2022 2nd International Conference on Intelligent Technologies (CONIT)
Karnataka, India. June 24-26, 2022

Short Term Load Forecasting using Machine Learning Techniques

Sonakshi Dua[1], Shaurya Gautam[2], Mahi Garg[3], Rajendra Mahla[4], Mrityunjay Chaudhary[5], Dr. Shelly Vadhera[6]
[1,2,3,4,5] B.Tech, [6] Professor
[1,2,3,4,5,6]Department of Electrical Engineering, National Institute of Technology, Kurukshetra, India
[1]sonakshi_11814040@nitkkr.ac.in

2022 International Conference on Intelligent Technologies (CONIT) | 978-1-6654-8407-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/CONIT55038.2022.9848160

***Abstract*—With recent technological and scientific advancements in the power systems, there has been a tandem need for load forecasting. This paper mainly discusses short-term load forecasting, which refers to the prediction of the system load demand over an interval ranging between minutes ahead to one week ahead. With advent of Machine Learning, the process of demand prediction has become easier and cost effective. The challenge of predicting the future demand can be characterized as a regression problem, hence the method of Support Vector Regression is used, as it has proved to be a robust method in the recent research. Different Neural Networks are also being used in several domains; hence Deep Neural Network has also been used to test the accuracy, The paper discusses the results obtained by two different methods. The comparison between the outcomes of the different algorithms has been discussed, in order to get a thorough understanding. The methods are explained vastly. The paper also discusses the factors affecting load forecasting directly.**

Keywords— Short term load forecasting, DNN, Support vector regression

I. INTRODUCTION

Electrical forecasting refers to the technique by which load demand of end users can be predicted, well before time, to provide uninterrupted power supply while minimizing over-generation. The procedure is extremely important because electrical energy cannot be stored and thus needs to be generated whenever required. Also, another advantage of Load Forecasting is reduced financial losses, as only the required amount of power gets generated. Electrical load forecasting plays a key role in power system planning and operation procedures.

Load Forecasting can be divided into three basic types, depending upon the future time period, up to which demand is predicted:

1. **Short term Load Forecasting:** The method of Medium-Term load forecasting is used when demand needs to be predicted for a period ranging from few minutes ahead to hour ahead. It is important for the smooth operation of power grids, also plays a crucial role in energy market analysis and load flow analysis.
2. **Medium Term Load Forecasting**: The method of Medium-Term load forecasting is used when demand needs to be predicted for a period ranging from one week to one year. It is useful in case of scheduling maintenance and outages in the power system.
3. **Long Term Load Forecasting:** The method of Medium-Term load forecasting is used when demand needs to be predicted for a period greater than one year. It is used to assess the power systems and also for forecasting the limitations of power systems in the coming future, in order to upgrade the systems.

Load Forecasting depends on several factors, these can be divided into the following parent reasons:

- Weather parameters:
  - Sunshine
  - Wind speed and direction
  - Sky cover
  - Humidity
  - Temperature
- Time factor:
  - Season
  - Weekly cycle
  - Day of week
  - Hour of day
  - Holiday
- Type of Customer:
  - Domestic Load
  - Commercial Load
  - Industrial Load
- Economic Factors:
  - GDP growth
  - Per capita income growth
- Random Events:
  - Strike
  - Thunderstorm or any other disaster
- Electricity Price

So far, a variety of techniques have been employed for electrical load forecasting. This paper takes a dive into various techniques and does a brief comparison among them.

II. DATA CONDITIONING AND PREPROCESSING

The dataset of load demand was taken from Rajasthan Vidyut Prasaran Nigam (RVPN) for a period of 12 months from February 2021 to January 2022, and sampled every 15 minutes. The dataset consisted of 15-minute demand for the entire state of Rajasthan. In order to take the weather parameters into account, the data of Jaipur city was taken as a normalized value for the state of Rajasthan. Further, weather data for the city of Jaipur was scraped from various sites and later processed for missing entries. The data of weather was available for every 30 minutes and hence, assuming that weather conditions do not fluctuate every 15 minutes, data was processed accordingly. During data preprocessing, the data was standardized using Min-Max scaling, while features such as Time, Day, Month and Year were extracted from the Date feature. Another feature added was of Weekday, which was given by the value 1 while weekend was given value of 0. In total, there were 9 features namely, Day, Month, Year, Time, Temperature, Wind, Humidity, Pressure and Weekday. These features are a prerequisite for Short Term load forecasting. The weather conditions of the area under consideration changes throughout the year affecting the load demand, and hence, it was mandatory to consider this as a feature for the dataset.

