Power Consumption Prediction

Krishna Prasanna
kbhamid2

Pururaj Jyotindra Dave
pdave2

Sharva V. Hiremath
shirema

Anirudh Nori
nnori

1. BACKGROUND

1.1 PROBLEM STATEMENT

Forecasting the electric power consumption for individual households is an important aspect in developing sustainable and efficient energy consumption strategies, and it has huge implications on capacity and resource planning,distribution grid-load planning, devising plant operations and maintenance, and setting revenue goals.

On the production side it alleviates inefficient power production issues by allowing these facilities to plan their operations around the peak and idle demand, thereby reducing waste across the grid. Knowing the dynamics of the power consumption helps distributors devise smart-grids capable of handling fluctuations in load demand and supply, economically feasible investment strategies for future generation and transmission, while reliably supplying all consumers with the required energy. Knowing the patterns of usage balances the amount of power needed and the power produced which will in turn stabilises the costs per unit of power during different periods of time - thereby reducing the costs of electricity on consumer's end.

But electric power forecasting is a challenging issue because of the variability in power consumption across households, time-series nature of the data and changing landscape of energy sources. The forecasting periods must also be adjusted to different applications based on end-users.

In is project, we aim to predict/estimate the electric consumption at different time scales i.e. daily, weekly, monthly and compare the performance of different models across selected metrics.

1.2 RELATED WORK

There has been a lot of work done in being able to predict electric power consumption. Typically they involve techniques such as ARMA (Autoregressive Moving Average), ARIMA (Autoregressive Integrated Moving Average) popular in fields like finance where they frequently deal with time series data [1],[2]. SVR (Support Vector Regression) in combination with DE optimization algorithms for parameter selection has also shown promising results for some use cases of time-series data in the areas of energy consumption forecasting [3].

Though these approaches perform moderately well their fundamental limitation is their inability to fully accommodate unexplained inputs which may be contributing to the variable being predicted or handling outliers which cause a sudden spike or dip in the data. Another limitation is their inability to take into account the trends that underlay the data. One of the methods that overcomes this limitations is SARIMA (Seasonal Autoregressive Integrated Moving Average) which take into account the seasonal and cyclical trends in the data [4]. SSVR (Seasonal Support Vector Regression) is another proposed technique which exploits the decomposition techniques possible for time-series data, thereby outperforming simple SVR with GA [5].

Contemporary works on this problem make use of deep learning approaches such as LSTM (Long Short-Term Memory) and CNN (Convolutional Neural Network) . These approaches have the benefit of not only looking at the target variable but also the predictor variables. The work done by [6],[7] and [8] reveals that using a deep learning approach improves the prediction accuracy as these models are better quipped to handle outliers and also take into consideration the context for the behaviour by looking at different variables.

The first phase of our project involves implementing non-deep learning machine learning models with ARIMA and SVR to establish baseline values for our evaluation metrics, and further improve those values by exploring LSTM models in our second phase.

2 METHODS

2.1 APPROACH

We first explain our approach and then the rational behind our approach:

Webscraping

A utility program that scrapes weather data pertaining to the location of this household and duration of forecasting to enhance our dataset was developed. Selenium web driver was used for scraping this data which contains daily weather information.

Preprocessing

The power consumption measurements data is continuous on a minute-level but it has nearly 1.25 percent rows of miss-

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

ing values, and these have been handled by replacing with numerical estimates using forward linear interpolation. The treated minute-level data was re-sampled at daily, weekly and monthly intervals.

Exploratory Data Analysis (EDA)
EDA consists of univariate and bi-variate analysis which was implemented using matplotlib.pyplot and seaborn libraries for result visualization.

1. Univariate analysis : The univariate analysis was performed to understand the statistical and distribution trends for pre-processed individual variables. Histograms were created for weather data attributes. Time-series plots were plotted at different time-scales to identify seasonality, cyclicity and data anomalies.

2. Bivariate analysis: Bivariate analysis for predictive modeling consists of computing correlation between weather data attributes (predictor variables) against global active power (response variable). Attributes with weak correlation values are dropped as they do not significantly explain variations in response variable. The same was implemented in this project. Scatter plots validated our corr-plot results.

Autoregressive Integrated Moving Average (ARIMA)
ARIMA model is a class of statistical models for analyzing and forecasting time series data. It is a generalization of the ARMA model with an added Integration part and this makes it capable of handling non-stationary data.

