**TAXI DEMAND PREDICTION USING BIG DATA ANALYTICAL TECHNIQUES**

Abstract. This empirical paper compares the accuracy of six univariate methods for short-term taxi demand forecasting for lead times up to a week-ahead. The methods are compared using a time series of half hourly demand for New York.

**Tentative plan:**

• Introduction

• Data Set Information

• Exploratory Data Analysis

• Data Preparation

• Supervised Learning Models – Linear Regression, Conditional Inference Decision Tree, Random Forest, Conditional Inference Random Forest, XG Boost

• Time Series Models – Holt-Winters seasonal method

• Deep Learning Model - LSTM

• Conclusions

1. **Introduction**

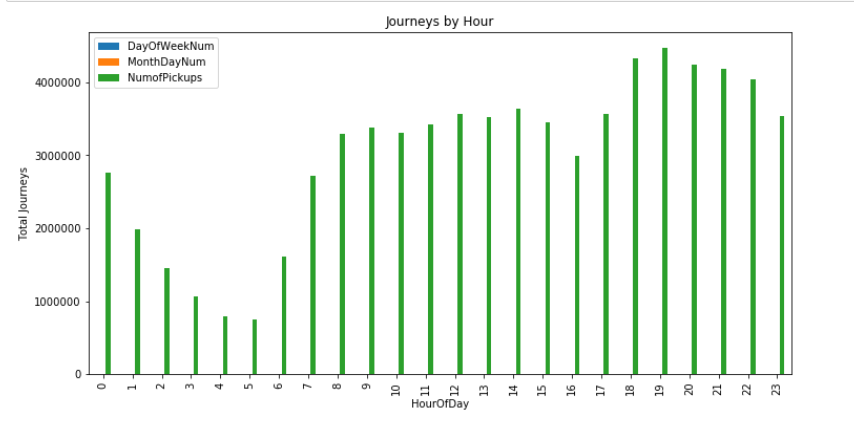
Taxi-supply planning requires eﬃcient management of existing taxis and optimization of the decisions concerning additional capacity. Demand prediction is an important aspect in the development of any model for taxi planning. The form of the demand depends on the type of planning and accuracy that is required; hence it can be represented as an annual demand, a peak demand or demand like daily, weekly etc. Short-term demand forecasts are required for the control and scheduling of taxis. The focus varies from minutes to several hours ahead. The predictions can help in optimizing taxi supply at a given location and time. In the short run, the taxi demand is mainly inﬂuenced by seasonality (daily - weekly cycles, calendar holidays) and special events. Weather related variation is certainly critical in predicting taxi demand for lead times beyond a day ahead. In this paper, we compare the accuracy of simple benchmarks and four more sophisticated methods. We evaluate the methods using 24 weeks data for the New York city. We consider lead times up to a week ahead.

1. **Data Set Information**

The data set is Yellow Taxi trip data of New York for the duration of Jan-2015 to Jun-2015. The dataset has 146 million rows, each row represents for one taxi ride ordered. We have also considered enrichment of data with weather and holiday details of New York for the same calendar duration.

