# Experimentation

## Hardware

The tests were carried out on a PC with a Windows OS. The computer's features are described in the following:

* GPU: Intel Nvidia RTX 2070
* CPU: Intel Core i7 8th Generation
* RAM: 16GB

## Software

Python is the programming language utilised. The experiment is carried out in a virtual environment generated with Visual Studio Code and Python 3.9.0. The relevant packages are utilised in the experiment. A Jupyter Notebook was used to write and run the code. The network was trained using the Nvidia RTX 2070 GPU, which has 16 GB VRAM and CUDA version 11.6.

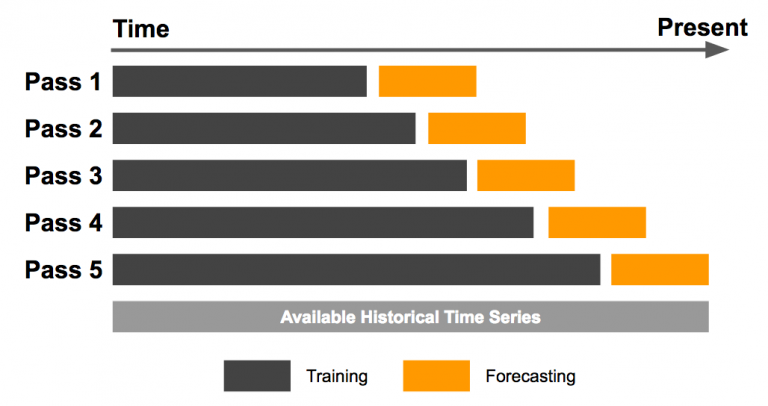
* TensorFlow
* Keras
* Scikit-learn
* Pandas
* Prophet
* Matplotlib
* NumPy
* Seaborn

## Settings

The challenge is to create a model using the S2S LSTM as well as BLST technique for correctly predicting a structure's energy usage using Machine Learning, ensemble, and even the Facebook Prophet methodologies. Predictions of energy use across various time periods and train/test splits are among the experiments. This time-series regression issue, if successful, can be used as proof for employing a learning algorithm in energy consumption prediction by applying past data of energy (Electricity, Water, and Gas) usage and weather conditions.

# Training

A typical cross-validation method, in which random points are utilised in each fold, cannot be performed in this situation due to the time constraint. The time series must be preserved; thus, the folds do not include random selection; rather, every fold increases the volume of training data. Each fold has its own test set, and the preceding fold's test set is appended to the current fold's training set for the next fold. An example of the expanding window is below:



# Experiments and Results

This chapter will go about performing ‘black-box’ testing to evaluate or created models and afterwards a visualization of the models will be shown ranking by the model with the best performance. All the models will be tested against a baseline prediction calculated using the simple persistence algorithm. The Zero Rule Algorithm is by far the most frequent baseline method for monitoring ML. In the instance of classifications, the algorithm correctly forecasts most classes. With regression cases, it predicts the expected average. This can be utilized with time-series data, but the sequence correlation architecture in the data set is not considered. The persistence algorithm is a similar technique for time series data sets. The persistence method predicts the expected outcome for the next step based on the last step . The following baseline forecast conditions must be fulfilled:

* **Simple**： A technique that does not need extensive training or understanding.
* **Fast**： Predicting in a quick and computationally simple manner.
* **Repeatable**： A deterministic technique gives the expected output when given the same input.

## Queen’s Building

### Baseline

Table showing the Queens Building Hourly Baseline

|  |  |  |  |
| --- | --- | --- | --- |
|  | Electric | Gas | Water |
| RMSE | 4.265 | 152110.65 | 0.057 |
| MSE | 18.190 | 23137650036.14 | 0.003 |
| MAE | 2.4481 | 726.80 | 0.034 |
| R2 | 0.944 | -2.000 | 0.895 |

From the table above we can see the gas dataset metrics could not accurately be measured which means a further investigation of the plot is needed.

Chart, line chart

Description automatically generated

The plot of Queen’s Building Gas Hourly Dataset

Based on the figure above we can see that the hourly dataset could not be plotted due to the fact we are dealing with data lower than the program can compute. The electric and water datasets produced very good metrics that could help train the desired algorithm, but with the gas dataset producing such results, it could degrade the performance.

Table showing the Queens Building Daily Baseline

|  |  |  |  |
| --- | --- | --- | --- |
|  | Electric | Gas | Water |
| RMSE | 8.217 | 53.023 | 0.071 |
| MSE | 67.521 | 2811.472 | 0.005 |
| MAE | 4.996 | 26.286 | 0.043 |
| R2 | 0.644 | 0.540 | 0.744 |

The daily baseline shows very good results and a perfect univariate baseline to try to beat for the chosen algorithms. These are all baseline metrics from a univariate model in hopes that adding regressors would improve it.