This model consists of three main aspects which are:

- AR: Autoregression. This uses the dependent relationship between an observation and some number of lagged observations.

- I: Integrated. This makes use of differencing of raw observations (i.e. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.

- MA: Moving Average. This uses the dependency between an observation and residual errors from a moving average model applied to lagged observations.

Each of these components are specified in the model as a parameters 'p','d','q' respectively. A brief description of the parameters is given below:

- p: The number of lag observations included in the model, also called the lag order.

- d: The number of times that the raw observations are differenced, also called the degree of differencing.

- q: The size of the moving average window, also called the order of moving average.

A Dickey-Fuller test was conducted to test for stationarity of the data and in case it was non-stationary we used differencing to stationarise it. Following this, Autocorrelation function (ACF) and Partial Autocorrelation (PACF) plots of the original or differenced data to determine the values of 'p' and 'q' were plotted

Support Vector Regression (SVR)
The support vector regression technique , proposed by Vapnik[9] is based on the structured risk minimization principle (SRM). SVR aims to minimize an upper bound of the generalization error, comprising of training error and a confidence level, rather than empirical error which has only training error, which gives it a better generalization performance over other traditional regression techniques. Given a set of data $(x_i,y_i)_{i=1}^N$ where x_i is input vector of predictors, y_i is the corresponding response value and N denotes the total number of data samples, the regression equation is given as:

$$G = w\phi(x_i) + b$$

where $\phi(x_i)$ denotes the feature of the inputs, w and b are coefficients and G is the predicted value of response variable for given input. The coefficients ((w_i) and b) are calculated by minimizing the regularized risk equation shown below:

$$P(G) = C\frac{1}{N}\sum_{i=1}^{N} L_\varepsilon(y_i, G_i) + \frac{1}{2}||w||^2$$

where C and ε are user-defined parameters. ε is the error margin, which is calculated as the difference between actual values and predicted values from regression equation.ε forms a tube around the actual regression line. The points outside the tube are viewed as training errors. $L_\varepsilon(y_i, G_i)$ is called an ε-insensitive loss function given as:

$$L_\varepsilon(y_i, G_i) = \begin{cases} 0, & \text{if } |y_i - G_i| \le \varepsilon \\ |y_i - G_i| - \varepsilon, & \text{otherwise} \end{cases}$$

The points with forecast values within the ε tube have a zero error and they do not contribute to the risk equation. $\frac{1}{2}||w||^2$ is the regularization parameter.Hence the parameters C is tuned to adjusted the trade-off between the empirical risk and the model flatness (as given by regularization expression).

Long Short-Term Memory (LSTM)
LSTM model is based on RNN architecture. Unlike traditional feed-forward neural networks it has feedback connections and thus could process entire sequences of data. Thus, LSTM models can store data over a period of time. Hence, it is very well suited in forecasting timeseries data by considering the past sequences of data. We can also decide what information will get stored and what will get discarded. LSTM model comprises of series of cells that serve the above purpose. There are three type of gates in each cell that handle that :-

- Input Gate: It is used to update the cell state by considering the new information.

- Forgot Gate: It will decide what information will be discarded and what will be kept.

- Output Gate: It decides the value of next hidden state which has information based on previous inputs and eventually be used in predictions.

2.2 RATIONALE
The suitability of the models implemented is discussed below:

Webscraping
It is intuitive to understand that weather patterns will have a correlation with the power consumption rates. For example, during winters when it is cold there will be a spike in the usage of room heating appliances which will increase the net power usage. By augmenting such information we are able to provide a context to DL-models for the power consumption trends. Thus, we needed to scrape the weather data so that DL-models could accommodate weather effects.

Preprocessing
Short-term, medium term and long-term forecasting of energy consumption are done at daily, weekly and monthly intervals. Forecasting at each level presents its own challenges like robustness to anomalies, handling missing values or incorrect meter readings and capturing effects of seasonality across different intervals. This calls for a need to develop different models for different intervals, with different hyper-parameter values rather than a single generalized model.

Exploratory Data Analysis (EDA)
Exploratory analysis of data gives us a better picture of statistical properties of our attributes and the temporal dynamics of the power consumption and weather trends at different time scales. Correlations between weather data scraped and the power consumption will help us identify factors most affecting the meter readings.

Autoregressive Integrated Moving Average (ARIMA)
ARIMA is a traditional approach to time-series forecasting when data is non-stationary - it uses the auto-regressed values of its target variable to forecast future value estimates.This is beneficial in cases where the predictor variables are unknown or can't be obtained.

