Modeling of Thermal Power Plant for multi-output prediction using Artificial Neural Networks

Md Safwan Mondal MMONDA4@UIC.EDU
University of Illinois Chicago, USA

Subramanian Ramasamy SRAMAS21@UIC.EDU
University of Illinois Chicago, USA

Partha Sarathi Dutta PDUTTA3@UIC.EDU
University of Illinois Chicago, USA

Hadis Anahideh HADIS@UIC.EDU
University of Illinois Chicago, USA

Abstract

Modeling of thermal power plant performance has become a necessity in order to generate the power output more efficiently. In this paper, the team propose a methodology to model the performance of thermal power plant with respect to its operational conditions with the use of Artificial Neural Networks (ANN). The team had implemented three different neural networks such as Deep feed-forward Neural Network (DNN), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) with different architectures for each network by performing the data preparation, network modeling and assessing the results obtained to identify the best architecture suitable for this problem. The proposed RNN architecture achieves good results as compared to DNN and CNN models.

1. Introduction

Thermal power plant is a complex multi-variable system associated with severe nonlinearity, uncertainties and multi-variable couplings which become more evident when the system is operated at a higher level energy conversion capacity. It is almost impossible to build a mathematical model of the system using conventional analytical methods. Nonlinear modeling techniques are required for predicting the output power of a power plant and for getting an estimate of the important parameters like the pressure, temperature and water flow rate of different components like the reheater, superheater, etc of the power plant. Fig. 1 shows the components of a power plant. In particular, knowing the pressure and temperature is critical because all the components has it's own design limitations of temperature and pressure tolerance. Further, the feedwater used in the cooling system is very costly and we can work in an optimal range of feedwater flow so that the cost of the feedwater can be minimized. Earlier, two kind of modeling methods were primarily used: the experimental method(Åström & Bell, 2000)(Kocaarslan & Cam, 2007) and the first principle based method (Flynn & O'Malley, 1999) (Wei et al., 2007). The recursive least squares (RLS) were also used to model the powerplant but one of the basic assumption of the RLS was that the power plant performs linearly around the operating point. However as there are more uncertainties, non-linearities and dynamics in the system, the RLS performs poorly (Liu et al., 2003) . More advanced techniques are now currently developed to account for the complexities and dynamic behavior of all the components of a modern power station.

For the case of a power plant, neural network is considered as the most suitable technique for capturing all the non-linearities and dynamic behavior of the system.

1.1. Related works

Previous literatures which were focused on predicting the output power of a power plant using a neural network had got high degree of prediction accuracy. One

Appearing in *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, UK, 2012.

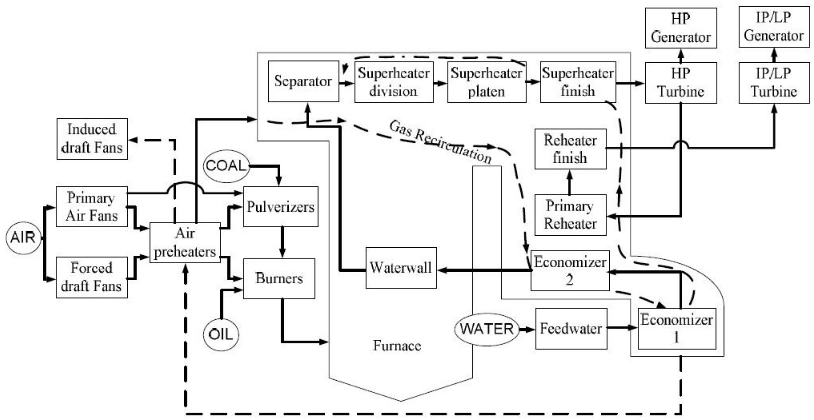

Figure 1. Schematic diagram of a power plant layout

popular neural network used for these kind of problem are Feedforward Neural Network (FNN). Most literatures used feed forward network and a "tapped delay" to deal with the readings of the time after a certain interval of few seconds. These networks used backpropagation to correct all the weights and biases of the network after each iteration.