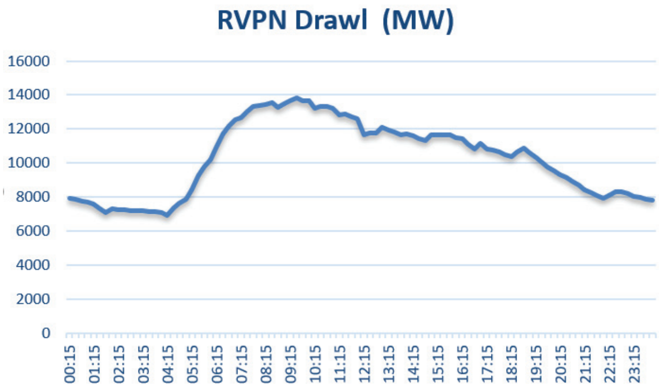

Fig. 1: Load demand curve of Rajasthan state for 01-Dec-2021 at intervals of 15 minutes.

III. TECHNIQUES USED

The problem of short-term load forecasting as seen from the viewpoint of Machine learning is that of Regression. Two different techniques have been used – Deep Neural Networks and Support Vector Machines.

A. *Support Vector Machines*

Support Vector Machines can be defined as supervised learning models having specific learning algorithms that analyze data and then use the data for classification and regression analysis [2]. In case of Support Vector Regression, a straight line fits the data and it is called as Hyperplane. The SVM algorithm aims to find a hyperplane such that the data points present in the n-dimensional space are classified distinctly. The term Support vectors defines the points present closest to the hyperplane on each side. Support vectors are the backbone of the Support Vector Machine and hence are used to build the SVM.

The Support Vector Regression works on the same principle as Support Vector Machine.

The set of training samples:

$D = \{(x_1 , y_1), (x_2 , y_2), \ldots , (x_m , y_m)\}, y_i<\varepsilon , y_i>-\varepsilon$.

The equation form of the trained SVR model is $f(x)=w^Tx+b$.

The type of hard-margin SVR optimization problem is expressed in the following form:

$$\max(\mathrm{w},\mathrm{b})\ \min(x_i, i=1,2 \ldots N)\ \frac{1}{\| w \|} \mid w^T x_i + b \mid$$
$$s.t \begin{cases} w^T x_i + b < \varepsilon\ ,\ y = \varepsilon \\ w^T x_i + b > \varepsilon,\ y = -\varepsilon \end{cases} \quad (1)$$

To have an effective SVR model, the selection of the kernel is an extremely important criteria. The selection of kernel changes according to the prediction problem and dataset

There are four different kernels available

- Linear: Capable of rapidly resolving linear problems with comparatively few parameters.
- RBF: It more stably processes nonlinear problems[3]
- Polynomial: It is useful for solving nonlinear problems.
- Sigmoid: Here, the SVR is a perceptron neural network with multiple layers involved. It prevents falling into the local optimum, in case of a convex quadratic optimization problem.[4]

B. *Deep Neural Network*

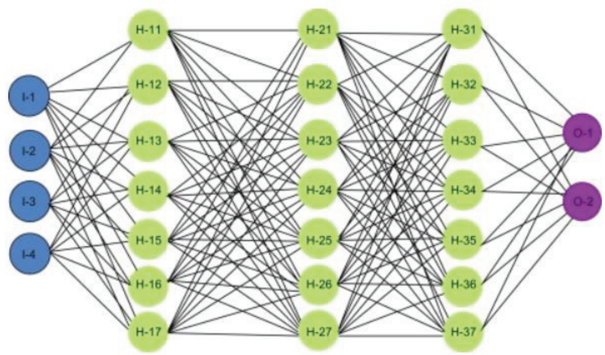

Fig. 2: Deep Neural Network having Three Hidden Layers [6]

As the name suggests DNNs are neural networks involving more than one hidden layer inspired by homo sapiens and the movement of the network is only in the forward direction. Each layer consists of many neurons. Neuron contains a value between 0 to 1 called as activation. Activation of a layer depends on the activation of previous layer. These neural networks can prove to be good for both classification and regression.

