HouseHold Power Consumption Prediction

CS725 - 2022 Course Project

Bhavani Shankar (22M0743), Sai Kumar Atluri (22M0745), Raghava Mudddu (22M0764), Pavan Kumar Yalavarthi (22M0776), Naradasu Hemanth Kumar (22M0777)
Indian Institute of Technology, Bombay

ABSTRACT

We are predicting Household Power Consumption using the Individual Household Electric Power Consumption Dataset obtained from UCI Machine Learning Repository. We have implemented five models: Linear Regression, Support Vector Regression, Random Forest Regression, Feed Forward Neural Network, LSTM and we have also implemented a paper where the goal has been achieved using a CNN LSTM. The dataset we have used is Individual Household Power consumption dataset obtained from UCI Machine Learning Repository.

KEYWORDS

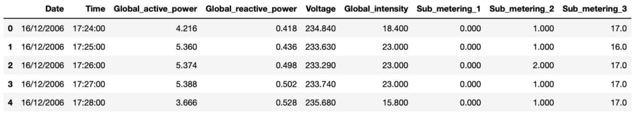
Power Consumption, Data Analysis, Model, Linear Regression, Support Vector Regression, Feed forward Neural Network, LSTM, CNN LSTM, MSE, RMSE

1 DATA SET

The data set for our project was taken from UCI Machine Learning Repository: Individual household electric power consumption Data Set .

The data set contains 2075259 measurements gathered in houses located in Sceaux (7km from Paris, France) between December 2006 and November 2010 (47 months). Different electrical quantities and some sub-metering values are available.

Below is a snapshot of the data set.



	Date	Time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
0	16/12/2006	17:24:00	4.216	0.418	234.840	18.400	0.000	1.000	17.0
1	16/12/2006	17:25:00	5.360	0.436	233.630	23.000	0.000	1.000	16.0
2	16/12/2006	17:26:00	5.374	0.498	233.290	23.000	0.000	2.000	17.0
3	16/12/2006	17:27:00	5.388	0.502	233.740	23.000	0.000	1.000	17.0
4	16/12/2006	17:28:00	3.666	0.528	235.680	15.800	0.000	1.000	17.0

The descriptions of various variables in the dataset are:

(1) **Date**: Date given in format dd/mm/yyyy
(2) **Time**: Time given in format hh:mm:ss
(3) **global_active_power**: It is the household global minute-averaged active power (in kilowatt)
(4) **global_reactive_power**: It is the household global minute-averaged reactive power (in kilowatt)
(5) **Voltage**: minute-averaged voltage (in volt)
(6) **global_intensity**: It is the household global minute-averaged current intensity (in ampere)
(7) **sub_metering_1**: energy sub-metering No. 1 (in watt-hour of active energy). It corresponds to the kitchen electronics, containing mainly a dishwasher, an oven and a microwave (hot plates are not electric but gas powered).
(8) **sub_metering_2**: energy sub-metering No. 2 (in watt-hour of active energy). It corresponds to the laundry room electronics, containing a tumble drier, a washing machine, a light and a refrigerator.
(9) **sub_metering_3**: energy sub-metering No. 3 (measured in watt-hour of active energy). It corresponds to an electric water-heater and an air-conditioner.

2 DATA ANALYSIS

Before Data analysis, data set is cleaned by filling the NA values. Now we have to fix on a appropriate dependent variable. As our goal is find the house hold power prediction, dependent variable is either global active power or global reactive power.

Global reactive power is the total power lost due to all the appliances in a household, and naturally it should be randomly distributed, i.e, it should not be dependent on features like voltage and global intensity (current). And the data cements the fact:

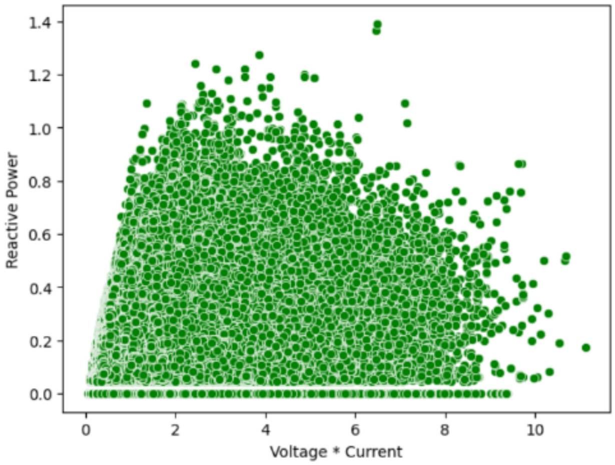


Figure 1: Reactive power vs VoltagexCurrent

Global active power should now be proportional to voltage * current, otherwise we can conclude that the data is erroneous because the data does not comply with the laws of physics (power = voltage * current).

