

Load forecasting: STLF using FFNN Model with Adam Optimization Algorithm

Devyani Singh¹ and Dr. Vipin Kakar²

¹⁻²Shri. Mata Vaishno Devi University/Department of electronics and communication, Jammu, India
Email: devyani97singh@gmail.com, vipan.kakar@smvdu.ac.in

Abstract— Short-Term Load Forecasting (STLF) is the process of predicting the amount of electricity demand that will be required over a short period of time, ranging from a few hours-days. The purpose of STLF is to help utility companies and power system operators to optimize the allocation and usage of energy resources, ensuring that enough electricity is generated and delivered to meet the expected demand at a given time. This involves analyses of factors that influences energy demand, such as weather conditions, time of day, weekdays, special events and other techniques to generate accurate and reliable predictions of future energy demand. Feed Forward Neural Network (FFNN) model is used for STLF in order to meliorate load accuracy. The data of three regions has been reckoned for training proposed model. Parameters-temperature, relative humidity, liquid precipitation and wind speed at a distance of 2m of every place is manipulated as training data set for presented model. Adam Optimization is applied for training model as it has robust learning rate, which makes this most pertinent algorithm to be applied for deep learning. This algorithm exhibits faster computational time and entails less tunning variables that embodies Adam Optimization as better algorithm and hence will improve accuracy outcomes of FFNN model.

Index Terms— Adam Optimization, Artificial Neural Network, Feed Forward Neural Network, Short-Term Load Forecasting, weather conditions.

I. INTRODUCTION

Abrupt headway in power generation is driving the need for a secure and cost-effective grid to continuously meet the user requirements under economic restrictions [1]. The strain on conventional grids has enhanced significantly through sudden rise in load demand due to distinct factors including residential, transportation, industrial and commercial [3]. Certain parametrical conditions such as temperature, humidity, wind direction, dirt, presence or absence of sunlight, increase or decrease of current and voltage, cloudy weather conditions, and many more increasing load values thus, leading disruptions in load accuracy.

Load forecasting is described as the difference between the actual value of load demand and predicted value of load demand of future. Load forecasting plays an integral role in the energy management system. The techniques used for load forecasting are branched into two types – parametric technique (examples auto regressive moving average (ARMA), stochastic time series, linear regression and general exponential techniques) and non- parametric technique (example artificial intelligence) [8]. load forecasting is an efficacious technique for boosting load accuracy and load prediction. Load forecasting is categorized into three different types (a)short-term load forecasting, which forecasts load from few minutes to one week (b)medium-term load forecasting, which forecasts

load from several weeks to few months (c)long-term load forecasting, which forecast load up-to few years [5]. This research is bounded to short-term load forecasting. The main motif of Short-term load forecasting is to forecast load from several minutes to one week. A robust STLTF aids to utilize and energy suppliers in resolving the challenges brought on by the increased use of renewable energy sources and the growth of the electricity market with more intricate pricing schemes in upcoming smart grids [4]. STLTF gets affected by several factors such as load value differs during day and night, special occasions, festive days, weekends, weekdays, weather conditions and many more [10]. Over a decade, research in STLTF has drastically increased because of reduction of amount of power produced by power industries. STLTF can benefit many sectors and industries by yielding the precise value of future load which will result into scrupulous planning, approximation of supply numbers and govern the value of prices. This will cause a drop in the operating cost, rise in profit and a more dependable supply of electricity [9]. Numerous machine learning, artificial intelligence and other computer science technologies are being adopted for STLTF. These involve Artificial Neural Network (ANN), Support Vector Machine (SVR), the random forest (RF), Long-Short Term Memory (LSTM), Convolutional Neural Network (CNN), Least square support vector machine (LS-SVM) many more can be utilized [6][7]. ANN or its' types are the most preferred model for STLTF.

A. Main Contributions of Work

Main contributions of this work

- Feed forward Neural Network (FFNN) model is proposed for STLTF.
- Model is being trained for parameters- humidity, temperature, barometer pressure and wind speed of Panama City for year 2013.
- Adam optimization algorithm is contemplated for training FFNN model.