During the use of the Deep neural network as a classifier, input nodes match the input features and output classes are matched by the output nodes.

Weights, bias, backpropagation and nonlinear activation are some of the most important aspects in case if DNN.

Each *pre-activation function* $\mathbf{a}^{(l)}(\mathbf{x})$ is typically a linear operation with matrix $\mathbf{W}^{(l)}$ and bias $\mathbf{b}^{(l)}$, which can be combined into a parameter θ:

$$a^{(l)}(x) = W^{(l)}x + b^{(l)},$$
$$a^{(l)}(x`) = \theta^{(l)}x`, L = 1$$
$$a^{(l)}(h^{`(l-1)}) = \theta^{(l)}h^{`(l-1)}, L > 1$$

The "dash" notation x` indicates that 1 has been appended to the vector $\mathbf{x}$. Hidden-layer activation functions $\mathbf{h}^{(l)}(\mathbf{x})$ often have the same form at each level.

To achieve the required output, weights and biases need to be adjusted in such a way that the activation function works to activate the nodes hidden in the network. Random initialization us done for weights and bias. After initialization, tens of thousands of inputs are taken for the network training. The weights and bias are adjusted by back propagation of error, so that hidden neurons are activated by suitable values. A unique set of weights and biases are used to identify a specific output, these are known as the kernel or feature set.

In cases when the traditional statistical methods fail, Deep Neural network comes to the rescue.

C. Recurrent Neural Network

In short, also called as RNN, a recurrent neural network can be understood as a type of sensory network in which nodes create an undirected or digraph using the temporal links. This enables it to show dynamic behaviour that is only present for a short period of time. As a result, applications like unsegmented, linked patterns, speech recognition and handwriting recognition might benefit from them. RNNs allow to run and process arbitrary input sequences and programs. RNNs, which are based on feedforward neural networks, can handle input sequences of varied lengths by using their internal state (memory).

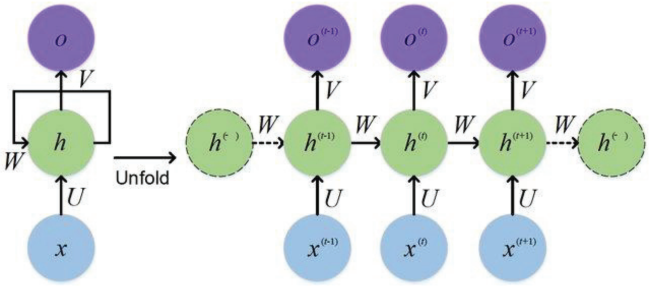

Fig. 3: The standard RNN and the unfolded RNN[8]

Although, a CNN or convolutional neural network tends to show temporal dynamic behaviour akin to RNN, a CNN has a finite impulse response whereas RNN refers to networks with infinite impulse response. This infinite response network can be defined as a directed acyclic graph which cannot be unrolled. However, a finite impulse response network can be unrolled and also replaced with FNN or feedforward neural network.

D. Convolutional Neural Network

A Convolutional Neural Network (CNN) is a Deep Learning algorithm which can take images as input. an image consist of many pixels and each block contain a value between 0 and 1 (weight) which signifies the occurrence of object which helps to classify object from each other. The number of iterations and the pre knowledge required to execute CNN is much lower than other algorithms like support vector and image segmentation. CNN is inspired from human brain neural network and consist of a number of hidden layers. With the time, it keeps improving its performance and hence give sharp results.

The structure of CNN can be interpreted in a similar way to the communication pattern of neurons in the human brain and is stimulated by the Visual Cortex organization. Individual neurons pass the information in a limited region or area of view called receptive field. The collection of such fields extends to cover the entire visible area. CNN is considered to be more powerful than RNN as CNN is a type of feed-forward artificial neural network transmitting with a multilayer perceptron designed to use small amounts of pre-processing.

