Short-Term Forecasting in Smart Electric Grid using N-BEATS

Neelesh Pratap Singh
Department of Electrical Engineering
NIT Warangal
Warangal, India
npee20303@student.nitw.ac.in

Aniket Ramendrakumar Joshi
Department of Electrical Engineering
NIT Warangal
Warangal, India
aree20310@student.nitw.ac.in

Mahamad Nabab Alam
Department of Electrical Engineering
NIT Warangal
Warangal, India
mnalam@nitw.ac.in

2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T) | 978-1-6654-5858-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICPC2T53885.2022.9776757

***Abstract*—In recent years, the idea of a smart grid is being projected in real life. In various countries which constitutes the main idea of deregulation which comes with the conversion of the consumer as a prosumer which affects electricity prices, demand, and power required. In this article Neural Basis Expansion Analysis for Interpretable Time Series Forecasting (N-BEATS) algorithm is used for forecasting of uni-variate data with data-preprocessing stages and multivariate with feature engineering stage, also various other benchmark methods are implemented using python for more flexibility to know robustness of proposed method on the particular case study for power system which is Ontario demand, hourly electricity price, wind speed in Ontario to have precise forecasting which helps in various tasks like demand response to conventional source management especially by detecting sharp spikes in data.**

***Index Terms*—N-BEATS, Deep Learning, Forecasting.**

NOMENCLATURE

Parameters

ϵ	Margin of tolerance where no penalty is given to errors
C	Regularization parameter
d	Differencing
f	number of filters
H	Forecast horizon (H=1, for this case study)
n_o	Total number of outputs
p	Auto-Regression (AR) model lags
q	Moving average lags
s	Strends (used while convolution)
t	Time vector

Variables

$\hat{y}_g$	Forecasted output for a example in sample g
$\hat{y}_{g_{mean}}$	Mean of all forecasted examples of sample g
$\hat{y}_{s,l}$	Forecast result of stack s of layer l
$\theta^f_{s,l}$	Polynomial coefficient predicted by Fully Connected (FC) network of layer l of stack s
y_g	Actual output for a example from sample g

I. INTRODUCTION

Artificial Intelligence (AI) becoming a new normal in every field and domain are known to us as it's a deciding factor for the next upcoming decade and emerging future in the electrical engineering domain.While explaining smart grid it somewhere consists of various applications of AI. As simple as intelligent devices for making products like smart meter and ambient assisted Learner. One of the important applications of AI in electrical engineering is forecasting which is moreover useful for managing the grid in a smart way. There can be several types of forecasting based on time scale either be very short (minute to daily), short term (hourly to weekly), medium term (weekly to year), long term (>1 year). In this article, short term forecasting is discussed where accuracy is the main concern as it is done for the short notice of time so must be accurate to take decisions according to that. Short term is beneficial for the precise dynamic scheduling like charging of electric vehicle when price is low, managing conventional sources of power according to the generation of renewable sources (to have maximum use of Renewable sources while having minimum use of conventional sources) using the forecast demand and weather forecast (like wind speed or Solar irradiance and temperature forecasting) this all shows short term forecasting helps to have optimal management of modern power system.

In previous studies, statistical techniques are used for short term forecasting of time-based series but due to their presumed structure for analysis for any type of data, which induces deficiency in the model while forecasting. Moreover statistical methods e.g Auto-Regressive Integrated Moving Average (ARIMA) very much useful for short term forecasting but not able to detect sharp spikes accurately, [1] as it is taking moving average which averages out the peak. To overcome this problem various hybrid model with ARIMA is proposed but they somewhere lacks in accuracy while comparing to Machine Learning models followed by deep learning models.

Data engineering helps proper understanding of data by analyzing its features and so with a machine learning algorithm, forecasting becomes easier and able to cope up with highly fluctuated data. Primary machine learning methods are Support Vector Machine (SVM) which is a powerful machine for both classification and regression [2], for forecasting Support Vector Regressor (SVR) is used which is able to maps data in new space separated by a soft margin by use of support vectors. Which clearly shows its robustness, but due to high computational burden and lack of memory unit, it does not work well with large data and accuracy diminishes with high dimensional data. As most of the problems in real-world are non-linear in nature (non-linear relation between features and target) as for forecasting problems need to have a deep understanding of High dimensional data so to analyze the complex pattern of non-linear problems for this understanding of deep learning algorithms plays a significant role in forecasting.

