Downloaded from orbit.dtu.dk on: Aug 16, 2024

DTU Library

Comparison of the Effect of Regularization Techniques and Lookback Window Length on Deep Learning Models in Short Term Load Forecasting

Kahraman, Aysegül; Hou, Peng; Yang, Guangya; Yang, Zhile

Published in:
Proceedings of 2021 International Top-Level Forum on Engineering Science and Technology Development Strategy

Link to article, DOI:
10.1007/978-981-16-7156-2_45

Publication date:
2022

Document Version
Early version, also known as pre-print

Link back to DTU Orbit

Citation (APA):
Kahraman, A., Hou, P., Yang, G., & Yang, Z. (2022). Comparison of the Effect of Regularization Techniques and Lookback Window Length on Deep Learning Models in Short Term Load Forecasting. In *Proceedings of 2021 International Top-Level Forum on Engineering Science and Technology Development Strategy* (pp. 655-669). Springer. https://doi.org/10.1007/978-981-16-7156-2_45

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Comparison of the Effect of Regularization Techniques and Lookback Window Length on Deep Learning Models in Short Term Load Forecasting

Aysegul Kahraman[1, 2][0000-0001-7673-7218], Peng Hou[3][0000-0002-5837-8431], Guangya Yang[1][0000-0003-4695-6705] and Zhile Yang[4][0000-0001-8580-534X]

[1] The Technical University of Denmark (DTU), Department of Electrical Engineering, 2800 Lyngby, Denmark
[2] Sino-Danish College (SDC), University of Chinese Academy of Sciences
[3] SEWPG European Innovation Center, Brendstrupgårdsvej 13, 8200 Aarhus, Denmark
[4] Shenzhen Institute of Advanced Integration Technology, Chinese Academy of Science, Shenzhen, China

Abstract. Management of electric power balance requires accurate forecasting of load and generation, especially in the context of renewable energy adoption. In this context, forecasting electric load requires more attention to decrease the uncertainties in the system operation. There have been many studies under this context, however, the effect of the lookback window for both deep learning and regularization techniques has not been fully investigated in the literature. In this study, we developed a comparative study based on 4 typical deep learning techniques, namely MLP, 1D-CNN, LSTM, and a hybrid model that is a combination of 1D-CNN and LSTM to forecast the electrical load. The effect of both regularization methods and lookback window length has been investigated in detail and found that they improved the forecasting performance based on the complexity and features of the networks. The methods are evaluated in terms of 4 different metrics namely MSE, MAE, MAPE and, R^2. The results show LSTM outperformed the other methods in general, and the increase of lookback length improved its performance with the average MAPE less than 2%.

Keywords: Deep Learning, Load Forecasting, Regularization, Lookback window.

1 Introduction

Energy consumption has been increasing in the last decades while the trend will continue in the forthcoming period, especially in the light of electrification of the transportation sector. In accompany to this is the increasing adoption of renewable energy sources such as solar and wind [1]. The variability and uncertainty in the generation and consumption challenge the security of the power supply. It is therefore necessary to develop accurate forecasting models on these quantities based on the relevant horizon to decrease the planning and operational costs and to ensure an efficient and reliable

power system operation [2]. Limited information about the data characteristics, dynamic frame, seasonality, and weather condition dependencies are the main factors of deficiency of energy management. Among these, weather dependencies are the major factors in the uncertainty of the data set [3]. These factors underpin the challenges of demand forecasting to enhance the power system operation. The temporal correlation within the data contains valuable information which might yield better forecasting performance. The decision of past data length which is expressed as lookback window length is another important decision to take.

Load forecasting is a popular topic in recent years since accurate forecasting is crucially desired for power system planning and operation. The most used methods are based on conventional statistical approaches such as Auto Regressive Moving Average (ARIMA) and linear regression. These methods are fast and easy to implement if there is a linear relationship between data points. However, these methods are not efficient when dealing with non-linear and more complex relationships in the data sets [4]. Therefore, Artificial Intelligence (AI) techniques, namely Artificial Neural Networks (ANNs), Support Vector Regression (SVR), and Fuzzy Logic (FL), have been used in literature. AI methods give relatively better performance since they can capture nonlinear and more complex relations [4]. Among these forecasting techniques implemented, namely ARIMA and one of the most common techniques in the DL field which is Long Short-Term Memory (LSTM) are applied and compared for both one-step and multi-step ahead load forecasting based on Open Power System Data set. It is identified that LSTM outperformed in both cases due to its high performance in time series forecasting [5].