### LSTM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Layers** | **Neurons** | **Epochs** | **Lag Variable** | **Metrics** | **Beaten?** |
| 2 | 50 | 50 | 6 hours | RMSE: 4.089  MSE: 16.723  MAE: 2.639  : 0.944 | Yes |
| 50 | 200 | 6 hours | RMSE: 4.180  MSE: 17.473  MAE: 2.595  : 0.942 | Yes |
| 150 | 50 | 6 hours | RMSE: 5.066  MSE: 25.664  MAE: 3.940  : 0.914 | No |
| 6 | 50 | 50 | 6 hours | RMSE: 9.510  MSE: 90.433  MAE: 7.489  : 0.698 | No |
| 8 | 32 | 50 | 6 hours | RMSE: 11.859  MSE: 140.642  MAE: 9.564  : 0.531 | No |

We can observe from the table above that if the LSTM model contains two layers, it improves the baseline RMSE. Another interesting fact is that the number of neurons is capped at 50; much greater and the RMSE begins to exceed the baseline. Due to time constraints, the remaining tests will be tested using the Daily dataset.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Layers** | **Neurons** | **Epochs** | **Training** | **Metrics** | **Beaten?** |
| 2 | 50 | 50 | 4 years | RMSE: 6.864  MSE: 47.117  MAE: 5.471  : 0.662 | Yes |
| 50 | 500 | 4 years | RMSE: 6.039  MSE: 36.465  MAE: 4.404  : 0.738 | Yes |
| 150 | 500 | 5 years | RMSE: 5.619  MSE: 31.576  MAE: 4.162  : 0.772 | Yes |
| 300 | 1000 | 5 years | RMSE: 6.707  MSE: 44.980  MAE: 4.972  : 0.675 | Yes |
| 1000 | 500 | 5 years | RMSE: 5.546  MSE: 30.761  MAE: 4.247  : 0.778 | Yes |
| 6 | 50 | 50 | 5 years | RMSE: 10.842  MSE: 117.539  MAE: 9.111  : 0.150 | No |
| 50 | 500 | 5 years | RMSE: 5.923  MSE: 35.081  MAE: 4.442  : 0.746 | Yes |
| 150 | 500 | 5 years | RMSE: 7.003  MSE: 49.041  MAE: 5.424  : 0.645 | Yes |
| 300 | 1000 | 5 years | RMSE: 6.623  MSE: 43.861  MAE: 5.032  : 0.683 | Yes |
| 8 | 50 | 500 | 5 years | RMSE: 7.200  MSE: 51.838  MAE: 5.470  : 0.625 | Yes |

Based on the extensive tests carried out, we can see that the LSTM performs very well and improves the RMSE. The LSTM with 2 layers and 1000 input neurons had the better metric. From this, we can assume that the BLSTM would also show similar results or even better.

### BLSTM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Layers** | **Neurons** | **Epochs** | **Train Variable** | **Metrics** | **Beaten?** |
| 2 | 50 | 50 | 5 years | RMSE: 5.480  MSE: 30.025  MAE: 4.173  : 0.783 | Yes |
| 50 | 200 | 5 years | RMSE: 4.817  MSE: 23.206  MAE: 3.508  : 0.832 | Yes |
| 200 | 200 | 5 years | RMSE: 4.807  MSE: 23.104  MAE: 3.591  : 0.833 | Yes |
| 6 | 50 | 50 | 5 years | RMSE: 7.200  MSE: 51.839  MAE: 5.744  : 0.625 | Yes |
|  | 150 | 500 | 5 years | RMSE: 6.624  MSE: 43.883  MAE: 5.012  : 0.683 | Yes |
| 8 | 50 | 200 | 5 years | RMSE: 11.280  MSE: 127.247  MAE: 9.271  : 0.080 | No |

Based on the extensive tests carried out, we can see that the LSTM performs very well and improves the RMSE. The BLSTM with 2 layers and 200 input neurons had the better metric. This proves the statement made in the literature that for a bidirectional LSTM, only 2 layers are needed for it to work effectively. Therefore, the rest of the test would only be carried out using only 2 layers of both the LSTM and BLSTM.