Support Vector Regression (SVR)
The use of SVR in time series forecasting is increasing very fast especially in forecasting non-stationary time series. SVR provides the flexibility to set how much error is acceptable in our model. Its goal is to minimize the coefficients more specifically l2-norm of coefficient vector. Thus, it is different from traditional model and does not try to minimize squared error. So, it can solve the problem of over-fitting very effectively. We also used both linear based and RBF-based kernels in SVR model in order to cover both linear and non-linear aspects of time-series data.

Long Short-Term Memory (LSTM)
We have picked this method because these models are better quipped to handle outliers and also take into consideration the context for the behaviour by looking at different variables thereby improving model accuracy.

3 PLAN AND EXPERIMENTS

3.1 DATASETS

The household energy consumption dataset is a multivariate, minute-level, numerical time-series data which contains power consumption measurements gathered in a household between December 2006 and November 2010 (47 months) in the city of Sceaux, France. It was taken from the UCI machine learning repository. There are 2,075,259 measurements with 9 attributes that capture various meter readings. The weather dataset is a multi-variate, numerical time-series data which captures different factors describing the daily weather conditions for Sceaux, France from December 2006 to November 2010. It has 1465 rows and 6 columns.

In this project, only the global active power reading was predicted. For ARIMA models, auto-regressive information on this attribute was used as its predictor variables, whereas for SVR weather data was used as the predictor variables.

3.2 HYPOTHESES

In this project, we compared and contrasted the forecasting performance of the traditional machine learning models against deep learning (DL) models for different time intervals on selected evaluation metrics, with and without external predictor variables. Our reason for doing so was based on three assumptions that we also sought to verify:

- DL methods offer superior performance over traditional methods for time series forecasting because they can handle asymmetries or trends in data, and are robust to anomalies.
- Augmenting weather data improves performance by enabling deep learning models to exploit the external context governing the power consumption trends.
- Medium term energy forecasting is robust to data fluctuations and anomalies as compared to short-term forecasting, and is more effective in exploiting changing weather patterns which may not be captured in detail for long-term forecasting. This also helps power distribution facilities understand demand estimates and set monthly revenue goals.

3.3 EXPERIMENTAL DESIGN

Exploratory Data Analysis (EDA)
Our exploratory data analysis consisted of univariate analysis on all attributes of energy consumption data and weather data, and bivariate analysis of all weather data points as predictor variables and global active power as the response variable.

1. Univariate analysis
    - Histograms for most attributes were right-skewed and multi-modal for energy consumption data. Most attributes for weather data have slight left-skew. The precipitation attribute had taken a single value for all columns and hence was dropped from the list of predictor variables due to absence of variability.
    - Prominent seasonal-trends were observed for all plots with a seasonal window of 1 year. Power consumption was least during the summer months (July, August) and highest during winter months (December, January).Power consumption is highest between 7 am and 9am, between 8 pm and 10pm on a given day. Meter readings were higher during weekends.
2. Bivariate analysis
    - Temperature and dew-point were strongly correlated to one another. Global active power was almost independent of pressure.Hence we dropped dew point,pressure from our predictor attributes.

[Plots for EDA are in the appendix.]

ARIMA

In case of ARIMA resampled power consumption data was used for forecasting. A Dickey-Fuller test was performed prior to check for stationarity for our daily, weekly and monthly data.

Following this the ACF and PACF plots were checked. Once we had the ACF and PACF, we then inferred the necessary values for 'p','d','q' for our models. Three separate ARIMA models were implemented for each time-scale. A description of these models is given below.

The 3 models of ARIMA implemented are:

- **Daily data:** For the daily data we noticed that there was a periodicity in PACF and ACF plots at intervals of 7 units. Hence we used 'p' = 7 and 'q' = 7 for our model. Using Dickey-Fuller test also revealed that the data was stationary so didn't have to use differencing to stationarise the data, so we used 'd' = 0. The configuration of the final model implemented is ARIMA(7,0,7)
- **Weekly data:** For the weekly data we noticed that there was a periodicity in PACF and ACF plots at intervals of 4 units. Hence we used 'p' = 4 and 'q' = 4 for our model. Using Dickey-Fuller test also revealed that the data was stationary so didn't have to use differencing to stationarise the data, so we used 'd' = 0. The configuration of the final model implemented is ARIMA(4,0,4)
- **Monthly data:** For the monthly data it was noticed that there was a periodicity in PACF and ACF plots at intervals of 12 units. Hence we used 'p' = 12 and 'q' = 12 for our model. Using Dickey-Fuller test also revealed that the data wasn't stationary so had to use differencing to stationarise the data, so we used 'd' = 1. The configuration of the final model implemented is ARIMA(12,1,12)

  Since we have approximately 4 year of data,the first 3 years of data was used for training and the last one year of data for testing. The models were evaluated using a scheme called walk-forward validation and the evaluation metrics used were RMSE, MAE and MAPE.