Multi-Layer Perceptron (MLP), LSTM, and Convolutional Neural Networks (CNN) are commonly used DL-based ANNs in load prediction [6]. The paper demonstrated that deep learning neural networks (DNN) are superior to traditional methods namely Moving Average (MA), Linear Regression (LR), Regression Tree (RT), SVR and MLP for short-term load forecasting. By representing weekdays and weekends forecasting performance separately, the paper shows that DNN is more capable of predicting these particular daily characteristics [7]. The last category for forecasting is hybrid methods which combine various methods from different areas and gather their advantages for more accurate results [8].

However, by increasing the number of layers and complexity of ANN, it can capture hidden information in the data set but can also cause overfitting problems [9]. The general solution is to implement one of the regularization methods to negate this problem by making slight changes to the learning algorithm such as early stopping and dropout [9]. Zhu et al. illustrated that regularization implementation boosts air pollutant forecasting performance [10]. The major difficulty is specifying model features (inputs) which can change in a wide range for load forecasting. However, in general, the electric load data that is recorded in the past is used for almost every single technique as these consist series of knowledge relating to output. The significant difference in the way that is handled of recorded load data in the feature set brings various effects to the forecasting performance. Also, if there are some weather data like temperature, wind speed, humidity and some other extracted data such as holiday, time, and month can be used [11]. In [12], the authors identified that “DropBlock” is a structured form of dropout in

convolutional networks and proved to have better accuracy and was more robust to hyper parameter selection than both dropout and without applying regularization. In this study, we focus on 1) the selection of deep learning network, which affects the overall electrical load forecasting performance greatly, 2) in addition to previous networks, the performance of the hybrid network by combining 1D-CNN and LSTM, 3) evaluation of different lookback window for each network by keeping all other hyper parameters same, and 4) application of various regularization methods to assess the effects on different deployed ANN models. The main contribution here is exploring a comprehensive observation by changing the length of the lookback window which is using parts of recorded load data as a feature for MLP, 1D-CNN, LSTM, and hybrid networks in addition to five regularization methods. The results show that adjusting the lookback window can tackle without increasing the network complexity.

2 Proposed Methods

In this section, we examine different types of DNNs under several network designs. One of the noticeable issues is selecting the DL method and deciding its hyper parameters. The network architecture is crucially effective for the performance of the network. Many DL structures may work properly but they may also present poor performance due to incorrect parameter selection or insufficient data set. In these situations, overfitting is a quite common case in DL literature. Thus, we also employ particular regularization techniques and examine the change of lookback window to check the structure and improve their working performance.

2.1 Deep Learning Techniques

MLP is one of the most well-known ANN structures. The main idea is to mimic the process of the brain by using perceptrons. The output (y) of the perceptron is equal to the multiplication of weights (w) and inputs (x). During the training, the network tries to learn correct weights which gives good performance for testing. The mathematical representation is given by Eq (1):

$$y_i = w_i^T x \tag{1}$$

Perceptron is generally sufficient enough for linear problems, but it lacks for more complex and non-linear problems. Therefore, MLP is applied as a first network. MLP has an input, output, and at least one hidden layer. The main difference is the hidden layer which has non-linear activation functions. If there is more than one hidden layer, then the network becomes deep.

The training and updating of the weights start randomly, and the network aims to minimize the error in every epoch. Applying normalization to the feature set helps to improve the performance because various inputs in a different range can be inexplicable

for the network. We prefer to apply the following Z-score normalization technique in our feature set to avoid outliers:

$$x' = \frac{x - \mu}{\sigma} \tag{2}$$

which μ represents mean and σ is the standard deviation. During the training period, overtraining, creating complex models (high number of parameters, layers, neurons), memorizing the data, or employing a high number of epochs can see a decrease in the training error, but in many cases, it jeopardizes the validation performance since the network lost the generalization ability. That is why stopping criteria play a crucial role to keep the generalization ability of the obtained network by preventing overtraining and saving simulation time. It is one of the regularization ways to handle these issues. In the literature, there are different regularization techniques and, in this work, we implement early stopping, weight decay, noise injection, dropout, and capacity reducing.

MLP includes fully connected (dense) layers. Every neuron has a connection to all neurons in the next dense layer and the network tries to find optimum weights and bias terms to minimize the loss function of the network. Consequently, these bring a remarkable number of parameters into the network. For this reason, the reduction in terms of parameters can be procured by using only local connections between the neurons instead of having all the connections. Thus, an alternative layer is introduced to a fully connected layer which calls a convolutional layer that helps to decrease the number of parameters for keeping computation more simplified. The resulted ANN is called CNN which is a powerful network that is commonly used in image processing. Additionally, CNN can also be used on a one-dimensional (1D) time-series signal. The general idea of CNN is the same for 1D, 2D, or 3D data. The difference is about their input dimension and the feature selection by a filter. In our problem, we use CNN for two sequential steps which are feature selection and time series forecasting. The feature extraction part generally includes convolution layers and pooling layers that help to extract major features of input sets, afterwards the fully connected layers can use these selected features to learn and perform the forecasting task. Furthermore, we use 1D-CNN to build hybrid networks since the superiority of CNN networks is extracting features from a given input set [8].

