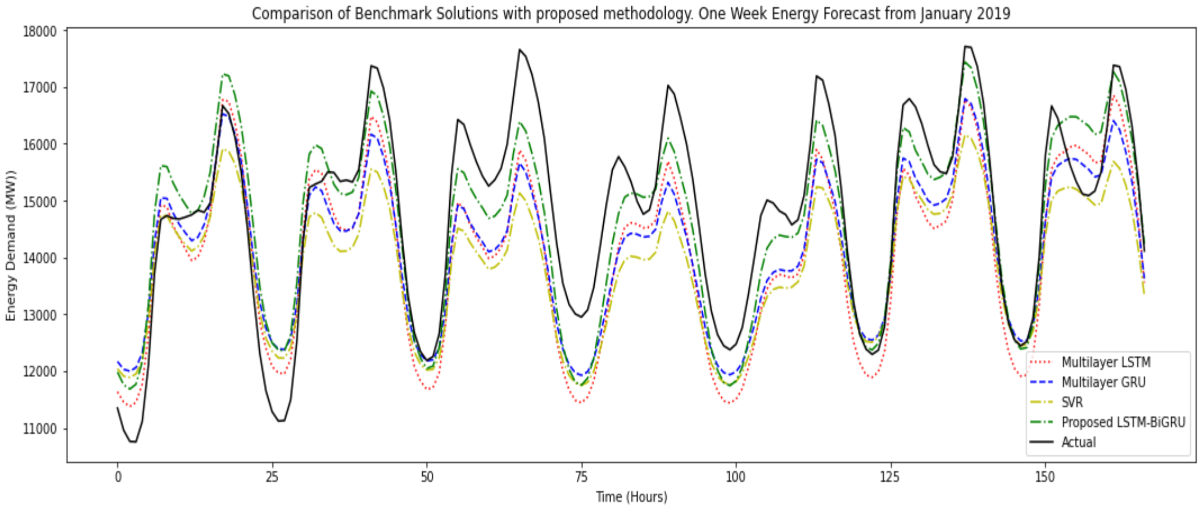

Figure 29: Comparison graph for Load Forecast 1st Week January 2019

Considering the above graph, it is clearly seen that the proposed solution performs better than the other benchmark solution with the decreased computational time. The computational graph is attached at the end of the section. Please consider the below table for MAPE error in the forecasting

Consider the MAPE calculated in the above-mentioned graph.

Table 11: MAPE on 1st Week of January 2019 Load Forecasting

Hr	LSTM	GRU	SVR	LSTM-BiGRU
1-168	6.15%	5.58%	7.04%	4.07%

It is clearly seen that the MAPE predicted in one day forecasting is only 4.0% that is the best result among the presented comparison result.

Consider the MAE calculated in the above-mentioned graph.

Table 12: MAE on 1st Week of January 2019 Load Forecasting

Hr	LSTM	GRU	SVR	LSTM-BiGRU
1-168	910.65	829.72	1062.81	585.57

It is clearly seen that the MAE predicted in one day forecasting is only 1110.89 that is the best result among the presented comparison result

For considering the deviation at hourly level, we have calculated RMSE, please consider the following graph

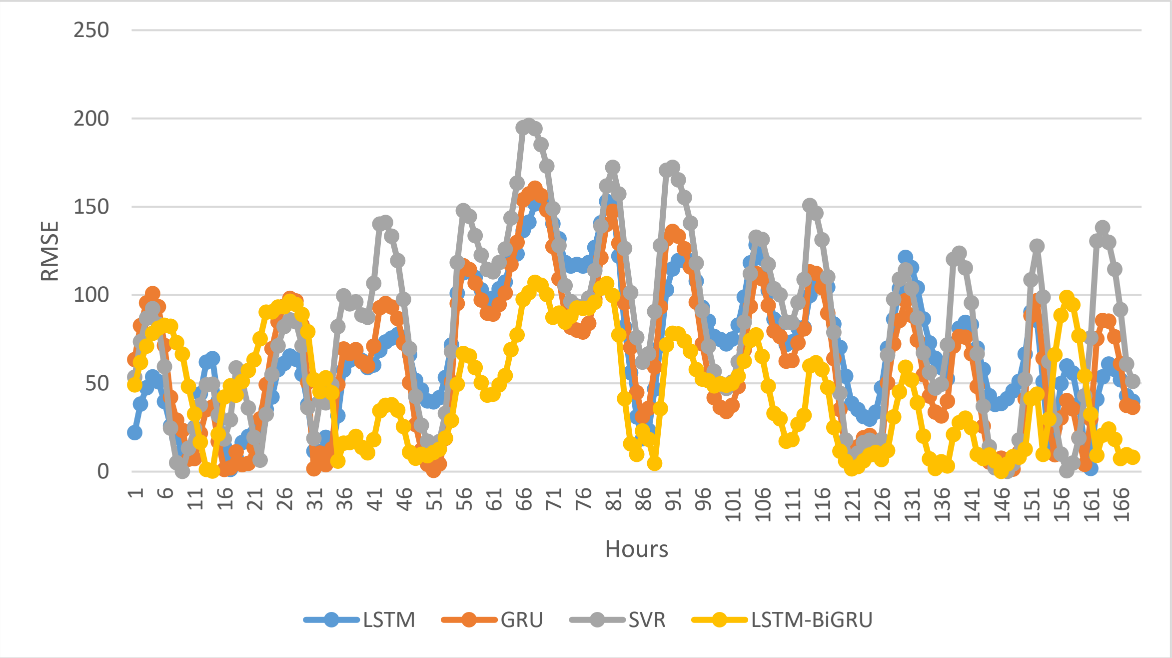

Figure 30: RMSE hourly deviation for 1st week of Jan, 2019 Load Forecasting

However, we have also calculated the RMSE of the whole 24hrs together to get a clear picture. Please consider the table below:

Table 13: RMSE on 1ST Week of January 2019 Load Forecasting

Hr	LSTM	GRU	SVR	LSTM-BiGRU
1-168	1029.53	983.23	1238.55	711.422

It is obvious from the above table that RMSE calculated for BI-GRU at each hour is less than the comparison models. There may be some hours where RMSE of the proposed solution be higher than the comparison models. We consider it alright as we are comparing them with the best benchmark solutions available and our aim of writing this thesis is to make a fairly accurate rate between forecasting and computational time. On average or considering load forecasting for the day it is alright of the proposed solution missed accuracy in a very few hours. In rest of the cases, the accuracy of forecasting is very close to original values.

4.4.2.3.3 One Day Load Forecast 2nd July 2019

Consider the following graph, solid black graph representing the actual forecast values while green is representing the proposed LSTM-BiGRU model forecast.

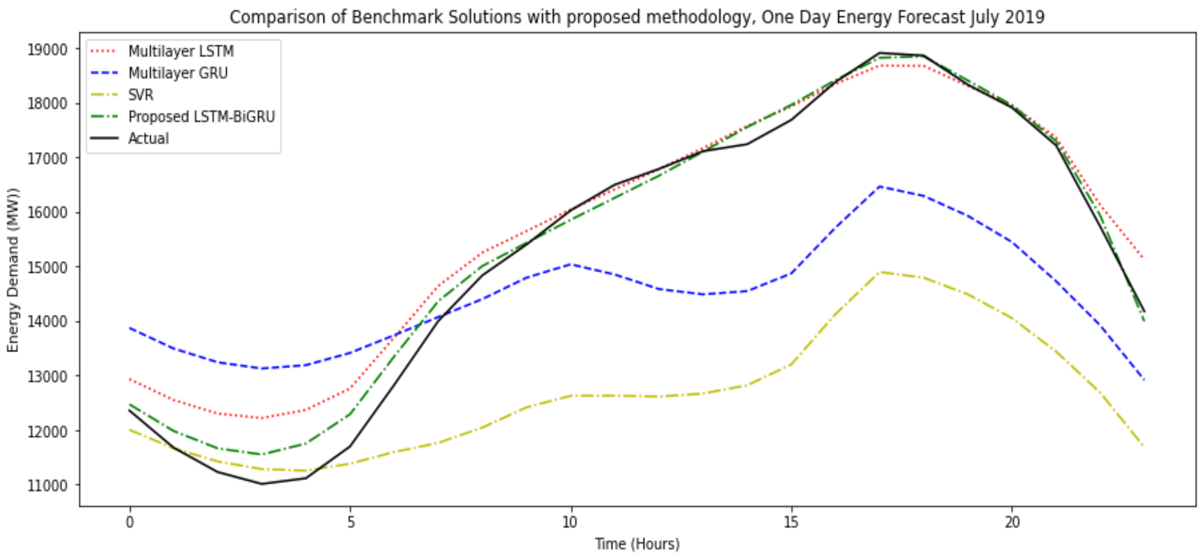

Figure 31: Comparison graph for Load Forecast 2nd July 2019

Considering the above graph, it is clearly seen that the proposed solution performs better than the other benchmark solution with the decreased computational time. The computational graph is attached at the end of the section. Please consider the below table for MAPE error in the forecasting

Consider the MAPE calculated in the above-mentioned graph.