We also found that the sub metering values are a part of the active power. We plot active power vs Sum of the sub meterings:

Now we investigate whether time has any impact on the active power.

We have only two columns: date, and time in hh:mm:ss. And after due investigation of the data we can find that there are only 16

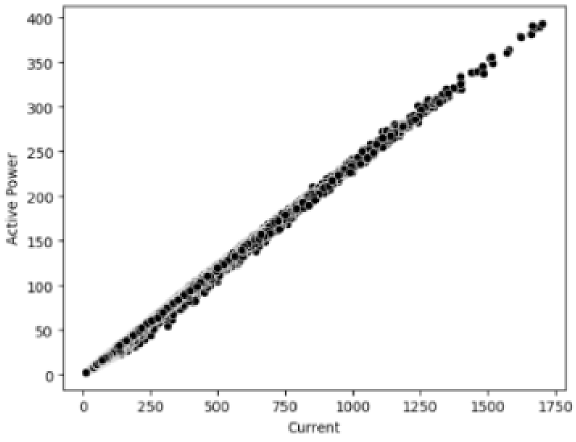


Figure 2: Relation between active power and current(amp)

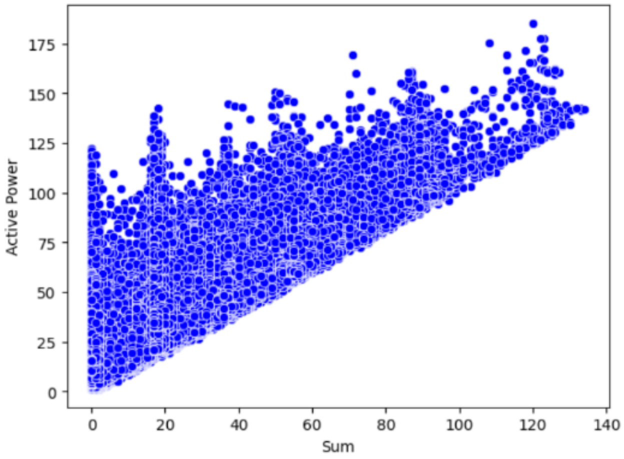


Figure 3: Active power vs Sum of the submeterings

records of 2006, which might deceive our analysis, if we want some statistics where the x-axis is 'year'. Hence we drop those records for analysis purposes. From the above graphs, we observe that Global Active power, (hence, power consumption) is highest at about 9 PM, with the second peak at 9 AM.

On a monthly scale, we notice that August has, on avg, the lowest power consumption. We can see clearly that less power is consumed in June, July, August, September, whereas more power is consumed during December, January, February. We can also see the power consumption starts decreasing almost after March till August - September post which the power consumption increases gradually. Annual Trends show that overall average power consumption is decreasing somewhat in this Time-Period.

Overall, we can say that time is a huge factor in determining power.

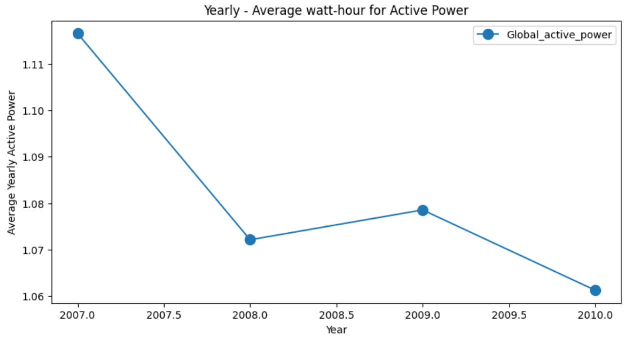


Figure 4: Power Consumption in different years(2007-2010)