SVR

- Steps before SVR model implementation
  1. Prior to implementing the model, the energy consumption data was resampled at date, week and month levels using summation for aggregation. A similar resampling was performed at week and month levels for our daily weather data using mean as aggregation function.
  2. Dew point and precipitation attributes were dropped for the reasons explained in EDA section.
  3. Resampled weather datasets were combined with global active power at corresponding level of observation to create our modeling dataset at daily, weekly and monthly levels. The weather data variables form predictor variables, and global active power is the response variable which we are trying to predict.
  4. Date column information was added as additional attribute by encoding monotonically increasing constant value for each incremental date. E.g. 1st day of the year takes date constant value of 1, 2nd day takes 2 and so on.
  5. Owing to the time-series nature of our data, data cannot be split into train and test datasets using randomized split.Hence in our project have manually split the first 3 years of data for training and the last one year of data for testing.
  6. After creating train-test splits for predictor and response variables, we have normalized the weather data attributes and response variable.
- SVR model implementation and evaluation
  1. We fit the SVR model for selected intervals using linear and radial-basis function kernels on normalized training datasets.
  2. The free parameters in the model are C and ε. C is the regularization parameter and the corresponding penalty is the squared l2 penatly. Strength of the regularization is inversely proportional to C. ε specifies the epsilon-tube within which no penalty is associated in the training loss function with points predicted within a distance epsilon from the actual value.
  3. Hyper-parameters C and ε were selected by implemented using grid-search based CV on test dataset. The primary evaluation metric for grid-search based hyper-parameter tuning was RMSE score for each grid-cell selection,and adjusted R-squared value the secondary evaluation metric.
  4. The final choice of hyper-parameters is given below, as per this order $(\varepsilon_{rbf}, \gamma_{rbf}, C_{rbf}, \varepsilon_{lin}, C_{lin})$:-
     - **Daily data:** Model used is SVR (0.001, 0.05, 0.1, 0.001, 0.01)
     - **Weekly data:** Model used is SVR (0.01, 0.01, 4, 0.01, 0.01)
     - **Monthly data:** Model used is SVR (0.001, 0.01, 4, 0.001, 4)
  5. SVR model was fit for above parameter selections using sklearn.svm modules, and evaluation metrics RMSE, adjusted R squared, MAPE, MAE were calculated for test data using metrics module of sklearn library. The plots for linear, rbf kernel predictions are in figure 2,3,4 and evaluation metric values are in figure 1.

LSTM

Two major variants of LSTM were implemented, one being a univariate LSTM which uses only the power consumption data to make a prediction and the other being a multivariate LSTM which uses the power consumption data as well as the weather data.

The reasons for implementing two variations is that with the univariate LSTM we wanted to see whether deep learning approaches perform better than traditional approaches when working on the same data (which in this case is the power consumption data) and with the multivariate LSTM we want to check if providing models with additional information about the context for the power consumption helps improve performance.

While the data fed to the two models are different the model configuration used is the same for corresponding time scales. This means for example that model configuration of the univariate LSTM for daily data and the multivariate LSTM for daily data is the same, only the number of features used is different.

Since we have approximately 4 years of data, the first 3 years of data was used for training and the last one year of data for testing. A course and fine search approach was used for setting the hyper-parameters. The data fed to the models was scaled before training and inverse scaled after predicting. The evaluation metrics used are RMSE, MAE and MAPE.

Adding more than one layer didn't improve the performance, which may be due to the fact that there aren't as many instances of data and adding complexity will not help improve the performance.