Table 14 : MAPE on 2nd July 2019 Load Forecasting

Hr	**LSTM**	**GRU**	**SVR**	**LSTM-BiGRU**
1-24	3.58%	12.15%	16.19%	1.76%

It is clearly seen that the MAPE predicted in one day forecasting is only 1.76% that is the best result among the presented comparison result.

Consider the MAE calculated in the above-mentioned graph.

Table 15: MAE on 2nd July 2019 Load Forecasting

Hr	**LSTM**	**GRU**	**SVR**	**LSTM-BiGRU**
1-24	456.93	1848.52	2694.62	231.72

It is clearly seen that the MAE predicted in one day forecasting is only 231.72 that is the best result among the presented comparison result.

For considering the deviation at hourly level, we have calculated RMSE, please consider the following graph.

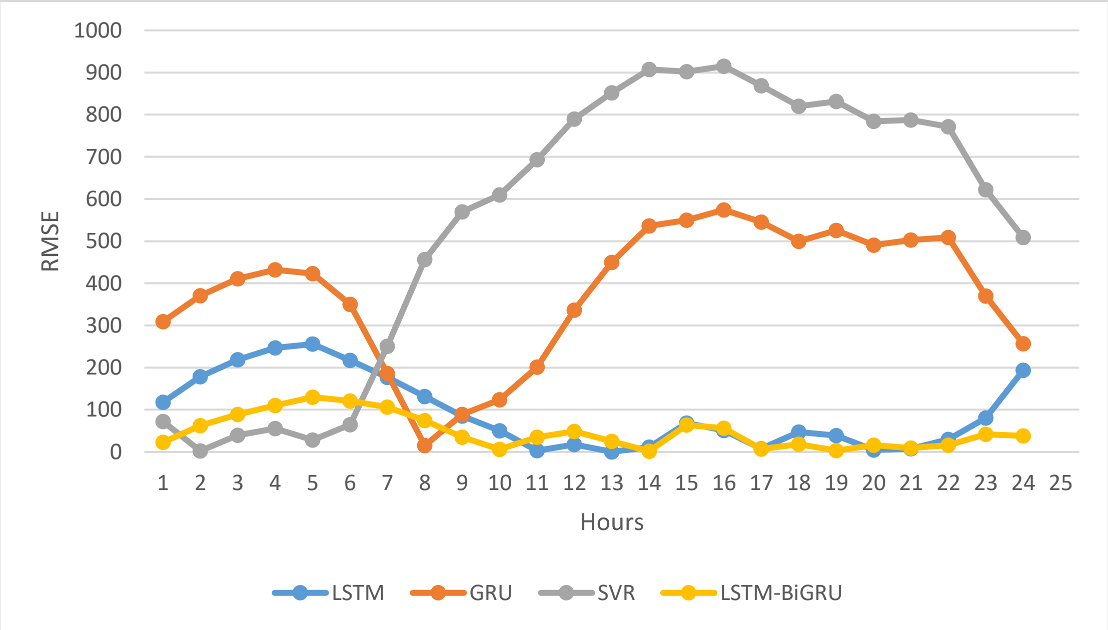

Figure 32: Hourly RMSE deviation for 2nd July, 2019 Load Forecasting

However, we have also calculated the RMSE of the whole 24hrs together to get a clear picture. Please consider the table below:

Table 16: RMSE on 2nd July 2019 Load Forecasting

Hr	LSTM	GRU	SVR	LSTM-BiGRU
1-24	617.64	2000.17	3143.06	299.26

It is obvious from the above table that RMSE calculated for BI-GRU at each hour is less than the comparison models. There may be some hours where RMSE of the proposed solution be higher than the comparison models. We consider it alright as we are comparing them with the best benchmark solutions available and our aim of writing this thesis is to make a fairly accurate rate between forecasting and computational time. On average or considering load forecasting for the day it is alright of the proposed solution missed accuracy in a very few hours. In rest of the cases, the accuracy of forecasting is very close to original values.

4.4.2.3.4 One Week Load Forecast July 2019

Consider the following graph, solid black graph representing the actual forecast values while green is representing the proposed LSTM-BiGRU model forecast.

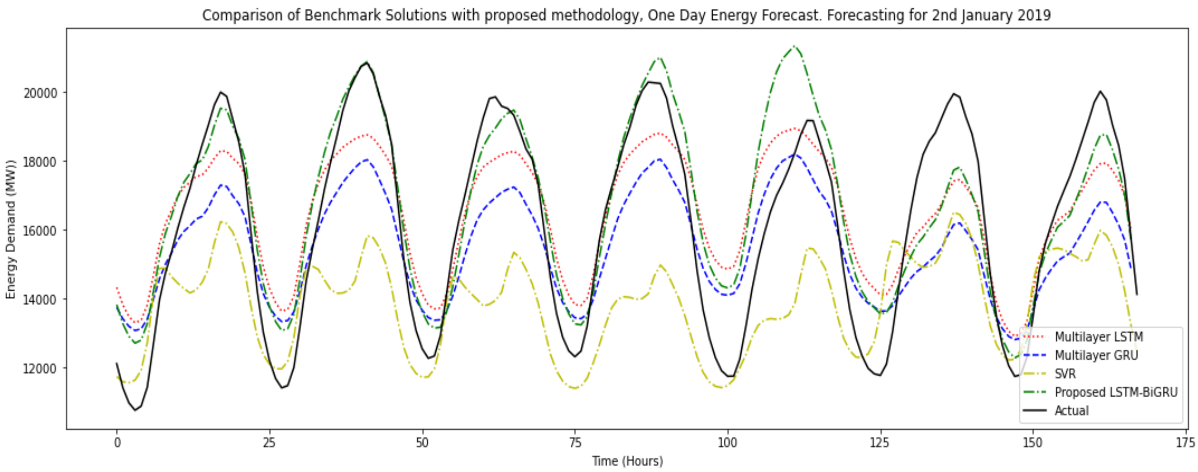

Figure 33: Comparison graph for Load Forecast 1st Week July, 2019

Considering the above graph, it is clearly seen that the proposed solution performs better than the other benchmark solution with the decreased computational time

The computational graph is attached at the end of the section. Please consider the below table for MAPE error in the forecasting

Consider the MAPE calculated in the above-mentioned graph.

Table 17: MAPE on 1st Week July 2019 Load Forecasting

Hr	LSTM	GRU	SVR	LSTM-BiGRU
1-168	9.02%	9.85%	14.73%	7.42%

It is clearly seen that the MAPE predicted in one day forecasting is only 7.42% that is the best result among the presented comparison result.

Consider the MAE calculated in the above-mentioned graph.

Table 18: MAE on 1st Week July Load Forecasting

Hr	LSTM	GRU	SVR	LSTM-BiGRU
1-168	1353.5	1605.21	2573.35	1110.89

It is clearly seen that the MAE predicted in one day forecasting is only 1110.89 that is the best result among the presented comparison result.

For considering the deviation at hourly level, we have calculated RMSE, please consider the following graph.

