Energy Analytics

I. Introduction

In this report, we will introduce our final forecasting model for estimating the energy load demand in the past weeks of March. The model was designed through several iterations throughout the forecasting rounds, focusing on improving its accuracy and reliability using historical data on temperatures, sunshine duration and other relevant indicators. We will provide an overview of the forecasting model, how we arrived at our final model and the model's performance in predicting the demand.

II. Forecasting Model Iterations

In reaching our final forecasting model, we went through three main iterations to refine the model which will be described in greater detail below.

First iteration

In the first iteration, we mainly explored the possible forecasting models and the potential parameters to include in the model, mainly for those involving regression. As we performed our exploratory analysis, we found that a 2nd order polynomial for temperature could be introduced to capture the non-linear relationship between temperature and electricity demand as shown in figure 1 below.

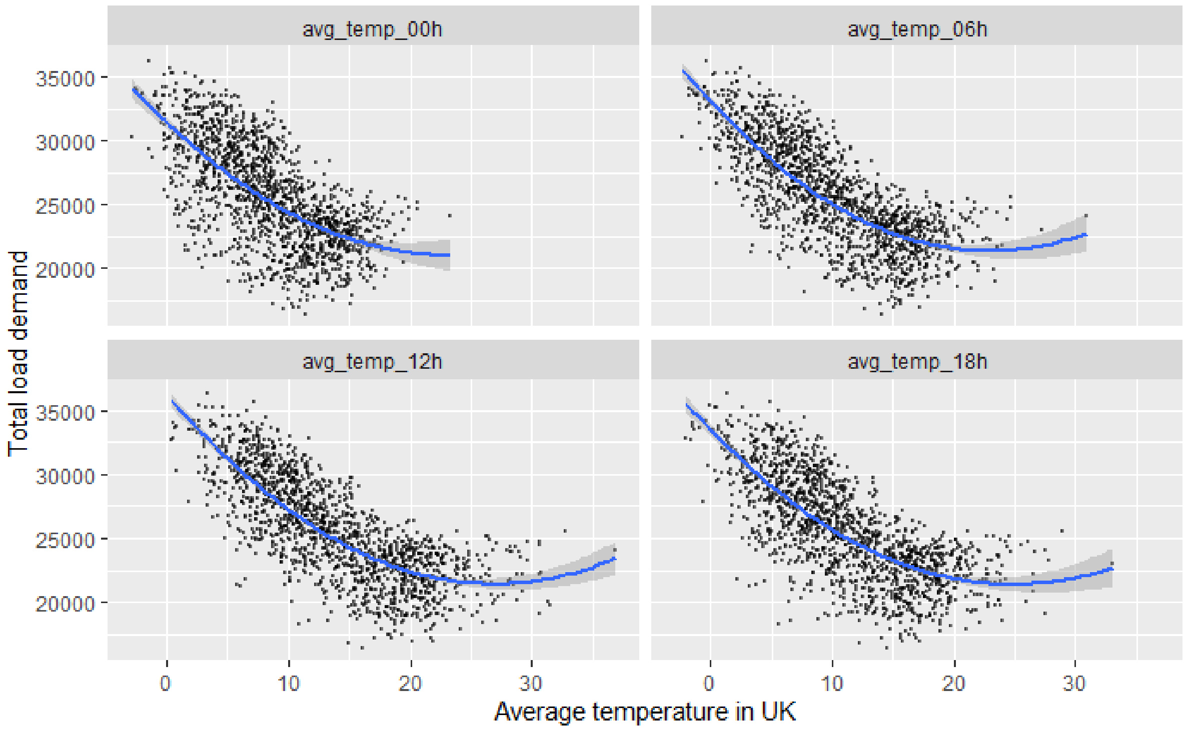

Figure 1. Average temperature from London, Bristol and Leeds against average total electricity load demand at different times of the day

Furthermore, we also introduced various calendar parameters to capture the monthly seasonality and the effect of non-working days on the electricity demand as observed in figure 2. Hence, we added month binary indicators, a weekend indicator and a holiday indicator based on the UK bank holidays extracted from the official GOV.UK website.