The details of the model configuration for the different timescales is given below:

- **Daily data:** This model uses an LSTM layer with 70 neurons followed by a dropout layer set at a value of .2 which is then connected to the final output layer. The model is trained for 100 epochs with a batch size of 40 and uses ADAM optimizer with a learning rate of .0001.
- **Weekly data:** This model uses an LSTM layer with 60 neurons followed by a dropout layer set at a value of .2 which is then connected to the final output layer. The model is trained for 70 epochs with a batch size of 5 and uses ADAM optimizer with a learning rate of .0001.
- **Monthly data:** This model uses an LSTM layer with 50 neurons followed by a dropout layer set at a value of .3 which is then connected to the final output layer. The model is trained for 50 epochs with a batch size of 2 and uses ADAM optimizer with a learning rate of .0001.

4 RESULTS

4.1 RESULTS

The results for the models at daily, weekly and monthly time scale are tabulated in the figures 1-4 below. The evaluation metrics used were - Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE).

Some observations about the results

- Amongst traditional predictive models, univariate ARIMA-based predictions have best performance values. The weekly MAPE value of ARIMA is 20% which is almost 60% better than SVM models. However, the RMSE value of ARIMA is higher than SVM based models.
- Linear kernel maps variables to linear space for constructing regressive hyper-plane and RBF kernel maps them to a radial basic function derived curve (on a higher dimensional space). RBF based SVR has marginally higher performance as compared to linear kernel SVR (1.94%) - this result may be an indicator that all our predictor variables may have non-linear interactions between them which can be mapped and regressed on a higher dimensional plane. Further significance tests need to be conducted to establish the validity of this indication.
- For all models the performance improves when we move from short term to long term time scale. This can be attributed to the reduction in the amount of fluctuation in the data as moving to a longer time scale tends to average out the peaks and the dips in the shorter term time scale.

4.2 DISCUSSIONS

- **Deep learning models vs. traditional methods**
  From the results we can observe that the deep learning approaches perform better in general. In particular the multivariate LSTM performs best across all time scales.The MAPE value for LSTM models are 16.3% and 13.27% which is almost half the MAPE of SVM model and 25% less than ARIMA's MAPE. This validates our hypotheses that deep learning approaches perform better than traditional predictive modeling strategies.
- **Augmenting weather data to provide context**
  When we compare the difference in performance between univariate and multivariate LSTM models, it becomes evident that the multivariate LSTM performs better. Multivariate LSTM's MAPE value is 13.27% which is 22.8% lesser than univariate LSTM's MAPE. By setting up the experiments such that univariate and multivariate LSTM models which have the same architecture for the same time scales only differ in the data that they use, we were able to confirm that augmenting weather data helps improve performance since it provides additional information about the context for the power consumption trends.
- **Short term vs. Medium term forecasting**
  The medium term energy forecasting - done at week-level in our project is more robust to extreme fluctuations,unlike daily level forecasting. On an average, weekly data has a 6% lower MAPE score than daily data for LSTM models. Also, weekly forecasting captures changing weather patterns in required detail, unlike long term (monthly fore-casting). There is only a slight improvement of MAPE from medium to long-term forecasting (1%) for LSTM models,at the trade-off of reduced detail in predicted values.

5 CONCLUSIONS

The results validate all three of our initial hypothesis. Performance evaluation scores are best for multivariate LSTM at all time scales, though their difference in performance decreases as the time-scale increases since the fluctuations also decrease.

Some key takeaways from the project are:

1. Time series forecasting for short term duration is hard regardless of the model. In our case though the performance improves when we use additional information like

Model	Daily - Error (RMSE) (MAE) (MAPE)	Weekly - Error (RMSE) (MAE) (MAPE)	Monthly - Error (RMSE) (MAE) (MAPE)
ARIMA	408.4 Kw 312.89 Kw 24.3%	2126.84 Kw 1543.13 Kw 20%	8714.43 Kw 7629.57 Kw 16.86%
SVM_linear	421.51 Kw 332.39 Kw 37.44%	1966.17 Kw 1444.51 Kw 32.85%	6348 Kw 4479.96 Kw 28.09%
SVM_RBF	414.29 Kw 321.96 Kw 34.14%	1928.93 Kw 1378.54 Kw 31.9%	6131.32 Kw 4342 Kw 26.52%
LSTM_univariate	381.23 Kw 292.3 Kw 21.64%	1956.21 Kw 1413.54 Kw 16.3%	6410.5 Kw 6554 Kw 15.41%
LSTM_multivariate	352 Kw 267.6 Kw 19.9%	1687.61 Kw 1177.69 Kw 13.27%	6031.32 Kw 4459 Kw 12.3%

Figure 1. Performance of models for different time scales.