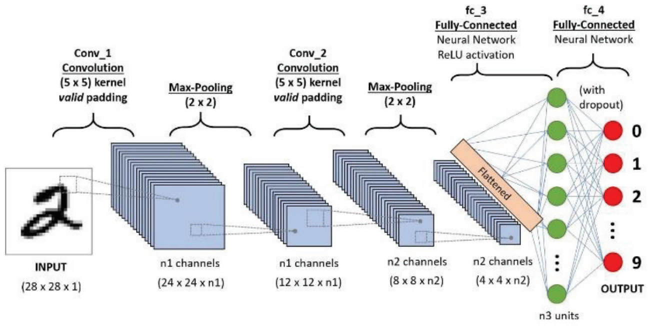

Fig. 4: A CNN sequence to classify handwritten digits [7]

IV. RESULTS AND COMPARISONS

A. Models Description

The **DNN model** was built using Keras library with TensorFlow framework. It is a Sequential model and uses ADAM optimizer which is based upon Gradient Descent algorithm. A basic DNN with 2 layers with 4 and 6 neurons respectively. It was trained and evaluated using MSE and MAE. After basic DNN model, 4 more models were made and compared.

The specifications are:

- Number of neurons were increased in Model 2.
- Sigmoid function was used as activation function in Model 3.
- Number of layers were increased in the fourth model.
- Dropout layer was added in Model 5.

SVR on other hand was built using Scikit-learn library. All the for different kernels were used and compared. This was evaluated using MAPE metrics.

B. Error metrics for evaluation

To assess the accuracy of the forecasting model, two metrics

are used: mean absolute error (MAE), mean

square error (MSE). The formulae are as follows:

$$MAE = \frac{\sum_{i=1}^{N} |y' - y|}{N}$$

$$MSE = \frac{\sum_{i=1}^{N} (y' - y)^2}{N}$$

where, y' = predicted output

y = actual output

N = total number of data points

MAE = Mean Absolute Error

MSE = Mean Square Error

C. *Final Evaluation and Figures*

Following is the loss graph comparing each model during training dataset using MSE Metrics.

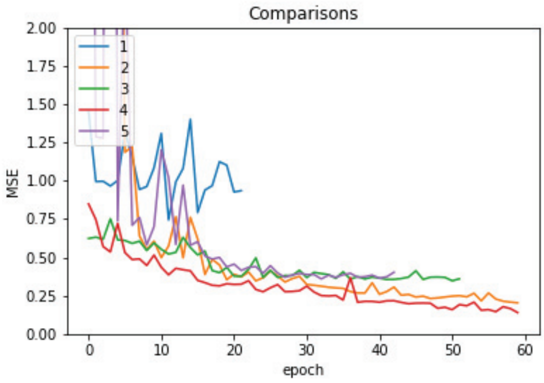

Fig. 5: Comparison of all models based on MSE metrics

As seen, Model 4 performs the best amongst all other trained models according to MSE.

Next, the loss graph below compares each model during training dataset using MAE Metrics.

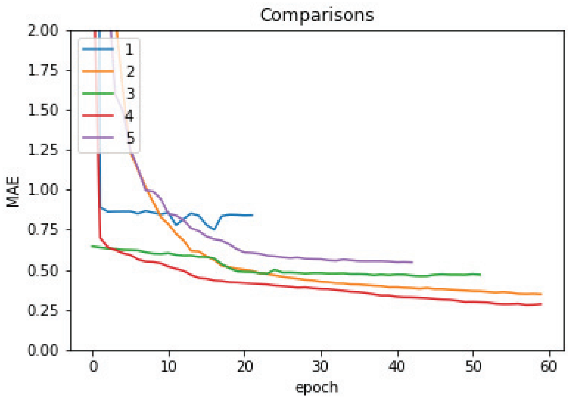

Fig. 6: Comparison of models based on MAE metrics

As seen, Model 4 performs the best amongst all other trained models according to MAE.

The following table gives a brief on all DNN models trained-

TABLE I. RESULTS FOR DNN MODELS

Model Description		MSE	MAE
Activation Function	**No. of Neurons**		
All relu	4-6-1	2.66	1.36
All relu	100-200-1	0.21	0.332
Sigmoid- relu-relu	100-200-1	0.35	0.461
All relu	100-500-300-200-1	0.12	0.27
Relu	Dropout-100-200-1	0.39	0.54

The following graphs show the performance of each model for prediction of 7 days from 25th Jan' 22 to 31st Jan' 22.