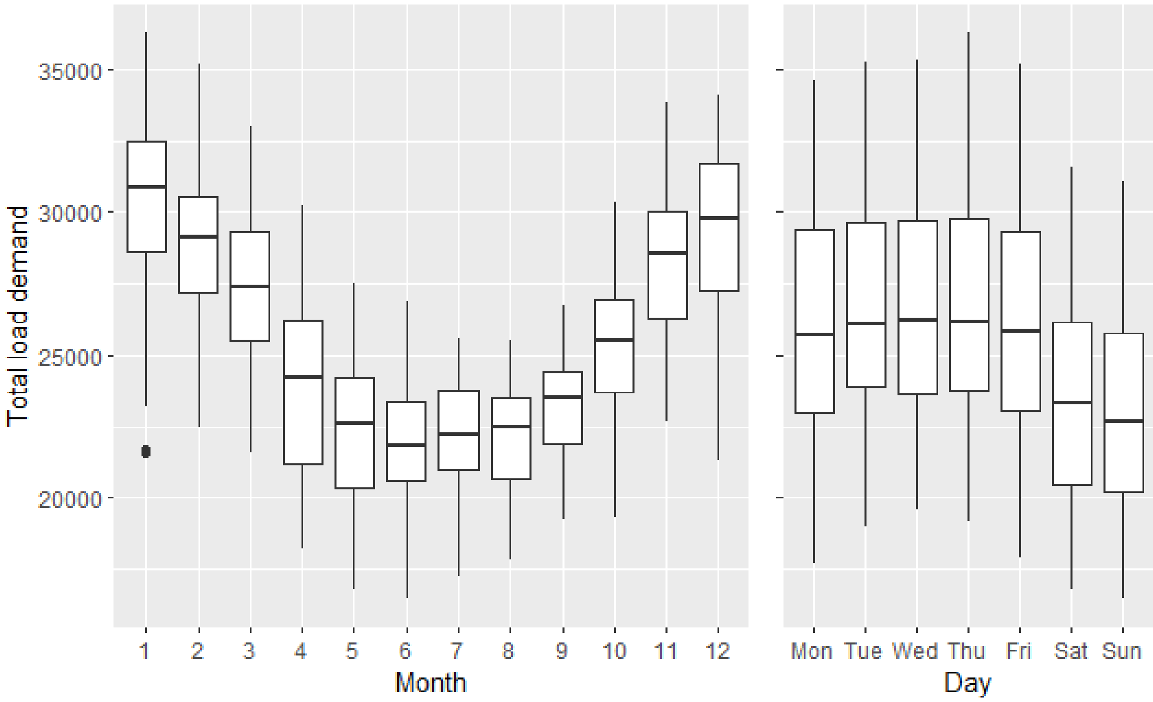

Figure 2. Average total electricity load demand in (i) left: different months of the year and (ii) right: different days of the week

We also generated a Fourier series consisting of sine and cosine terms to capture seasonality for periods of 7, 30, 90 and 365 days (weekly, monthly, quarterly and yearly), i.e.

$$s_{k,t} = sin(\frac{2k\pi t}{m}), \quad c_{k,t} = cos(\frac{2k\pi t}{365})$$

where k is based on the number of terms to include in the series, t is the period of seasonality to be captured and m is the time-series frequency which in this case is set to be $m = 365$ or yearly (we assume the data is a yearly time-series with a frequency of 365 days in a year).

Based on this, we estimated the number of Fourier terms to use by minimising the AIC through iterations in the linear regression model and ARIMA with the regression model. We found that using the following numbers of Fourier terms: {3, 1, 2, 3} for the respective periods of {7, 30, 90, 365} produced the lowest AIC, i.e.

$$F(t = 7) = \sum_{k=1}^{3} s_{k,7} + c_{k,7} \quad , \qquad F(t = 30) = s_{1,30} + c_{1,30}$$

$$F(t = 90) = \sum_{k=1}^{2} s_{k,90} + c_{k,90} \quad , \qquad F(t = 365) = \sum_{k=1}^{3} s_{k,365} + c_{k,365}$$

Where $F(t)$ is the Fourier series for seasonal period $t \in \{7, 30, 90, 365\}$.

Additionally, before feeding the model with the aforementioned parameters, we applied a correction factor to the temperature forecast provided to us to predict the next day's demand. We observe a noticeable discrepancy between the forecasted temperature data provided and the actual temperature. As the future demand will be determined by the forecasted temperature, we would want to minimise this error. Hence, we introduce an additive correction term to the forecasted temperature based on a naive seasonal average of the forecast errors.