However, the disadvantage of a FNN is that it involves large number of neurons and it cannot capture the dynamic behavior of the system(Lee et al., 2007). A Recurrent Neural Network (RNN) is the perfect fit for modeling of a power plant as it involves a delay or feedback connection to capture the time delay. The output of RNN not only depends on the input but it also develops a memory in the training process. The delay in the network stores the history of the input and the output data and makes it suitable for dynamical system modeling. (Ku & Lee, 1995) have used Diagonal Recurrent Neural Network (DRNN) for modeling the power plant. The advantage of DRNN includes less training time and lesser number of neurons as it has only one hidden layer of self-recurrent neurons . The other popular RNN used in power plant modeling is Long Short Term Memory (LSTM). (Irwin et al., 1995) had modelled a 200 MW power plant using neural network. There were also other techniques like fuzzy logic, genetic algorithm and neuro-fuzzy network which were used for power plant modeling. A comparison of FNN, FRNN (Fully-connected Recurrent Neural Network) and DRNN was also previously studied. Peng et al used LSTM to predict the changing trend of dissolved gases in transformer oil using LSTM. And(Zhang et al., 2020) used Dynamic Particle Swarm Optimization (DPSO) and bidirectional LSTM with adaptive learning methods to balance the local and global search abilities. Convolutional Neural Networks (CNN) were also used in power plant modeling to mainly visualize the datasets(El-Sefy et al., 2021) .(Kim, 2014) had used CNN-LSTM network to predict the housing energy consumption where the CNN was used to analyze the spatial features and the LSTM was used to study the temporal features.

In a power plant the boiler spray desuperheater plays a crucial role to cool down and regulate the high temperature superheated steam flowing through the boiler system (Nie Yu & Shuhong, 2013). It is important to keep the temperature of the steam within normal operating range to guarantee the safe operation of turbine and other auxiliary equipment of the power plant. A sudden change of the operating conditions can influence the temperature control unit of the power plant thus affecting the power output of the turbine (Wen, 2020).Thus, treating these operating parameters as the inputs and the temperatures, desuperheater feedwater flow and the power plant output as the outputs, in this work we tried to model the complex multivariable power plant system with the help of deep learning applications to predict the output behaviour of the power plant w.r.t. it's input operational conditions.

2. Data preparation

Data preparation consists of three integrated phases: data acquisition, data preprocessing and sampling the data for the training and testing on the machine learning model.

2.1. Data acquisition

The dataset was collected from open-source data science platform (Kaggle) contains a large amount of power plant data captured continuously by the control panel or power plant monitoring device. The dataset contains total 116 power plant parameters with 17,280 observations. Due to human error, faulty sensor measurement there might be some missing values or faulty observations called 'outliers' in the dataset which are needed to be detected and replaced by preprocessing of the data.

2.2. Data preprocessing

For easy understanding and interpretation the parameter column names were changed to the proper power plant terminologies. The whole dataset was scrutinized if it contains any missing values, null or invalid observations. The non-natural outliers present in the dataset can confuse the machine learning model hence it is important to replace the outliers with proper data replacing method. We replaced the outliers of a variable with their previous row's values. The features in the dataset have different ranges; hence the feature values were normalized before test training splitting.

2.3. Training, Testing dataset

The dataset was pre-randomized and divided into training and testing data in 70:30 ratio. The training observations were fed to machine learning model to learn the nonlinear relationships between the variables and testing data were used to verify the performance of the model. As per our objective we choose total 18 multiple outputs including the power output of the turbine, desuperheater feedwater flow and the inlet-outlet temperatures of the valves of the power plant; whereas the remaining 98 features describing the process parameters were considered to be the input parameters of the model.

3. Methods

In order to make sure that the dataset in hand is a time-based sequential dataset, the team performed the autocorrelation analysis to graphically summarize the strength of the relationship of a certain observation with previous observations. Out of 18 output response variables, two important variables like the Generator output and the steam temperature are chosen. Figures Fig. 2 and Fig. 3 represent the corresponding autocorrelation plots for those two variables. From those plots, it can be seen that for a lag of 50 across the observations and with 95% confidence interval, there exists strong dependencies or correlations between the observation values. Hence, it can be said that this is a time-based sequential dataset.