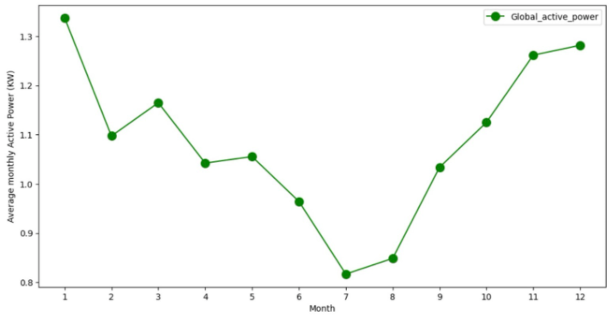


Figure 5: Power consumption in various months of the year

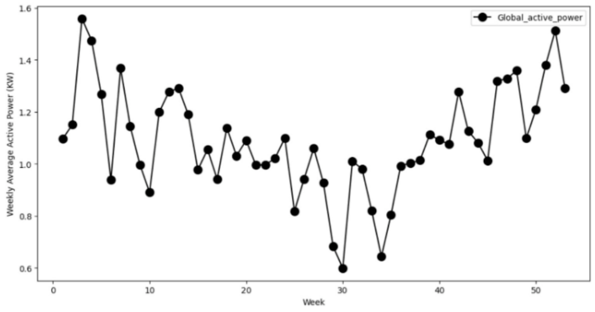


Figure 6: Power Consumption in week

From the above analysis, we have fixed global active power as the dependent variable. And we have also seen that time and the three sub meterings along with Voltage*global intensity can be considered as features for the model.

3 MODELS

We have implemented various models: Linear Regression, Support Vector Regression, Feed Forward Neural Network, LSTM and CNN LSTM. The details and results are shown.

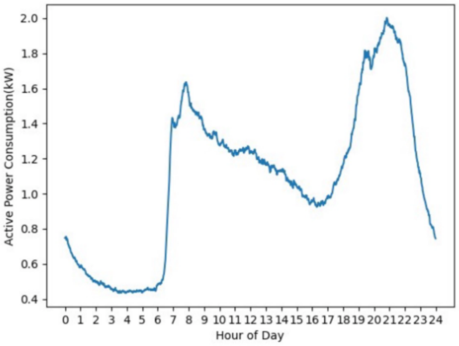
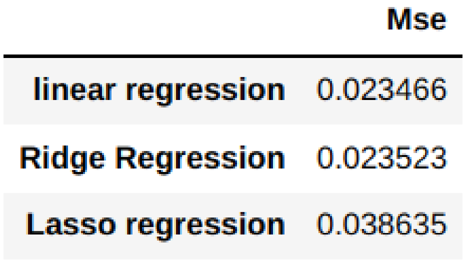


Figure 7: Power Consumption in a day

3.1 Linear Regression

We have trained the model with linear regression taking first 70% of the data and then tested the model against rest of the 30% of the data. Later ridge(L2 normalization) and Lasso(L1 normalization) variations are added to the model and trained again.These are results obtained:



	Mse
linear regression	0.023466
Ridge Regression	0.023523
Lasso regression	0.038635

Figure 8: MSE in linear regression

We can see this relation between Active power and voltage x current :

So we have done feature engineering to an extra column in our data set, i.e VI, which is product of voltage and current to capture that relation. After modifying the data set and trained the model again with Linear regression along with Ridge and Lasso variation, these are the results obtained:

3.2 Support Vector Regression(SVR)

Support vector regression is the most common application form of SVM and SVR uses the very same principle as SVM. SVR finds the best-fitting hyperplane that has the maximum number of points on the boundary with the margin ϵ . Svr and linear regression solve the same problems, but svr explores the non-linear relations in the data by not restricting itself to linear models, so svr works on both linear and non-linearly distributed data. SVM and SVR are popular for handling the non-linear data by using a kernel trick.