Recurrent Neural Network (RNN) allows to include past or self-connections and it is widely used if there is sequential data in different relational architectures such as one-to-one, many-to-one, one-to-many, and many-to-many in text, audio, and video applications [13]. The main difference of RNN from MLP is the output relations with past computations and past experiences which is called past memory in addition to current input at a particular time step. RNN uses BPTT (Back Propagation through Time), and it is good at capturing short-term dependencies, but computing and repeating the gradient to the initial cell state causes exploding gradient problems if the values are bigger than one and vanishing gradient problems for values smaller than one. There are methods to treat this gradient problem which are mainly changing activation function to

Rectified Linear Unit (ReLU) function, returns the output as 0 for any negative inputs and same value if non-negative, initializing the weight matrix, but the most robust way is introducing gated cells namely LSTM and GRU [13]. We use LSTM which is better to handle gradient problems as well as tracking information for long-term dependencies. Fig. 1. represents the LSTM model with its gates.

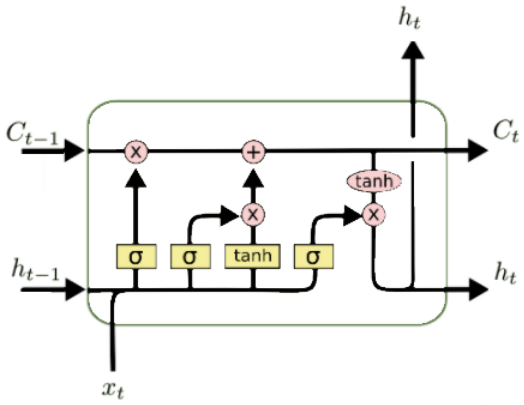

Fig. 1. The representation of the LSTM model [14].

LSTM cells can follow the information throughout many steps, and they can detect and forget irrelevant past information by forgetting gates f_t (Eq. (3)), keep the relevant new information to update by input gates i_t Eq. (4) and having new potential cell state $\widetilde{C_t}$ by using tanh layer with Eq. (5), update previous information by update cell state C_t Eq. (6) and finally produce the output h_t by using output gate o_t Eq. (7) and Eq. (8). In this way, LSTM can capture both long- and short-term dependencies. In the following equations, σ is the sigmoid layers, W and b are the weight and bias for the related gates.

$$f_t = \sigma(W_f, [h_{t-1}, x_t] + b_f \tag{3}$$

$$i_t = \sigma(W_i, [h_{t-1}, x_t] + b_i \tag{4}$$

$$\widetilde{C_t} = tanh(W_c, [h_{t-1}, x_t] + b_c \tag{5}$$

$$C_t = f_t . C_{t-1} + i_t . \widetilde{C_t} \tag{6}$$

$$o_t = \sigma(W_o, [h_{t-1}, x_t] + b_o \tag{7}$$

$$h_t = o_t . \tanh(c_t) \tag{8}$$

LSTM is a promising network for sequential data, as well as playing a key role in building hybrid network structures. We create a hybrid network by using an LSTM and a 1D-CNN network. The hybrid model is a combination of 1D-CNN and LSTM networks. While the integration of 1D-CNN aids feature selection, LSTM increases the capability of storing relevant past information. The model architecture can be seen in Fig 2. The CNN layer is followed by max-pooling and flatten layers to improve the feature extraction process.

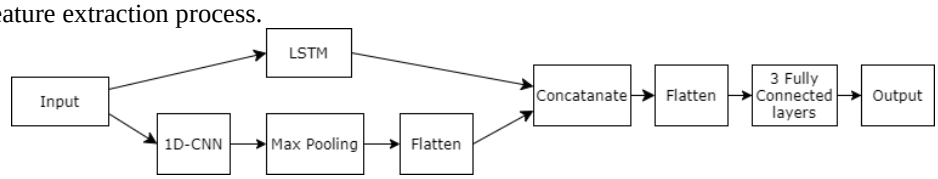

Fig. 2. Structure of the Hybrid Network

2.2 Regularization Methods

The regularization techniques are developed to ensure DNNs' performance not only on the training set but also on the test data. Regularization includes various techniques, in this study we applied early stopping, weight decay, Gaussian noise injection, dropout, and reducing network capacity to the original model for every DL algorithm constituted. One of the most common techniques is early stopping. It interferes with the number of epochs by checking both training and validation errors whether they are decreasing or not. If the decrease of validation error is stopped for one or more epochs while training error is still decreasing, the network is stopping the training to avoid overfitting and memorizing the network. Weight decay defends removing unnecessary connections by adding a penalty term to the training. Another simple implementation is noise injection during training, wherein this paper we tested Gaussian noise. Dropout is a fast technique that can be applied during training. It is mainly essential for large networks therefore here it is applied to compare the performance of the DNNs. During the implementation of dropout, the various non-output units with defined probability are removed in every epoch of the training process. The final method is dimensionality reduction, intending to reduce the number of parameters and complexity in the network.