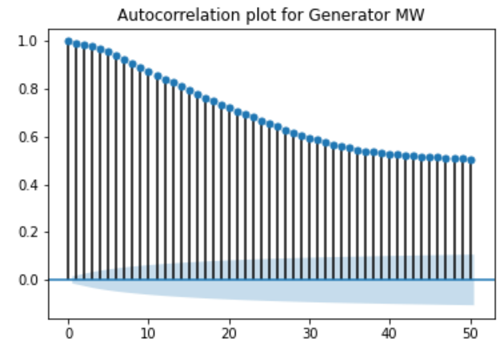

Figure 2. Autocorrelation plot for Generator MW output

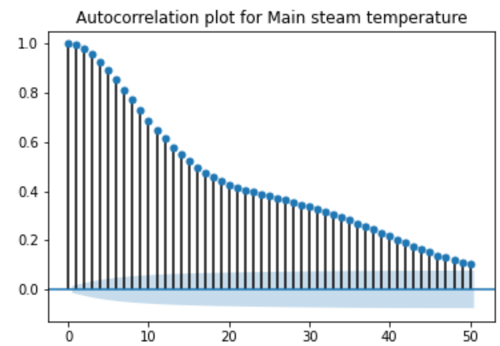

Figure 3. Autocorrelation plot for Main steam outlet temperature output

3.1. Deep Neural Network Modeling

The simple algorithm to start with was the deep feedforward neural network (DNN) to model the complex nonlinear relationship between the input and output parameters. The DNN consists a number of layers namely, the input layer: $X = [X_1, X_2, ...X_p]^T$, the output layer: $y = [y_1, y_2, ...y_L]^T$ and a number of hidden layers connected in a feedforward way as shown in Fig. 4. The hidden layers contains a number of neurons which computes the weighted sum of the incoming signals, bias terms to pass through an activation function g(.) to produce the output of the neurons. The learning algorithm used for training was the backpropagation learning algorithm which tried to minimize the error between the predicted output and actual output. The Mean Squared Error (MSE) is defined by Eq. 1 as the error metrics for evaluating the performance of the model. We also used R^2 value defined as Eq. 2 for measuring the accuracy of different models.

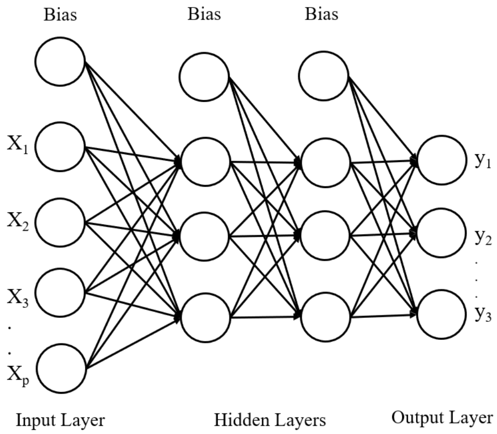

Figure 4. DNN architechture

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 \quad (1)$$

$$R^2 = 1 - \frac{RSS}{TSS} \quad (2)$$

3.2. Convolutional Neural Networks

Convolutional Neural Network (CNN) was first proposed by (LeCun et al., 1995) for image processing, and one of the major advantages of using CNN for image processing analysis is its ability to process very large images or video sequences. Though CNN are widely popular for image classification, some of the works like (Abdel-Hamid et al., 2012) and (Kim, 2014) had introduced CNNs for sequential data as well. The team had found from the correlation analysis that the dataset for the problem in hand is a time-series sequential dataset. Moreover, it is also proven from (Wang et al., 2019) and (Mehtab & Sen, 2020) that CNN performs well on a time series dataset. This is because, unlike feed-forward networks, CNN learns to recognize existing patterns in the dataset. Although CNN is not being popularly used for solving the particular problem that we have in hand, the team was interested in exploring to see how well the convolutional neural network predicts the output results by keeping the sequence of the data into account.

Convolutional Neural Networks are similar to ordinary neural networks in the sense that, there are several layers and neurons in each layer are connected to the neurons in the next layer. These neurons have learnable weights and biases similar to feed-forward neural networks, except that the CNN architecture allows to encode certain properties like spatial-sharing of weights and spatial pooling thereby vastly reducing the number of trainable parameters in the network.

There are three main types of layers to build the CNN.

- Convolutional layer
- Pooling layer
- Fully-Connected layer

Each set of layers use appropriately fixed-size kernels as filters to recognize patterns in the dataset. Use of such kernels greatly reduces the computational burden on the fully connected layers and thus predicts the output efficiently.