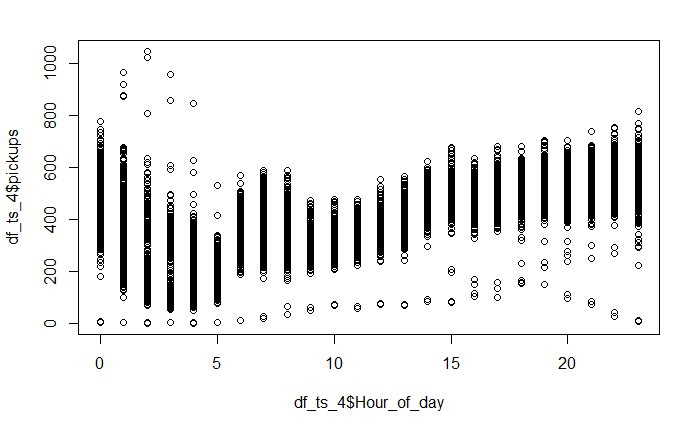
1. **Exploratory Data Analysis**

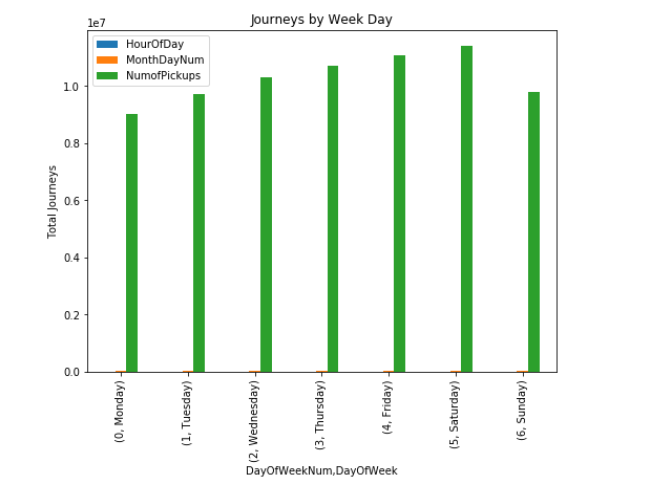
We have plotted multiple graphs to analyze the demand on hourly, weekly, monthly and demand variation within the hour respectively.



Hourly analysis during the day shows least demand during early morning hours and maximum during evening hours.

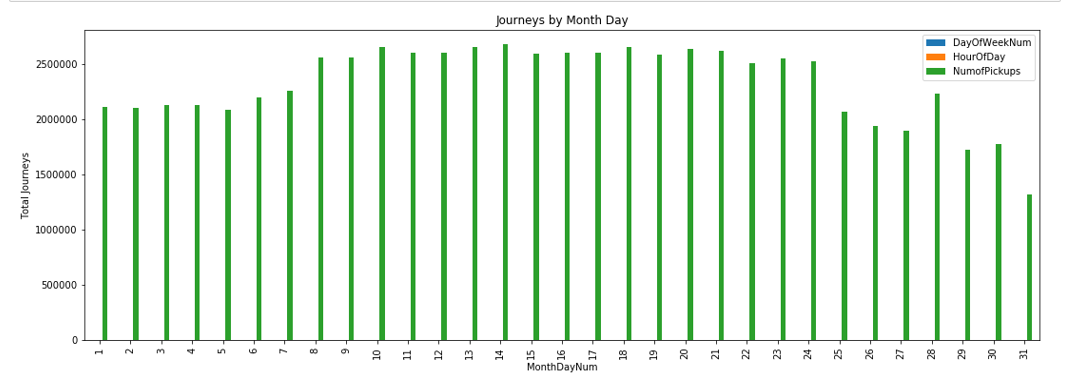
The graph below gives a deeper detail about the demand variation within the hour.





Week-day analysis shows an uptrend towards the weekend and falling on Sunday with minimum demand on Monday.

Month-Day analysis shows consistent high demand during the second and third week of the month while lesser in the first week and last week being minimum.



1. **Data Preparation**

The raw data is available in the following format:



As the desired input is taxi pickups every half an hour, following transformation steps are executed on data:

1. Data Imputation: Replace the missing values by average of pickups at 1 previous and 1 ahead time stamps
2. Round-up the location (latitude and longitude) to 1 decimal place
3. Group the data based on concatenated latitude and longitude. This will result in 4 datasets, one for each combination of latitude and longitude
4. Replace outliers by 5th or 95th percentile using box plot
5. Group the data for each time stamp in each dataset and count rows as no of pickups for that timestamp
6. Aggregate taxi pickups for each dataset on half hourly basis
7. Figure below shows the final dataset for one location



1. The dataset prepared is univariate data used for time series models and LSTM model
2. To apply Supervised learning models further data enrichment followed by feature engineering is performed
3. Figure below shows the enriched and engineered data

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. **Approach and Algorithms**

After aggregation of data on half hourly basis, we have in total 8064 data points for 24 weeks duration. 23 weeks of data has been used to train all the models and 24th week pickups have been used as test dataset.

From Step 3 above, we will have 4 datasets based on Location within New York.