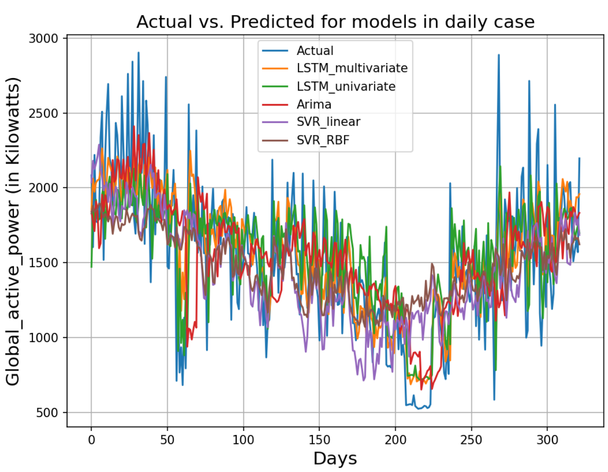

Figure 2. Performance of models for different time scales.

the weather data there is still a gap in what can be explained by weather alone. This gap can be possibly attributed to human factors like the working hours of the residents of the house. Knowing these details can help improve the performance.

2. Most models are able to predict the dips in the data but are unable to handle the sudden peaks. This was an interesting observation about the performance of the models which causes most of the error in the prediction values. This maybe caused by the fact that the peaks in training data are not prominently defined and consequently peak predictions in test data are affected. Whereas the training data has sharp

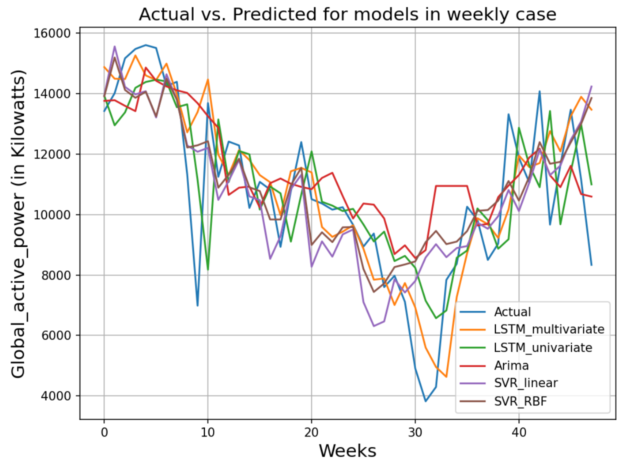

Figure 3. Performance of models for different time scales.

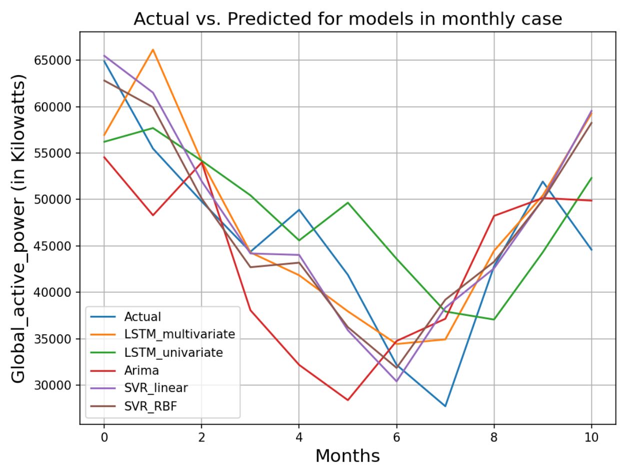

Figure 4. Performance of models for different time scales.

troughs or dips, and test predictions for these dips come close to actual test response values.

Few things that we would have liked to implement are models like a Bidirectional-LSTM and Prophet. The Bidirectional-LSTM would allow the model to look at the future data and learn the weights accordingly which can be useful in our case. Whereas using Prophet might be useful where non-linear trends are fit with seasonality effects.