We use four error metrics to evaluate the model performances: Mean Square Error (MSE) which is also our loss function, Mean Absolute Error (MAE), Mean Absolute Percentage Error, and R^2 which is the exposition of evaluating the feature set can express the output. If the R^2 is close to 1, then it can be said the feature set is chosen appropriately. The equations are given by Eq. (9)-(12). While x_s indicates the actual load consumption, $\hat{x}_s$ represents the forecasted load consumption value.

$$MSE = \frac{1}{L}\sum_{s=1}^{L}(\hat{x}_s - x_s)^2 \tag{9}$$

$$MAE = \frac{1}{L}\sum_{s=1}^{L}|\hat{x}_s - x_s| \tag{10}$$

$$MAPE = \frac{100}{L}\sum_{s=1}^{L}\frac{|\hat{x}_s - x_s|}{|x_s|} \tag{11}$$

$$R^2 = 1 - \frac{\sum_{s=1}^{L}(\hat{x}_s - x_s)^2}{\sum_{s=1}^{L}(x_s - \mu)^2} \tag{12}$$

The lookback window is related to several previous time steps for forecasting the next step. The decision of the lookback length affects the network's performance since the closest previous data express the behavior of the set. Thus, we investigate the network performance under different lengths of lookback windows.

3 Case Studies and Results

3.1 Data Preparation

There are 10 various features that are mainly based on the extracted load and temperature features from the data set. In detail, daily average load, maximum and minimum values for both load and temperature, weekend/weekday and hour of the day, the previous day, week, and average load. Temperature data is one of the most useful meteorological inputs among all and the results are also supporting this point. It is worth mentioning that only historical temperature and load data were used as inputs, which means that we can execute the short-term load forecasting without having any problem. The data set includes 277 days. Fig 3. presents all electrical load consumption which is in a resolution of 15 minutes and the relevant extracted features from this set.

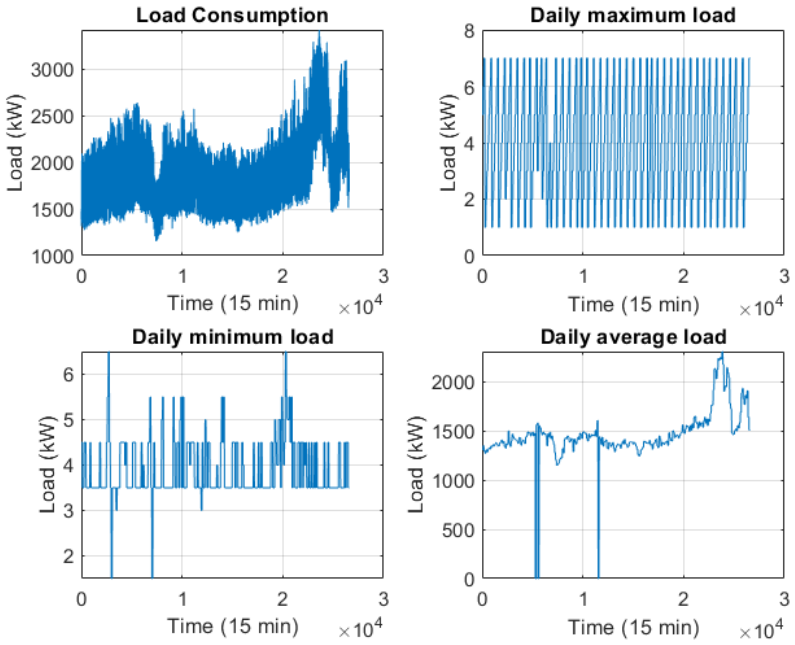

Fig. 3. Load consumption and other features are extracted from the load data set.

The work in the paper was to implement various DL techniques incorporating the stated regularization methods and evaluate which method was more effective in terms of performance improvement. Furthermore, DNNs allowed to use of sequential historical data, and the length of the lookback window affected the forecasting performance as well. Thus, studies were developed based on selected network architectures and afterward the effects of the lookback window on performance were recorded. All the mentioned architectures were created using the Keras library in the Python environment.

3.2 Implementation of Deep Learning and Regularization Techniques in Comparison with Lookback Window

In the first stage, different DL techniques including various regularization methods are applied and then the best regularization is identified for individual DL architectures. In the second stage, the performances of the selected regularization methods for each model architecture are compared under various lookback window lengths.