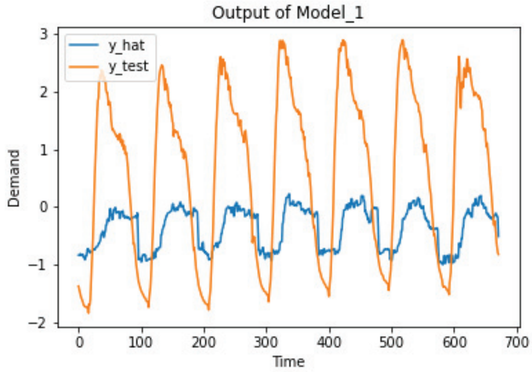

Fig. 7: Prediction of Model_1

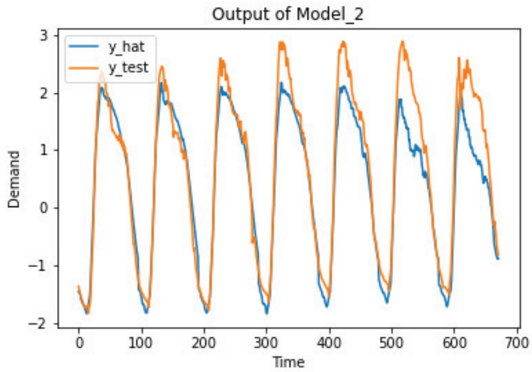

Fig. 8: Prediction of Model_2

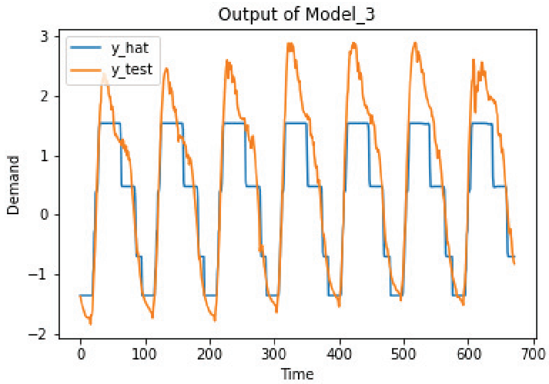

Fig. 9: Prediction of Model_3

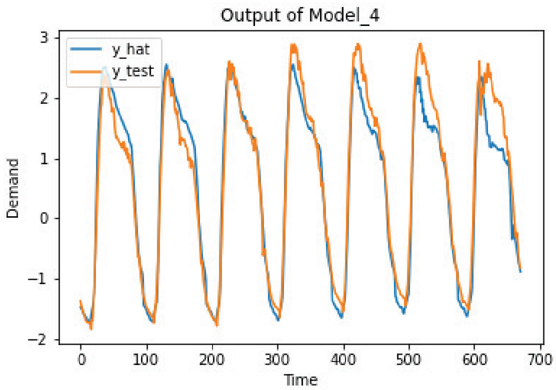

Fig. 10: Prediction of Model_4

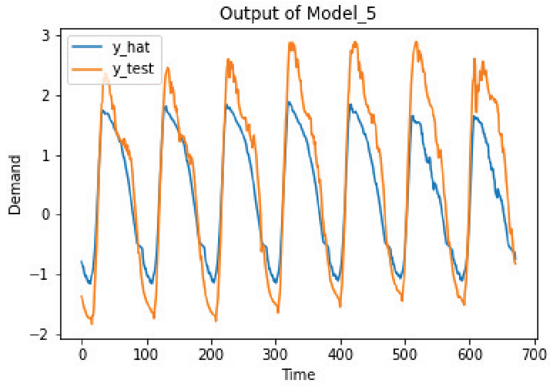

Fig.11: Prediction of Model_5

For SVR, the following table shows the performances with different kernels-

TABLE II. RESULTS FOR SVR MODELS

Kernel used	MAPE %	ACCURACY %
Linear	23.068	76.93
RBF	23.497	76.50
Polynomial	19.835	80.16
Sigmoid	24.551	75.45