### XGBoost

|  |  |  |  |
| --- | --- | --- | --- |
| **Iterations** | **Learning Rate** | **Metrics** | **Beaten?** |
| 50 | 0.001 | RMSE: 51.521  MSE: 2654.385  MAE: 49.731  : -12.798 | No |
| 0.01 | RMSE: 33.547  MSE: 1125.394  MAE: 31.791  : -4.850 | No |
| 0.1 | RMSE: 7.542  MSE: 56.885  MAE: 5.843  : 0.704 | Yes |
| 200 | 0.001 | RMSE: 44.606  MSE: 1989.701  MAE: 42.860  : -9.343 | No |
| 0.1 | RMSE: 7.575  MSE: 57.374  MAE: 5.791  : 0.702 | Yes |
| 500 | 0.001 | RMSE: 33.618  MSE: 1130.203  MAE: 31.863  : -4.875 | No |
| 0.1 | RMSE: 7.701  MSE: 59.306  MAE: 5.851  : 0.692 | Yes |
| 2000 | 0.001 | RMSE: 10.989  MSE: 120.751  MAE: 9.065  : 0.372 | No |
| 0.01 | RMSE: 7.552  MSE: 57.028  MAE: 5.772  : 0.704 | Yes |
| 5000 | 0.001 | RMSE: 7.607  MSE: 57.871  MAE: 5.916  : 0.699 | Yes |
| 0.1 | RMSE: 7.819  MSE: 61.134  MAE: 5.931  : 0.682 | Yes |
| 25000 | 0.001 | RMSE: 7.528  MSE: 56.678  MAE: 5.757  : 0.705 | Yes |
| 0.1 | RMSE: 7.819  MSE: 61.134  MAE: 5.931  : 0.682 | Yes |

From this test, we can see that as the number of iterations and learning rate increases it improves the RMSE. In the iterations 50, 200 and 500, the best learning rate is 0.1 but in the later iterations especially 25,000, it was discovered that the learning was instead 0.001 producing the lowest RMSE.

### LightGBM

|  |  |  |  |
| --- | --- | --- | --- |
| **Iterations** | **Learning Rate** | **Metrics** | **Beaten?** |
| 50 | 0.001 | RMSE: 13.443  MSE: 180.715  MAE: 10.954  : 0.061 | No |
| 0.01 | RMSE: 10.676  MSE: 113.977  MAE: 8.554  : 0.408 | No |
| 0.1 | RMSE: 7.576  MSE: 57.400  MAE: 5.907  : 0.702 | Yes |
| 200 | 0.001 | RMSE: 12.322  MSE: 151.829  MAE: 9.977  : 0.211 | No |
| 0.1 | RMSE: 7.637  MSE: 58.325  MAE: 5.885  : 0.697 | Yes |
| 500 | 0.001 | RMSE: 10.689  MSE: 114.254  MAE: 8.565  : 0.406 | No |
| 0.1 | RMSE: 7.848  MSE: 61.598  MAE: 6.050  : 0.680 | Yes |
| 2000 | 0.001 | RMSE: 8.116  MSE: 65.873  MAE: 6.374  : 0.658 | Yes |
| 0.01 | RMSE: 7.647  MSE: 58.477  MAE: 5.886  : 0.696 | Yes |
| 5000 | 0.001 | RMSE: 7.554  MSE: 57.061  MAE: 5.861  : 0.703 | Yes |
| 0.1 | RMSE: 8.047  MSE: 64.761  MAE: 6.202  : 0.663 | Yes |
| 25000 | 0.001 | RMSE: 7.617  MSE: 58.014  MAE: 5.833  : 0.698 | Yes |
| 0.1 | RMSE: 8.048  MSE: 64.763  MAE: 6.202  : 0.663 | Yes |

From this test, we can see that as the number of iterations and learning rate increases it improves the RMSE. In the iterations 50, 200 and 500, the best learning rate is 0.1 but in the later iterations especially 5,000, it was discovered that the learning was instead 0.001 producing the lowest RMSE.

### Prophet

|  |  |  |  |
| --- | --- | --- | --- |
| **Trainable Days** | **Parameter Added** | **Metrics** | **Beaten?** |
| 4700 | Interval\_width: 0.9 | RMSE: 4.900  MSE: 24.011  MAE: 3.811  : 0.330 | Yes |
| Interval\_width: 0.9  UK holidays | RMSE: 5.136  MSE: 26.382  MAE: 3.930  : 0.264 | Yes |
| Interval\_width: 0.9  UK holidays  Monthly seasonality | RMSE: 5.117  MSE: 26.188  MAE: 3.935  : 0.269 | Yes |
| 4000 | Interval width: 0.9 | RMSE: 7.209  MSE: 51.964  MAE: 5.597  : 0.514 | Yes |
| Interval\_width: 0.9  UK holidays | RMSE: 7.263  MSE: 52.753  MAE: 5.639  : 0.507 | Yes |
| Interval\_width: 0.9  UK holidays  Monthly seasonality | RMSE: 7.268  MSE: 58.829  MAE: 5.650  : 0.506 | Yes |
| 3500 | Interval\_width: 0.9 | RMSE: 7.329  MSE: 53.179  MAE: 5.928  : 0.547 | Yes |
| Interval\_width: 0.9  UK holidays | RMSE: 7.295  MSE: 53.211  MAE: 5.907  : 0.551 | Yes |
| Interval\_width: 0.9  UK holidays  Monthly seasonality | RMSE: 7.267  MSE: 52.814  MAE: 5.880  : 0.555 | Yes |
| 3000 | Interval\_width: 0.9 | RMSE: 21.462  MSE: 460.623  MAE: 18.605  : -2.748 | No |