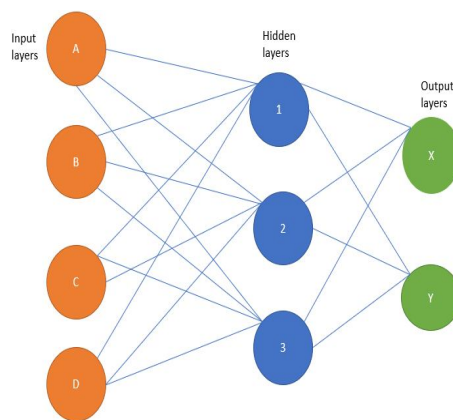
This paper is catalogued into: section 2 has abridged description of ANN, Adam Optimization and FFNN, section 3 contains the methodologies and data collection, section 4 comprises of simulations and results and section 5 culminates the paper.

II. ANN, ADAM OPTIMIZATION AND FFNN

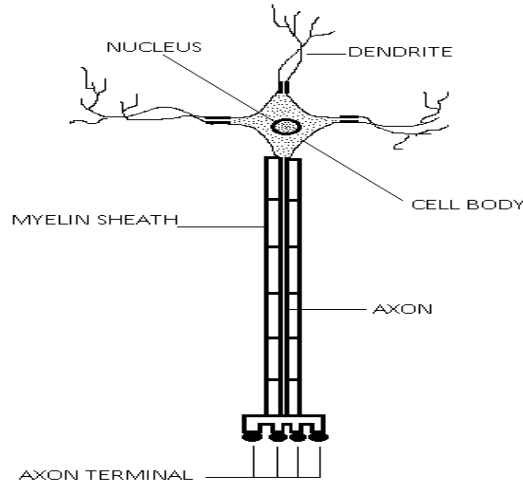
A. Ann (Artificial Neural Network)

The design of ANN is extracted from the structure of human neuron. ANN is incorporated of three layers – input, hidden and output layers. Usually, ANN is the preferred model for Short-Term Load Forecasting (STLTF). For extricating the relationship between load values and weather variables, ANN model is being considered. The generalization ability of ANN should be as high as possible for attaining accurate forecasting results. For this purpose, ANN needs adequate training data set with convenient generalization. However, enough data is not obtainable as the seasonal effects or time of the day differs the load value also varies with the variations in weather situations [11,12].

The input weights are applied to the input layer of ANN which will process the information and will lead it to next layer for further processing that will generate output with help of activation function or algorithm for which ANN is being trained. The desired output will be contrasted with the obtained output if both show dissimilarities, then there exhibits an error.