In [3], modified form of Convolutional Neural Network (CNN) is introduced where morlet wavelet function is used as an activation function in place of traditional activation functions (basic work of activation function is to extract complex non-linear behaviour of data) but due to lack of memory unit and gradient vanishing problem it gives less accuracy for a forecasting problem. Need of working with big data and to learn the patterns of non-linear complex problems the more deeper network is required along the feature of memory unit. The new version of deep neural network is Recurrent Neural Network (RNN) [4] which acts as a memory unit with the use of weights but due to the problem of gradient vanishing RNN is not that much efficient, due to these facts various sequence to sequence models like Long Short Term Memory (LSTM) [5] is proposed to solve the problem of gradient vanishing.

In [6], residential load forecasting is done with the use of LSTM which acts as a memory unit with the predefined operation of gates so each layer acts as a memory unit instead of hidden units (which previously only processes the data not memorizing the features in it for further use), this all shows that LSTM solves the problem of gradient vanishing but due to error accumulation (as the output of the previous unit fed to next) and due to its unidirectional memory temporal features LSTM alone are not used efficiently. Machine Learning (ML) algorithms are stochastic in behaviour because it can be possible that easier algorithm will take over complex algorithms in terms of accuracy and obviously impose less calculation burden on the machine. As in [7] a study of comparison between SVR and LSTM is done which shows that SVR is good for stationary time series and gives promising results in uni-variate data and on the other hand LSTM perfectly detects the peak and reduces Root Mean Square Error (RMSE) [8] and gives more accurate results. Although, it is also mentioned that algorithms result is dependent on the case study on which analysis is done.

In [9], Micro-Clustering (MC) is done using k-means. This clustered data is fed to Bidirectional Long Short Term Memory (B-LSTM:advanced version of LSTM in which future features are being considered which gives the ability to analyse whole temporal behaviour) to have forecasting of one day ahead but due to the more weightage given to MC module e.g if MC not done properly results are worse than the basic ML algorithms. This makes it more complex way of analyzing forecasting problem and can lead to higher inaccuracy. In the proposed method, Neural Basis Expansion Analysis for Interpretable Time Series Forecasting (N-BEATS) [10] is used for analysing the uni-variate data after some data pre-processing and also employing it for multivariate data to compare accuracy for both methods.

In the proposed method, N-BEATS [10] is used for analyzing the uni-variate data but in the proposed method instead of feeding the algorithm directly, data is fed after some data pre-processing because N-BEATS is not employed for electricity demand, wind speed, or electricity price all at the same place till now and also employing it for multivariate data (using feature engineering) also compares accuracy for both methods. All the methods are programmed using python instead of Application Programming Interface (API) because basic implementations through python have more flexibility for variations.

The main structure of the paper is defined as follows. In section II, Data pre-processing describes the body for the proposed method which consists of the data pre-processing module & training and forecasting module. In section III data description and comparison with different methods of forecasting is done for a case study are explained. In section IV numerical results for the proposed algorithm are discussed for both uni-variate and multivariate analysis. Section V contains the conclusion and outcomes of the studied algorithm along with the future scope for analyzing the uni-variate forecasting problem.

II. PROPOSED METHODOLOGY

The proposed method is defined based on two modules: the first is the data pre-processing module and the second one is the training & forecasting Module, in this section these modules are discussed in full detail.

A. Data Pre-Processing Module

As the very first stage of every AI technique is data filtering and processing making it able to learn with algorithm because there is a difference between learning and memorizing things e.g learning means getting feature-rich insights from data so can work well on diversified test data and in memorizing it only trains for lower training error due to which can't able to perform better on test data. Extracting data, outlier analysis for time-series using methods like interpolation, removing or adding an extra column (feature) according to requirement e.g without doing uni-variate (single column time series data) or multivariate analysis (multiple columns of features and target ready data). Making index as date-time makes work easier to analyze time series problem, and plotting it to analyze seasonality, using null hypothesis testing for stationarity test. For uni-variate directly split it (to supervised split) and for multivariate including extra features (cos or sin) according to seasonality for a time period and training the single model with all time series so lot of data was there to extract pattern. Then batching and pre-fetching ensure loading of data from Central Processing Unit (CPU) (who prepare data) to Graphics Processing Unit (GPU) (done computation on data) so process time required must be small.