We can see from the extensive test that the Prophet algorithm performed extremely well in terms of RMSE and shows that it could be used for energy forecasting. Training for 3500 days (roughly 10 years) yielded lower RMSE but an score of 55% (depicted below).

Chart, histogram

Description automatically generated

## Hugh Aston

### Baseline

Table showing the Hugh Aston Hourly Baseline

|  |  |  |  |
| --- | --- | --- | --- |
|  | Electric | Gas | Water |
| RMSE | 7.713 | 27.48 | 0.131 |
| MSE | 59.483 | 755.156 | 0.017 |
| MAE | 4.646 | 12.024 | 0.083 |
| R2 | 0.917 | 0.827 | 0.853 |

Table showing the Hugh Aston Daily Baseline

|  |  |  |  |
| --- | --- | --- | --- |
|  | Electric | Gas | Water |
| RMSE | 13.573 | 25.191 | 0.172 |
| MSE | 184.23 | 634.58 | 0.03 |
| MAE | 8.354 | 14.7 | 0.115 |
| R2 | 0.34 | 0.803 | 0.36 |

### LSTM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Layers** | **Neurons** | **Epochs** | **Training** | **Metrics** | **Beaten?** |
| 2 | 50 | 50 | 4 years | RMSE: 8.774  MSE: 76.976  MAE: 6.613  : 0.731 | Yes |
| 50 | 500 | 4 years | RMSE: 8.408  MSE: 70.702  MAE: 5.981  : 0.752 | Yes |
| 150 | 500 | 4 years | RMSE: 8.739  MSE: 76.368  MAE: 6.174  : 0.739 | Yes |
| 300 | 1000 | 4 years | RMSE: 8.132  MSE: 66.137  MAE: 6.015  : 0.774 | Yes |
| 1000 | 500 | 4 years | RMSE: 7.862  MSE: 61.816  MAE: 5.953  : 0.788 | Yes |
| 6 | 50 | 50 | 4 years | RMSE: 16.893  MSE: 285.381  MAE: 15.113  : 0.023 | No |
| 50 | 500 | 4 years | RMSE: 11.091  MSE: 123.015  MAE: 7.805  : 0.579 | Yes |
| 150 | 500 | 4 years | RMSE: 9.345  MSE: 87.331  MAE: 6.707  : 0.701 | Yes |
| 300 | 1000 | 4 years | RMSE: 10.099  MSE: 101.982  MAE: 7.738  : 0.651 | Yes |
| 8 | 50 | 500 | 4 years | RMSE: 9.602  MSE: 92.192  MAE: 6.738  : 0.684 | Yes |

Based on the test carried out, this building’s LSTM statistics show that the 2-layered structure provided the best RMSE results. Unlike the Queen’s Building LSTM results, it only had a single instance of the network not outperforming the baseline results. On the 2-layered structure, the higher the input neurons the better the results with a computation time trade-off.

### BLSTM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Layers** | **Neurons** | **Epochs** | **Training** | **Metrics** | **Beaten?** |
| 2 | 50 | 50 | 4 years | RMSE: 7.873  MSE: 61.977  MAE: 5.692  : 0.783 | Yes |
| 50 | 500 | 4 years | RMSE: 7.582  MSE: 57.484  MAE: 5.581  : 0.799 | Yes |
| 150 | 500 | 4 years | RMSE: 7.413  MSE: 54.955  MAE: 5.354  : 0.808 | Yes |
| 300 | 1000 | 4 years | RMSE: 8.783  MSE: 77.142  MAE: 6.449  : 0.730 | Yes |
| 1000 | 500 | 4 years | RMSE: 8.268  MSE: 68.360  MAE: 5.910  : 0.761 | Yes |
| 6 | 50 | 50 | 4 years | RMSE: 12.031  MSE: 144.750  MAE: 9.379  : 0.505 | Yes |
| 50 | 500 | 4 years | RMSE: 11.579  MSE: 134.073  MAE: 8.638  : 0.541 | Yes |
| 150 | 500 | 4 years | RMSE: 8.535  MSE: 72.843  MAE: 5.893  : 0.751 | Yes |
| 300 | 1000 | 4 years | RMSE: 9.362  MSE: 87.643  MAE: 6.998  : 0.700 | Yes |
| 8 | 50 | 500 | 4 years | RMSE: 9.738  MSE: 94.829  MAE: 7.050  : 0.668 | Yes |

Based on the test carried out, this building’s BLSTM statistics show that the 2-layered structure provided the best RMSE results. On the 2-layered structure, the input neuron is capped at 150 as any higher the RMSE starts to trail off.