“Fig. 1” basic architecture of artificial neural network



“Fig. 2” biological structure of human neuron

Performance parameters for ANN model-

Mean Absolute Percentage Error: $MAPE = \frac{1}{n} \sum \frac{|e_t|}{d_t}$

Mean Absolute Error: $MAE = \frac{1}{n} \sum |e_t|$

Root Mean Square Error: $RMSE = \sqrt{\frac{1}{n} \sum e_t^2}$

Mean Squared Error: $MSE = \frac{1}{n} \sum e_t^2$

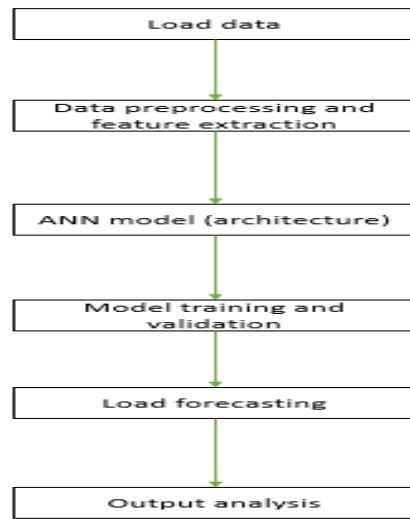
A. Adam Optimization

Adam Optimization gradient-based optimization algorithm. It is attained from adaptive moment estimation as it enumerates independent adaptive learning rates for numerous parameters starting with the first and second moments of gradients [14]. The usage of this algorithm is quite popular for optimizing neural networks [16]. This maintains a running average of the first and second moments of the gradient. The first moment is the mean of the gradient, while the second moment is the variance of the gradient. The Adam algorithm is implied by using the few steps for each weight in the network: determine the gradient of the loss function with respect to weight. Update the first moment estimate by taking a weighted average of the current gradient and the previous first moment estimate. Update the second moment estimate by taking a weighted average of the current gradient squared and the previous second moment estimate. Calculate the bias-corrected first and second moment estimates to avoid bias towards zero in the early stages of training. Update the weight using the bias-corrected first and second moment estimates and the learning rate. The learning rate in Adam is adaptive, meaning it is adjusted for each weight based on the first and second moment estimates. The momentum in Adam is also adaptive, which means it takes into account the first and second moment estimates when computing the momentum.

B. FFNN (Feed-Forward Neural Network)

Feed Forward Neural Network (FFNN) model is proposed for STLf. When a model indicates the existence of feedback network between input- output layers of the network then that model is illustrated as Feed Forward Neural Network (FFNN) [13].

It is called "feed forward" because information flows through the network in one direction, from input to output, without any feedback loops. A feed forward neural network consists of an input layer, one or more hidden layers, and an output layer. The input layer receives the input data, which is then processed through the hidden layers using a set of weights and biases. Each neuron in the hidden layers performs a weighted sum of the input values, followed by the application of an activation function. The output of each hidden layer is then passed as input to the next layer until the output layer is reached. The output layer produces the final result of the network. The main purpose of the hidden layers in a feed forward neural network is to learn the underlying patterns in the input data. Each hidden layer represents a set of learned features that capture different aspects of the input data. As the number of hidden layers increases, the network becomes more capable of capturing complex patterns in the input data.



“Fig. 3” block diagram of proposed model

III. PROBLEM DEFINITION

Analysing the accuracy and other performance factors of short-term load forecasting by using certain machine learning model with specific algorithm.

IV. PROBLEM FORMULATION

Most of the existing models present in literature survey escalate the performance of smart grid by reducing STLF. STLF is reduced by improving the load prediction accuracy and minimizing errors (difference between desired load and output load) under all environmental conditions (temperature, humidity, cloudy days, dust accumulation) or other special occasions. There is need to design multi-layer ANN model that can increase the load prediction accuracy and minimize error.

V. SOLUTION METHODOLOGIES

The data evaluated for STLF:

- Online data of three places- Tocumen, Santiago and David City.
- Data of every hour is utilized.

A. Initialization, Training and Adaption of ANN

Numerous aspects that comprehend electric load are financial, climatological, time and many more, hence spotted in all dissipating units- industrial, domestic and residential. Usually, load forecasting is classified into three- short-term, medium-term and long-term load forecasting. The research of this paper will be bounded to STLF [15]. The parameters utilized for STLF are temperature, humidity, wind speed and barometer pressure. These variables are input to the proposed model.

Feed Forward Neural Network (FFNN) is proposed model as it composed of numerous neurons that are connected through synaptic weights, arranged in the form of layering format [17].

Adam optimization algorithm is adopted from adaptive moment estimation as it enumerates independent adaptive learning rates for numerous parameters starting with the first and second moments of gradients [14]. The main aim of utilizing Adam optimization is its' better learning rate for deep learning techniques (ANN models). The parameters of all the three places (from years 2015-2020) function as input to the model and after data cleaning. The output will be forecasted data with better accuracy outcomes, attained by implementing Adam Optimization algorithm on ANN model. With the help of Google Collab FFNN model has been established.

B. Illustration of block diagram of FFNN

The hourly data of power load of Tocumen, Santiago and David city is supplied as input to the load data block. The missing or inaccurate data points are eradicated by data pre-processing and feature extraction is being

performed to include features like weather conditions. ANN model is created by defining the number of Input, Hidden and Output layer neurons. Numerous activation functions are utilized to enhance functioning of model. For training proposed model, pre-processed data is considered and model performance is evaluated by validation set. The training process is required to diminish the load error between actual and forecasted load.