B. Training & Forecasting

This article discusses the modelling of pipeline from training to forecasting, as in RNN current state (input) & internal update considered to act as memory, similarly in LSTM gate structures in such a way to act as a memory unit. Whereas N-BEATS possesses a certain different approach which takes the whole window in one go and combines many forecasts in a single iteration with the use of fully connected layers (layers that are connected with weights to learn). It takes the idea of the residual block in Fig. 1 where its blocks are connected in a residual way (idea of implementing skip connection to know the significance of intermediate layers e.g useful or not).

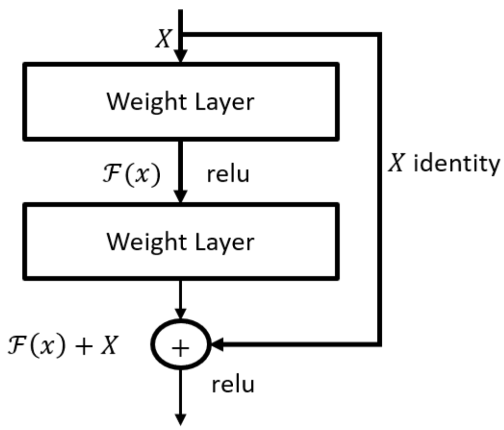


Fig. 1: Concept of residual block.

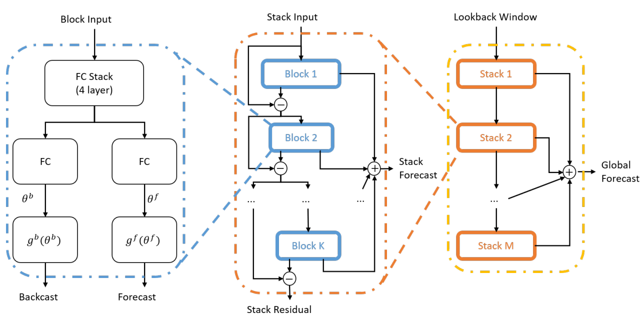


Fig. 2: N-BEATS architecture.

Here in figure 2, basic building block consists of multiple layers of FC network with Rectified Linear Unit (ReLU) as activation to predict basis expansion coefficients θ^f (forward) and θ^b (backward), blocks are organised in fashion of doubly residual stacking concept, these coefficients are fed to basis layers e.g g^f and g^b respectively to project them and produce the backcast ($\hat{x}_l$) and forecast ($\hat{y}_l$) respectively.

Block I gives the backcast which is modelling of a past window and forecast output which is modelling of future values. Where block II tries to model residual error happening while backcast & forecast (reconstruction of time series) then this error work as input to the next block to further update forecast according to this error. Using the residual block architecture problem of gradient vanishing is solved even for a larger lookback window shown by results later in this paper. It decomposes trend & seasonality e.g making the output more interpretable while keeping accuracy good by inducting several biases.

Trend modelling can be considered as an monotonic function or slowly varying function so in order to model its behaviour an accumulation function shows trend model shown by equation 1

$$\hat{y}_{s,l} = \sum_{i=0}^{p} \theta^f_{s,l,i} t^i \tag{1}$$

and seasonality modeling which can be cyclical, regular, or recurring fluctuations by using the fourier series as it can be modeled as periodic function described by equation 2

$$\hat{y}_{s,l} = \sum_{i=0}^{[H/2-1]} \theta^f_{s,l,i} cos(2\pi it) + \theta^f_{s,l,i+[H/2]} sin(2\pi it) \tag{2}$$

It regularizes the results by inherent ensembling because it takes average prediction (forecasting results) from various sub-blocks. The first block tries to capture the main trend whereas the second block specializes in smaller errors. The algorithm (N-BEATS) which is used for analysing the proposed case study works well is because of the meta-learning concept which explains that to decompose learning process into an inner and outer training loop, where the inner training loop focuses on task-specific knowledge (where reconstruction updated and outer training loop focus on an across-task knowledge entire network parameter updated). Faster training on GPU, simple architecture, and lightweight network which is not externally dependent along with that do various tasks on its own.

C. Overall Structure

In figure 3, after data-preprocessing data is used to train the model using the predefined values of hyper-parameter in Table I and after training using test data to forecast the data according to lookback data, if finding the single step (next 24 hours in single prediction) take forecast period directly as required but if doing multi-step (take prediction of the previous prediction to act as new test data until prediction required).