Prepare train and test set for time series

train\_data = [1:7728,]

test\_data = [7729:8064,]

Clarifications:

The demand trend graph shows relatively poor demand during 24th week so having a conservative approach we have taken 24th week data as Test set. Secondly, for Supervised Learning models random splitting would be more appropriate but as these models are going to be compared with Time Series Models so having different data set for comparison would not be justified.

The reason behind dividing the New York map in 4 sections is to get better detail of the demand which may vary because of demography variations.

N.B.: Demography details have not been added in the data set.

We have used multiple algorithms to find the one performing best to predict the demand with the past data available to us.

Please refer to below to get details about them.

Linear Regression

Algorithm:

1. Let us create a Linear model on train data with input parameters (pickups ~ hour\_min + Temp.+ Visibility + Precip + Conditions + Day\_of\_week + isholiday + one\_week\_lag\_pickups) using lm().
2. Apply the model on Test Data to Predict using predict().
3. Store the results for all 4 locations

If (I == 1)

{

Result\_lm = combine pickups of test dataset with pred

}

else

{

result\_lm = add pickups of test dataset and pred with result\_rf

}

1. Calculate the Mean Absolute Percentage Error (MAPE).

Conditional Inference Decision Tree

Refer to the link <https://cran.r-project.org/web/packages/partykit/vignettes/ctree.pdf> for details about Conditional Inference Decision Tree.

Algorithm:

1. Let us create a tree model on train data and call that as model\_rpart.

model\_rpart = create tree model on train\_data (pickups ~ hour\_min + Temp.+ Visibility + Precip + Conditions + Day\_of\_week + isholiday + one\_week\_lag\_pickups) using ctree().

1. Prune the tree using prune function and call it as Pfit, where cp=0.01149.
2. pred = predict(model\_rpart,test\_data)
3. Store the results for all 4 locations

If (I == 1)

{

result\_dt = combine pickups of test dataset with pred

}

else

{

result\_dt = add pickups of test dataset and pred with result\_dt

}

1. Calculate the Mean Absolute Percentage Error (MAPE).

Random Forest

Logical flow of the model has been same as in Decision Tree model above, the number of trees we have considered as additional parameter. The value of ntree=500 for best result in this case.

Conditional Inference Random Forest

Refer to the link <https://rdrr.io/cran/partykit/man/cforest.html> to get details about Conditional Inference random Forest.

Algorithm:

1. Let us create a model on train data with input parameters (pickups ~ hour\_min + Temp.+ Visibility + Precip + Conditions + Day\_of\_week + isholiday + one\_week\_lag\_pickups) using cforest().
2. Predict the value on test data, predict (model, test\_data, OOB = TRUE, type="response").
3. Store the results for all 4 locations

If (I == 1)

{

result\_crf = combine pickups of test dataset with pred

}

else

{

result\_crf = add pickups of test dataset and pred with result\_crf

}

1. Calculate the Mean Absolute Percentage Error (MAPE).

XGBoost

Refer to the link <http://xgboost.readthedocs.io/en/latest/model.html> for details about XGBoost.

Holt-Winters Seasonal Method

The time series model used for prediction is **Holt-Winters seasonal method**. To brief what Holt-Winters method is, it comprises the forecast equation and three smoothing equations — **level**, **trend** and the **seasonal component**, with smoothing parameters.

There are two variations to this method that differ in the nature of the seasonal component. The **additive method** is preferred when the seasonal variations are roughly constant through the series. The **multiplicative method** is preferred when the seasonal variations are changing proportional to the level of the series.

1. To optimize alpha(level), beta(trend) and gamma(seasonality) parameters in Holt Winter model, 3 nested loops were used as below:

y = train time series

For alpha in 0.1 to 1

For beta in 0 to 1

For gamma in 0 to 1

model = HoltWinters(y, alpha,beta,gamma,start.periods = 336)