For example, the average forecast error in London every 12 PM is estimated to be -1.466745°, so our correction term is +1.466745° which we apply to the forecasted London temperature at 12 PM. If we refer to the example shown in table 1 below, London's temperature on 5 March 2023 at 12 PM is forecasted to be 4.5°, hence our corrected temperature on that particular day, time and location would be 5.966745°, which turns out to be very close to the actual temperature of 5.828078° (or at least closer to actual temperature than the forecast provided). Since we do not know the actual temperature on the next day, we can only rely on the given forecasted temperature data. By feeding the model with this adjusted temperature, we found that it minimises the discrepancy between the actual temperature and the temperature fed into the model parameter (error of -0.3552° on average), as compared to simply using the given forecasted temperatures (error of -0.8948° on average). We update this correction term every week, by taking the latest average forecast errors.

Table 1. London temperatures comparison between the actual temperature, given forecasted temperature and the corrected temperature used for prediction on March 5th, 2023

Datetime	Actual Temp.	Given Temperature Data		Temperature Parameter for Prediction		
		Temp.	Errors	Correction	Temp.	Errors
5-Mar-23 00h	3.908078	5.5	-1.59192	-1.55626	3.943745	0.035667
5-Mar-23 06h	6.329744	5.5	0.829744	0.468411	5.968411	-0.36133
5-Mar-23 12h	5.828078	4.5	1.328078	1.466745	5.966745	0.138667
5-Mar-23 18h	2.598078	2	0.598078	0.538356	2.538356	-0.05972

After pre-processing and preparing the data, we split the data into train-test split based on 80-20 ratio. We then fit these parameters into the various models we tested, such as multivariate linear regression (MLR), harmonic multivariate linear regression with Fourier terms (MLRF), basic ARIMA model, dynamic ARIMA with regression, dynamic harmonic ARIMA (Fourier terms), seasonally adjusted ETS (STL-ETS) model and neural network. We evaluated the model using the root of mean squared errors (RMSE) and selected the model which gives the lowest RMSE which we found to be the MLR model as seen in table 2.

Second Iteration

As we evaluated our model performance in the first iteration, we noticed that the model did not perform so well. We found that there are confounding variables in our regression which we fail to consider, leading to endogeneity in the current electricity demand specifically the post-pandemic situation due to the spread of Covid-19 and the hiking gas prices caused by the Russia-Ukraine war as seen in figure 3 below where overall level decreased by 7% to 8% from the pre-covid period.