3.3. RNN modelling

As has been previously found in the literature, RNNs are the most suited for modeling a power plant because the data are collected in the sensors sequentially after a certain delay. One of the popular architecture among all the RNNs is the LSTM. It keeps that portion of the history in its memory that is most relevant and discards the rest because of its special gated network. It has an input gate, output gate, and forget gate for the information flow. This specific architecture of LSTM prevents gradient explosion or vanishing gradient problems and the long-term dependency issue. The advantage of this method includes high accuracy, robustness, parallel computation ability, and the capacity to fully understand the complex non-linear property of the different components which are present in the plant. However, training an LSTM requires a large dataset like the one for this problem which has more than 17,000 data taken at different timestamps for all the 115 features. Also, (Chen et al., 2020) in their paper on "very short term power prediction of the PV power plant" had shown that among all the RNNs, LSTM performs the best for forecasting or predicting the output power. In the next section the different experiments with different model architecture have been discussed in details.

4. Experiments

4.1. DNN architechture

The DNN consists of 98 input units in the input layer and 18 output neurons in the output layer. The number of hidden layers and the neurons in the hidden layers were varied experimentally to check their variation in the performance. (Alnaimi & Al-Kayiem, 2011) showed that sigmoid, hyperbolic tangent and linear activation function has the capability to produce the best results for power plant modeling. We tried these activation functions with upto 3 hidden layers to check the performances of the models.

4.2. CNN architecture

In the present literature, there exists a plethora of architectures for CNN model. Out of the existing CNN models, the Residual Network (ResNet) is one of the popular networks which outperforms classic CNN models. This is because, ResNets are one of the most efficient Neural Network architectures that mitigates the vanishing gradient problem. Despite being a deep network, the number of trainable parameters are comparable to other CNN architectures because of its shortcut residual layers and the error rate is too low compared with other CNN models. Fig. 5 represents an example ResNet for an 1-D Convolutional time series data.

4.3. Residual Network

In this section, the team describes about the CNN architecture that had been used to solve this problem. The proposed CNN architecture to tackle this problem is a relatively deep Residual Network (ResNet) adapted from (Ismail Fawaz et al., 2019). For creating a prediction model for this problem, this architecture contains 11 layers in total, out of which, the first 9 layers are the convolutional layers followed by a Global Average Pooling (GAP) layer, and then these are connected to a Fully Connected (FC) layer that performs the prediction of the output values. The key characteristic of the ResNet is the presence of residual layers which basically facilitates the shortcut residual connection between two consecutive convolutional layers. Fig. 5 represents the ResNet with skip residual connections for our regression problem.

The network consists of three residual blocks and inside each residual block, there are three convolutional layers combined with batch normalization layers wrapped up in a ReLU activation function. Each residual block can skip and bypass the connection of those convolutional layers, and connect with the next residual block via convolutional and batch normalization layers wrapped up similarly in a ReLU activation function. The number of filters for first residual block is fixed to 64, whereas the rest of the blocks contain 128 filters. The kernel sizes for each residual blocks are fixed. The kernel sizes for the first, second and third convolutions within a residual block are 8, 5 and 3 respectively. These 3 residual blocks are followed by the GAP layer and the FC layer, which predicts the 18 multiple output values. The loss function for the problem is Mean-Squared Error (MSE) and ADAM optimizer is used for the optimization of the loss function.

4.4. RNN architecture

A sequential model was implemented and the LSTM package was imported from the TensorFlow library to construct the LSTM network. The dataset was reshaped into the 'number of samples, timesteps, features' format. In this case, as we are modeling and predicting the output variables and we are not forecasting against time, the timestep is considered one. 256 layers of LSTM were followed by another layer of 512 layers of LSTM.

Activation functions ReLU and tanh were used in the hidden layer. It is to be mentioned that the proper choice of the activation function is still an active field of research but ReLU was chosen as the objective of the project was regression and not classification. The other variants of ReLU like leaky ReLU, parametric ReLU, and randomized ReLU could also have been tested but were not done because the usage of ReLU itself gave a regression coefficient (R2) value over 94 percent. The final output layer was chosen as linear. Also, bagging is a popular technique in a neural network for reducing the generalization error. The idea involves training several possible combinations of the network and then each combination voting for the output at the end. But doing this on a very large dataset like ours involves lots of computation time. So a dropout technique is used which removes the non-output units of the network. A dropout of 20 percent was introduced after the first set of LSTM layers and 10 percent on the next set.