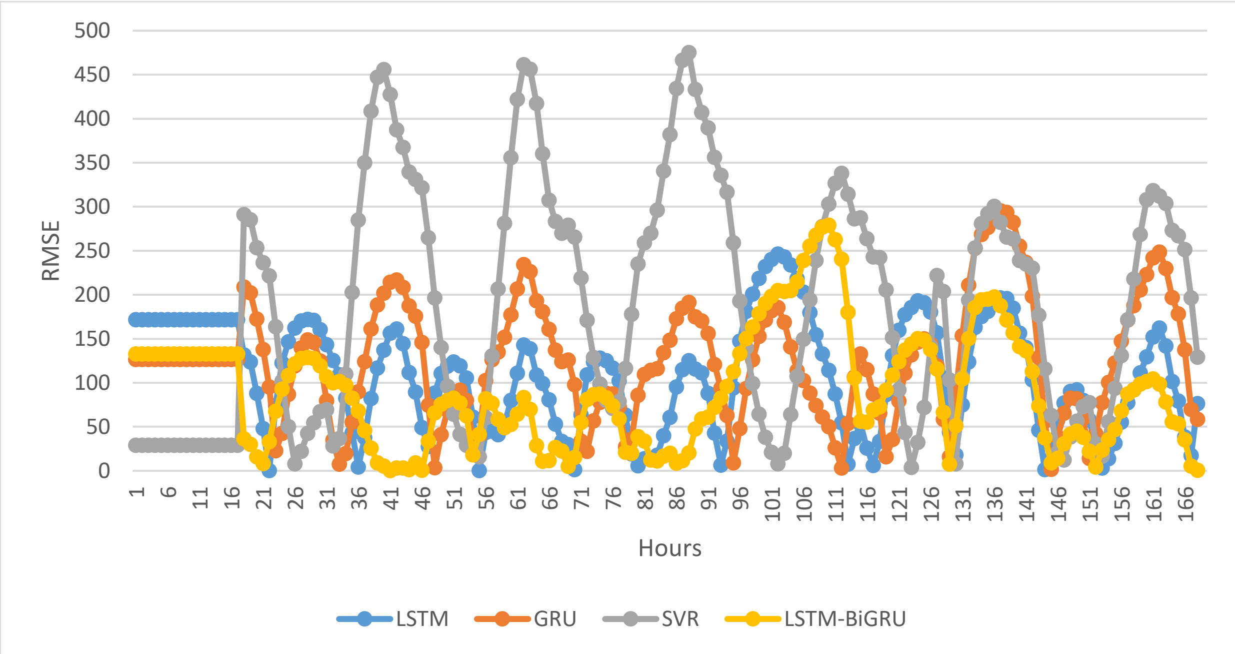

Figure 34: Hourly RMSE deviation for 1st week July, 2019 Load Forecasting

However, we have also calculated the RMSE of the whole 24hrs together to get a clear picture. Please consider the table below:

Table 19: RMSE on 1st Week July 2019 Load Forecasting

Hr	LSTM	GRU	SVR	LSTM-BiGRU
1-168	1574.06	1843.82	3067.18	1400.11

It is obvious from the above table that RMSE calculated for BI-GRU at each hour is less than the comparison models. There may be some hours where RMSE of the proposed solution be higher than the comparison models. We consider it alright as we are comparing them with the best benchmark solutions available and our aim of writing this thesis is to make a fairly accurate rate between forecasting and computational time. On average or considering load forecasting for the day it is alright of the proposed solution missed accuracy in a very few hours. In rest of the cases, the accuracy of forecasting is very close to original values.

4.4.2.3.5 Yearly Load Forecast 2019

We have computed the load forecast for the year 2019, that is as follows:

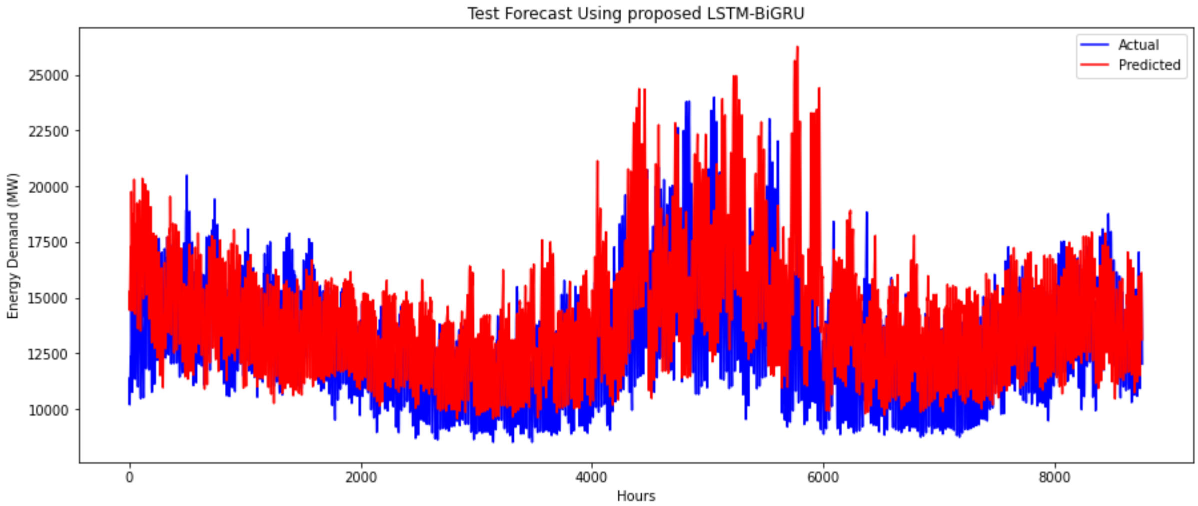

Figure 35: Yearly Load Forecast for Year 2019

The scope of this manuscript is for short term load forecasting, we have computed the whole year just to get an idea of overall efficiency of the proposed solution. The error metrices are as follows:

Hr	MAE	RMSE	MAPE (%)	Execution time
1-8760	1610.40	2129.050	12.21	34.55 (sec)

4.4.2.4 **Computational Time for January Forecast**

Here we are summarizing the computational time for one day forecast and one week forecast together. We are doing this together because our model is trained once and hence it predict forecast value of one day or week in almost same time. The computational Time here is inclusive of feature section (10.07 sec). Consider the table below:

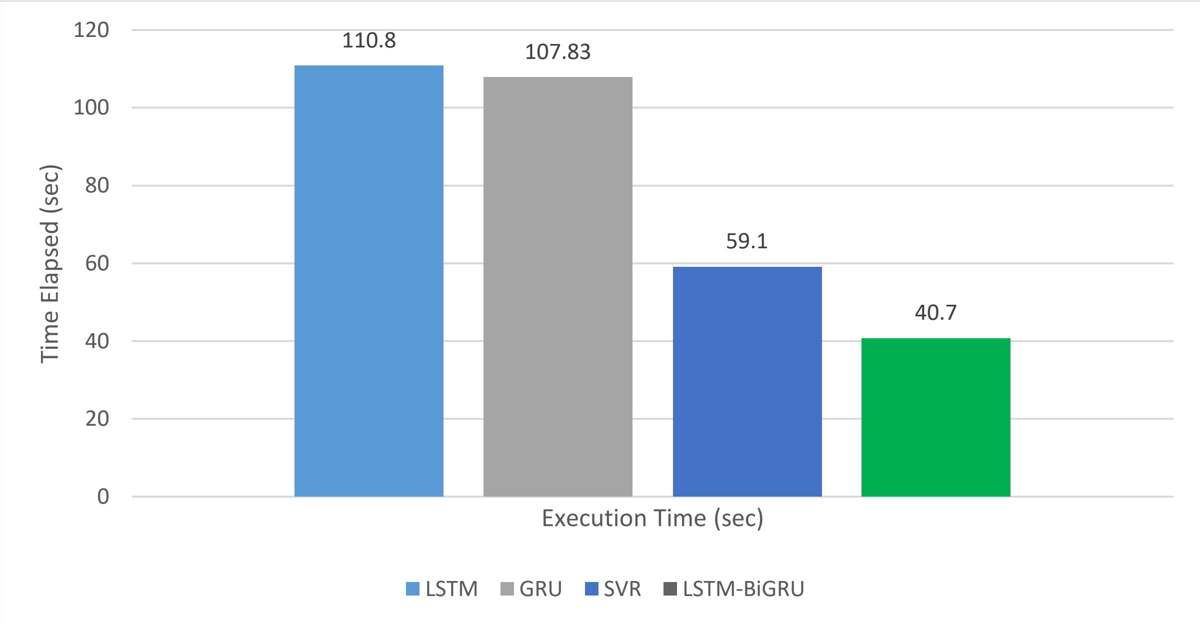

Figure 36: Model Computational Time for Load Forecasting January 2019

The calculated execution time is in seconds. It is clear from the above-mentioned table the execution time of the proposed solution is almost half of the other proposed solutions. It means that the proposed solution is providing the best forecasting results in the least possible time, hence accomplishing the objective of this research.