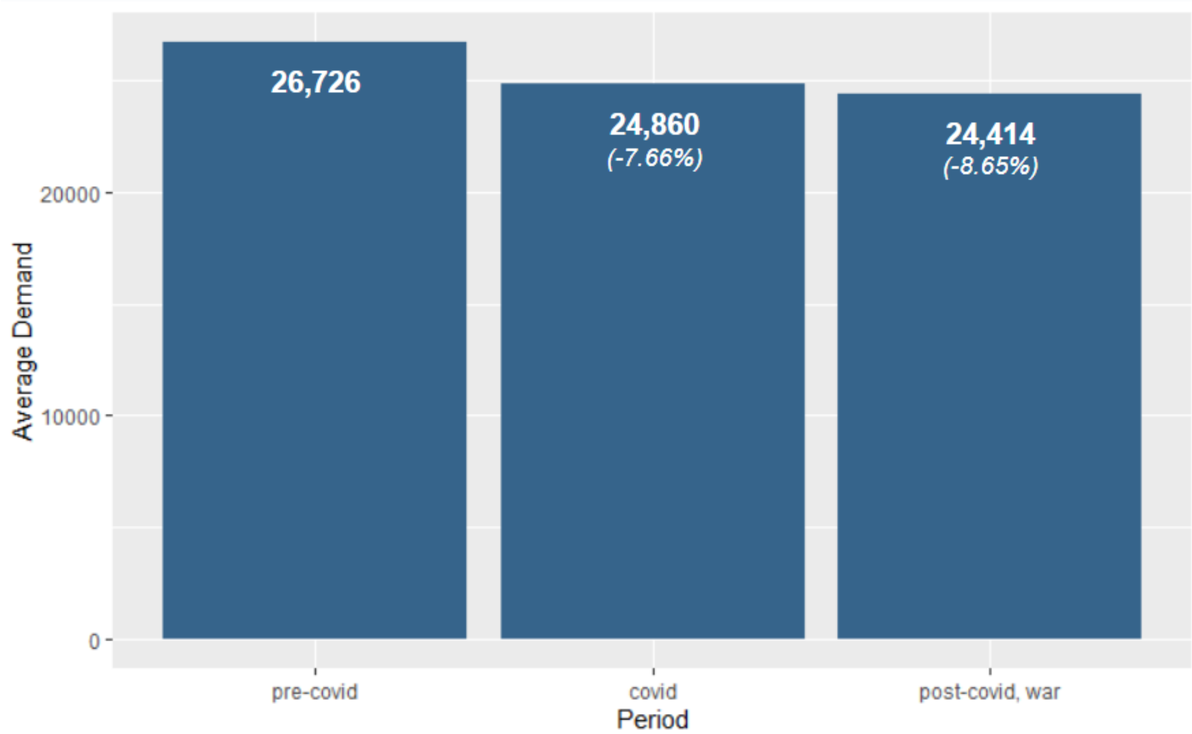

Figure 3. One-year average load demand calculated over three 1Y-periods: (i) Pre-covid (1 Jan to 31 Dec 2019), (ii) Covid (1 Jan to 31 Dec 2020), and (iii) Post-covid, war period (1 Jan to 31 Dec 2022)

The pandemic in the past 2 years results in businesses to operate remotely or in a hybrid mode which led to a decrease in commercial electricity usage. The skyrocketing gas prices due to the war also led to a decrease in the level of electricity demand. To account for both these factors, we introduced the covid stringency index and the average monthly gas prices.

We also note that daily fluctuations in gas prices may not immediately affect the daily load demand. Instead, there is a time-lag effect, as consumers observe the past few weeks or months' data before adjusting their electricity usage, for example, consumers do not necessarily increase their electricity usage just because today's gas price falls. Therefore, we decided to aggregate the gas prices into a monthly average.

To increase our selection of models, we also introduced the TBATS model in this round. Lastly, we tried to combine the forecasts of various models and find that combining model forecasts, even by means of simple average, leads to significant improvement in forecasting accuracy. This aligns with the findings and recommendations by Bates (1969) and Clemen (1989) that a combination of forecasts yields better forecasting results.

Therefore, we used the simple average combination of 5 model forecasts with the lowest RMSE, i.e. multivariate linear regression (MLR), multivariate linear regression with Fourier terms (MLRF), dynamic ARIMA with regression term, seasonal ETS model (STL-ETS) and TBATS model. This hybrid forecast leads to a significantly lower RMSE as compared to any individual model that was tested as observed in table 2.

Table 2. Summary of Test RMSE at different stages of the model iteration

Model	First Iteration	Second Iteration	Third Iteration
MLR	0.07832419	0.09974969	0.09915717
MLR (w/ Fourier)	0.08070631	0.10138777	0.10104959
ARIMA (w/ Reg)	0.09972001	0.10243300	0.10098244
ARIMA (w/ Fourier)	0.21961439	0.22441252	0.23556765
STL-ETS	0.11302033	0.11283138	0.10333952
Neural Network	0.15587746	0.17653157	0.17402071
TBATS	-	0.09155158	0.09709712
Combination (All)	-	0.08813160	0.08849985
Combination I	-	0.08254498	0.07741888
Combination II	-	-	0.07734754

Note: cells highlighted green indicates the model with the lowest forecast error (RMSE) in the given stage; combination (all) is the average forecast of all models, combination I is the average forecast of 5 models (MLR, MLRF, ARIMA w/ Reg, STL-ETS and TBATS), combination II is the weighted average using inverse RMSE of the same models used in combination I.

Third Iteration

To further improve the model, we decided to introduce weightage to each model's forecasts instead of taking a simple average. Our weights are based on the inverse value of the model RMSE, which can be expressed as follows:

$$w_i = \frac{1/RMSE_i}{\Sigma_i 1/RMSE_i}$$

For model i, where $i \in \{MLR, MLRF, ARIMA\ (w\ Reg), STL-ETS, TBATS\}$.