The MLP architecture contains one input layer, 4 hidden layers with 64, 64, 48, and 24 neurons, and one output layer. The CNN contains input, 1D convolution, max pooling, 3 fully connected, and output layer. The filter size is 16 and the kernel is 2. Similarly, the LSTM network has an input layer, an LSTM layer, 3 hidden layers, which have the same number of neurons as the last hidden layers of MLP, and then the output layer. The hybrid network is constituted by using both 1D-CNN and LSTM layers. The

CNN layer is followed by max-pooling and flatten then, the LSTM and output of the 1D-CNN layer are concatenated to be used in the following 3 fully connected layers.

We choose the default features as follows: the optimizer was Adam, the batch size was 32, the activation function was ReLU, and the epoch was 250 for every network. We then applied 5 regularization techniques for each architecture: early stopping, weight decay, noise injection, dropout, and capacity reducing. We compared all performances with the one without implemented any regularization. We evaluated all the results based on 4 given criteria for training, validation, and test sets, which are divided into 80%, 10%, and 10%, respectively. We look back at the 24 steps for all networks. The set has 15-minutes intervals; thus, we mainly use the previous 6 hours in this case. The results are given in Table 1, the results show that the LSTM network outperforms the others without having any regularization. The Hybrid network demonstrated similar performance with the implementation of early stopping and capacity reduction. Consequently, although the LSTM network demonstrates good performance for time-series with its high complexity, the hybrid network can have accurate performance with less complexity and fast convergence performance. The results also showed that dropout was not effective for any DL approach with this data set. The main reason is that the data set does not have considerable overfitting. The other regularizations improved for some specific cases. For instance, while early stopping help to improve the 1D-CNN and hybrid models, Gaussian Noise injection improved the performance for MLP and LSTM networks.

Table 1. Deep learning model results under various regularization methods for training, validation, and testing.

		1D-CNN				**MLP**				**LSTM**				**HYBRID (1DCNN – LSTM)**			
		MSE	MAE	MAPE	R^2	MSE	MAE	MAPE	R^2	MSE	MAE	MAPE	R^2	MSE	MAE	MAPE	R^2
Orginal Model	Training	3241.76	44.61	2.33	0.98	1633.97	31.79	1.65	0.99	816.34	22.84	1.21	0.99	5882.68	56.76	2.83	0.96
	Validation	3435.67	45.66	2.38	0.98	2079.28	36.11	1.89	0.99	1184.20	27.78	1.47	0.99	5968.64	57.32	2.88	0.96
	Testing	3185.18	44.53	2.30	0.98	2083.81	36.32	1.68	0.99	**1205.12**	**27.63**	**1.45**	**0.99**	5807.15	56.93	2.83	0.96
Early Stopping	Training	2721.53	39.94	2.03	0.98	2370.58	36.02	1.78	0.98	10735.99	73.72	3.76	0.93	1099.25	26.34	1.40	0.99
	Validation	2957.44	41.76	2.12	0.98	2720.64	39.92	2.02	0.98	13522.50	76.05	3.89	0.92	1506.89	30.60	1.63	0.99
	Testing	**2754.48**	**40.80**	**2.07**	**0.98**	2879.61	40.41	2.01	0.98	11032.59	73.97	3.78	0.92	**1461.67**	**30.35**	**1.60**	**0.99**
Weight Decay	Training	2868.53	41.55	2.13	0.98	1311.73	28.45	1.48	0.99	768463.75	706.33	38.11	-4.29	1805.35	33.49	1.72	0.99
	Validation	3222.23	43.77	2.25	0.98	1717.92	32.73	1.72	0.99	753222.69	704.61	38.06	-4.04	2036.73	35.46	1.83	0.99
	Testing	3159.23	43.61	2.22	0.98	1655.49	32.20	1.67	0.99	744978.62	701.36	37.55	-4.23	2027.21	35.10	1.79	0.99
Injection Noise	Training	3572.38	44.79	2.26	0.98	1301.25	28.34	1.47	0.99	804.41	22.72	1.20	0.99	5269.10	53.48	2.70	0.96
	Validation	3850.55	46.46	2.35	0.97	1713.31	33.25	1.75	0.99	1213.51	28.08	1.49	0.99	5377.95	54.16	2.72	0.96
	Testing	3428.49	44.34	2.23	0.98	**1747.02**	**33.17**	**1.72**	**0.99**	1210.84	27.93	1.46	0.99	4687.72	51.58	2.60	0.97
Dropout	Training	19041.84	101.87	4.65	0.88	19041.84	101.87	4.65	0.88	237659.22	477.28	24.41	-0.61	237807.97	477.06	24.36	-0.61
	Validation	18456.02	100.39	4.61	0.88	18456.02	100.39	4.61	0.88	237959.42	477.23	24.44	-0.57	238208.52	477.01	24.39	-0.58
	Testing	20724.70	107.49	4.91	0.86	20724.70	107.49	4.91	0.86	242222.45	481.90	24.46	-0.69	242810.34	482.01	24.42	-0.70
Reducing Capacity	Training	3666.39	44.97	2.27	0.98	3666.39	44.97	2.27	0.98	2433.77	38.17	1.98	0.98	1227.29	27.59	1.45	0.99
	Validation	3966.35	46.80	2.38	0.97	3966.35	46.80	2.38	0.97	2510.18	39.06	2.03	0.98	1506.01	30.93	1.62	0.99
	Testing	3748.62	46.12	2.33	0.97	3748.62	46.12	2.33	0.97	2484.74	38.23	1.97	0.98	1509.34	30.79	1.61	0.99

