Mobile network traffic prediction

Problem statement

Mobile phone devices such as smartphones, tablets, wearable devices as well as mobile phone subscribers are increasing rapidly.According to a report presented by Ericsson, the mobile devices have surpassed the world population. Due to such a huge growth in mobile devices and mobile phone subscribers, the congestion of mobile networks is not unusual. This increases traffic congestion on the base station and energy consumption as well.

To overcome this network traffic congestion the traditional approach or brute force method is deploying more base stations, and adding more data processing units (increased capital expenditure) thus increasing energy consumption and maintenance spending (increased operating expenditure). This project proposes to use call detail record (CDR) to analyse the various patterns and predict the network utilization so that an informed network expansion can be taken up by the mobile service provider resulting in improved Quality of Service (QoS) to the end customers.

Data Wrangling

In this project, a multi-source dataset released by Telecom Italia in 2015 is used. The dataset is one of the most comprehensive collections from an operator and also publicly available. Originally, the collection was created for a big data challenge with projects ranging from mobile networking to social applications. Provided data points include records in telecommunication, weather, news, social network, and electricity from the city of Milan and Trento during November and December 2013.

Dataset Description:

For mobile Internet traffic forecasting, we focus on telecommunication records. Geographical grids are first defined for data recording. The city is divided into 100×100 areas with aggregated call detail record (CDR) data. Each grid has a unique square ID covering an area with the size of 235 × 235 meters. The telecommunications dataset contains the following information used in this work:

- **Square ID**: the identification of the square of Grid.
- **Time interval**: Data is aggregated for 10 minute time interval, the beginning of the time interval of the record is given. The end interval time can be obtained by adding 10 minutes to this value
- **Internet traffic activity**: the number of CDRs generated during the time interval in a square id

- **Received SMS** a CDR is generated each time a user receives an SMS
- **Sent SMS** a CDR is generated each time a user sends an SMS
- **Incoming Call** a CDR is generated each time a user receives a call
- **Outgoing Call** a CDR is generated each time a user issues a call

Data source: telecom italia open big data challenge dataset - A multi-source dataset of urban life in the city of Milan and the Province of Trentino Dataverse

Wrangling:

The Telecom Italia dataset chosen for this project consists of 59 TSV files (one file for each data, data was collected for 2 months from Nov to Dec 2013) consisting of spatially aggregated Telecommunication activity (Internet, sms and calls) at an interval of 10 mins, each file is about 300 MB. The total size of the dataset is 18.6 GB, due the huge size of the chosen dataset all of the data cannot be loaded at once into python (RAM constraints of a single system). To overcome this, each file was loaded into a pandas DataFrame in a loop.

Once one TSV is loaded to a temporary dataframe, following steps are performed on this dataframe:

1. Time interval column is converted to CET timezone so that further resampling can be done with ease
2. The NaN values are updated with 0 assuming that there were no Telecommunication activity during that time interval, since all of the values (SMSin, SMSout, callIn, callOut, Internet) are integers.
3. The Internet, SMS and Call activity is aggregated to 10 mins time interval ignoring the GridId and CountryCode. GridId and CountryCode is ignored for this analysis since the size of the file is too big

The processed dataframe is appended to a consolidated dataframe, which will contain aggregated information of all the other dataframe processed previously.