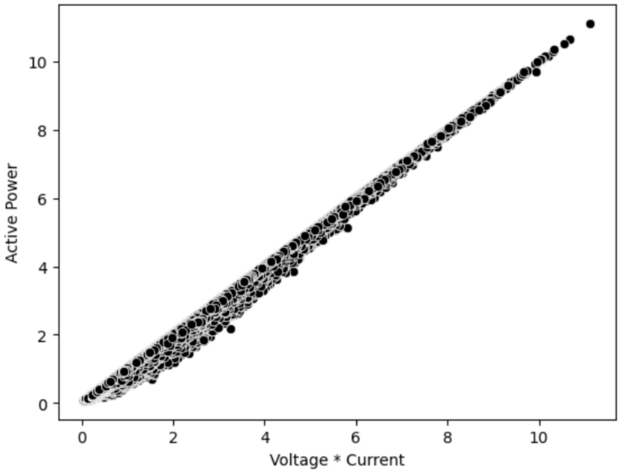
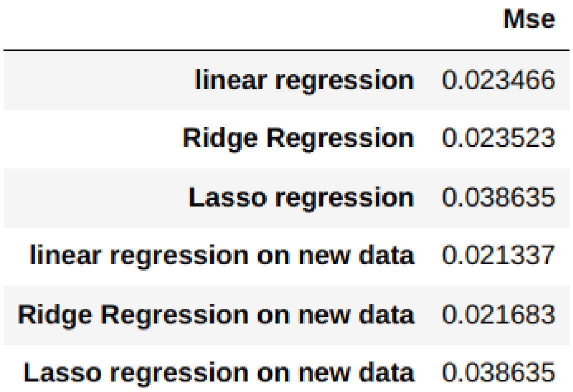


Figure 9: Active power vs Voltage x Current



	Mse
linear regression	0.023466
Ridge Regression	0.023523
Lasso regression	0.038635
linear regression on new data	0.021337
Ridge Regression on new data	0.021683
Lasso regression on new data	0.038635

Figure 10: MSE in Linear Regression after adding Voltage x Current new Column

we have applied Linear kernel to the data to predict the active power from the given dataset, Linear kernel takes two hyperparameters ***C*** and epsilon(ϵ), after fine tuning of the hyperparameters linear kernel has given the mse of 0.023, results are far better compared to what was anticipated surprisingly linear kernel worked well with this data. But there are other kernels ,they can exploit the non linear relationships in the data to get a better model

Radial Basis Function(RBF) is a general purpose kernel which can be used when data is non-linear or distribution of the data is not known. RBF kernel have the three hyperparameters epsilon(ϵ),***C*** and gamma(σ).After hyperparameter tuning rbf kernel gave much improved results of mse 0.018 as anticipated.

3.3 Random Forest Regression

Random Forest is made of multiple individual Regression Trees. The output of each tree was taken into consideration by calculating their average.

No. of trees and depth of each tree are the hyperparameters here.

The number of trees in the forest has been fixed to 20 after observing the performance, and we did a grid search on the depth.

We noticed overfitting after depth 20. Hence that was chosen as the depth.

MSE was 0.56 on test data (when reactive power was included and voltage x current was not considered.) When reactive power was dropped as a feature and voltage x current are included, RMSE drops to 0.009.

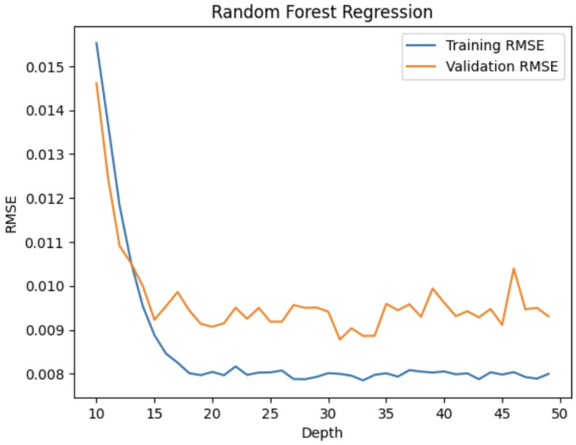


Figure 11: Training vs validation loss for RFR

We have picked up some estimators and we have noticed that the initial splits were on "Global_intensity (Current)" and "Voltage". We have also noticed the sub_metering3 was also among the initial splits and this says that submetering3 is an important feature for power consumption - submetering3 corresponds to power consumption of Air conditioner and water heater. One of the tree is visualised :