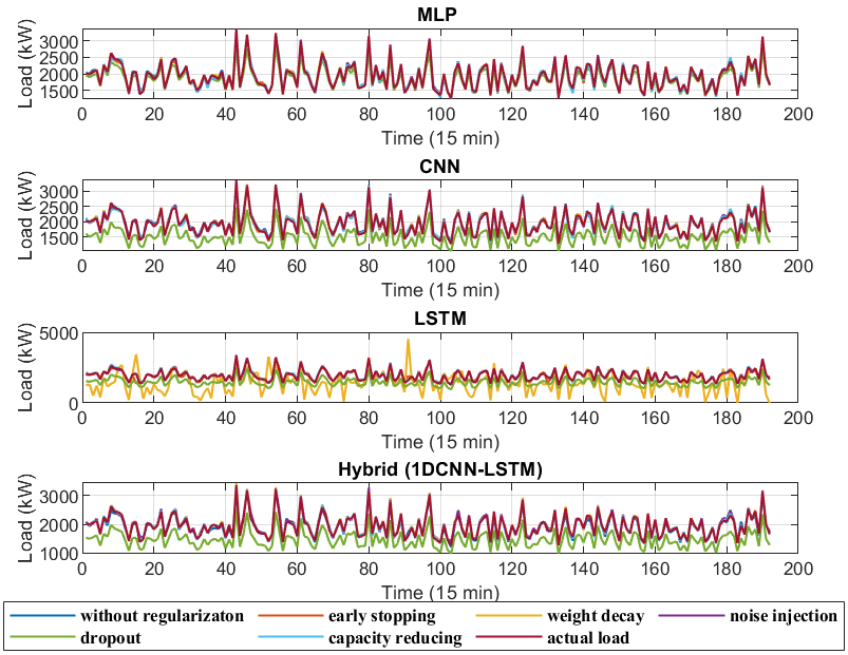

Fig. 3. The performance change for capturing the load consumption under various regularization techniques for MLP, CNN, LSTM, and Hybrid networks.

We take these architectures without making any change in network arrangements and move to the second part to obtain the simulations for observing the change of lookback window for 6, 12, 24, and 36. The detailed results are illustrated in Table 2.

In Fig. 4 the effect of the regularization techniques is demonstrated by plotting predictions and the actual load for representative 2 weeks. MLP has exhibited less variation under these techniques, while CNN and Hybrid network exhibit similar behavior under the regularizations, and especially dropout did not improve their performance at all. However, while LSTM was not imposed by both weight decay and dropout, it was affected positively by noise injection. Regularization methods are generally used to prevent overfitting. Here we do not see the effectual change since the set did not have an overfitting problem in the first place. However, we still observe the change in networks' performance. Based on these results, we specify a regularization type for each DL model to use in the next application, which is an investigation of the lookback window.

As can be seen in Table 2, changing the length of the lookback window has a realizable effect on the forecast results. Fig. 5 presents the change of average MSE results under variation of 4 lookback window horizons. LSTM has outperformed all others except when the lookback was equal to 36 because it has caused a serious increase in the network complexity. The lowest forecasting performance belongs to 1D-CNN which tells the necessity of using it in Hybrid forms instead of operating separately for time series data forecasting.