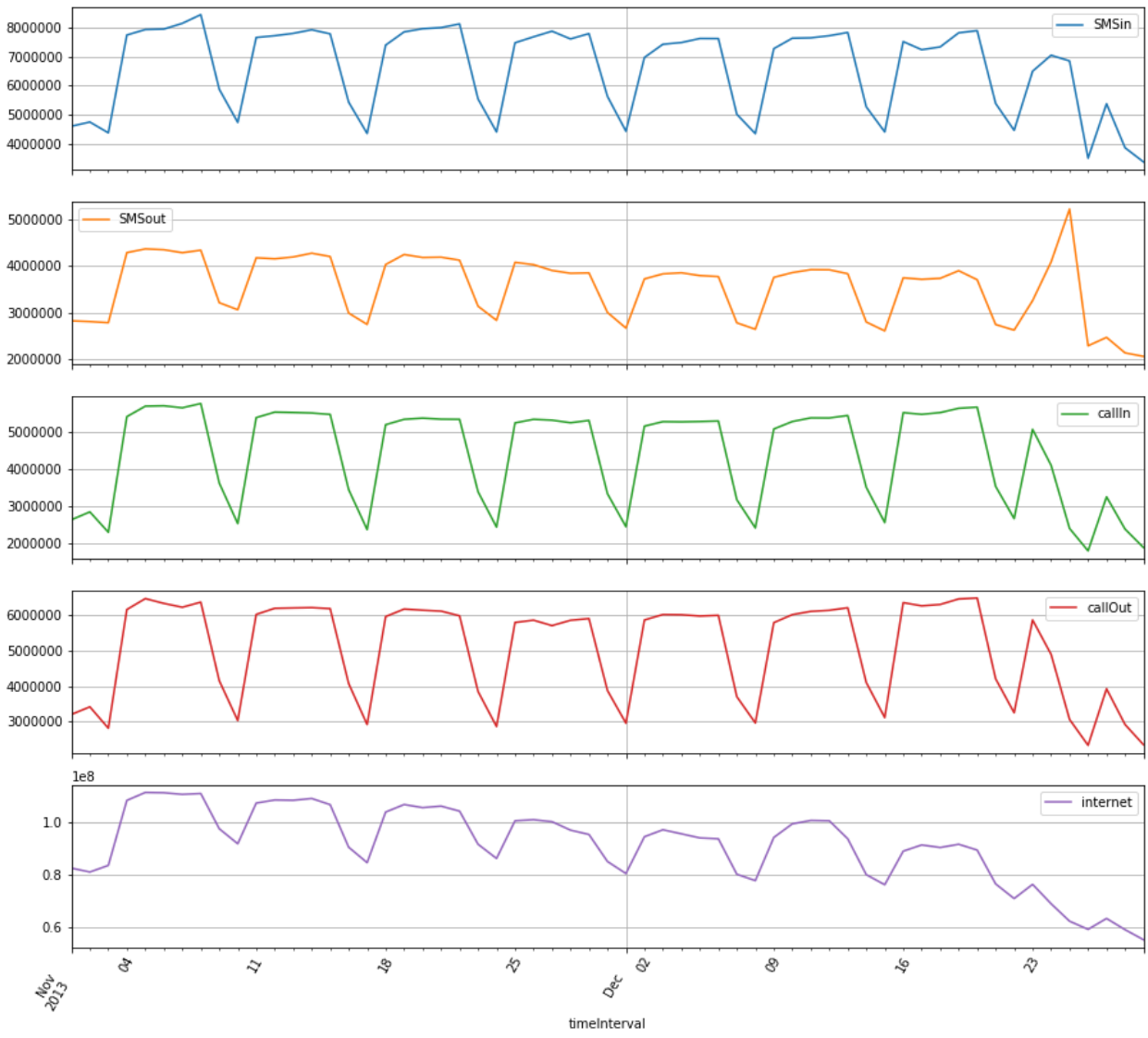
Feature Engineering:

Following variables were derived on the daily aggregated dataframe (cdfDaily):