4.4.2.5 Computational Time for July Forecast

Here we are summarizing the computational time for one day forecast and one week forecast together. We are doing this together because our model is trained once and hence it predict forecast value of one day or week in almost same time. The computational Time here is inclusive of feature section (10.07 sec). Consider the table below:

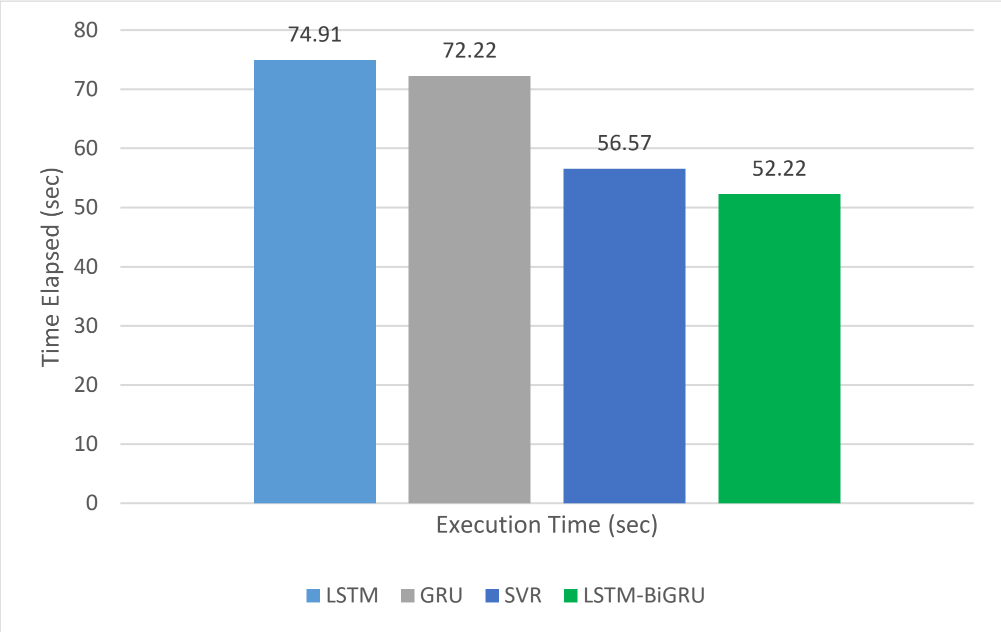

Figure 37: Model Computational Time for Load Forecasting July, 2019

The calculated execution time is in seconds. It is clear from the above-mentioned table the execution time of the proposed solution is almost half of the other proposed solutions. It means that the proposed solution is providing the best forecasting results in the least possible time, hence accomplishing the objective of this research.

4.3 Price Forecasting

Price Forecasting is like Load forecasting, except we changed the target variable. For load forecasting the target variable that we are going to predict is DEMAND while in price forecasting the target value is RT_LMP that is Real Time Locational Marginal Price

4.3.1 Feature Engineering for Price Forecasting

Feature Engineering for Price Forecasting follows the same steps as Load forecasting. It includes removal of constant, duplicate and correlated features. We performed MI and SFS side by side and then took common features from them. For feature selection, we used the dataset of 2018 only. Let us discuss the feature selection below:

4.3.1.1 Mutual Information

We calculated MI of each feature with load consumption. We plot the following graph with the obtained value. Please consider the figure

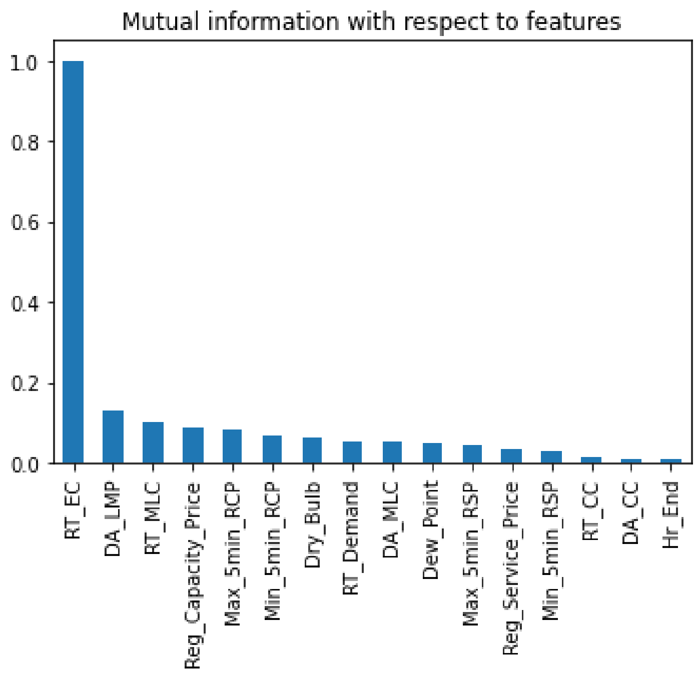

Figure 38: Mutual Information for Feature Selection (Price Forecasting)

Next, we performed SFS, the insights are explained below:

4.3.1.2 Sequential Forward Selection

We have already explained the SFS in CHapter3 and in Section Feature Engineering for Load Forecasting.

As described in the methodology section, here we must select 5 features out of all. Here in the attached graph below, blue lines are the calculated performance values against each feature while light below the region shows the standard deviation. Whereas we can’t see light blue section for all data points as they don't deviate from the performance value.

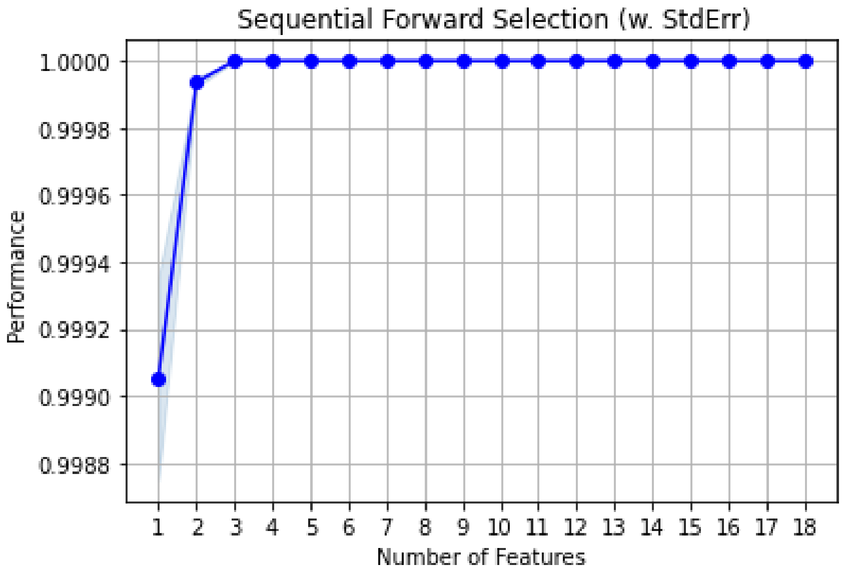

Figure 39: Sequential Foreword Selection for Feature Selection (Price Forecasting)

The algorithm begins with 0 features, and it keeps on adding a feature and calculating the importance. Here in the attached graph, we have importance of each feature marked with its standard deviation value. We have fixed a threshold of 5 features. So, the 5 selected features came out are

"DA_LMP, DA_EC, RT_EC, RT_CC, RT_MLC"

4.3.1.3 Common Features

We select features from both section methods described above. As we are working on short term load forecasting, we must make everything to select with best possible precision. We select common features from the feature selecting methods. They were 3 in number. These are as follows

"DA_LMP, RT_EC, RT_MLC"

Now, we use these features in the next data determination step and forecasting steps.