### XGBoost

|  |  |  |  |
| --- | --- | --- | --- |
| **Iterations** | **Learning Rate** | **Metrics** | **Beaten?** |
| 50 | 0.001 | RMSE: 62.211  MSE: 3870.216  MAE: 60.174  : -13.611 | No |
| 0.01 | RMSE: 40.385  MSE: 1630.946  MAE: 38.390  : -5.127 | No |
| 0.1 | RMSE: 9.326  MSE: 86.973  MAE: 7.073  : 0.672 | Yes |
| 200 | 0.001 | RMSE: 53.812  MSE: 2895.756  MAE: 51.827  : -9.932 | No |
| 0.1 | RMSE: 9.361  MSE: 87.622  MAE: 7.093  : 0.669 | Yes |
| 500 | 0.001 | RMSE: 40.477  MSE: 1638.348  MAE: 38.400  : -5.185 | No |
| 0.1 | RMSE: 9.518  MSE: 90.600  MAE: 7.214  : 0.658 | Yes |
| 2000 | 0.001 | RMSE: 13.015  MSE: 169.388  MAE: 10.837  : 0.361 | Yes |
| 0.1 | RMSE: 9.617  MSE: 92.492  MAE: 7.309  : 0.651 | Yes |
| 5000 | 0.001 | RMSE: 9.270  MSE: 85.926  MAE: 7.063  : 0.676 | Yes |
| 0.1 | RMSE: 9.621  MSE: 92.554  MAE: 7.312  : 0.651 | Yes |
| 25000 | 0.001 | RMSE: 9.270  MSE: 85.938  MAE: 7.034  : 0.676 | Yes |
| 0.1 | RMSE: 9.621  MSE: 92.554  MAE: 7.312  : 0.651 | Yes |

From this test, we can see that as the number of iterations and learning rate increases it improves the RMSE. In the iterations 50, 200 and 500, the best learning rate is 0.1 but in the later iterations especially 25,000, it was discovered that the learning was instead 0.001 producing the lowest RMSE.

### LightGBM

|  |  |  |  |
| --- | --- | --- | --- |
| **Iterations** | **Learning Rate** | **Metrics** | **Beaten?** |
| 50 | 0.001 | RMSE: 15.796  MSE: 249.509  MAE: 13.255  : 0.06 | No |
| 0.01 | RMSE: 12.537  MSE: 157.167  MAE: 10.338  : 0.407 | Yes |
| 0.1 | RMSE: 9.268  MSE: 85.891  MAE: 7.039  : 0.676 | Yes |
| 200 | 0.001 | RMSE: 14.475  MSE: 209.528  MAE: 12.059  : 0.209 | No |
| 0.1 | RMSE: 9.283  MSE: 86.177  MAE: 7.034  : 0.675 | Yes |
| 500 | 0.001 | RMSE: 12.548  MSE: 157.457  MAE: 10.349  : 0.406 | Yes |
| 0.1 | RMSE: 9.573  MSE: 91.635  MAE: 7.253  : 0.654 | Yes |
| 2000 | 0.001 | RMSE: 9.639  MSE: 92.903  MAE: 7.519  : 0.649 | Yes |
| 0.1 | RMSE: 9.779  MSE: 95.625  MAE: 7.411  : 0.639 | Yes |
| 5000 | 0.001 | RMSE: 9.227  MSE: 85.147  MAE: 7.021  : 0.679 | Yes |
| 0.1 | RMSE: 9.791  MSE: 95.868  MAE: 7.420  : 0.638 | Yes |
| 25000 | 0.001 | RMSE: 9.298  MSE: 86.458  MAE: 7.079  : 0.674 | Yes |
| 0.1 | RMSE: 9.791  MSE: 95.869  MAE: 7.420  : 0.638 | Yes |

From this test, we can see that as the number of iterations and learning rate increases it improves the RMSE. In the iterations 50 and 200, the best learning rate is 0.1 but in the later iterations especially 5,000, it was discovered that the learning was instead 0.001 producing the lowest RMSE.