C. Adam Optimization with FFNN

The main motive of forecasting demand of load grounded on load data and other data related to environmental conditions. Because of which regression is needed and for this purpose FFNN is the most preferred model with Adam Optimization algorithm. Adam optimization is used as it has great performance and fast convergence over a broad range of problems.

TABLE I. PARAMETERS DETAIL OF FEED-FORWARD NEURAL NETWORK

Number of input neurons	6(temperature, relative humidity, wind speed, liquid precipitation, data-time, net demand)
Number of output neurons	1
Neural network model	Feed Forward Neural Network
Number of hidden layers	2
Algorithm	Adam Optimization
Number of neurons in hidden layers	64 and 32
Activation function for hidden layer 1	ReLU (Rectified Linear Unit)
Activation function for hidden layer 2	Tanh (Hyperbolic Tangent)
Performance variables	MAE, MSE, RMSE, R2
Maximum number of epochs	25
Percentage of using information	Training dataset (80%), Testing dataset (20%)
Learning rate	0.001
Weight update method	Adam Optimization

VI. DATA MODEL

TABLE II. DATA OF TOCUMEN CITY

date and time	Net demand	Temperature at 2m, Tocumen	Relative humidity at 2m, Tocumen	Liquid precipitation at 2m, Tocumen	Wind speed at 2m, Tocumen
3-1-2015,1:00	970.345	25.86525879	0.018576382	0.016174316	21.85054582
3-1-2015,2:00	912.1755	25.89925537	0.018653292	0.016418457	22.16694428
3-1-2015,3:00	900.2688	25.93728027	0.01876786	0.0154800415	22.45491088
3-1-2015,4:00	889.9538	25.95754395	0.018890057	0.016273499	22.11048087
3-1-2015,5:00	893.6865	25.97384033	0.018981151	0.017280579	21.8608852
3-1-2015,6:00	879.2323	26.03414307	0.019079996	0.014541626	20.06203782
3-1-2015,7:00	932.4876	26.6914917	0.019331587	0.0066452026	21.62349567
3-1-2015,8:00	1048.972	27.67406616	0.019369703	0.0068626404	23.775317
3-1-2015,9:00	1167.9074	28.76040039	0.019171301	0.010231018	24.6361515
3-1-2015,10:00	1257.5069	29.76665649	0.018759238	0.009017944	25.86267136
3-1-2015,11:00	1254.583	30.52376709	0.018408278	0.004322052	26.82808151
3-1-2015,12:00	1216.9004	30.98180542	0.018133579	0.0027980804	26.65431045
3-1-2015,13:00	1202.1556	31.21120605	0.017904576	0.0026474	25.75416055
3-1-2015,14:00	1197.2616	31.15651855	0.017744431	0.0014648438	24.90217224
3-1-2015,15:00	1169.0034	30.81676636	0.01765308	0.00040137768	24.21445514
3-1-2015,16:00	1136.7054	30.05721436	0.017706675	2.70E-05	23.76680031

3-1-2015,17:00	1101.9447	28.80245361	0.018042427	8.37E-13	22.75511413
3-1-2015,18:00	1107.0406	27.36635742	0.018568844	8.37E-13	20.67880942
3-1-2015,19:00	1142.1548	26.61394653	0.018851262	0.0004069805	19.92638635
3-1-2015,20:00	1097.2334	26.26726685	0.018889746	0.0014262199	19.42842398
3-1-2015,21:00	1074.6544	26.08352051	0.0188859386	0.0024576187	19.5695168
3-1-2015,22:00	1041.3244	25.95629272	0.018775864	0.0028734207	19.65992258
3-1-2015,23:00	999.634	25.8612915	0.01866179	0.0031547546	19.70297711