Table 2. Lookback window length results for DL architectures.

lookback window		**6**				**12**				**24**				**36**			
		MSE	MAE	MAPE	R^2	MSE	MAE	MAPE	R^2	MSE	MAE	MAPE	R^2	MSE	MAE	MAPE	R^2
1D-CNN	Training	5629.40	55.94	2.81	0.96	5928.65	56.51	2.80	0.96	2721.53	39.94	2.03	0.98	2137.75	36.22	1.85	0.99
	Validation	5687.34	56.47	2.85	0.96	5930.08	56.33	2.80	0.96	2957.44	41.76	2.12	0.98	2596.77	39.41	2.01	0.98
	Testing	6230.14	57.98	2.91	0.96	6044.12	56.88	2.82	0.96	2754.48	40.80	2.07	0.98	**2487.27**	**39.05**	**1.99**	**0.98**
MLP	Training	2263.51	37.48	1.94	0.98	1878.37	34.17	1.76	0.99	1301.25	28.34	1.47	0.99	1322.01	28.80	1.51	0.99
	Validation	2629.24	40.27	2.07	0.98	2145.07	34.17	1.88	0.99	1713.31	33.25	1.75	0.99	1765.66	33.15	1.74	0.99
	Testing	2818.37	41.22	2.12	0.98	2382.00	38.54	1.98	0.98	**1747.02**	**33.17**	**1.72**	**0.99**	1797.43	33.54	1.75	0.99
LSTM	Training	853.66	23.32	1.23	0.99	803.08	22.57	1.19	0.99	804.41	22.72	1.20	0.99	2523.52	38.09	2.02	0.98
	Validation	1288.98	28.37	1.49	0.99	1243.18	28.01	1.48	0.99	1213.51	28.08	1.49	0.99	3189.45	41.82	2.23	0.98
	Testing	1410.18	29.63	1.54	0.99	1237.09	28.13	1.47	0.99	**1210.84**	**27.93**	**1.46**	**0.99**	2903.25	41.16	2.17	0.98
Hybrid	Training	1503.00	31.16	1.61	0.99	1071.96	26.02	1.37	0.99	1099.25	26.34	1.40	0.99	1736.66	32.51	1.69	0.99
	Validation	1707.72	33.05	1.71	0.99	1441.20	29.87	1.57	0.99	1506.89	30.60	1.63	0.99	2054.11	35.31	1.83	0.99
	Testing	1956.98	35.77	1.83	0.99	**1450.27**	**30.58**	**1.59**	**0.99**	1461.67	30.35	1.60	0.99	1908.43	34.11	1.76	0.99

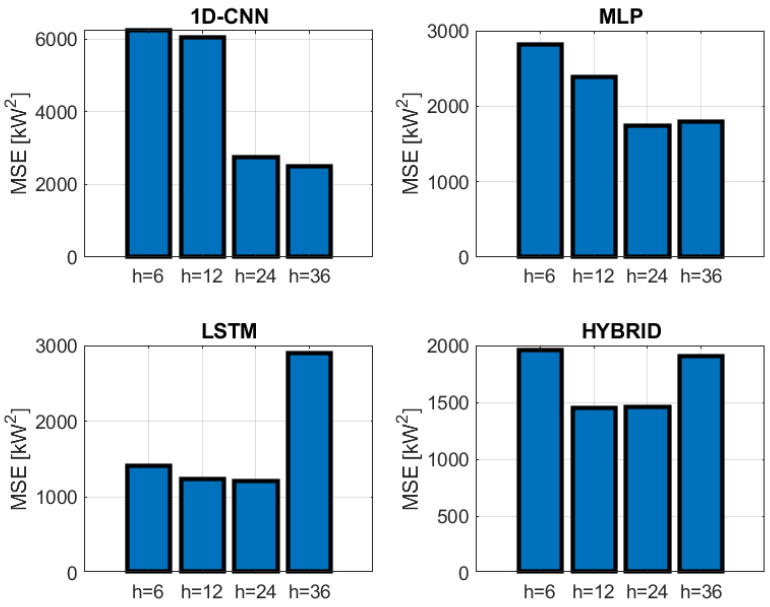

Fig. 4. MSE change with respect to length of lookback window for deep learning techniques.

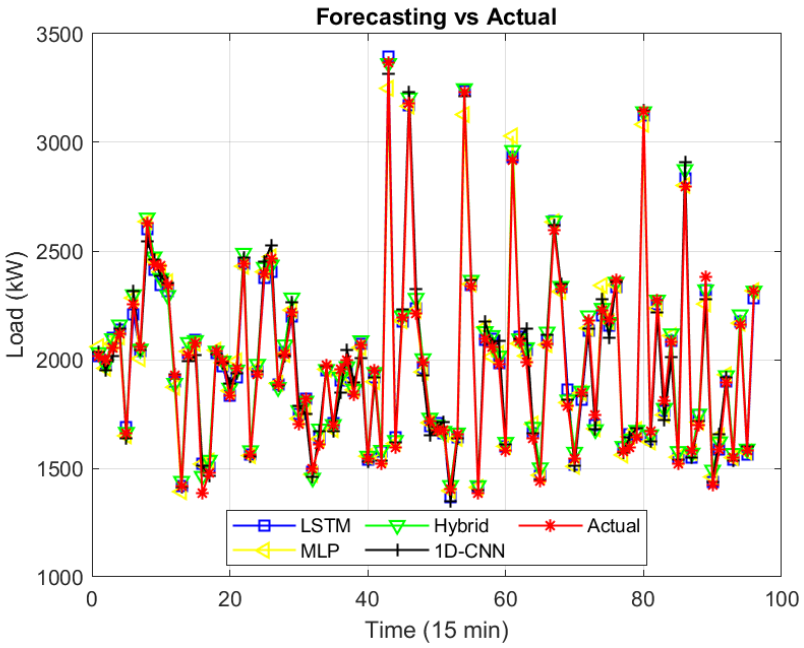

Fig. 5. The forecast results after taking the best regularization method of each DL model for a day ahead when the lookback window is 24.