4.3.2 Model Training Loss Graph

We have used a dataset from ISO NE CA. The dataset is hourly based. We used a dataset of three years 2016, 2017 and 2018 for training and evaluating the model and performing testing in 2019's dataset, respectively. We are using 90% data in training and 10% in validation. We will be using separate data from 2019 for testing models. Considering the scope of our project we will test the model for one day, and one week forecasting. The composition of the proposed hybrid model has been explained in Chapter 3. However, the hyper parameters selected during training and testing of the model are as follows:

Table 20: List of Optimized Hyper Parameters Selected during model building

Input Sequence Size	24 or 168
Output Sequence Size	24 or 168
Training Batch Size	70% (24,544 rows)
Validation Batch Size	30% (10,519 cols)
Testing Batch Size	Separate time series of Price (24 or 168)
Batch size	1024
Activation Function	ReLU
Optimizer	Adam
Learning Rate	0.024 to 0.001
Dropout	0.2
Number of LSTM Units	100
Number of Bidirectional GRU Units	50

The graph attached below is the graph for calculating training loss during the model training phase. The error is measured in MSE.

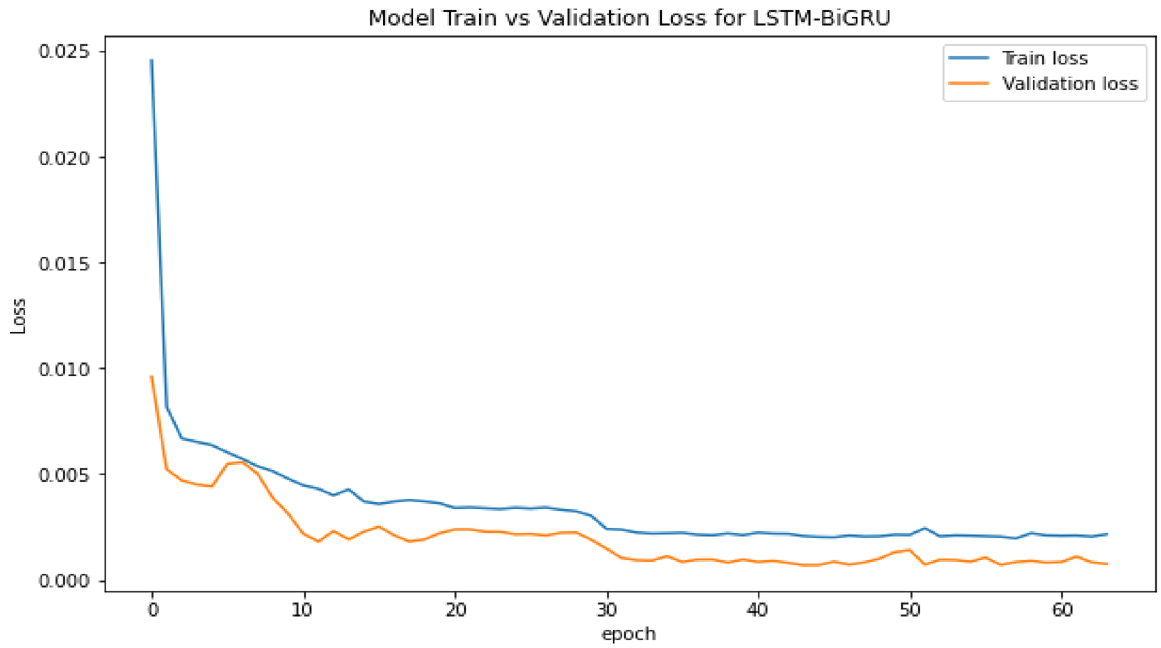

Figure 40: Training Loss graph for Price Forecasting

It can be seen in the graph that the training MSE has decreased from 0.025 to below 0.004. Similarly, on validation set the MSE value decreases from 0.01 to below 0.001. Both the cures are seen very near to each other that means error values are least at that time. It means the model got trained very well. The minor fluctuations are okay as validation loss is fluctuating while training loss is very stable. The second observation is the use of EarlyStopping criterion that makes the model stop further training after 63 iterations. This helps us in decreasing the training time and composition of the model helps in robust and more precise forecasting.

After training the model, now it's time for forecasting. So, here we are going to present the results of 1 day and 1-week graphs of Price forecasting. We train the model once and then on base of that training we predict the time series. The predictions of one day and one week are predicted at the same time, and we just draw graphs to display the accuracy of the proposed solution.

4.3.2.1 LSTM-BiGRU based Price Forecasting

4.3.2.1.1 One Day Price Forecast 2nd January 2019

The attached graph below is the graph for one day forecast for the second of January.

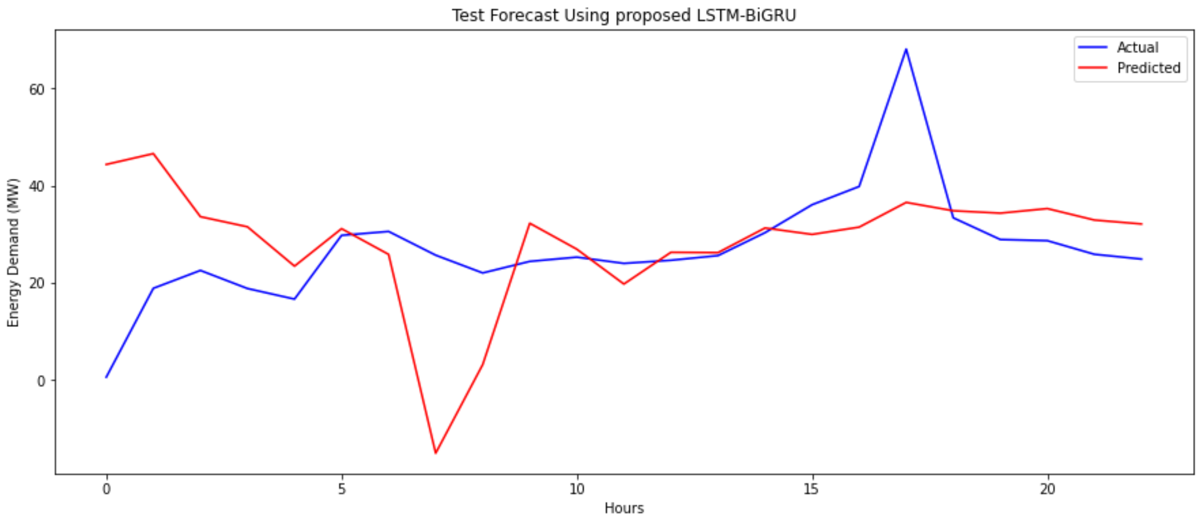

Figure 41: Price Forecasting for 2nd January 2019

It is clearly seen that the predicted values follow the pattern as in the actual data graph. The MAE error calculated is 11.23 and the deviations from the actual values in terms of RMSE is calculated 16.70.