We found that this method of weighting the forecast gives a slightly better, if not comparable, accuracy to simple averaging. It is also theoretically more reliable and sound as compared to a simple average, which tends to oversimplify the forecast.

Additionally, we also observed a consistently noticeable degree of overprediction in the previous rounds' forecasts. Hence, we decided to also add a correction multiplier term in our forecast. The correction multiplier is obtained by using a naive method of taking the ratio of actual demand against the predicted value for the previous day's demand. The idea behind this is that there are factors that we fail to capture in our models, causing an overprediction in our forecasts.

For example, let's say we want to forecast the demand for round 7, on 21 March 2023. Our model predicted the log demand on 19 March to be 10.10610 but the actual demand is 10.02361, therefore our correction multiplier is $\gamma = \frac{10.02361}{10.10610} = 0.991838$. We then apply this correction multiplier to our model's prediction for 21 March, which was initially 10.191346, and adjusted it to 10.10816 which turned out to be really close to the actual log demand of 10.06528.

By using this multiplier term, we hoped to approximate the effects that cause our model to overpredict in a practical and convenient manner. In general, our correction multiplier term normally ranges between 0.991 to 0.995. Multiplying this correction multiplier term reduces the overprediction as expected and results in more accurate prediction overall, especially in round 6 and 7.

Final Forecasting Model

The forecasting model that we have finalised is a combination of forecasts from 5 individual models that we trained. We have tested various models such as neural network and pure ARIMA model, however, we found that there are only a few models that perform well and have relatively low forecast errors. Through three rounds of model iterations aforementioned, we concluded the final forecasting process in the steps as shown in figure 4 below.

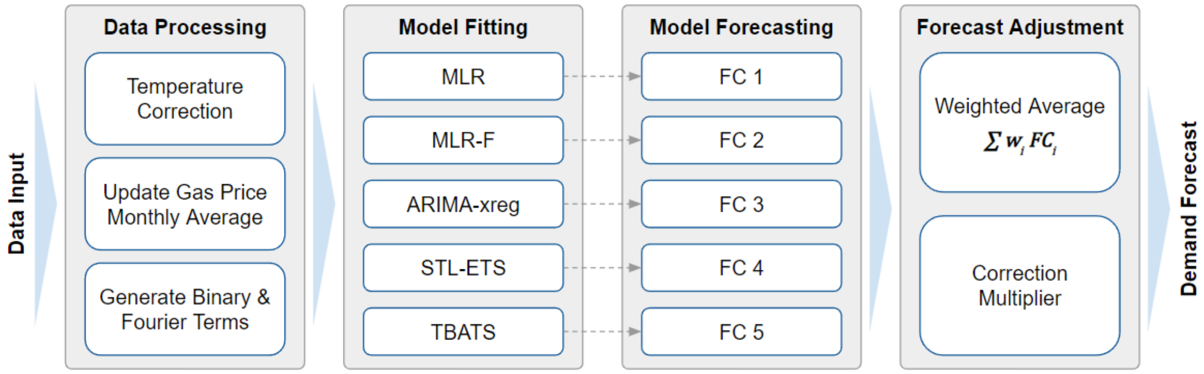

Figure 4. Finalised forecasting process; note: FC_i indicates the forecast of model i, and w_i indicates the weightage of the forecast i where w is calculated as the inverse RMSE of the model divided over the total inverse RMSE of all selected models.

The individual sub-models that we have selected and combined are as follows:

Model 1: Multiple Linear Regression (MLR)

Multivariate Regression is a method used to measure the linear relationship between the multiple independent factors and the dependent variable. This statistical method is broadly used to predict the outcome of the dependent variables according to changes in the independent factors.

In this case, the dependent variable is the log demand of electricity. The independent factors mainly include temperature, sunshine duration, several calendar binary indicators and two external shock indicators.

(i) For temperature, we included all the temperature indicators in London, Bristol and Leeds at 00h, 06h, 12h, 18h as well as the 2nd order polynomial transformation for those temperature indicators, in order to capture the non-linear pattern between temperature and demand as seen in figure 1.

(ii) Calendar parameters include a binary indicator for holidays which takes into account UK bank holidays that were extracted from the official GOV.UK websites, a weekend indicator, and month indicators. We decided to include these calendar binary parameters as we observed significant impact on demand as seen in figure 2.