forecast = predict 336 (1 week ahead values) using computed model

result = combine forecast and test data

calculate MAPE

If computed MAPE < last MAPE then accept the solution

1. The above loop is run separately for additive and multiplicative Holt Winter model

Deep Learning Model - LSTM

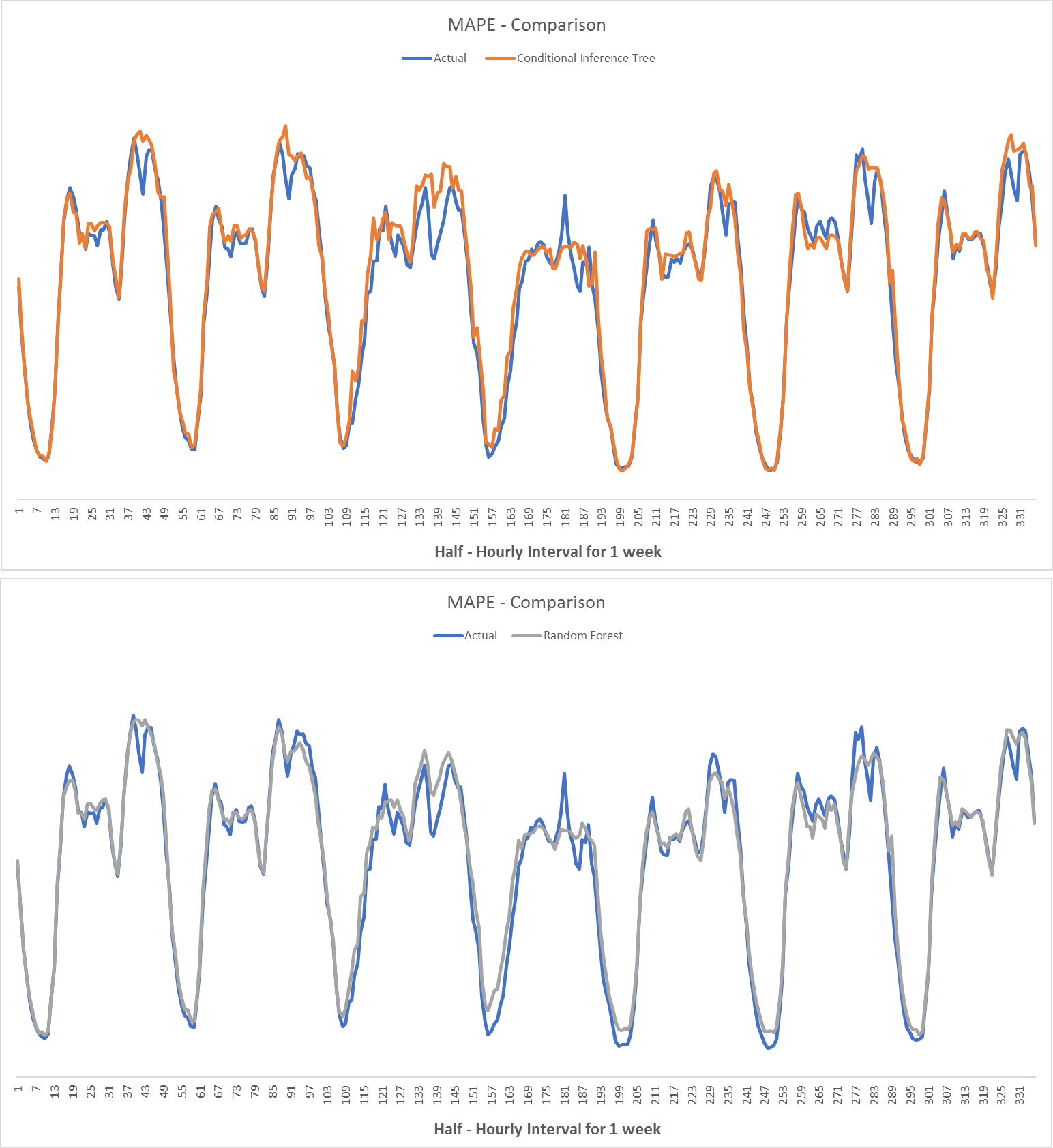
Refer to the link <https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/> for details about LSTM.