The training of the model is done using the Jupyter Notebook platform (python3) with force to run on the local system with 16 GB RAM AMD Ryzen 7, CPU with 2.9 GHz clock frequency, 6 GB of GPU with a training time of 8 hours. Python libraries used are sklearn for ML and Keras with Tensor-flow for Deep learning (the Deep learning part can also be done using PyTorch library).

III. CASE STUDY

A. Data Description

Data is taken from the Canadian website for Ontario province [11] [12] for three years 1st Jan 2016 to 31st Dec 2018 which further splits into training and testing data but in a sequential manner not in a random manner otherwise problem of future leak to past comes into existence. The first one is the electricity price of Ontario which is very highly dynamic,

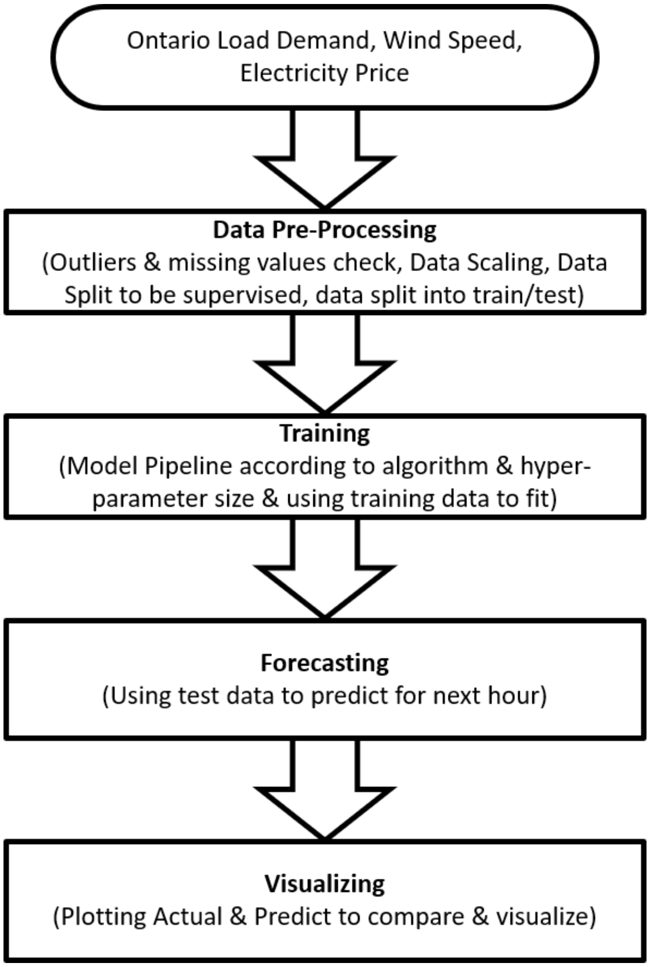


Fig. 3: Overall flowchart for proposed method.

second is wind speed because Ontario is having a high number of wind farms & load demand. Each data-set is split into 80% (training) & 20% (testing) sequentially.

B. Previous Benchmark Methods

To present the robustness of the proposed method the brief comparison with other standard forecasting techniques in the same environment as of proposed method. In this comparison ARIMA (is an statistical method for time series forecasting),CNN (Image processing algorithm so for short term forecasting by converting data into matrix form),SVM (an powerful machine with high ability of feature extraction map points in high dimensional space with respect to different kernels e.g Linear, Gaussian etc) kernel selection also effects results and provides more accurate results due to presence of soft margin formed by state vectors, Random Forest (RF) [13] (RF is an ML algorithm working on concept of decision tree in ensemble manner to implement an idea of decision at every node of splitting of features to solve non-linear problems), LSTM (Used for time series analysis like audio, speech etc work as an memory unit which makes it more accurate for time series data), MC-BLSTM (MC using k-means with silhouette index for optimal clusters then training BLSTM increases accuracy only when clustering done perfectly otherwise gives worse results).