4.3.2.1.2 One Week Price Forecast January 2019

The attached graph below is the graph for one week forecast for the month of January

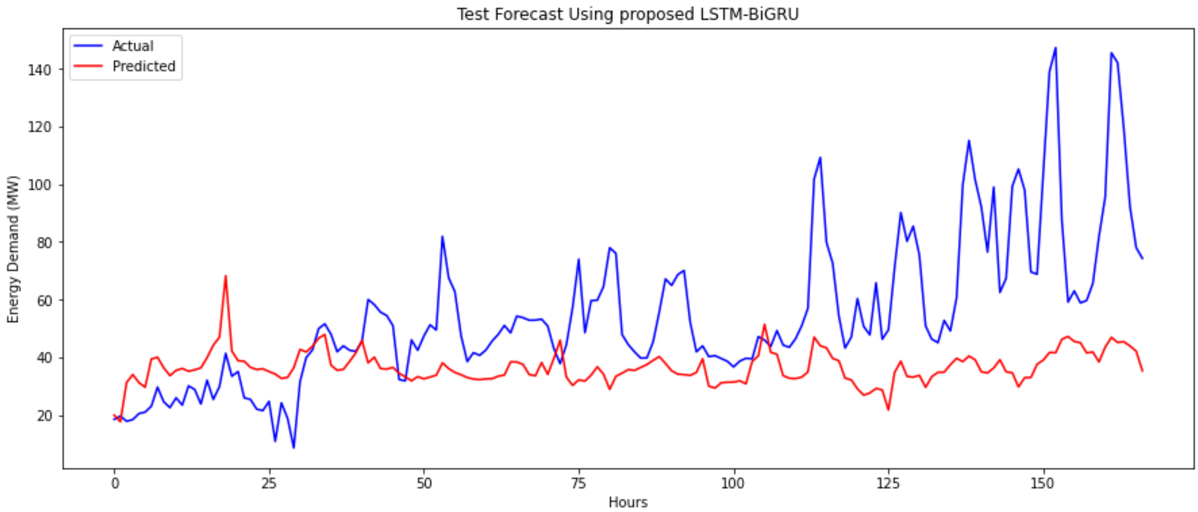

Figure 42: Price Forecasting for 1st Week January 2019

It is clearly seen that the predicted values follow the pattern as in the actual data graph. The MAE error calculated is 22.91 and the deviations from the actual values in terms of RMSE 30.79 calculated and attached in the graph below.

4.3.2.2. Comparison graphs

4.3.2.2.1 One Day Price Forecast January 2019

Consider the following graph, solid black graph representing the actual forecast values while green is representing the proposed LSTM-BiGRU model forecast.

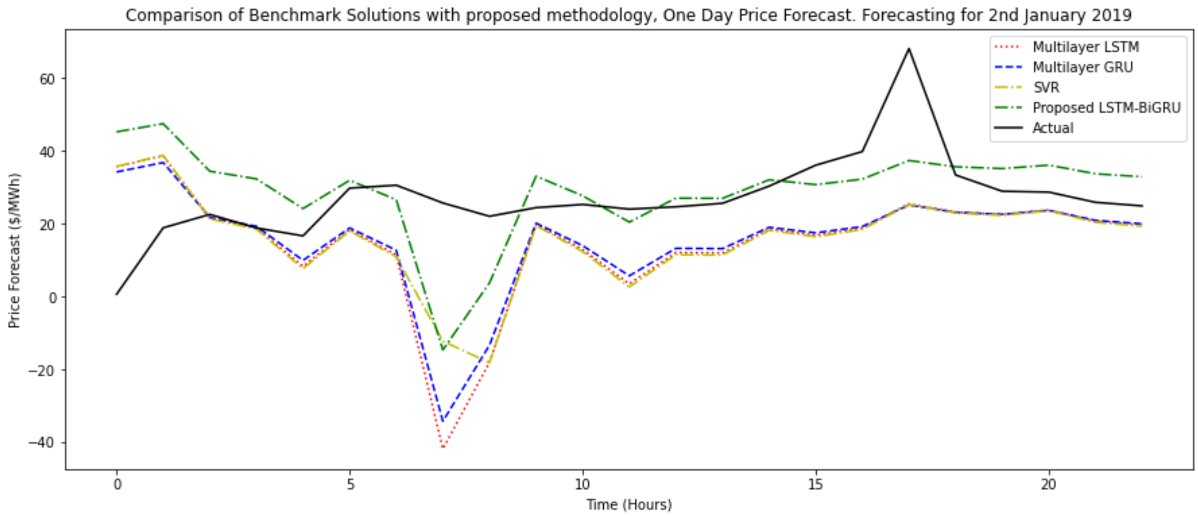

Figure 43: Comparison graph for Price Forecast 2nd January, 2019

Considering the above graph, it is clearly seen that the proposed solution performs better than the other benchmark solution with the decreased computational time.

The computational graph is attached at the end of the section. Please consider the below table for MAE error in the forecasting.

Table 21: MAE on 2nd January 2019 Price Forecasting

Hr	LSTM	GRU	SVR	LSTM-BiGRU
1-24	17.07 $/MWH	15.91 $/MWH	16.14 $/MWH	2.35 $/MWH

It is clearly seen that the MAE predicted in one day forecasting is only 2.35 that is the best result among the presented comparison result.

However, we have also calculated the RMSE of the whole 24hrs together to get a clear picture. Please consider the table below:

Table 22: RMSE on 2nd January 2019 Price Forecasting

Hr	LSTM	GRU	SVR	LSTM-BiGRU
1-24	23.12 $/MWH	21.38 $/MWH	20.2 $/MWH	16.85 $/MWH

It is obvious from the above table that RMSE calculated for BI-GRU at each hour is less than the comparison models. There may be some hours where RMSE of the proposed solution be higher than the comparison models. We consider it alright as we are comparing them with the best benchmark solutions available and our aim of writing this thesis is to make a fairly accurate rate between forecasting and computational time. On average or considering load forecasting for the day it is alright of the proposed solution missed accuracy in a very few hours. In rest of the cases, the accuracy of forecasting is very close to original values.

4.3.2.2.2 One Week Price Forecast January 2019

Consider the following graph, solid black graph representing the actual forecast values while green is representing the proposed LSTM-BiGRU model forecast.

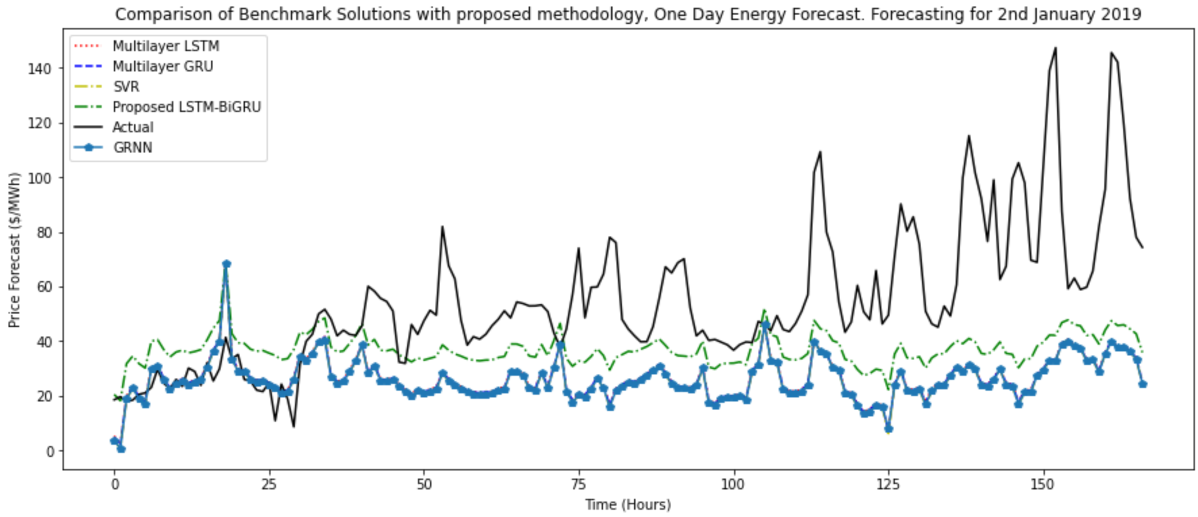

Figure 44: Comparison graph for Price Forecast 1st Week January 2019

Considering the above graph, it is clearly seen that the proposed solution performs better than the other benchmark solution with the decreased computational time.

The computational graph is attached at the end of the section. Please consider the below table for MAE error in the forecasting

Consider the MAE calculated in the above-mentioned graph.

Table 23: MAE on 1st Week January 2019 Price Forecasting

Hr	LSTM	GRU	SVR	LSTM-BiGRU
1-168	29.47$/MWH	29.58 $/MWH	30.04$/MWH	22.63 $/MWH

It is clearly seen that the MAE predicted in one day forecasting is only 22.63 that is the best result among the presented comparison result.