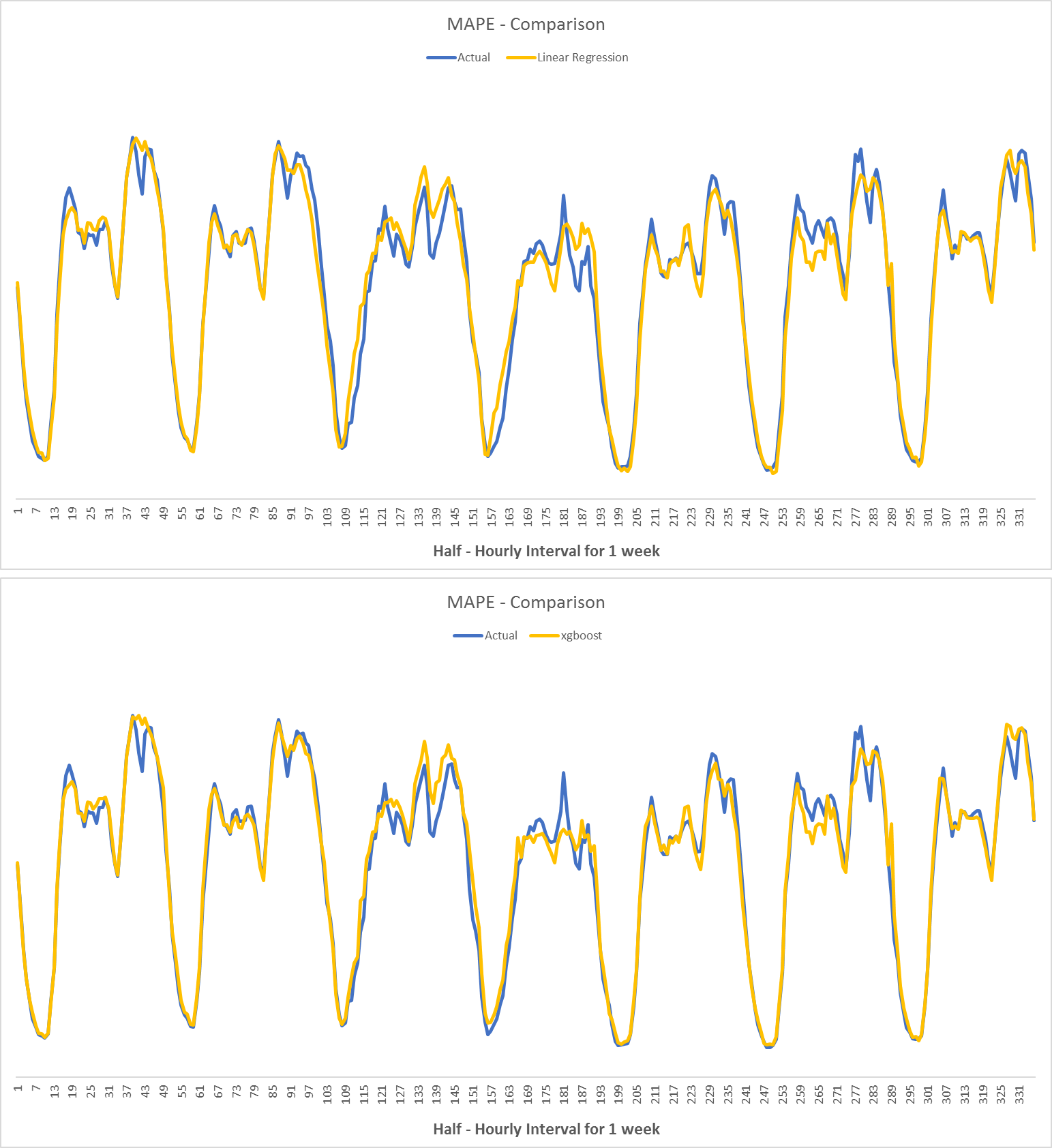
To model LSTM data is re-arranged in the following manner:

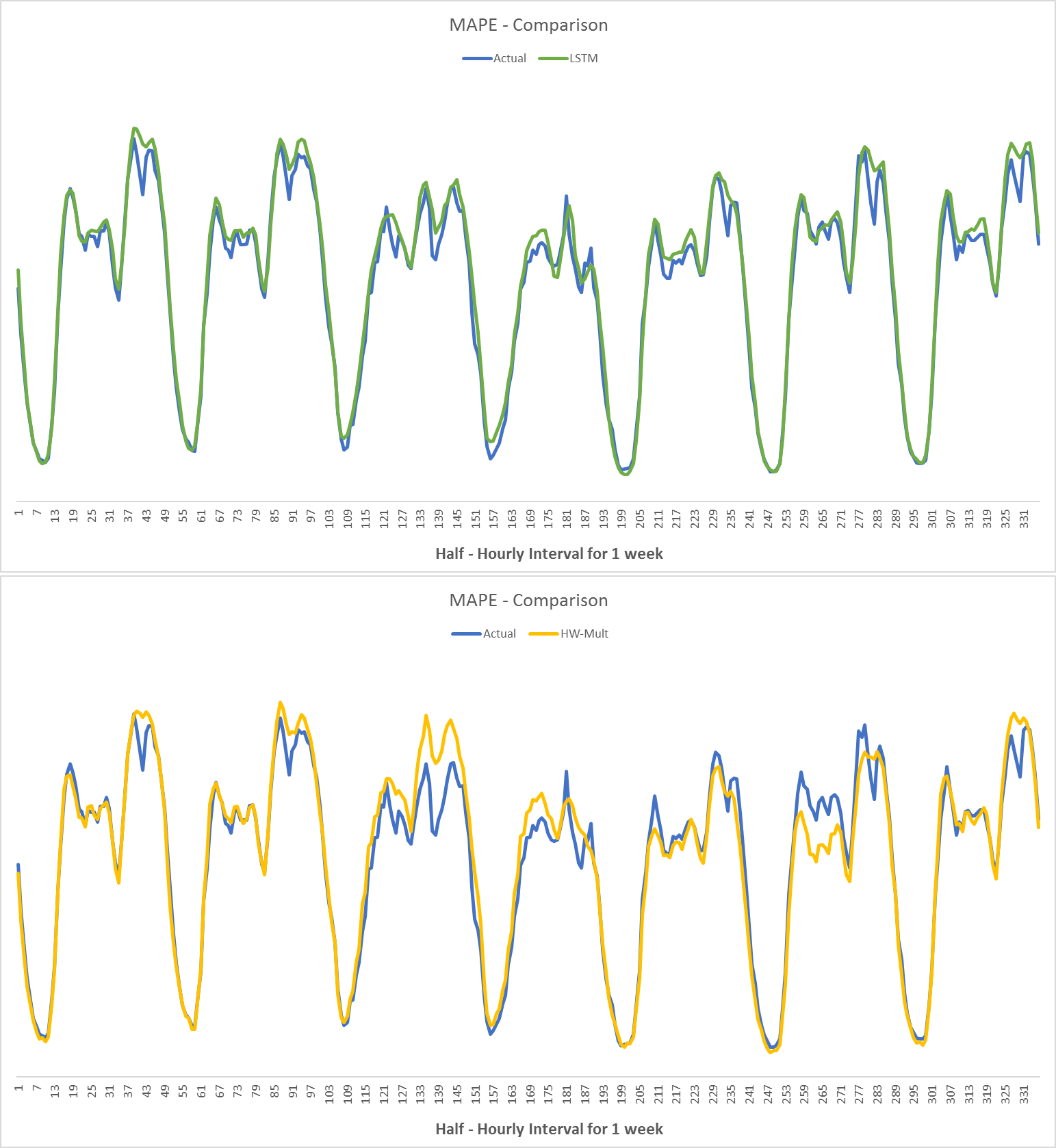
1. Every value starting from position 337 is put into 1-D array. This is Y-variable.
2. 336 values prior to the value added to Y-variable are added to X variable
3. X variable is an array with dimensions 8046,336
4. Y variable is an array with dimensions 336,336
5. Layers used 1 input, a hidden layer with 4 LSTM blocks and an output layer that makes a single value prediction, followed by a dense layer
6. The default sigmoid activation function is used for the LSTM blocks. The network is trained for 20 epochs and a batch size of 1 is used.
7. **Graphical Results:**