IV. Results

Firstly different error criteria metrics are discussed so to get an accurate idea of analysis while comparing the proposed method with other benchmark methods

A. Error Metrics in Forecasting

Three metrics are considered in this study to mean absolute percentage error [14] (MAPE – shows how the prediction is different from its actual taking average), mean absolute error [15] (MAE – Average error produced), RMSE (how largely the residuals are spread out in space) [16] are as follows:

$$MAE = \frac{1}{n_o}\sum_{g=1}^{n_o}|\hat{Y}_g - Y_g| \tag{3}$$

$$MAPE = \frac{1}{n_o}\sum_{g=1}^{n_o}\left|\frac{\hat{Y}_g - Y_g}{\hat{Y}_{g_{mean}}}\right| \tag{4}$$

$$RMSE = \sqrt{\frac{1}{n_o}\sum_{g=1}^{n_o}|\hat{Y}_g - Y_g|^2} \tag{5}$$

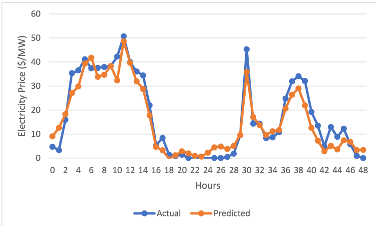
B. Forecasting Results

The first step in a case study is to do data pre-processing which is used to have data filtering (outlier and missing value check and also data scaling) then to split that data into time lag nature e.g making feature and target to make that uni-variate data usable for model learning and forecasting. After that several algorithms are applied to check their accuracy before discussing the N-BEATS which is a state of art technology algorithm that gives much better results with a simpler structure and fast speed. Table I shows the hyper-parameters used after several attempts with respect to the taken case study because they can vary according to the case study to be analyzed. Below shown Table II compares different algorithms based on evaluation matrices MAE, MAPE, RMSE from equation 3,4 & 5 respectively. Actually with sufficient data deep learning algorithms performs better because it is able to extract deep features which are not in the case of RF, SVM & ARIMA however if data is limited RF performs better than the other deep learning algorithms.

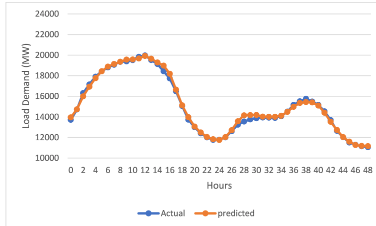
TABLE I: Hyper-parameter Selection for the Model's Pipeline

Algorithm	Window Size	Hyper-Parameter
ARIMA	1	P,d,q = (1,1,1)
SVM	48	Linear Kernel, C = 0.1, ϵ = 0.2
RF	48	Default
ANN	72	3 layers, 50 neurons, 20 epochs
CNN	164	Functional API, CONV1D (f=64, size=2), Max Pool (s=2) Dropout = 0.25, Dense
LSTM	48	3 layers (2 LSTM & 1 Dense), 20 epochs
MC-LSTM	1	15 layers, Rate = 0.01-0.005, dropout = 0.5, 500 iterations
N-BEATS (Multi)	72	5000 epochs, 512 neurons, 4 layers & 30 Stacks
N-BEATS (Uni)	72	5000 epochs, 512 neurons, 4 layers & 30 Stacks

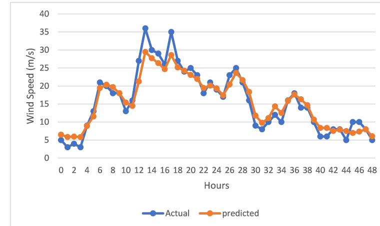
The load demand and wind speed accuracy is bit much higher than the electricity price (highly dynamic in nature)



(a) Electricity price

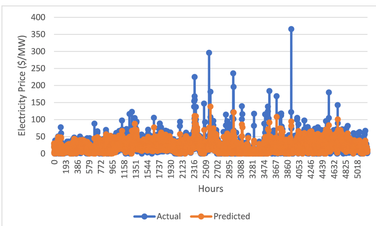


(b) Load demand

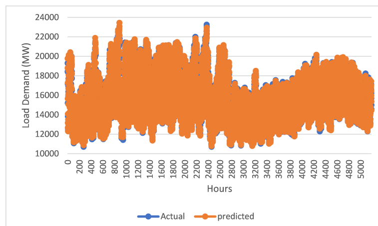


(c) Wind speed

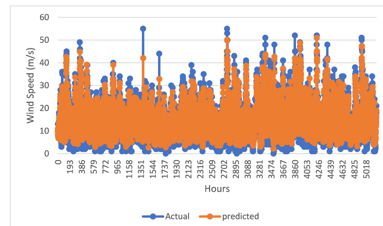
Fig. 4: Forecasting results of 48 hours.