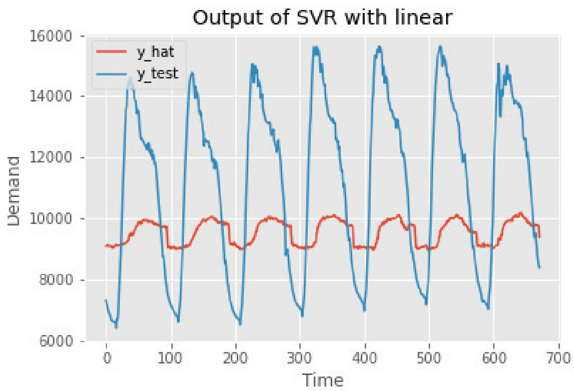

Fig. 12: Prediction of SVR with linear kernel

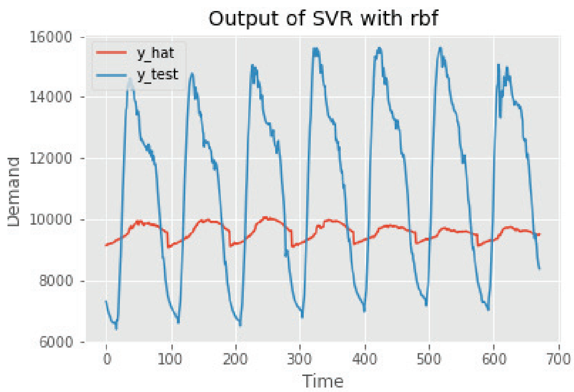

Fig. 13: Prediction of SVR with RBF kernel

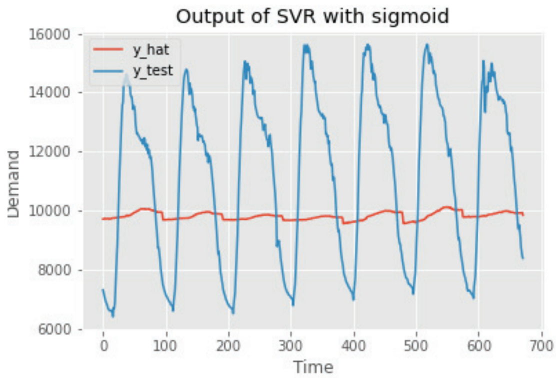

Fig. 14: Prediction of SVR with sigmoid kernel

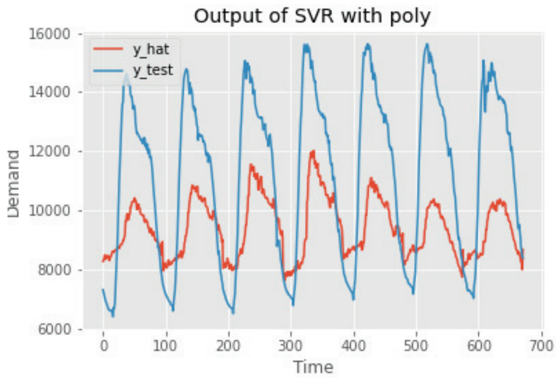

Fig. 15: Prediction of SVR with poly kernel

Among SVR, polynomial kernel gives the best result, yet inferior to DNN model 4. As can be seen, DNN gives much better results compared to SVR. Improving hyperparameters and data features will provide much better results.

V. FUTURE SCOPE

We aim to increase the number of features in the upcoming models, which will include the information such as demand on same time at previous week and notable special events throughout the year. Also, we aim to increase the size of dataset. In case of SVR, some optimization techniques will be used. Parameter tuning is required in case of DNN model.

ACKNOWLEDGEMENT

All authors would like to express special thanks to Dr. Om Prakash Mahela for his continuous support to complete this research and providing data of recorded peak loads in Rajasthan.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (references)

[2] https://towardsdatascience.com/unlocking-the-true-power-of-support-vector-regression-847fd123a4a0

[3] P. Jiang, F. Liu and Y. Song, "A hybrid forecasting model based on date-framework strategy and improved feature selection technology for short-term load forecasting", Energy, vol. 119, pp. 694-709, Jan. 2017.

[4] Z. Tan, J. Zhang, Y. He, Y. Zhang, G. Xiong and Y. Liu, "Short-Term Load Forecasting Based on Integration of SVR and Stacking," in IEEE Access, vol. 8, pp. 227719-227728, 2020, doi: 10.1109/ACCESS.2020.3041779.R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[5] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[6] H.S. Das, P. Roy, “A Deep Dive into Deep Learning Techniques for solving spoken language identification, Problems” 2019, Intelligent Speech Signal Processing Academic Press, pp. 81-100.

[7] S.K. Sharif, C.S. Varshini, G. Sreekanth, G. Hruday, M. Chandu, “Sign Language Recognition” International Journal of Engineering Research & Technology, Vol. 9, Issue 05, pp. 1132-1137, May – 2020.

[8] W. Feng, N. Guan, Y. Li, X. Zhang and Z. Luo, "Audio visual speech recognition with multimodal recurrent neural networks," 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 681-688.