4.5. Results for DNN model

As the team varied the number of hidden layers with different activation functions, the DNN model with 3 hidden layers and with sigmoid activation function in the hidden layers and linear activation function at the output layer produced the best result as shown in table 1. But as the mean squared error (MSE) of testing dataset was still higher 0.1344 the team further experimented with CNN model.

4.6. Results for CNN model

In order to train the ResNet model for this power plant dataset, the team fixed the batch size to be 100 and the number of epochs to be 50. After training the model, it was observed that the MSE value was low and both the training and testing accuracies were 98.5% and 91.6%,

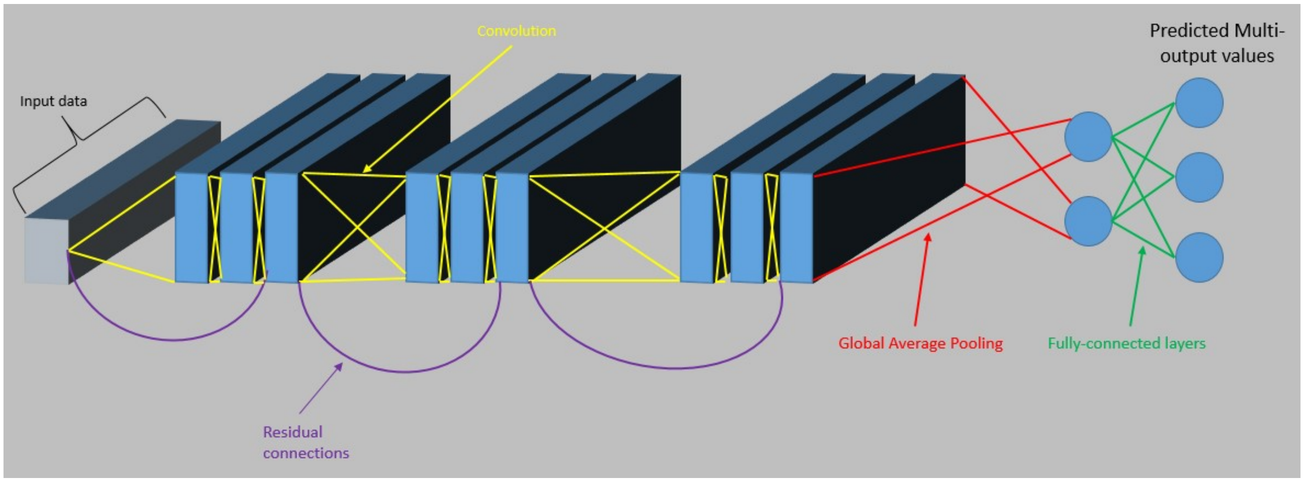

Figure 5. Residual Network

Table 1. Model evaluation - Metrics obtained from the different DNN architectures for regression prediction.

Hidden Layers	Activation functions	R^2 score (in %) Training	Testing
1 HL(50)	Tanh	83	81
1 HL(50)	Sigmoid	84	82
1 HL(50)	ReLu	81	82
2 HL(50-25)	Tanh	85	83
2 HL(50-25)	Sigmoid	86	85
2 HL(50-25)	ReLu	84	83
3 HL(60-40-25)	Tanh	87	84
3 HL(60-40-25)	Sigmoid	89	86
3 HL(60-40-25)	ReLu	86	85

Table 2. Model evaluation - Metrics obtained from the different CNN architectures for regression prediction.

Model	Set	MSE	R^2 score (in %)
Simple CNN	Training set	0.06	93.8
	Testing set	0.12	88.3
ResNet	Training set	0.01	98.5
	Testing set	0.08	91.6

Table 3. Model evaluation - Metrics obtained from the different RNN architectures for regression prediction.