### Prophet

|  |  |  |  |
| --- | --- | --- | --- |
| **Trainable Days** | **Parameter Added** | **Metrics** | **Beaten?** |
| 3500 | Interval\_width: 0.9 | RMSE: 17.288  MSE: 298.868  MAE: 14.134  : 0.220 | No |
| Interval\_width: 0.9  UK holidays | RMSE: 17.005  MSE: 289.160  MAE: 13.950  : 0.245 | No |
| Interval\_width: 0.9  UK holidays  Monthly seasonality | RMSE: 17.010  MSE: 289.344  MAE: 13.972  : 0.245 | No |
| 2900 | Interval width: 0.9 | RMSE: 12.882  MSE: 165.940  MAE: 9.857  : 0.486 | Yes |
| Interval\_width: 0.9  UK holidays | RMSE: 12.859  MSE: 165.365  MAE: 9.663  : 0.488 | Yes |
| Interval\_width: 0.9  UK holidays  Monthly seasonality | RMSE: 12.912  MSE: 166.719  MAE: 9.683  : 0.484 | Yes |
| 2100 | Interval\_width: 0.9 | RMSE: 16.271  MSE: 264.751  MAE: 11.897  : 0.119 | No |
| Interval\_width: 0.9  UK holidays | RMSE: 16.284  MSE: 265.174  MAE: 11.734  : 0.118 | No |
| Interval\_width: 0.9  UK holidays  Monthly seasonality | RMSE: 16.524  MSE: 273.052  MAE: 11.907  : 0.091 | No |
| 2200 | Interval\_width: 0.9 | RMSE: 16.222  MSE: 263.169  MAE: 11.944  : 0.122 | No |

We can see from the extensive test that the Prophet algorithm did not perform as expected and could only surpass the baseline by a hair. The number of trainable days that were successful is 2900 days and the RMSE achieved was through the addition of the in-built Prophet function of country holidays as regressors. (See diagram below). This shows the forecast in blue and the actual dataset in red; the algorithm seemed to struggle from April 2020 due to the country going into sudden lockdown, but Prophet shows to an extent what the normal consumption rate would have been.

Chart, histogram

Description automatically generated

## Gateway House

### Baseline

|  |  |  |  |
| --- | --- | --- | --- |
| Hourly Baseline | | | |
|  | Electric | Gas | Water |
| RMSE | 16753.53 | 34.104 | 0.134 |
| MSE | 280680631.65 | 1163.092 | 0.018 |
| MAE | 83.353 | 10.998 | 0.047 |
| R2 | -1.999 | 0.702 | 0.741 |

|  |  |  |  |
| --- | --- | --- | --- |
| Daily Baseline | | | |
|  | Electric | Gas | Water |
| RMSE | 16.602 | 31.317 | 0.124 |
| MSE | 275.639 | 980.762 | 0.015 |
| MAE | 9.609 | 14.89 | 0.072 |
| R2 | 0.54 | 0.411 | 0.271 |

### LSTM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Layers** | **Neurons** | **Epochs** | **Training** | **Metrics** | **Beaten?** |
| 2 | 50 | 50 | 7 years | RMSE: 10.537  MSE: 111.028  MAE: 8.845  : 0.582 | Yes |
| 50 | 500 | 7 years | RMSE: 12.892  MSE: 166.216  MAE: 10.957  : 0.375 | Yes |
| 150 | 500 | 7 years | RMSE: 9.995  MSE: 99.891  MAE: 8.245  : 0.624 | Yes |
| 300 | 1000 | 7 years | RMSE: 7.710  MSE: 59.437  MAE: 5.942  : 0.776 | Yes |
| 1000 | 500 | 7 years | RMSE: 7.196  MSE: 51.777  MAE: 5.720  : 0.805 | Yes |
| 6 | 50 | 50 | 7 years | RMSE: 13.081  MSE: 171.117  MAE: 10.644  : 0.356 | Yes |
| 50 | 500 | 7 years | RMSE: 13.872  MSE: 192.421  MAE: 11.322  : 0.276 | Yes |
| 150 | 500 | 7 years | RMSE: 10.423  MSE: 108.649  MAE: 8.039  : 0.591 | Yes |
| 300 | 1000 | 7 years | RMSE: 13.464  MSE: 181.281  MAE: 10.404  : 0.318 | Yes |
| 8 | 50 | 500 | 7 years | RMSE: 25.333  MSE: 641.757  MAE: 21.502  : -1.415 | No |

Based on the test carried out, this building’s LSTM statistics show that the 2-layered structure once again provided the best RMSE results. On the 2-layered structure, it showed that the best result occurred with 1000 input neurons and 500 epochs. 300 input neurons were also tested and showed the second-best results. After 8 layers, the model fails to adequately train and produces very bad results. This suggests more fine-tuning is needed.