(a) Electricity price

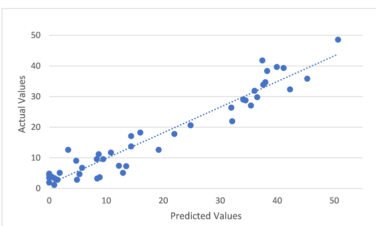


(b) Load demand

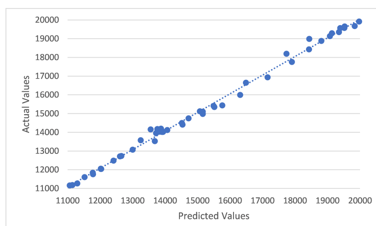


(c) Wind speed

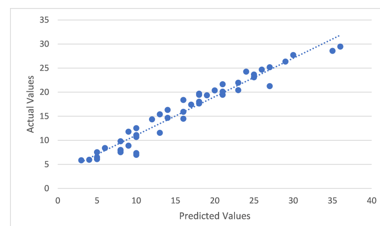
Fig. 5: Forecasting result for test dataset of 7 months.



(a) Electricity price (R^2 : 0.9141)



(b) Load demand (R^2 : 0.9948)



(c) Wind speed (R^2 : 0.9440)

Fig. 6: Regression plots of 48 hours.

TABLE II: Forecasting Result

Algorithm	Wind Speed MAPE (%)	Wind Speed MAE (m/s)	Wind Speed RMSE (m/s)	Price MAPE (%)	Price MAE (m/s)	Price RMSE (m/s)	Demand MAPE (%)	Demand MAE (m/s)	Demand RMSE (m/s)
ARIMA	42.43	8.59	9.8	30.18	8.21	12.37	7.01	1024.15	1201.03
SVM	38.63	2.75	3.50	36.21	8.56	14.13	2.62	240.88	322.7
RF	34.38	3.17	3.90	31.66	7.08	10.43	4.27	640.92	744.6
ANN	25.98	4.59	5.81	37.41	14.43	11.23	7.27	481.16	942.4
CNN	29.92	4.2	5.18	24.18	8.74	11.409	9.92	402.57	773.4
LSTM	18.11	5.11	8.42	16.42	7.81	11.42	1.84	449.2	596.4
MC-BLSTM	13.14	2.17	3.08	18.81	4.46	6.47	1.108	180	247.14
N-BEATS (Multi)	12.14	1.62	2.32	19.02	4.801	6.78	0.97	196.04	242.18
N-BEATS (Uni)	11.93	1.40	2.12	16.6	3.18	4.46	0.92	142.79	199.68

but overall accuracies are better in comparison with previous benchmark methods as shown in Table II, even with the MC-BLSTM model which is very complex to build upon and the chances of getting the wrong model is more. Below figures shows prediction and actual plot for complete test set which shows getting much accurate results in load demand and wind speed (ignoring some outliers which are somewhere stochastic in nature comprising of 0.9 % of the total test set). Also, uni-variate data converted to multivariate by taking the seasonal behavior into consideration e.g sin and cos features taken for a span of 24 hours, and also those results are compared. In fig. 4 actual and predicted results are plotted for the span of 48 hours with high stochastic behavior to visualize accuracy of the proposed method for the N-BEATS algorithm, along with that complete actual and predicted results for 7 months is shown in fig.5 to analyze results over a large time-frame to have an idea over a change in seasonality and trend for several months.

To verify the robustness of the case study in another way, as forecasting is analogous to regression so R^2 which is a well-known criterion that shows the correlation between forecasted and actual data [17] is also taken into consideration for this study as shown in Fig. 6, high R-squared indicates more accurate answer. Multivariate (features induced in data engineering) also gives accurate results but somewhere near to uni-variate not much variation was there which is not significant for which the extensive feature engineering is worth, this proves that the N-BEATS algorithm with uni-variate data for a particular case study works well (because can't generalize an algorithm and its hyper-parameter for all case studies there is some variation always required to get much accurate result on a particular case study).

This article proposes all analysis on uni-variate data which makes work simpler of collecting the data as it doesn't require other features data, along with that exploratory data analysis for time series was also done which consists various tasks from outlier analysis to data scaling to data replacement (interpolation) concepts which enhances the performance of the model and provides better accuracy in comparison to multivariate analysis on the same data. The unexpected result got is that the same hyper-parameter works very well for all three data selected, probably due to correlation between them and opens up further study on hyper-parameters selection which is normally a tedious task and stochastic in nature.