TABLE III. DATA OF DAVID CITY

date and time	Net demand	Temperature at 2m, David	Relative humidity at 2m, David	Liquid precipitation at 2m, David	Wind speed at 2km, David
3-1-2015,1:00	970.345	22.66213379	0.016562222	0.09609985	5.364147952
3-1-2015,2:00	912.1755	22.57894287	0.016509432	0.087646484	5.572471346
3-1-2015,3:00	900.2688	22.53103027	0.016479041	0.07873535	5.871183745
3-1-2015,4:00	889.9538	22.51223145	0.016486797	0.06838989	5.883621114
3-1-2015,5:00	893.6865	22.48165283	0.016455822	0.06436157	5.611723559
3-1-2015,6:00	879.2323	22.45601807	0.016409708	0.061538696	5.280351132
3-1-2015,7:00	932.4876	22.9493042	0.016569747	0.060897827	5.126910803
3-1-2015,8:00	1048.972	24.08812866	0.016676527	0.05619812	5.060610609
3-1-2015,9:00	1167.9074	25.47915039	0.016645972	0.051071167	4.915658463
3-1-2015,10:00	1257.5069	26.70415649	0.016607748	0.058685303	4.685579709
3-1-2015,11:00	1254.583	27.64876709	0.01673744	0.07723999	4.069101872
3-1-2015,12:00	1216.9004	28.13805542	0.017263828	0.095458984	3.122228324
3-1-2015,13:00	1202.1556	28.33620605	0.017370518	0.11364746	2.90053138
3-1-2015,14:00	1197.2616	28.2346355	0.01738585	0.1281128	2.914702659
3-1-2015,15:00	1169.0034	27.91051636	0.01740894	0.12640381	3.026640617
3-1-2015,16:00	1136.7054	27.29158936	0.017500682	0.11956787	3.247855434
3-1-2015,17:00	1101.9447	26.23995361	0.017706733	0.119659424	3.290524877
3-1-2015,18:00	1107.0406	24.96010742	0.017905086	0.11782837	3.331619551
3-1-2015,19:00	1142.1548	24.23894653	0.017653447	0.110961914	4.027541452
3-1-2015,20:00	1097.2334	23.73601685	0.017432531	0.105407715	4.50568921
3-1-2015,21:00	1074.6544	23.41164551	0.017356396	0.09338379	4.447957927
3-1-2015,22:00	1041.3244	23.15941772	0.017249985	0.08782959	4.26351307
3-1-2015,23:00	999.634	22.9394165	0.017105393	0.09085083	4.26351307