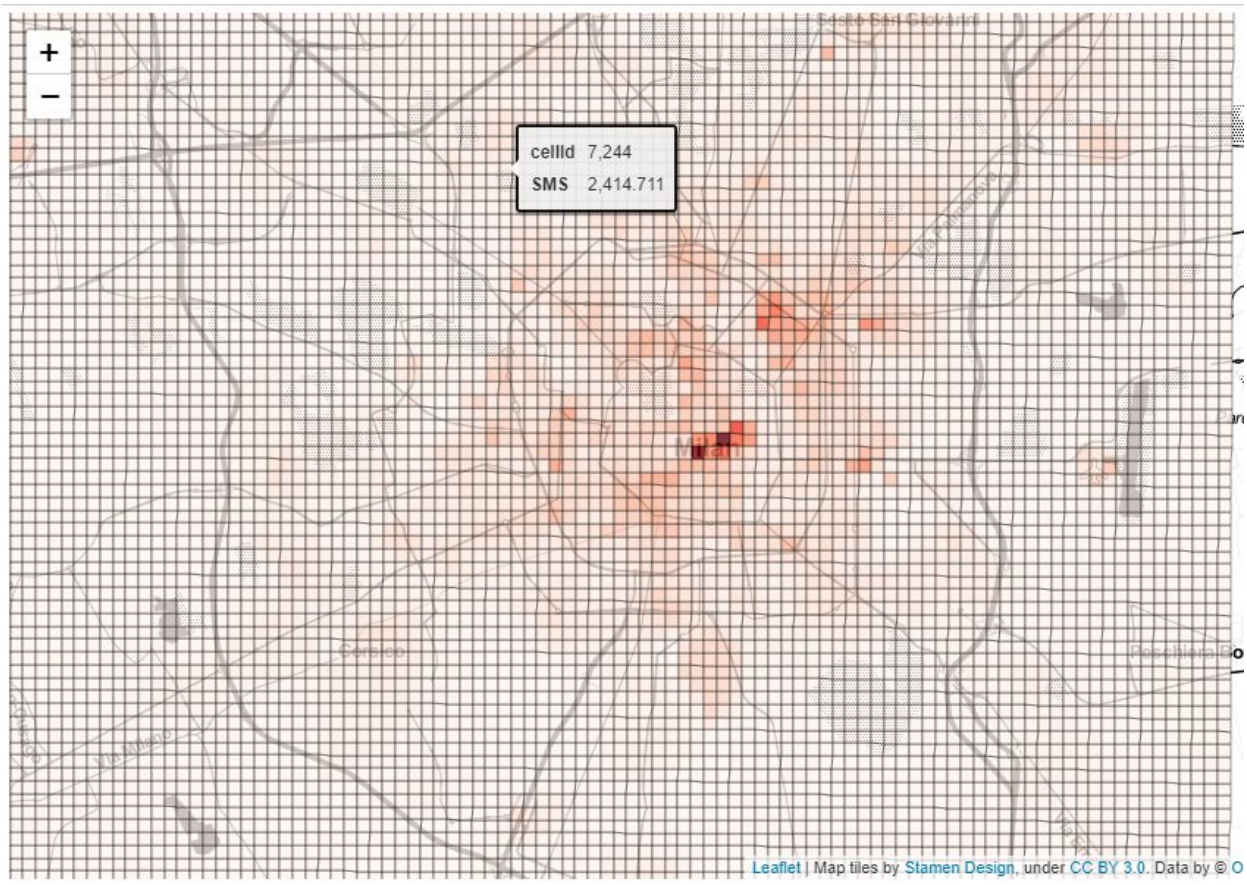
1. SMS: sum of SMSIn and SMSout. This will help in identifying the total SMS per day
2. Call: sum of callIn and callout. This will help in identifying the total calls per day
3. Day: day of the week (0-Monday, 6-Sunday). I have derived this variable to later identify if there is a correlation between the telecommunication activity and the day of the week
4. Weekend: whether a particular was recorded on a weekday or a weekend (derived based on the Day column, above, 5 / 6 belong to weekend)