(iii) Additional factors are added to account for endogeneity in electricity demand such as covid stringency index and the average monthly gas prices.

We fit all the variables into the linear regression model and use stepwise forward AIC to select the best model based on the lowest AIC score, which we obtain below.

$$\widehat{ldemand} = 10.55 - 0.00467T_{(LON,00h)} + 0.00032T^2_{(LON,00h)} - 0.00681T_{(LON,06h)} - 0.00859T_{(LON,12h)} + 0.00023T^2_{(LON,12h)} - 0.00402T_{(BRI,06h)} + 0.00004T^2_{(BRI,12h)} - 0.00327T_{(LEE,06h)} + 0.00026T_{(LEE,12h)} + 0.00026T^2_{(LEE,12h)} - 0.00006S_{LON} - 0.00005S_{LEE} - 0.00004S_{BRI} - 0.1550holiday - 0.1326weekend - 0.00004covid - 0.00020gas - 0.0120Feb - 0.0400Mar - 0.1207Apr - 0.1399May - 1.4650Jun - 0.1240Jul - 0.1257Aug - 0.1067Sep - 0.0510Oct - 0.0059Nov$$

Where $\widehat{ldemand}$ is the log of demand, $T_{(i,j)}$ is the temperature reading at city i and time j, S_i indicates sunshine duration at city i, $covid$ for covid stringency level and gas for average monthly gas prices.

Model 2: Multiple Linear regression with fourier terms (MLRF)

To further capture seasonality patterns in the time series, we generate Fourier terms in the period of 7, 30, 90, and 365 to approximate various effects of time seasonality (weekly, monthly, seasonally and yearly). To find the number of Fourier terms to use, we iterate the model over a range and calculate for the lowest AIC and found that the number of Fourier terms to use is {3, 1, 2, 3} for the period {7, 30, 90, 365} respectively.
After that, we combine all the covariates that were used previously (temperature, sunshine, calendar factor) and the Fourier terms together and fit them into a linear regression. Using the AIC score to select the best fit MLRF model and repeat all the following processes that we have done on the MLR model. We obtain the model regression as follows:

$$\widehat{ldemand} = 10.49 - 0.00479T_{(LON,00h)} + 0.00033T^2_{(LON,00h)} - 0.01119T_{(LON,06h)} + 0.00016T^2_{(LON,06h)} - 0.00273T_{(LEE,06h)} - 0.00584T_{(LEE,12h)} + 0.00032T^2_{(LEE,12h)} - 0.000004T^2_{(LEE,18h)} - 0.00017T^2_{(BRI,06h)} - 0.00638T_{(BRI,12h)} - 0.00021T^2_{(BRI,12h)} - 0.00006S_{LON} + 0.00003S_{BRI} + 0.00005S_{LEE} - 0.1508weekend - 0.1492holiday - 0.0002gas - 0.0005covid - 0.00242Feb - 0.03969Mar - 0.04068Apr - 0.02Jun - 0.03038Sep - 0.04136Oct - 0.02755Nov - 0.0212Dec - 0.00327s_{1,7} - 0.01359c_{2,7} - 0.00468s_{2,7} - 0.00807s_{3,7} - 0.0038s_{1,30} + 0.00591c_{1,90} - 0.00917s_{2,90} + 0.09313c_{1,365} - 0.00808c_{2,365} - 0.00624s_{3,365} - 0.02251c_{3,365}$$

Where $\widehat{ldemand}$ is the log of demand, $T_{(i,j)}$ is the temperature reading at city i and time j, S_i indicates sunshine duration at city i, $covid$ for covid stringency level, gas for average monthly gas prices and $s_{k,t}, c_{k,t}$ are the sine and cosine terms of the Fourier series for term k and seasonality period t.