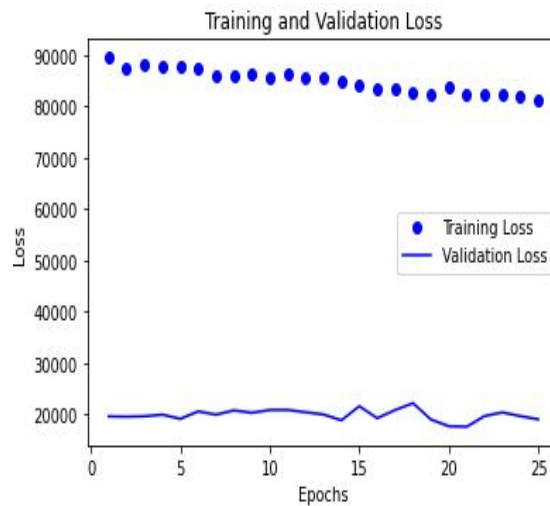
TABLE IV. DATA OF SANTIAGO

date and time	Net demand	Temperature at 2m, Santiago	Relative humidity at 2m, Santiago	Liquid precipitation at 2m, Santiago	Wind speed at 2m, Santiago
3-1-2015,1:00	970.345	23.48244629	0.017271755	0.0018553734	10.32894873
3-1-2015,2:00	912.1755	23.39925537	0.017264742	0.0013270378	10.68151714
3-1-2015,3:00	900.2688	23.34353027	0.017211463	0.0014281273	10.87492379
3-1-2015,4:00	889.9538	23.23879395	0.017127667	0.0025987625	10.51862016
3-1-2015,5:00	893.6865	23.07540283	0.017058544	0.0017290115	9.733588918
3-1-2015,6:00	879.2323	22.99508057	0.01702769	0.0014853477	9.08727316
3-1-2015,7:00	932.4876	24.2852417	0.017424239	0.0021762884	11.39539342
3-1-2015,8:00	1048.972	26.18969116	0.018072706	0.0045394897	12.872866
3-1-2015,9:00	1167.9074	27.91665039	0.018454138	0.0042915344	14.54802667
3-1-2015,10:00	1257.5069	29.17290649	0.018675314	0.0049209595	15.08168788
3-1-2015,11:00	1254.583	30.07064209	0.018820265	0.0074386597	15.28723923
3-1-2015,12:00	1216.9004	30.63805542	0.018843113	0.0098724365	15.28092683
3-1-2015,13:00	1202.1556	30.88308105	0.018789586	0.01342392	15.03815734
3-1-2015,14:00	1197.2616	30.82839355	0.018728623	0.016586304	14.69807932
3-1-2015,15:00	1169.0034	30.45739136	0.018713566	0.018455505	14.07604797
3-1-2015,16:00	1136.7054	29.77596436	0.018935008	0.018684387	12.47207795
3-1-2015,17:00	1101.9447	28.47432861	0.019415718	0.024085999	9.446227274
3-1-2015,18:00	1107.0406	26.56948242	0.019148678	0.028442383	7.084643851

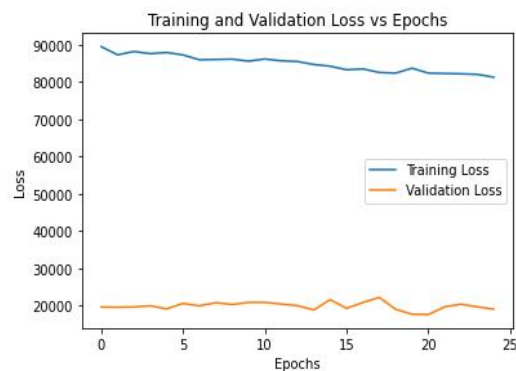
3-1-2015,19:00	1142.1548	25.81707153	0.018851262	0.024963379	7.028678289
3-1-2015,20:00	1097.2334	25.25164185	0.018554052	0.021995544	7.178384138
3-1-2015,21:00	1074.6544	24.77102051	0.018256664	0.018157959	7.502368873
3-1-2015,22:00	1041.3244	24.37035522	0.017967148	0.013534546	7.758983977
3-1-2015,23:00	999.634	24.056604	0.017792039	0.009838104	7.965630479

VII. RESULTS

- Google collab and python coding has been opted for FFNN model construction
- The data has been collected from online sources and after gathering all the data. Data cleaning is conducted for eliminating missing data points, after which data pre-processing will be performed to make smooth data running possible on machine learning techniques.
- Training model for dataset of three locations including-Tocumen, Santiago and David City (“TABLE 2”, “TABLE 3” & “TABLE 4”).
- Adam Optimization algorithm is used with FFNN because of huge dataset.
- Default weight update method of optimizer named Adam is utilized here.
- Python code is created and after running the code certain graphs can be plotted- graph between loss and epochs (“Fig.4” & “Fig.5”) and actual values vs predicted values (“Fig.6”).
- Values for MSE, RMSE, MAE, R-squared and Accuracy are displayed along-with “Fig.5”.

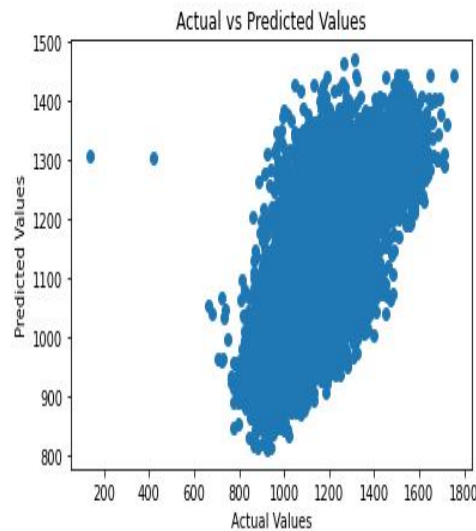


“Fig.4” graph of training and validation loss vs epochs



301/301 [=====] - 1s 2ms/step
MSE: 19117.552883475484
RMSE: 138.2662391311613
MAE: 112.27254811198134
R-squared: 0.48204300063453254
Accuracy: 90.70587905377863

“Fig. 5” graph between training and validation loss vs epochs and values of performance parameters are also displayed below the graph



“Fig.6” graph on actual vs predicted values

VIII. ACKNOWLEDGEMENT

No, acknowledgement has been proclaimed here.