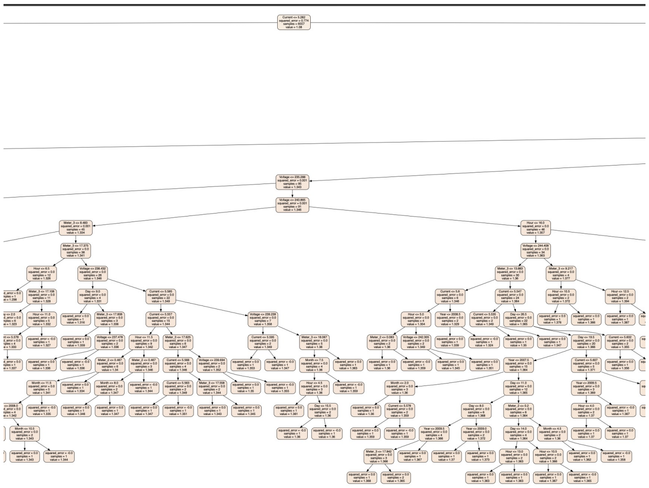


Figure 12: Visualization of Decision Tree

3.4 Feed Forward Neural Network

Regression and Forecasting is done using Feed forward neural network. For forecasting the data has been sampled to mean of an hour. Let's consider y(t) be the value of global_active_ power to be forecasted. The input layer of the feed forward neural network is given seven features.

The input features are are global_active_power(t-1), global_ reactive_power(t-1), voltage(t-1), global_intensity(t-1), submetering_1(t-1), submetering_2(t-1), submetering_3(t-1). Training was done using 3 years data and testing using 1 year data. After performing various hyper parameter tuning the best achievable Mean squared loss was 0.55. The below graph depicts how the training loss and validation loss is varying on each epoch.

The parameters for which the feed forward neural network gave the best loss are, the number of hidden layers = 3, number of neurons in each hidden layer = 32, optimizer algorithm used is Adam, activation function used is ReLU, and L2 regularization with lambda value 0.0001.

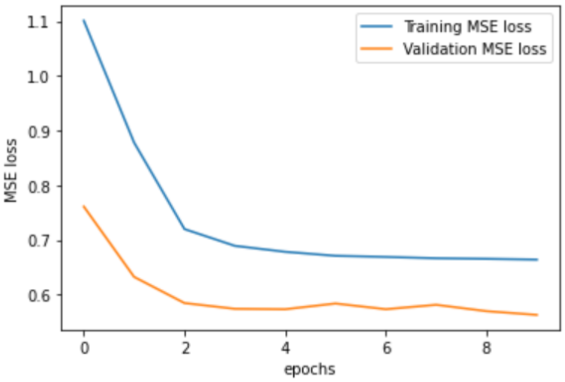
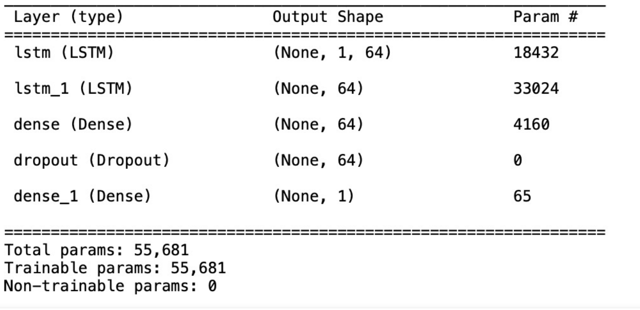


Figure 13: Training vs Validation Loss for FFNN

3.5 RNN and LSTM

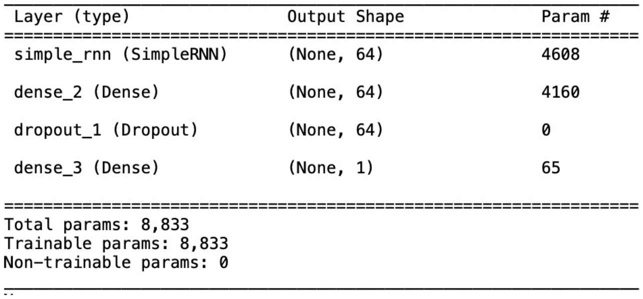
As temporal information is vital for this data, RNN and LSTM models would work better when compared to feed forward neural networks. The following are the models of RNN and LSTM we have implemented (with reactive power included and voltage x current not considered):



```
 Layer (type)                 Output Shape              Param #
=================================================================
 lstm (LSTM)                  (None, 1, 64)             18432

 lstm_1 (LSTM)                (None, 64)                33024

 dense (Dense)                (None, 64)                4160

 dropout (Dropout)            (None, 64)                0

 dense_1 (Dense)              (None, 1)                 65

=================================================================
Total params: 55,681
Trainable params: 55,681
Non-trainable params: 0
```