Visualization

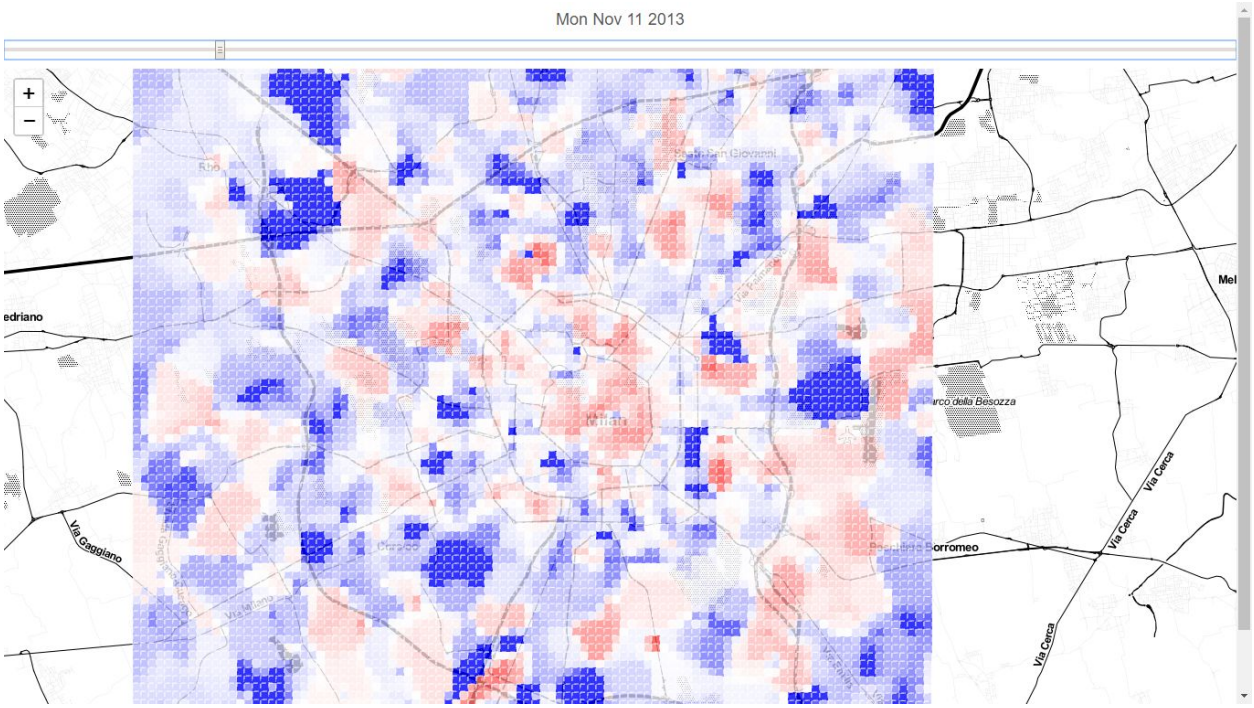
All telecom activities (SMS, calls, Internet) seem to follow a similar pattern with usage being relatively the same across the weekdays and a sharp fall during the weekends. The pattern is broken during the christmas period though.



Grids concentrated in and around the city center have highest usage, whereas the usage is minimal in the outskirts. Usage per grid seems to gradually decrease as the distance of the grid from the city center increases.



A time series interactive map indicated reduction in the cell phone activity during weekends in the city center and increase in the activity during weekends in sub-urban areas.

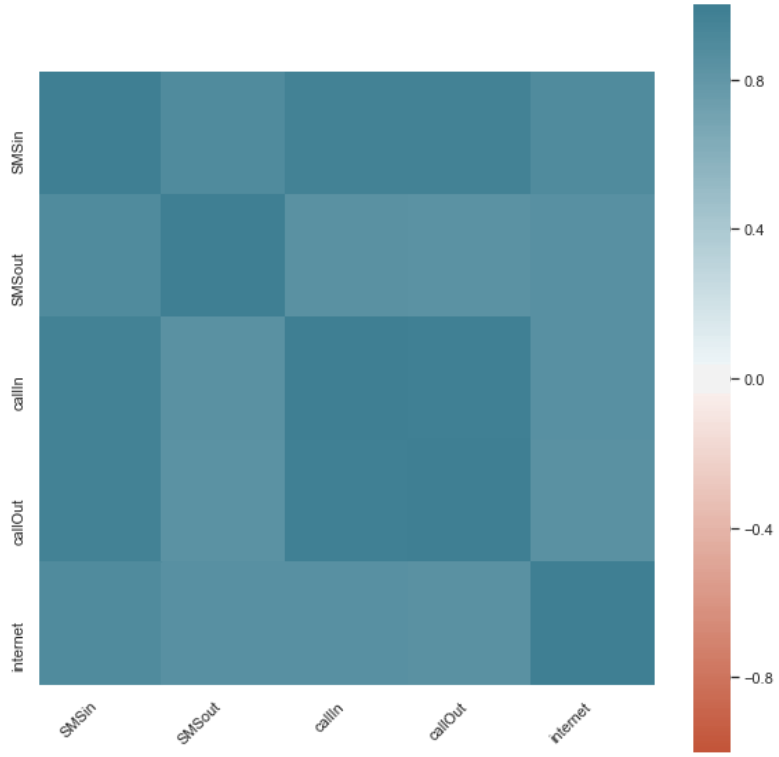


Inferential Statistics

Correlation between variables:

Following variables have high correlation:

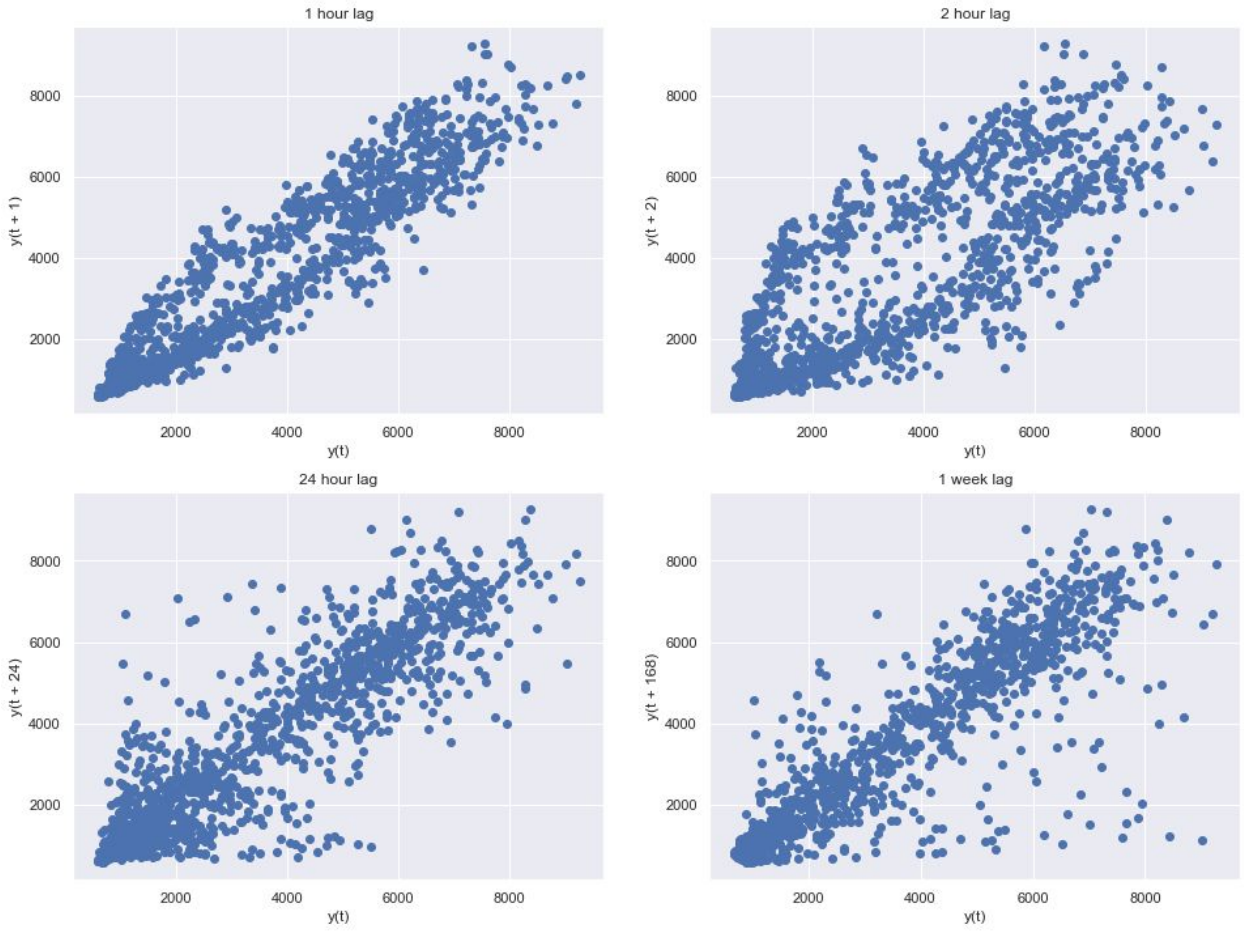
1. SMSIn and callIn
2. SMSIn and callOut
3. callIn and callOut