IX. CONCLUSION

In this paper, the use of ANN model has effectuated an effective success of STLF. The online data of locations-Tocumen, Santiago and David city (2015-2020) is contemplated to train suggested model. Adam Optimization was better algorithm to be applied to FFNN model as it has greater efficiency and faster convergence properties. FFNN showed better accuracy results for STLF after imposing Adam Optimization. The analysis of performance parameters was better here. 90.7% accuracy achieved after training FFNN model. This value can be upgraded more by adding continuous changes to number of layers of the model. This can be implied on bigger scale, area-wise. In future, more research work can be done on FFNN model by covering more environmental parameters.

REFERENCES

- [1] Tahir, Muhammad F., et al. "Short term load forecasting using bootstrap aggregating based ensemble artificial neural network." *Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering)* 13.7 (2020): 980-992.
- [2] Dagdougui, Hanane, et al. "Neural network model for short-term and very-short-term load forecasting in district buildings." *Energy and Buildings* 203 (2019): 109408.
- [3] Jindal, Anish, Mukesh Singh, and Neeraj Kumar. "Consumption-aware data analytical demand response scheme for peak load reduction in smart grid." *IEEE Transactions on Industrial Electronics* 65.11 (2018): 8993-9004.
- [4] Chen, Kunjin, et al. "Short-term load forecasting with deep residual networks." *IEEE Transactions on Smart Grid* 10.4 (2018): 3943-3952.
- [5] Islam, Kazi Rafiqul, et al. "An Effective Short-Term Electrical Load Forecasting Model: A Constructive Neural Network Approach." *European Journal of Engineering and Technology Research* 7.4 (2022): 14-20.
- [6] Shafiei Chafi, Zahra, and Hossein Afrakhte. "Short-term load forecasting using neural network and particle swarm optimization (PSO) algorithm." *Mathematical Problems in Engineering* 2021 (2021).
- [7] Guo, Weilin, et al. "Machine-Learning based methods in short-term load forecasting." *The Electricity Journal* 34.1 (2021): 106884.
- [8] Baliyan, Arjun, Kumar Gaurav, and Sudhansu Kumar Mishra. "A review of short-term load forecasting using artificial neural network models." *Procedia Computer Science* 48 (2015): 121-125.
- [9] Hosein, Stefan, and Patrick Hosein. "Load forecasting using deep neural networks." *2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2017.
- [10] Elgarhy, Shady Mahmoud, et al. "Short term load forecasting using ANN technique." *2017 Nineteenth International Middle East Power Systems Conference (MEPCON)*. IEEE, 2017.
- [11] Singh, Saurabh, Shueb Hussain, and Mohammad Abid Bazaz. "Short term load forecasting using artificial neural network." *2017 Fourth International Conference on Image Information Processing (ICIIP)*. IEEE, 2017.

- [12] Singh, Saurabh, Shoeb Hussain, and Mohammad Abid Bazaz. "Short term load forecasting using artificial neural network." 2017 Fourth International Conference on Image Information Processing (ICIIP). IEEE, 2017.
- [13] Sazli, Murat H. "A brief review of feed-forward neural networks." Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering 50.01 (2006).
- [14] Soydaner, Derya. "A comparison of optimization algorithms for deep learning." International Journal of Pattern Recognition and Artificial Intelligence 34.13 (2020): 2052013.
- [15] Abbas, Farukh, et al. "Short term residential load forecasting: An improved optimal nonlinear auto regressive (NARX) method with exponential weight decay function." Electronics 7.12 (2018): 432.
- [16] Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv preprint arXiv:1609.04747 (2016).
- [17] Ojha, Varun Kumar, Ajith Abraham, and Václav Snášel. "Metaheuristic design of feedforward neural networks: A review of two decades of research." Engineering Applications of Artificial Intelligence 60 (2017): 97-116.