6 CODE REPOSITORY

The github link for the code is found below.
[Link]

7 REFERENCES

[1] "Time series analysis of household electric consumption with ARIMA and ARMA models." Chujai, Pasapitch, Nittaya Kerdprasop, and Kittisak Kerdprasop. [Link]

[2] “Times Series Analysis Forecasting and Control,”. Box, G.E.P. and G. Jenkins, Holden-Day, San Francisco, CA, 1976

[3] "Time series forecasting for building energy consumption using weighted Support Vector Regression with differential evolution optimization technique."Fan Zhang, Chirag Deb, Siew Eang Lee, Junjing Yang, Kwok Wei Shah. [Link]

[4] "Artificial neural network and SARIMA based models for power load forecasting in Turkish electricity market.". Bozkurt, Ömer Özgür, Göksel Biricik, and Ziya Cihan Tayşi. [Link]

[5] "Time series forecasting by a seasonal support vector regression model." Ping-Feng Pai a,*, Kuo-Ping Lin b, Chi-Shen Lin c, Ping-Teng Chang [Link]

[6] "Predicting residential energy consumption using CNN-LSTM neural networks."72-81.Kim, Tae-Young, and Sung-Bae Cho. [Link]

[7] “Multi-Step Short-Term Power Consumption Forecasting with a Hybrid Deep Learning Strategy”.Ke Yan, Xudong Wang, Yang Du, Ning Jin, Haichao Huang and Hangxia Zhou. [Link]

[8] "Day-ahead electricity consumption prediction of a population of households: analyzing different machine learning techniques based on real data from rte in france." Theile, Philipp, et al. IEEE, 2018. [Link]

[9] V. N. Vapnik.: The Nature of Statistical Learning TheonJ. Springer, 1995.

8 ONLINE COLLABORATION

The list of meetings held are given below:

- 03/28/2021 - From 5:30 PM to 7:30 PM - Everyone attended
- 03/31/2021 - From 5:30 PM to 7 PM - Everyone attended
- 04/03/2021 - From 10 AM to 12 PM - Everyone attended
- 04/07/2021 - From 11 AM to 1 PM - Everyone attended
- 04/08/2021 - From 6:15 PM to 7 PM - Everyone attended
- 04/10/2021 - From 10 AM to 12 PM - Everyone attended
- 04/12/2021 - From 6:15 PM to 7:30 PM - Everyone attended
- 04/17/2021 - From 10 AM to 1 PM - Everyone attended
- 04/30/2021 - From 7 PM to 8:30 PM - Everyone attended

9 APPENDIX

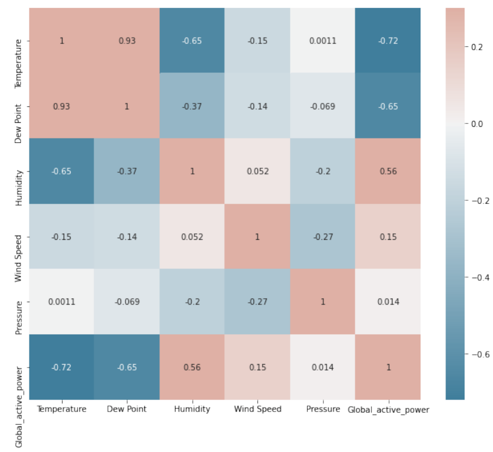

Figure 5. Correlation between predictor and response variables

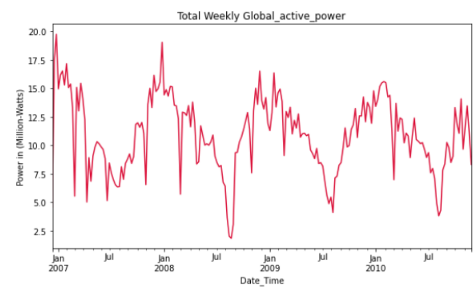

Figure 6. Time-series plot for total weekly global active power

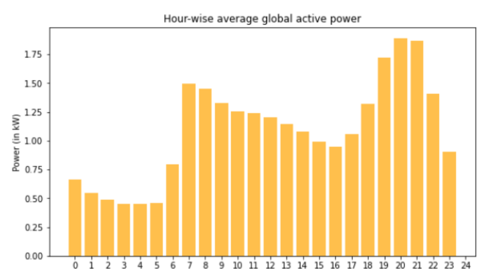

Figure 7. Average hourly distribution of global active power

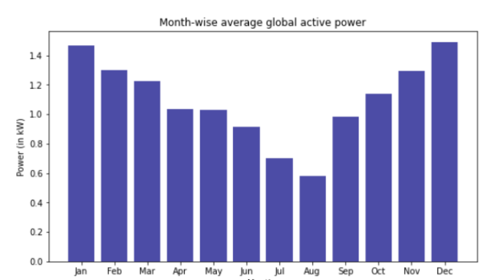

Figure 8. Average monthly distribution global active power