Model	Set	MSE	R^2 score (in %)
Simple RNN	Training set	0.09	91.3
	Testing set	0.08	91.7
LSTM	Training set	0.04	95.7
	Testing set	0.06	94.1

which implies that the model does a good job in predicting the right output values for the corresponding power plant inputs. The results of the MSE and the accuracies were tabulated in the Table 3. In the table, the Simple CNN model represents a simple CNN architecture with two convolutional and max-pooling layers and two fully connected layers. These MSE and R^2 values were calculated from the equations 1 and 2 The figure Fig. 7 represents the performance plots of the neural network. From these plots of losses and accuracies vs epochs, it can be seen that the model parameters converge closer to optimality.

4.7. Results for RNN model

The LSTM network was followed by a regular deeply connected network (dense) with 512, 256, 128, 64, 32 and 18 (same as the number of output variables) layers respectively. As it is a regression problem, the performance was evaluated using the R^2 score. The model was trained for 200 epochs as training for higher epochs did not led to a significant improvement in accuracy. Training for very less epochs like 10, 20, etc is also not a proper strategy as the loss continues to decrease and does not saturate for such a less number of epochs. The mean squared error (MSE) was also evaluated. The designed LSTM model performed very well and the R^2 score after performing 200 epochs was 94.05% and the validation loss was 0.0590 (see Fig. 8). A summary of the comparison of the performance of the RNN network implemented by LSTM and simple RNN is shown in table 3.