### BLSTM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Layers** | **Neurons** | **Epochs** | **Training** | **Metrics** | **Beaten?** |
| 2 | 50 | 50 | 7 years | RMSE: 9.369  MSE: 87.778  MAE: 7.550  : 0.670 | Yes |
| 50 | 500 | 7 years | RMSE: 9.395  MSE: 88.271  MAE: 7.769  : 0.688 | Yes |
| 150 | 500 | 7 years | RMSE: 10.064  MSE: 101.280  MAE: 8.375  : 0.619 | Yes |
| 300 | 1000 | 7 years | RMSE: 11.185  MSE: 125.098  MAE: 9.606  : 0529 | Yes |
| 6 | 50 | 50 | 7 years | RMSE: 10.866  MSE: 118.065  MAE: 9.052  : 0.556 | Yes |
| 50 | 500 | 7 years | RMSE: 14.330  MSE: 205.358  MAE: 12.438  : 0.227 | Yes |
| 8 | 50 | 50 | 7 years | RMSE: 14.745  MSE: 217.404  MAE: 12.782  : 0.182 | Yes |

Based on the test carried out, this building’s BLSTM statistics show that the 2-layered structure provided the best RMSE results. On the 2-layered structure, the input neuron is capped at 50 as any higher the RMSE starts to increase till a sudden drop at 150 neurons.

### XGBoost

|  |  |  |  |
| --- | --- | --- | --- |
| **Iterations** | **Learning Rate** | **Metrics** | **Beaten?** |
| 50 | 0.001 | RMSE: 65.773  MSE: 4326.104  MAE: 61.483  : -6.415 | No |
| 0.01 | RMSE: 43.137  MSE: 1860.776  MAE: 39.142  : -2.190 | No |
| 0.1 | RMSE: 12.981  MSE: 168.502  MAE: 9.833  : 0.711 | Yes |
| 200 | 0.001 | RMSE: 57.034  MSE: 3252.834  MAE: 52.909  : -4.576 | No |
| 0.1 | RMSE: 12.882  MSE: 165.958  MAE: 9.651  : 0.716 | Yes |
| 500 | 0.001 | RMSE: 43.226  MSE: 1868.479  MAE: 39.232  : -2.203 | No |
| 0.1 | RMSE: 13.089  MSE: 171.322  MAE: 9.788  : 0.706 | Yes |
| 2000 | 0.001 | RMSE: 16.102  MSE: 260.089  MAE: 12.626  : 0.554 | Yes |
| 0.1 | RMSE: 13.153  MSE: 173.003  MAE: 9.833  : 0.703 | Yes |
| 5000 | 0.001 | RMSE: 13.012  MSE: 169.305  MAE: 9.899  : 0.710 | Yes |
| 0.1 | RMSE: 13.155  MSE: 173.042  MAE: 9.836  : 0.703 | Yes |
| 25000 | 0.001 | RMSE: 12.894  MSE: 166.250  MAE: 9.679  : 0.715 | Yes |
| 0.1 | RMSE: 13.155  MSE: 173.042  MAE: 9.836  : 0.703 | Yes |

From this test, we can see that as the number of iterations and learning rate increases it improves the RMSE. In the iterations 50, 200 and 500, the best learning rate is 0.1. The iterations with the best RMSE are when the model is trained on 200 iterations.

### LightGBM

|  |  |  |  |
| --- | --- | --- | --- |
| **Iterations** | **Learning Rate** | **Metrics** | **Beaten?** |
| 50 | 0.001 | RMSE: 23.350  MSE: 545.244  MAE: 19.682  : 0.065 | No |
| 0.01 | RMSE: 18.126  MSE: 328.546  MAE: 15.167  : 0.437 | No |
| 0.1 | RMSE: 12.873  MSE: 165.706  MAE: 9.576  : 0.716 | Yes |
| 200 | 0.001 | RMSE: 21.242  MSE: 451.229  MAE: 17.849  : 0.227 | No |
| 0.1 | RMSE: 12.913  MSE: 166.783  MAE: 9.668  : 0.714 | Yes |
| 500 | 0.001 | RMSE: 18.142  MSE: 329.149  MAE: 15.184  : 0.436 | No |
| 0.1 | RMSE: 13.306  MSE: 177.046  MAE: 9.999  : 0.697 | Yes |
| 2000 | 0.001 | RMSE: 13.549  MSE: 183.568  MAE: 10.764  : 0.685 | Yes |
| 0.1 | RMSE: 13.660  MSE: 186.609  MAE: 10.339  : 0.680 | Yes |
| 5000 | 0.001 | RMSE: 12.873  MSE: 165.714  MAE: 9.765  : 0.716 | Yes |
| 0.1 | RMSE: 13.681  MSE: 187.171  MAE: 10.357  : 0.679 | Yes |
| 25000 | 0.001 | RMSE: 12.952  MSE: 167.762  MAE: 9.694  : 0.712 | Yes |
| 0.1 | RMSE: 13.681  MSE: 187.174  MAE: 10.357  : 0.679 | Yes |