Figure 14: LSTM Model

The MSE of RNN is about 0.52 and LSTM is about 0.49. LSTM worked slightly better when compared to RNN.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 simple_rnn (SimpleRNN)      (None, 64)                4608

 dense_2 (Dense)             (None, 64)                4160

 dropout_1 (Dropout)         (None, 64)                0

 dense_3 (Dense)             (None, 1)                 65

=================================================================
Total params: 8,833
Trainable params: 8,833
Non-trainable params: 0
_________________________________________________________________
```

Figure 15: RNN Model

3.6 CNN LSTM

We also have implemented a paper (Predicting residential energy consumption using CNN-LSTM neural networks - ScienceDirect). In the paper, the goal has been achieved by using a CNN LSTM. The features in the dataset are correlated and hence CNN would be able to exploit the relationship between the features. The goal of the CNN LSTM Model is to use CNN Layers to extract features from a variety of variables that influence the prediction of energy consumption. LSTM layers, which remember the trend factor of electrical energy consumption, can be fed the output of these CNN layers, which would have extracted key power consumption aspects and eliminated the noise. Passing the target variable's prediction to a fully connected layer from the final LSTM layer's output. As a result, the objective of CNN LSTM is to forecast the "spatial correlation" of the variables with time information in multivariate time series data on power consumption.

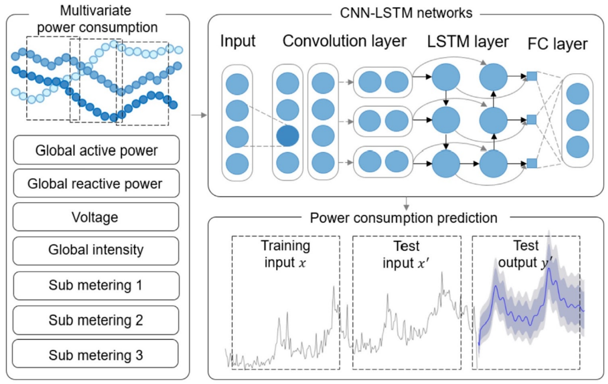


Figure 16: CNN LSTM Overview

A sliding window algorithm is used to preprocess the data that the CNN LSTM uses. Convolution layers and Pooling Layers of the CNN Layer are used to extract the spatial characteristics of the multivariate time series variable. These are next passed with noise removed, to the LSTM Layer. Irregular time information is modelled by the LSTM Layer. Finally, the CNN-LSTM method can generate predicted electrical energy consumption in a fully connected hierarchy.

The proposed CNN LSTM Architecture:

Type	# Filter	Kernel size	Stride	# Param
Convolution	64	(2, 1)	1	192
Activation (ReLu)	–	–	–	0
Pooling	–	(2, 1)	2	0
Convolution	64	(2, 1)	1	8256
Activation (ReLu)	–	–	–	0
Pooling	–	(2, 1)	2	0
TimeDistributed	–	–	–	0
LSTM (64)	–	–	–	180,480
Activation (tanh)	–	–	–	0
Dense (32)	–	–	–	2080
Dense (60)	–	–	–	1980
Total number of parameters				192,988

Figure 17: CNN LSTM Model