However, we have also calculated the RMSE of the whole 24hrs together to get a clear picture. Please consider the table below:

Table 24: RMSE on 1st Week January 2019 Price Forecasting

Hr	LSTM	GRU	SVR	LSTM-BiGRU
1-168	37.68 $/MWH	37.78 $/MWH	38.20 $/MWH	30.79 $/MWH

It is obvious from the above table that RMSE calculated for BI-GRU at each hour is less than the comparison models. There may be some hours where RMSE of the proposed solution be higher than the comparison models. We consider it alright as we are comparing them with the best benchmark solutions available and our aim of writing this thesis is to make a fairly accurate rate between forecasting and computational time. On average or considering load forecasting for the day it is alright of the proposed solution missed accuracy in a very few hours. In rest of the cases, the accuracy of forecasting is very close to original values.

As we have forecasted price for the month of January, similarly we forecasted energy for the month of July as well. And for forecasting energy for July, we train our model with time series of 3.5 years and then used that specific time series in forecasting, respectively.

4.3.2.2.3 One Day Price Forecast for July 2nd, 2019

The attached graph below is the graph for one day forecast for July 2nd. Here we are directly attaching the comparison graph below

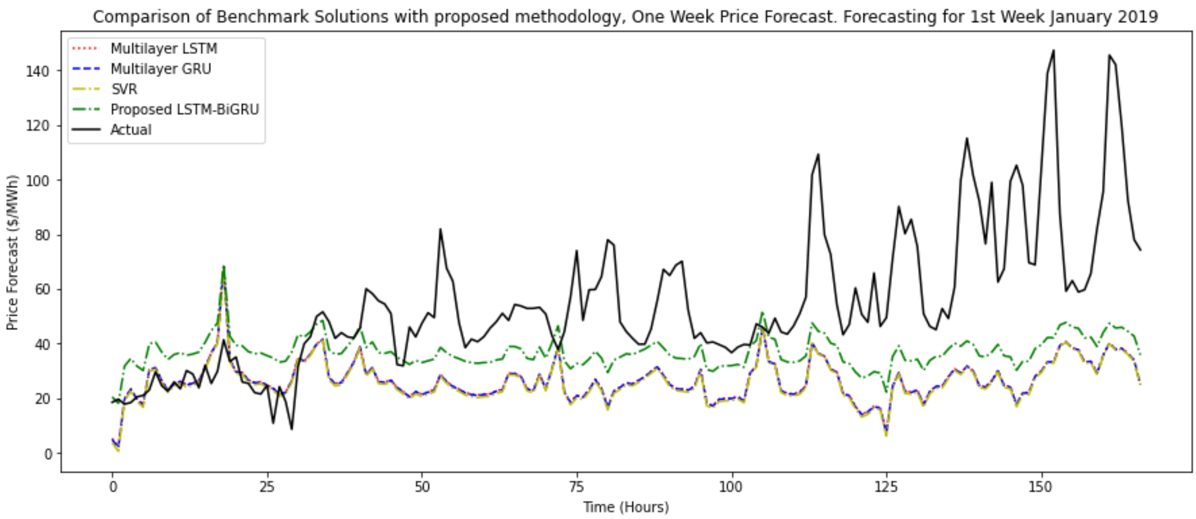

Figure 45: Comparison graph for Price Forecast 2nd July 2019

Considering the above graph, it is clearly seen that the proposed solution performs better than the other benchmark solution with the decreased computational time. SVR is showing the highest deviations in this scenario.

The computational graph is attached at the end of the section. Please consider the below table for MAE error in the forecasting

Consider the MAE calculated in the above-mentioned graph.

Table 25: MAE on 2nd July 2019 Price Forecasting

Hr	LSTM	GRU	SVR	LSTM-BiGRU
1-24	3.89 $/MWH	3.11 $/MWH	3.04 $/MWH	3.02 $/MWH

It is clearly seen that the MAE predicted in one day forecasting is only 3.02 that is the best result among the presented comparison result.

However, we have also calculated the RMSE of the whole 24hrs together to get a clear picture. Please consider the table below:

Table 26: RMSE on 2nd July 2019 Price Forecasting

Hr	LSTM	GRU	SVR	LSTM-BiGRU
1-24	4.55 $/MWH	3.84 $/MWH	4.06 $/MWH	3.02 $/MWH

It is obvious from the above table that RMSE calculated for BI-GRU at each hour is less than the comparison models. There may be some hours where RMSE of the proposed solution be higher than the comparison models. We consider it alright as we are comparing them with the best benchmark solutions available and our aim of writing this thesis is to make a fairly accurate rate between forecasting and computational time. On average or considering load forecasting for the day it is alright of the proposed solution missed accuracy in a very few hours. In rest of the cases, the accuracy of forecasting is very close to original values.

4.3.2.2.4 One Week Load Forecasting, July 2019

The attached graph below is the graph for one week forecast. Here we are directly attaching the comparison graph below

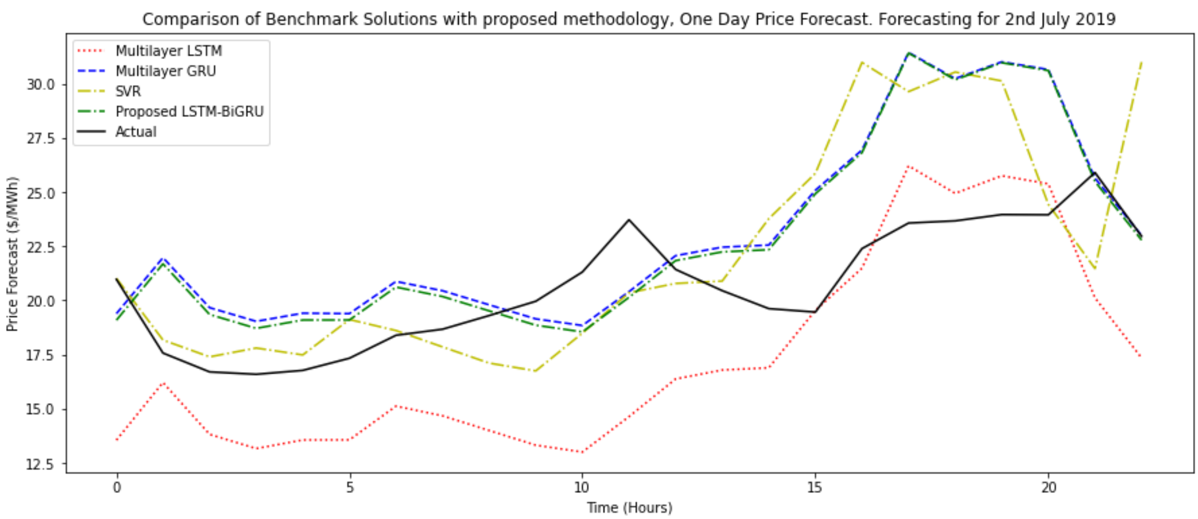

Figure 46: Comparison graph for Price Forecast 1st Week July 2019

Considering the above graph, it is clearly seen that the proposed solution performs better than the other benchmark solution with the decreased computational time. SVR is showing the highest deviations in this scenario.

The computational graph is attached at the end of the section. Please consider the below table for MAE error in the forecasting

Consider the MAE calculated in the above-mentioned graph.

Table 27: MAE on 1st Week of July 2019 Price Forecasting

Hr	**LSTM**	**GRU**	**SVR**	**LSTM-BiGRU**
1-168	11.14 $/MWH	10.13 $/MWH	10.22 $/MWH	9.60 $/MWH

It is clearly seen that the MAE predicted in one day forecasting is only 9.60 that is the best result among the presented comparison result.

For considering the deviation at hourly level, we have calculated RMSE, please consider the following table.