From this test, we can see that as the number of iterations and learning rate increases it improves the RMSE. Increasing iterations from 50 to 500 can see that the RMSE improves as the learning rate increases. The opposite is seen for iterations in the 1000s.

### Prophet

|  |  |  |  |
| --- | --- | --- | --- |
| **Trainable Days** | **Parameter Added** | **Metrics** | **Beaten?** |
| 4000 | Interval\_width: 0.9 | RMSE: 7.718  MSE: 59.564  MAE: 5.864  : 0.666 | Yes |
| Interval\_width: 0.9  UK holidays | RMSE: 8.572  MSE: 73.480  MAE: 6.592  : 0.588 | Yes |
| Interval\_width: 0.9  UK holidays  Monthly seasonality | RMSE: 8.662  MSE: 75.025  MAE: 6.670  : 0.579 | Yes |
| 4500 | Interval width: 0.9 | RMSE: 7.249  MSE: 52.525  MAE: 5.436  : 0.316 | Yes |
| Interval\_width: 0.9  UK holidays | RMSE: 7.649  MSE: 59.247  MAE: 5.829  : 0.229 | Yes |
| Interval\_width: 0.9  UK holidays  Monthly seasonality | RMSE: 7.736  MSE: 59.851  MAE: 5.846  : 0.221 | Yes |
| 3900 | Interval\_width: 0.9 | RMSE: 9.761  MSE: 95.271  MAE: 7.210  : 0.507 | Yes |
| Interval\_width: 0.9  UK holidays | RMSE: 10.625  MSE: 112.894  MAE: 7.998  : 0.416 | Yes |
| Interval\_width: 0.9  UK holidays  Monthly seasonality | RMSE: 10.859  MSE: 117.929  MAE: 8.190  : 0.390 | Yes |

We can see from the extensive test that the Prophet algorithm performed extremely well in terms of RMSE and shows that it could be used for energy forecasting. Training for 4500 days (roughly 12 years) yielded lower RMSE but an score of 31.6% (depicted below). Also found was that adding regressors yields higher RMSE for the Gateway House dataset. From the figure, it is observed that the algorithm seems to find an issue with data found in July 2021 which needs investigating.

Chart, histogram

Description automatically generated

Summary of Results

Chart, bar chart

Description automatically generated

The figure above shows all the chosen algorithms compared to the baseline scores. We can see that:

* In 2 out of 3 buildings, the bidirectional LSTM performed significantly better than all other algorithms as stated in the literature. Gateway House was not able to fully utilise the BLSTM which calls for further fine-tuning.
* In the case of Hugh Aston, the Prophet algorithm struggled to make sense of the chaotic data pattern occurring after the coronavirus outbreak. More testing needs to be undertaken and some hyper-parameter tuning.
* The means squared error (MSE) and the root mean squared error (RMSE) both share the same characteristics in terms of creating a bar chart.
* Lastly, this is by far not the end of as in ML, we are always testing to improve.

Conclusion

In terms of the average RMSE score, it was a surprise that the LSTM outperformed the bidirectional-LSTM as this was meant to be an improvement to the LSTM according to the work “Short term power load forecasting based on multi-layer bidirectional recurrent neural network ”[32]. The LSTM achieved an average RMSE score of 6.9 following the ‘less is best’ rule of the metric. Bidirectional LSTM came in second which was not a surprise due to its forward and backwards network traversal. The ensemble methods, XGBoost and LightGBM showed similar scores which means there needs to be more fine-tuning else a hybrid version of the algorithm needs to be explored. From the testing, the more the iterations there the better the predictive performance. Finally, the second algorithm produce surprising results was the Facebook Prophet as it achieved an average RMSE score of 8.4 surpassing both ensemble methodologies. There are very few research utilizing this algorithm for energy consumption prediction. The algorithm was very lightweight as fast to implement which serves as a great start for beginners and experts. The only reason for this score was due to the Hugh Aston dataset including lots of anomalies which could only be forward filled as we are dealing with a timeseries data. Nonetheless we foresee many studies to be carried out using Prophet.