Model 3: Dynamic ARIMA model (with regression terms)

The dynamic ARIMA model is a model which combines the ARIMA model and all the regressors that may affect the predicted values together. This model is introduced when we want to include external factors that might have an impact on the time series being modelled, as the ARIMA function only uses the historical time series data to make the future prediction, while in reality, many exogenous factors may affect the real outcome and have not been considered within the ARIMA model. Therefore, the dynamic ARIMA model allows us to simulate the relationship between time series and external variables over time. In detail, we take the covariates from the best-fit MLR model as the regressors of the dynamic ARIMA model. Then, using the *auto.arima* function to find the best fit, which in our case we found that ARIMA (1,1,1) (1,0,1) [7] is best, i.e. 1 AR term and 1 MA term with order 1 differencing, as well as 1 seasonal AR term and 1 seasonal MA term with lag of 7 days. The final ARIMA model with regression is expressed below.

$$\widehat{ldemand} = 0.5285AR_1 - 0.8026MA_1 + 0.9681SAR_1 - 0.8713SMA_1 - 0.0072T_{(LON,00h)} - 0.0003T^2_{(LON,12h)} + 0.0003T_{(LON,06h)} - 0.0051T_{(LON,12h)} + 0.0001T^2_{(LON,12h)} - 0.0029T_{(BRI,06h)} - 0.0022T_{(LEE,12h)} + 0.0001T^2_{(LEE,12h)} - 0.1216weekend - 0.1005holiday + 0.0001gas - 0.0012covid$$

Where AR_i is the auto-regression term i, SAR_i is the seasonal auto-regression term i, MA_i is the moving average term i, SMA_i is the seasonal moving average term i and $T_{(i,j)}$ is the temperature reading at city i and time j.

Model 4: TBATS Model

TBATS stands for Trigonometric seasonality, Box-Cox transformation, ARMA errors, Trend, and Seasonal components. The model approximates seasonality using trigonometric representation based on Fourier series, followed by a sequence of time series transformation and modelling as the name suggests. TBATS can handle multiple components of time series and large periods of seasonality generally well. We added bias adjustment to our TBATS model parameter to ensure that the model predicts the mean, instead of using median for the forecast.

The final TBATS model trained was BATS(1, {5,2}, 0.945, -), which has the following parameters: $\lambda = 0.999727 \approx 1$ (or approximately no Box-Cox transformation), $\phi = 0.944808 \approx 0.945$ (close to total dampening), with 5 AR terms with the coefficients {0.410437, -0.826076, -0.084469, -0.41583, -0.328359} and 2 MA terms with the coefficients {-0.86241, 0.691923}.

Model 5: STL-ETS Model

Since the usual ETS (Error, Trend, Seasonality) model is unable to capture long periods of seasonality, we use STL (seasonal-trend decomposition with Loess) in conjunction to decompose the seasonality before passing the seasonally adjusted time-series data into the ETS model. We found that this model fares quite well in capturing the seasonality. The final STL-ETS model that we trained has ETS parameter of (M, N, N) i.e. multiplicative error terms with no trend and seasonality addition in the ETS component. The STL would have captured the seasonality and trend in the model. The smoothing parameter used is $\alpha = 0.1603$.

To calculate the final forecast, we generate the forecast for the next day for all 5 models, get the weighted average based on inverse RMSE and finally multiply by the correction term, γ. The final forecast is therefore expressed as follows:

$$FC = \gamma \cdot \sum_{i=1}^{5} w_i \cdot FC_i$$

Where FC is the final forecast, FC_i is the forecast of the individual model i, w_i is the weight of each individual model i which can be expressed as $w_i = \frac{1/RMSE_i}{\Sigma_i 1/RMSE_i}$, and γ is the correction multiplier term.

In our model, we take the standard deviation to be the squared sum of errors of the combined model forecast divided by the degree of freedom (i.e. length of the time series data minus 1). To calculate this, we generate the fitted values of all the 5 models and average them based on the weightage aforementioned. We then take the difference between the combined fitted values and the actual load demand to obtain the combined model errors, u.

Therefore, the standard deviation is calculated as

$$\sigma = \sqrt{\frac{1}{N-1} \Sigma_j \, {u_j}^2} = \sqrt{\frac{1}{N-1} \sum_{j}^{N} [\, (\sum_{i=1}^{5} w_i \cdot \widehat{ldemand}_{i,j}\,) \, - ldmand_j \,]^2}$$

Where i indicates the model used and j indicates the period in the time series data.

Lastly, before finalising our forecast, we also perform sanity checks on the forecast by comparing it against the past week's demands and BBC's weather forecasts. For example, if the past week's log demand on Monday with an average temperature of 2° is 10.31 and BBC's weather forecast for next Monday stated that it would be sunny weather with a 10° average temperature, we should expect that our predicted log demand for the next Monday is lower than 10.31.