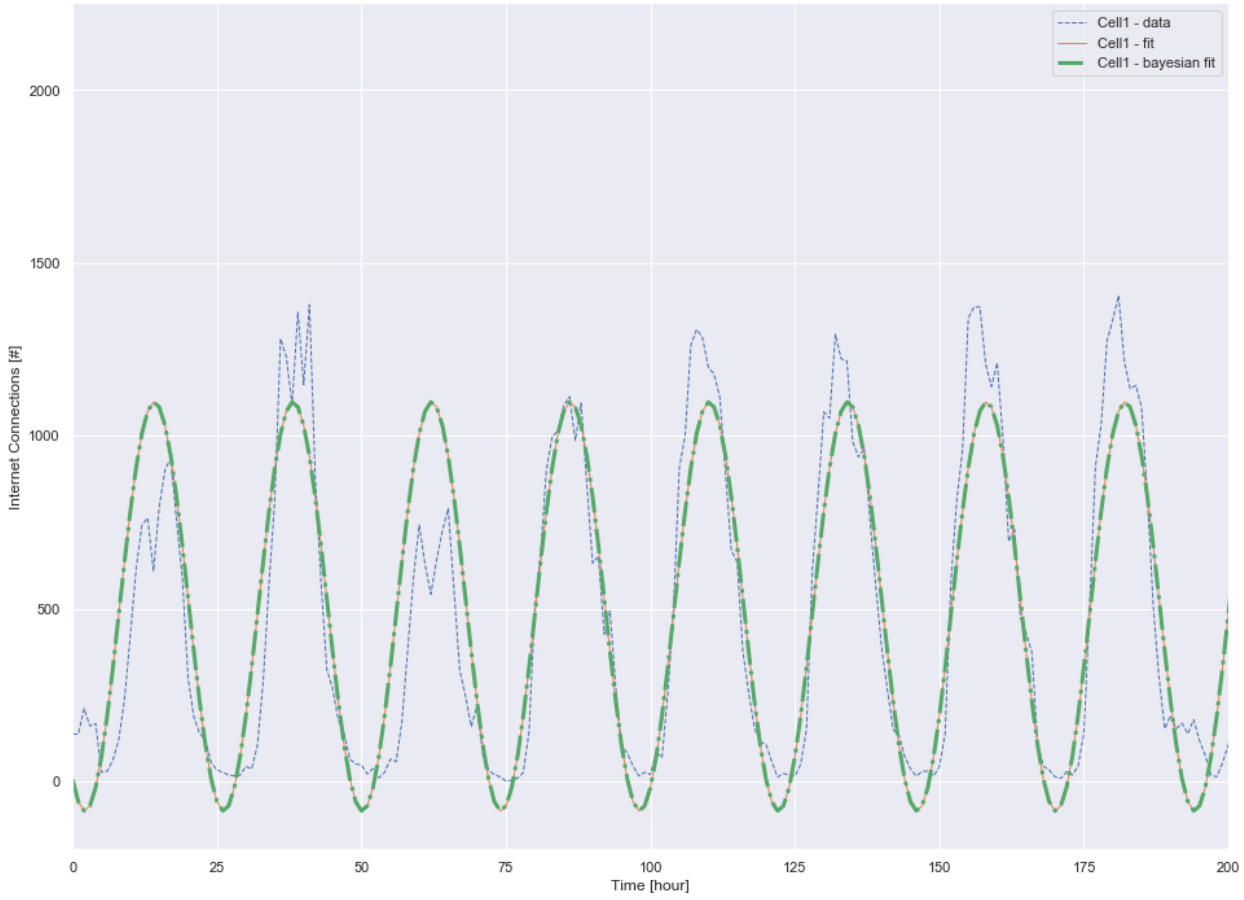
	SMSin	SMSout	callIn	callOut	internet
SMSin	1.000000	0.895334	0.960479	0.953618	0.895376
SMSout	0.895334	1.000000	0.842331	0.835209	0.843780
callIn	0.960479	0.842331	1.000000	0.978196	0.852959
callOut	0.953618	0.835209	0.978196	1.000000	0.836540
internet	0.895376	0.843780	0.852959	0.836540	1.000000

Model Fit:

On further analysis of the data it was identified that the time series data for two months was auto correlated with a time period of 24 hours lag. The data distribution seemed to be following a sine distribution.