However, we have also calculated the RMSE of the whole 24hrs together to get a clear picture. Please consider the table below:

Table 28: RMSE on 1st Week of July 2019 Price Forecasting

Hr	**LSTM**	**GRU**	**SVR**	**LSTM-BiGRU**
1-168	17.54 $/MWH	16.04 $/MWH	16.12 $/MWH	15.40 $/MWH

It is obvious from the above table that RMSE calculated for BI-GRU at each hour is less than the comparison models. There may be some hours where RMSE of the proposed solution be higher than the comparison models. We consider it alright as we are comparing them with the best benchmark solutions available and our aim of writing this thesis is to make a fairly accurate rate between forecasting and computational time. On average or considering load forecasting for the day it is alright of the proposed solution missed accuracy in a very few hours. In rest of the cases, the accuracy of forecasting is very close to original values.

4.3.2.2.5 Yearly Price Forecast 2019

We have computed the load forecast for the year 2019, that is as follows:

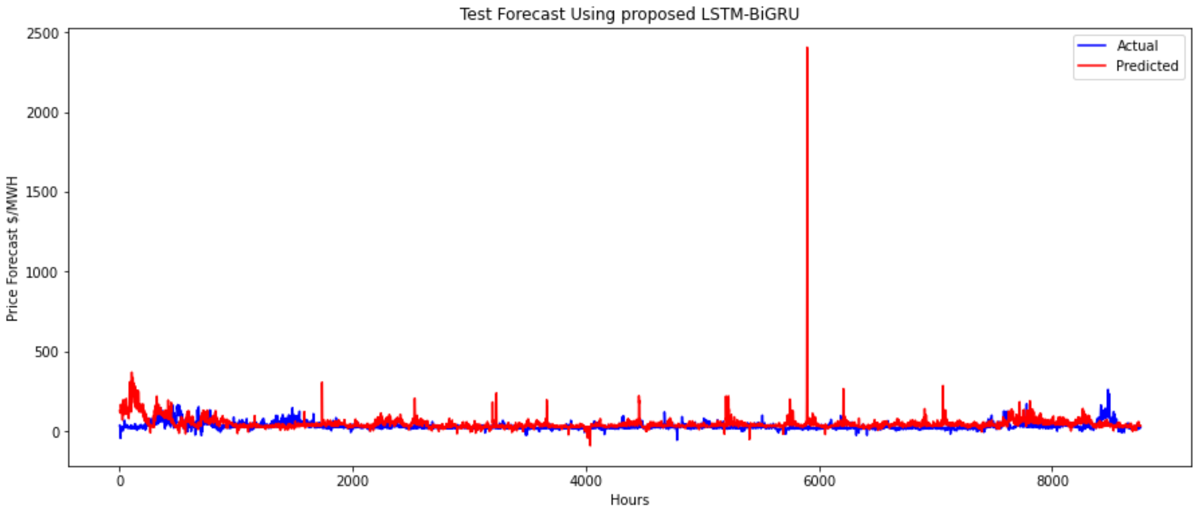

Figure 47: Yearly Price Forecast for year 2019

The scope of this manuscript is for short term price forecasting, we have computed the whole year just to get an idea of overall efficiency of the proposed solution. The error metrices are as follows:

Hr	MAE	RMSE	Execution time
1-8760	24.85	47.56	38.55 (sec)

4.3.2.3 Computational Time for January 2019 Price Forecast

Here we are summarizing the computational time for one day forecast and one week forecast together. We are doing this together because our model is trained once and hence it predict forecast value of one day or week in almost same time. The computational Time here is inclusive of feature section (9.73 sec). Consider the table below:

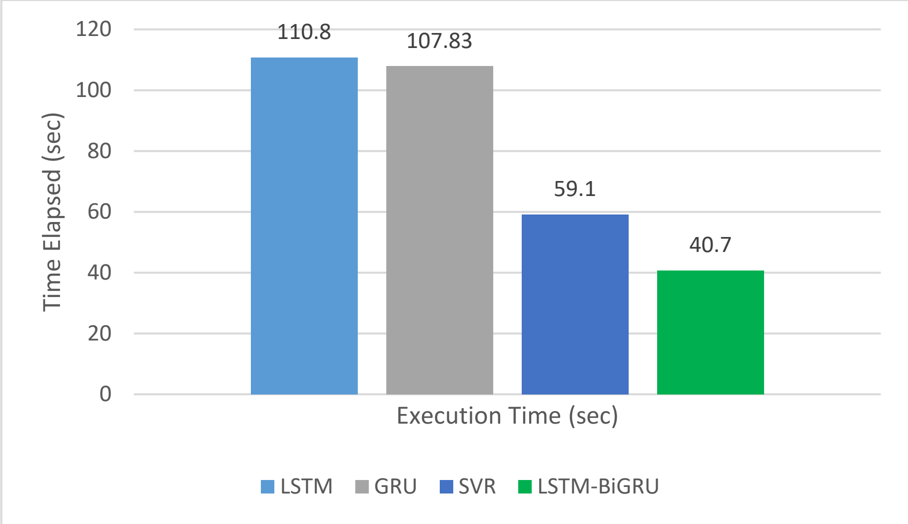

Figure 48: Model Computational Time for Price Forecasting January 2019

The calculated execution time is in seconds. It is clear from the above-mentioned table the execution time of the proposed solution is almost half of the other proposed solutions. It means that the proposed solution is providing the best forecasting results in the least possible time, hence accomplishing the objective of this research.

4.3.2.4 Computational Time for July 2019 Price Forecast

Here we are summarizing the computational time for one day forecast and one week forecast together. We are doing this together because our model is trained once and hence it predict forecast value of one day or week in almost same time. The computational Time here is inclusive of feature section (9.73 sec). Consider the table below:

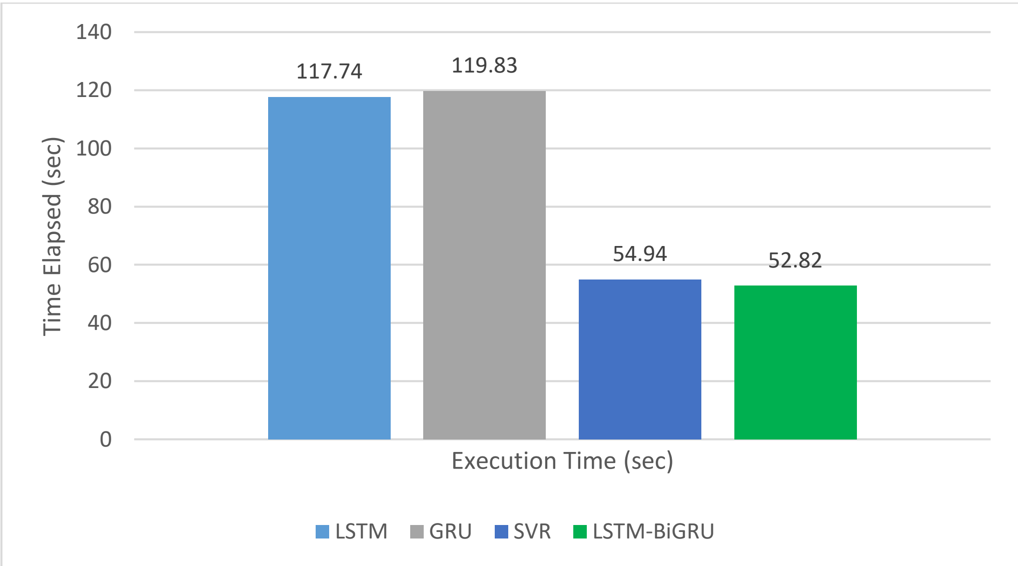

Figure 49: Model Computational Time for Price Forecasting July 2019

The calculated execution time is in seconds. It is clear from the above-mentioned table the execution time of the proposed solution is almost half of the other proposed solutions. It means that the proposed solution is providing the best forecasting results in the least possible time, hence accomplishing the objective of this research.

So, here we have presented the results of the proposed solution, and it is clearly seen that the proposed solution performed better in terms of forecasting accuracy and convergence rate and hence we successfully concluded our manuscript. The conclusion and the improvements in the proposed model are explained in the next section.

Chapter 5
Conclusion and Future Work

5.1 Conclusion

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.

5.2 Future Work

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. Ahmad, 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.