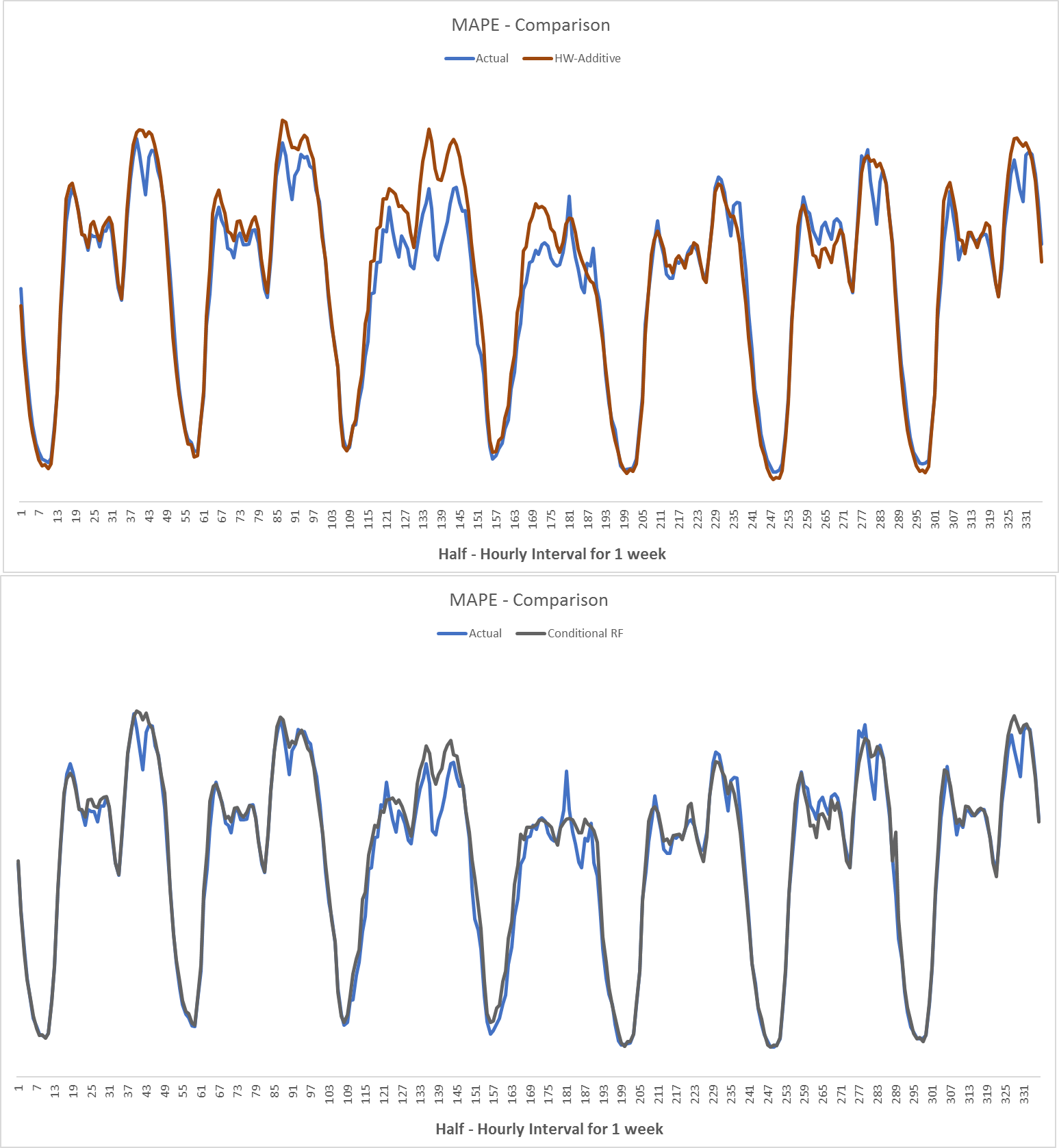
Conditional Random Forest suites best to predict the demand for the data available to us though Holt-Winter-Multiplicative has given best result for one section of the map out of the four. It is because…

When we compare multiple graphs we see that Conditional Random Forest almost imitates the actual demand curve with accuracy of 94% which should be highly acceptable.









1. **Conclusion**
2. **Bibliography**