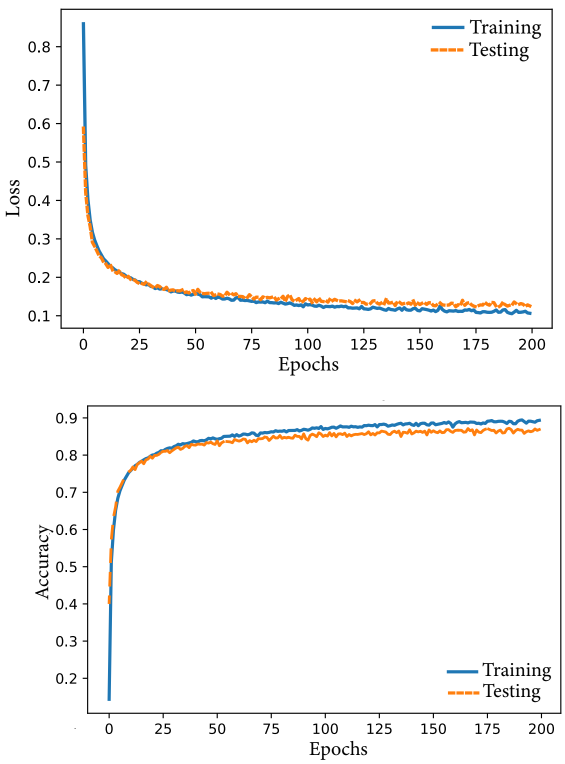

Figure 6. Training vs Testing for DNN model

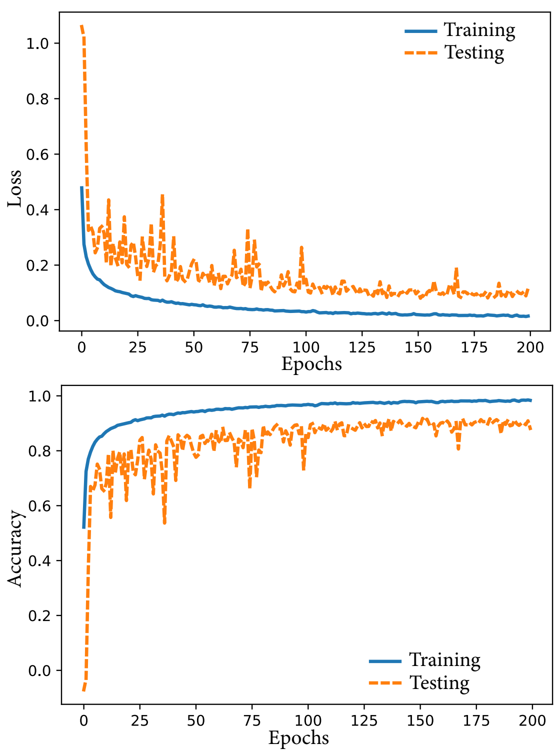

Figure 7. Training vs testing for CNN model

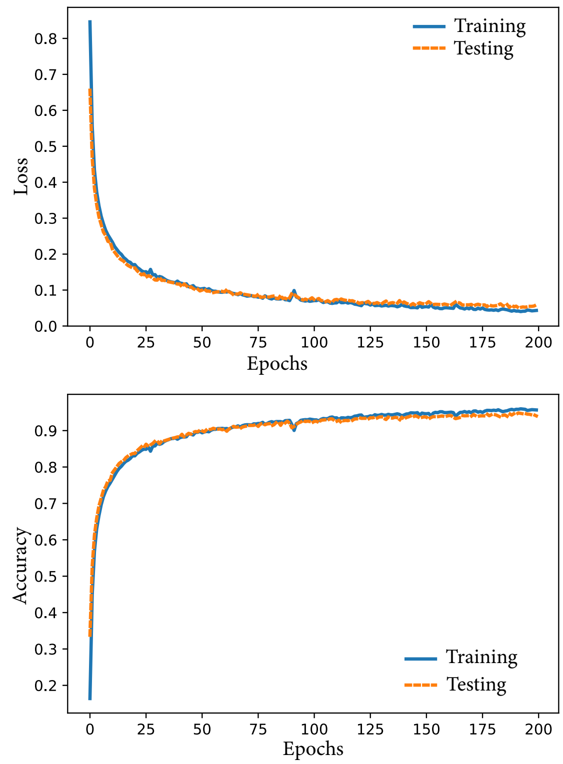

Figure 8. Training vs testing for RNN model

4.8. Comparison of Different models

Due to inefficiency of the traditional machine learning models to map the complex nonlinear relationship between the power plant operational parameters and the power plant outputs, the team incorporated different types of NN models. The relative comparison of different types of models have been shown in fig.9. From the figure, LSTM based RNN model proved to be the optimal NN model. Power plant output was also predicted using the optimal RNN model which has been depicted in fig. 10.

5. Conclusion

In this paper, the team proposed three different neural network models to to solve a problem of performance improvement of desuperheater temperature control system. The proposed regression-based model can predict multiple outputs based on certain inputs more accurately by considering the non-linearities and dynamic behavior of the system. Out of comparing the metrics of various architectures for the model, it was corroborated that the RNN-based LSTM model produced good results in terms of obtaining lower loss and higher R^2 score values as opposed to other models.

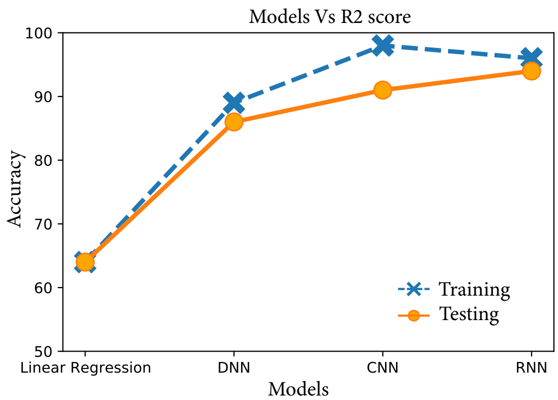

Figure 9. Accuracy of different models

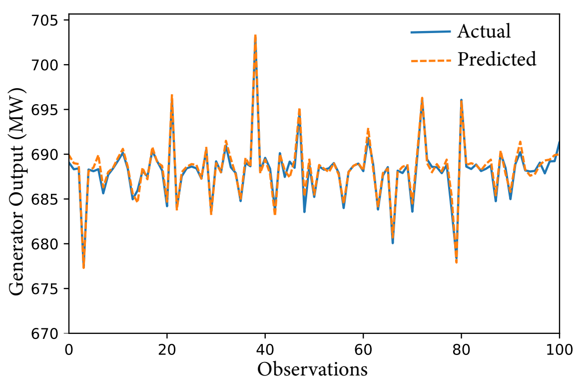

Figure 10. Actual output Vs predicted output

This proposed model can be extended further by performing a prediction analysis of different output variables from certain input variables in order to find an optimal range of input variables for which, the output results obtained will generate the electricity power output more efficiently by reducing the cost.

Acknowledgments

The team would like to thank Dr. Hadis Anahideh for her guidance and teaching different advanced Machine Learning concepts. The teammates declare that there was no conflict of interest during the course of this project.

References

Abdel-Hamid, Ossama, Mohamed, Abdel-rahman, Jiang, Hui, and Penn, Gerald. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In *2012 IEEE international conference on Acoustics, speech and signal processing (ICASSP)*, pp. 4277–4280. IEEE, 2012.