A regression curve fit model was applied on this sine function to identify optimal parameters. Based on the optimal parameter Bayesian inference was simulated for 100,000 traces to identify if the selected parameters were indeed optimal fit for the data. The curve fit and the Bayesian Inference reported very similar results and it was concluded that the data follows a sine distribution



Model evaluation:

Approach:

Since we have time series data one approach to generate predictions is using the **ARIMA model and LSTM** (we have seen in EDA that the data is auto regressive with 24 hours lag). The other approach is to generate additional variables t-1 and t-2 data i.e. take the **correlated variable** t's one hour and two hour lag data as features to derive the dependent variable (we have seen in EDA that some variables are highly correlated)

Problem with the time series approach is that the data becomes very limited as we have to consider the time series evaluation only on a per cell basis. As a result, the model predictions for both ARIMA and LSTM are not as good as models built on correlated variable approach.

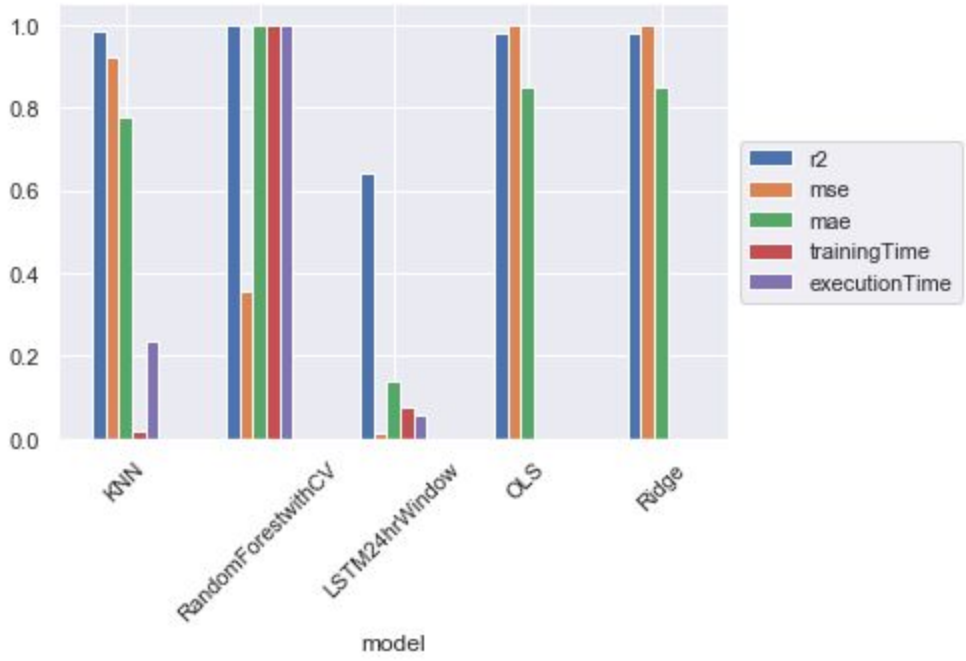
Result:

Of all the different models implemented it was observed that the Random Forest model shows the best R^2 score of 94.7% for the selected hyperparameters. Even though the accuracy of the model is very high, the result set for the given hyperparameters took **130** mins to produce this result, in contrast KNN (12) produced a R^2 score of 94% taking **12** mins which is much better than RandomForest. Linear Regression with Ridge regularization was very fast (300 ms) with an R^2 score of 93%.

Depending on the production application i.e. if the model training time is not relevant then RandomForest can be utilised.

For our scenario of predicting the network usage can be done beforehand and we don't require real time feedback. For this reason I have chosen KNN as the correct model since it is not taking very long to execute and also not sacrificing too much on the predictive power.

Below images show the various models implemented and their results.



	model	mae	mse	r2	trainingTime	executionTime
0	KNN	6.34	327.27	0.9333	149.00	42.90
0	RandomForestwithCV	8.16	127.44	0.9477	7800.00	180.00
0	LSTM24hrWindow	1.15	5.19	0.6078	600.00	10.00
0	OLS	6.91	355.92	0.9300	3.18	0.20
0	Ridge	6.91	355.92	0.9300	0.37	0.15