To validate the CNN-LSTM's operation principle, we looked within the model. The input window for power consumption is 60 minutes. Through two convolution and pooling layers, the input window is cut four times. The LSTM layer receives the output result as an input. Although there is less noise, the temporal trend is essentially unchanged as evidenced by the power consumption through the convolution layer. Through model internal analysis, we have shown that CNN-LSTM modelling is carried out with little information loss.

The following figure shows that the output of CNN (which is supposed to be the input to LSTM) preserves the temporal information, which important to the LSTM.

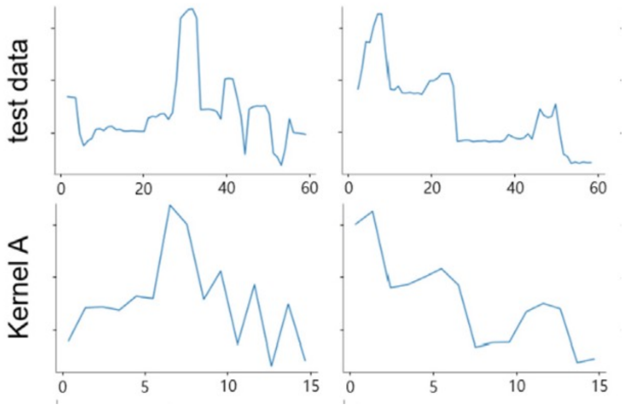


Figure 18: CNN Kernel Output

The MSE on the test data is 0.34 for hourly forecasting. Reactive power has not been dropped from the feature set.

The class activation map is used to obtain a weighted activation mapgenerated for each electric energy consumption window. It also localizes the receptive field of CNN. We can see that sub_metering1, sub_metering2 and sub_metering3 are important variables through the average value of the class activation score. In particular, submetering3 corresponds to an electric water heater and an air conditioner. The use of heating and cooling systems has a significant impact on energy consumption demand and is an important parameter in predicting consumption. The predicted vs actual for hundred hour forecating is given:

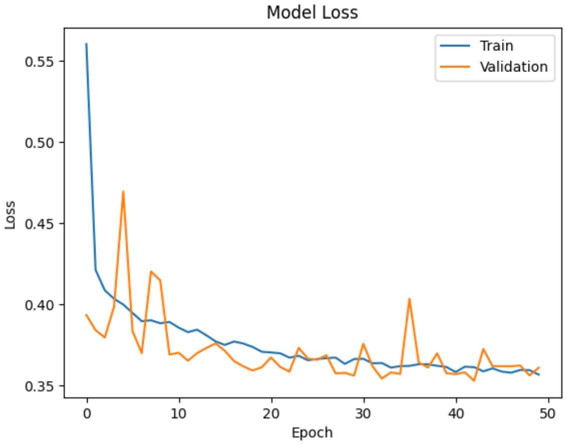


Figure 19: CNN Model Loss

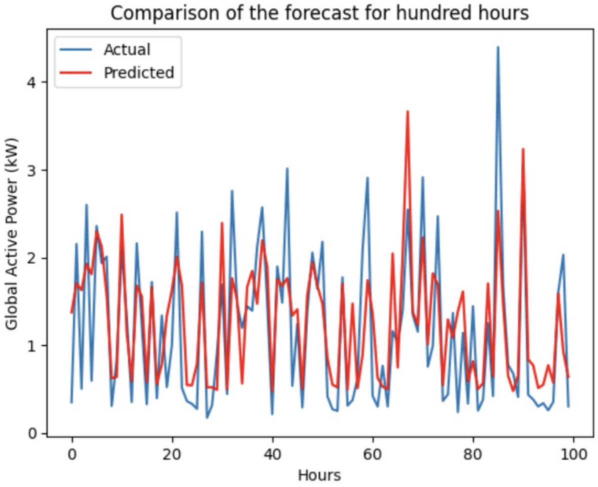


Figure 20: Comaparision of Forecast

4 CONCLUSION

We have implemented Linear Regression, Support Vector Regression, Random Forest Regression, FFNN, RNN, LSTM and CNN LSTM on the Household Power Consumption dataset. We have considered "Global_active_power" as the target variable. We have found out that even though there is a linear relationship between Voltage x current and Active Power, there is certain amount of interference caused by "Global_Reactive_Power". If Reactive Power is not dropped for training models, the linear relationship is beautifully captured by all the models. Without dropping Global Reactive Power, our CNN LSTM model works well among all the models with an MSE on test data of 0.39.

REFERENCES

[1] Tae-Young Kim and Sung-Bae Cho, *Predicting residential energy consumption using CNN-LSTM neural networks*, Energy, 182(2019), (Predicting residential energy consumption using CNN-LSTM neural networks - ScienceDirect).