V. Conclusion

In this article precise short term forecasting method for the Ontario case study based on the N-BEATS algorithm is implemented and its robustness and accuracy is verified by implementing benchmark algorithms and comparing their accuracy for the same case study and getting better accuracy with less complexity, faster and easy training to get around 97% accuracy along with detection of high fluctuation points (99.8% detection rate) with the use of sequence networks (act like memory element). N-BEATS gives much accurate results in comparison to the most recent work of forecasting which implements MC-BLSTM. In comparison to other benchmark methods, N-BEATS work very well on uni-variate data with easy architecture. It also proposes hyper-parameter for the model training and also explains the concept of taking the same hyper-parameters for the correlated data. This model itself extracts seasonality and trend so the only requirement is data engineering which depends on the type of data considered. With the integration of all these advancements, N-BEATS becomes a new benchmark for uni-variate data in comparison to recent benchmark algorithms. Precise estimation of power system situation motivates distribution system operator to make better decisions with fewer errors.

As forecasting of load demand, wind speed, electricity price is achieved with higher accuracy but due to very high dynamic behavior price accuracy can become more accurate so as future work, implementing the integration of various sequence to sequence networks to detect that much high fluctuation in price.

References

[1] J. Fattah, L. Ezzine, Z. Aman, H. El Moussami, and A. Lachhab, "Forecasting of demand using arima model," *International Journal of Engineering Business Management*, vol. 10, p. 1847979018808673, 2018.

[2] S. Huang, N. Cai, P. P. Pacheco, S. Narrandes, Y. Wang, and W. Xu, "Applications of support vector machine (svm) learning in cancer genomics," *Cancer genomics & proteomics*, vol. 15, no. 1, pp. 41–51, 2018.

[3] M. Rafiei, T. Niknam, J. Aghaei, M. Shafie-Khah, and J. P. Catalão, "Probabilistic load forecasting using an improved wavelet neural network trained by generalized extreme learning machine," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6961–6971, 2018.

[4] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of cnn and rnn for natural language processing," *arXiv preprint arXiv:1702.01923*, 2017.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on lstm recurrent neural network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, 2017.

[7] H. Bakir, G. Chniti, and H. Zaher, "E-commerce price forecasting using lstm neural networks," *International Journal of Machine Learning and Computing*, vol. 8, no. 2, pp. 169–174, 2018.

[8] S. Kameyama and K. Sugiura, "Estimating tree height and volume using unmanned aerial vehicle photography and sfm technology, with verification of result accuracy," *Drones*, vol. 4, no. 2, p. 19, 2020.

[9] H. Jahangir, H. Tayarani, S. S. Gougheri, M. A. Golkar, A. Ahmadian, and A. Elkamel, "Deep learning-based forecasting approach in smart grids with microclustering and bidirectional lstm network," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 9, pp. 8298–8309, 2020.

[10] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-beats: Neural basis expansion analysis for interpretable time series forecasting," *arXiv preprint arXiv:1905.10437*, 2019.

[11] [Online]. Available: https://www.ieso.ca/en/

[12] E. Canada and C. Change, "Government of canada," Nov 2021. [Online]. Available: https://climate.weather.gc.ca/historical_data/search_historic_data_e.html

[13] G. Dudek, "Short-term load forecasting using random forests," in *Intelligent Systems' 2014*. Springer, 2015, pp. 821–828.

[14] R. Beigaite, T. Krilavičius, and K. L. Man, "Electricity price forecasting for nord pool data," in *2018 International Conference on Platform Technology and Service (PlatCon)*. IEEE, 2018, pp. 1–6.

[15] T. Chai and R. R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)–arguments against avoiding rmse in the literature," *Geoscientific model development*, vol. 7, no. 3, pp. 1247–1250, 2014.

[16] A. Botchkarev, "Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology," *arXiv preprint arXiv:1809.03006*, 2018.

[17] H. Jahangir, H. Tayarani, S. Baghali, A. Ahmadian, A. Elkamel, M. A. Golkar, and M. Castilla, "A novel electricity price forecasting approach based on dimension reduction strategy and rough artificial neural networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2369–2381, 2019.