The length effect of the lookback window changes based on the data and problem. In general, the window size can be decided by implying grid search or evolutionary algorithms. Fig. 6 illustrates the forecasting results for 96 different steps by making one-step-ahead predictions of each proposed deep learning method with respect to the actual load. All the methods have converged to the actual values. However, we can also realize MLP and 1D-CNN seem to be less accurate in comparison with LSTM and Hybrid models.

4 Conclusion

Short-term load forecasting is getting more important not only for energy management in an optimal way, but also make reliable operations for the power system. In this study, MLP, CNN, LSTM, and integration of CNN-LSTM models are presented. The variations are presented when applying different regularization techniques to decrease the error. Each network is affected on a different scale. For instance, weight decay causes an increase in all network prediction performance, except for the LSTM. In general, LSTM is working successfully without applying any regularization which outperforms others, only Gaussian noise injection improved slightly the results. Next, we observe the effect of lookback window length over the forecasting performance. We specifically investigated the effect of historical data window length and resolution under various deep learning and regularization techniques. Naturally, increasing the historical data window length brings complexity to the network. Since the RNN architecture is already complex, the performance is affected negatively only when the length of the lookback window is adjusted to 36, we obtained improvement for the rest of the network performance. Thus, in general, we observe that the inclusion of more steps in the past helps to improve prediction performance, especially if it is not a complex neural network. In addition, the results showed that each regularization technique has an impact on the forecasting result and the selection of both regularization and model architecture details are completely relevant subjective to the data set. Even though one of the most common techniques is dropout, it negatively affects the result for deep learning methods like LSTM and hybrid networks significantly. Apart from these, numerical results show that forecasting error (MSE) can be under 2%.

5 Future Work

Load forecasting and deep learning have been developed and studied during the last few years. However, there is still much room for novel implementations and improvements. In this study, we see that the design of DL architecture is crucial. Every data needs attention to choosing the best method and architecture design. Thus, in future studies, we will employ more robust preprocessing techniques to make sure they will give more robust results. Also, we can create many architectures, but the optimal way is to choose all these prominent hyper parameters wisely. We will employ a differential evaluation to choose these parameters to optimize the forecast performance. Lastly, the

lookback window decision should be taken by implementing different search algorithms.

References

1. Energy Information Administration (EIA), https://www.eia.gov/outlooks/ieo/, last accessed 2021/04/14.
2. Notton, Gilles, et al. "Intermittent and stochastic character of renewable energy sources: Consequences, cost of intermittence and benefit of forecasting." Renewable and sustainable energy reviews 87: 96-105, (2018).
3. He, Wan. "Load forecasting via deep neural networks." Procedia Computer Science 122: 308-314, (2017).
4. Alpaydin, Ethem. Introduction to machine learning. MIT press, (2020).
5. Masum, Shamsul, Ying Liu, and John Chiverton. "Multi-step time series forecasting of electric load using machine learning models." International conference on artificial intelligence and soft computing. Springer, Cham, 2018.
6. Eskandari, Hosein, Maryam Imani, and Mohsen Parsa Moghaddam. "Convolutional and recurrent neural network based model for short-term load forecasting." Electric Power Systems Research 195: 107173, (2021).
7. Hosein, Stefan, and Patrick Hosein. "Load forecasting using deep neural networks." 2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT). IEEE, 2017.
8. Tian, Chujie, et al. "A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network." Energies 11.12 (2018): 3493.
9. Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." The journal of machine learning research 15.1: 1929-1958, (2014).
10. Zhu, Dixian, et al. "A machine learning approach for air quality prediction: Model regularization and optimization." Big data and cognitive computing 2.1:5, (2018).
11. Bouktif, Salah, et al. "Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches." Energies 11.7: 1636, (2018).
12. Ghiasi, Golnaz, Tsung-Yi Lin, and Quoc V. Le. "Dropblock: A regularization method for convolutional networks." arXiv preprint arXiv:1810.12890 (2018).
13. Goodfellow, Ian, et al. Deep learning. Vol. 1. No. 2. Cambridge: MIT press, 2016.
14. Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/, last accessed 2021/04/18.