III. Forecasting Results and Evaluation

Our forecasting results are summarised in table 3 below, including the forecast errors and the model used in each round. Overall, we believe that we have continuously improved our forecasts and achieved a stable result from round 4 onwards, where the forecast errors fell within the range of the standard deviation. However, we noted that the last forecasting round shows a slightly higher forecasting error at 4%, with the log demand higher than predicted despite the weather condition being similar to the previous forecasting round. This could mean that there is probably another endogenous factor we have yet to account for, causing a shock to the demand.

Table 3. Summary of forecasting rounds

Round	Model	Forecast	Demand	Error	Qualitative Evaluation
Round 1 *7 March 2023*	Multivariate Linear Regression	10.39272 *(0.0572746)*	10.291510	+0.1012103 *(+10.65%)*	Missing independent variables in the regression: (i) covid stringency, (ii) gas price index
Round 2 *9 March 2023*	Dynamic ARIMA (with regression)	10.31634 *(0.0335376)*	10.338284	-0.0219442 *(-2.17%)*	MLR performs poorly due to missing variables, use next best model ARIMA
Round 3 *11 March 2023*	Combination I (Simple Average)	10.21603 *(0.0346656)*	10.195270	+0.0207602 *(+2.10%)*	Combine model forecast: MLR, MLRF, ARIMA (xreg), STLF and TBATs with simple average
Round 4 *14 March 2023*	Combination I (Simple Average)	10.28061 *(0.0343804)*	10.197302	+0.0833084 *(+8.69%)*	Observed model's tendency to overpredict demand, to correct overprediction next round
Round 5 *16 March 2023*	Combination II (Weighted, Adjusted)	10.20503 *(0.0339595)*	10.196676	+0.0083524 *(+0.84%)*	Inverse of RMSE is used as weights and correction multiplier is added to manually tone down overprediction
Round 6 *18 March 2023*	Combination II (Weighted, Adjusted)	10.08793 *(0.0339504)*	10.065277	+0.0226532 *(+2.29%)*	Acceptable forecast performance, within standard deviation
Round 7 *21 March 2023*	Combination II (Weighted, Adjusted)	10.10816 *(0.0342660)*	10.129258	-0.0210977 *(-2.09%)*	Acceptable forecast performance, within standard deviation
Round 8 *23 March 2023*	Combination II (Weighted, Adjusted)	10.14059 *(0.0343902)*	10.100403	-0.0401876 *(-4.10%)*	Slightly outside standard deviation, despite temperature and sunshine similar to round 7

Note: column 'Forecast' contains the point forecast value and its standard deviation in brackets, column 'Error' contains the forecast error from the actual log demand and the corresponding percentage error based on actual demand (non-logarithmic term).

IV. Improvements and Extension

One possible extension to our model is to explore more forecasting models such as Support Vector Regression (SVR) model or Vector Auto-Regression (VAR) model in our range of sub-models. This way we have a greater range of potential models to combine. Furthermore, we may want to explore additional parameters which have yet to be accounted for in our regression model, such as macro- and micro-economic trends in the UK, which may affect business and commercial activities hence impacting the consumption of electricity in commercial buildings and facilities.

V. Conclusion

To conclude our report, we have introduced our final forecasting model which was formed by combining the forecast of the 5 best performing models based on RMSE values. The combined forecast was calculated by taking the weighted average based on their inverse RMSE values and multiplying a correction term to minimise the overprediction. The standard deviation of the model is calculated by assuming similarity with the standard error, taking the square root of the combined sum of squared error of the models divided by the degree of freedom. Overall, our model performed relatively well, except for certain days when demand spiked unexpectedly.

VI. References

[1] Santos, L.L. & Castilo F.S. (2019) *Introduction to Spatial Network Forecast with R*. https://laurentlsantos.github.io/forecasting/support-vector-regression.html. Accessed on 5 March 2023.

[2] Hyndman, R.J., & Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. Accessed on 12 March 2023.

[3] Bates, J. M., & Granger, C. W. J. (1969). The combination of forecasts. Operational Research Quarterly, 20(4), 451–468. [DOI]

[4] Clemen, R. (1989). Combining forecasts: A review and annotated bibliography. International Journal of Forecasting, 5(4), 559–583. [DOI]