Alnaimi, FB Ismail and Al-Kayiem, Hussain H. Artificial intelligent system for steam boiler diagnosis based on superheater monitoring. *Journal of Applied Sciences*, 11(9):1566–1572, 2011.

Åström, Karl Johan and Bell, Rodney D. Drum-boiler dynamics. *Automatica*, 36(3):363–378, 2000.

Chen, Biaowei, Lin, Peijie, Lai, Yunfeng, Cheng, Shuying, Chen, Zhicong, and Wu, Lijun. Very-short-term power prediction for pv power plants using a simple and effective rcc-lstm model based on short term multivariate historical datasets. *Electronics*, 9(2): 289, 2020.

El-Sefy, M, Yosri, A, El-Dakhakhni, W, Nagasaki, S, and Wiebe, L. Artificial neural network for predicting nuclear power plant dynamic behaviors. *Nuclear Engineering and Technology*, 53(10):3275–3285, 2021.

Flynn, ME and O'Malley, MJ. A drum boiler model for long term power system dynamic simulation. *IEEE Transactions on Power Systems*, 14(1):209–217, 1999.

Irwin, G, Brown, M, Hogg, B, and Swidenbank, E. Neural network modelling of a 200 mw boiler system. *IEE proceedings-Control theory and Applications*, 142(6):529–536, 1995.

Ismail Fawaz, Hassan, Forestier, Germain, Weber, Jonathan, Idoumghar, Lhassane, and Muller, Pierre-Alain. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

Kaggle. Kaggle dataset. https://www.kaggle.com/datasets/deepakburi062/power-plant-data-optimization-problem. Online; accessed March 24, 2022.

Kim, Yoon. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1181. URL https://aclanthology.org/D14-1181.

Kocaarslan, I and Cam, E. Experimental modelling and simulation with adaptive control of power plant. *Energy Conversion and Management*, 48(3):787–796, 2007.

Ku, Chao-Chee and Lee, Kwang Y. Diagonal recurrent neural networks for dynamic systems control. *IEEE transactions on neural networks*, 6(1):144–156, 1995.

LeCun, Yann, Bengio, Yoshua, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361 (10):1995, 1995.

Lee, Kwang Y, Heo, Jin S, Hoffman, Jason A, Kim, Sung-Ho, and Jung, Won-Hee. Neural network-based modeling for a large-scale power plant. In *2007 IEEE Power Engineering Society General Meeting*, pp. 1–8. IEEE, 2007.

Liu, Xiang-Jie, Lara-Rosano, Felipe, and Chan, CW. Neurofuzzy network modelling and control of steam pressure in 300 mw steam-boiler system. *Engineering Applications of Artificial Intelligence*, 16(5-6): 431–440, 2003.

Mehtab, Sidra and Sen, Jaydip. Stock price prediction using cnn and lstm-based deep learning models. In *2020 International Conference on Decision Aid Sciences and Application (DASA)*, pp. 447–453. IEEE, 2020.

Nie Yu, Zhang Yangpin and Shuhong, Huang. Artificial intelligent system for steam boiler diagnosis based on superheater monitoring. *Journal of Engineering for Thermal Energy and Power*, 28(04): 336–340+431–432, 2013.

Wang, Kang, Li, Kenli, Zhou, Liqian, Hu, Yikun, Cheng, Zhongyao, Liu, Jing, and Chen, Cen. Multiple convolutional neural networks for multivariate time series prediction. *Neurocomputing*, 360:107–119, 2019.

Wei, Jian-Lin, Wang, Jihong, and Wu, QH. Development of a multisegment coal mill model using an evolutionary computation technique. *IEEE Transactions on Energy Conversion*, 22(3):718–727, 2007.

Wen, Libin. Model of thermal power plant considering water spray desuperheater for power system analysis. In *Journal of Physics: Conference Series*, volume 1633, pp. 012029. IOP Publishing, 2020.

Zhang, Guoyin, Tan, Feng, and Wu, Yanxia. Ship motion attitude prediction based on an adaptive dynamic particle swarm optimization algorithm and bidirectional lstm neural network. *IEEE Access*, 8: 90087–90098, 2020.