APPROACH AND RESULTS

Introduction

Figure 1 shows the overall approach to build and visualise the machine learning model. A primary input to the process is Distributed Control System (DCS) Historian data provided in the form of a large CSV files.The time-series data set consisted of 422,449 rows with each row having 106 columns and covered a time period of approximately 4-years (January 2015 - January 2019) with temporal resolution of 5-minutes.

In order to assess equipment utilisation beyond capacity, additional equipment rating data was sourced from electrical power distribution system records.

The following sections describe in further detail the specific activities undertaken across the data preparation, model training, model testing, model optimisation, deployment and visualisation phases.

APPROACH

Data Preparation

Before the data could be used for training the models, the data was inspected for quality (missing values etc), as well as the presence of key input variables well understood to likely drive the process plant behaviour.

Domain Subject Matter Experts (SMEs) recommended 14 key input variables, also referred to as a feature set, which were likely to influence the performance of the operating plant and hence drive equipment power consumption. The key input variables identified represent plant inflow (available at 5-minute intervals) as well as flow quality parameters such as COD, NH3, Turbidity, pH among others (available once every one or two days).

There were a total of 91 output variables (labels) describing equipment power consumption all of which were generally available at 5-minute intervals. The power consumption data was available in different forms (Watts, amps, VA etc) which were normalised during model development to enable comparison with with equipment rating data.

Both the input feature set and output labels contained missing values for some time instances. As the data is continuously changing in time series data sets, the missing feature values were substituted with the last available value of the feature preceding the missing value. When training and testing machine learning models, it is important to maintain the integrity of the output labels to obtain valid results. For an output label, instances with missing values were discarded for both model training and testing.

To enable the prediction of the plant state 60minutes in advance, a commonly technique is to train the model utilising lagged data sets. To achieve this, a new data set lagged by a defined period of time (60min-115min) is created as the basis for training this use-case specific model. Within this new data set, the output labels also become features for the purpose predicting future plant states.

Model Development

Model development involved three main activities: training the model, testing the model and model optimization.

Training and Testing Data

The data set was divided into a training and testing data set. The data set was divided into monthly datasets. Each of the monthly datasets was then further divided into training and testing sets using a 90/10 split: 90% of data for training and 10% for testing. That resulted in data from the first 27 days of a month being in the training set and the remaining data in the testing set. The reasoning behind splitting the data set this way

is to eliminate the introduction biasing of recent data over older data. A model was trained and tested for each month and for each output variable.

Defining an Appropriate Performance Metric

The commonly used coefficient of determination (R2) was used to assess model performance. The coefficient of determination is used to assess how well the model predicts the real behaviour of the system. An R2 value of 0 indicates that the model does not explain any of the system behaviour, whilst a value of 1 indicates the model perfectly predicts the system behaviour.

Assessment of Co-Variance Influences

Before model training, it was necessary to determine which of the input variables had high covariances with the output variables. To assess this, Pearson's R (correlation coefficient) was calculated for each input-output pair. This did not yield many input/outputs pairs exhibiting a high Pearson's R correlation coefficient. Table 1 shows the top correlations found between input and output variables.

Table 1: Correlation Between Input and Output Variables

Output	Input	Correlation
PPE_MTR_POWER	PP68_PMP_FLOW	0.456
PPS_MTR.POWER	PP57_PMP_FLOW	0.419
PP4_MTR_POWER	PP24_PMP_FLOW	0.388
PPS_MTR_POWER	PP68_PMP_FLOW	0.359
PP2_MTR_POWER	PP24_PMP_FLOW	0.327
TOT_POWER	UAB_PRILNH3	0.312
POW_HV_CB1_WATTS	PP24_PMP_FLOW	0.3
AER_P1_MX1_CUR_U	PP24_PMP_FLOW	0.281
FAN6_POWER	LAB_PRILCOD	0.278
PP3_MTR.POWER	PP13_PMP_FLOW	0.275
P2_MX1_CUR_U	LAB_SEC_COD	0.261
Pa_MX1_CUR_U	PP24_PMP_FLOW	0.243
P3_MX1_CUR_U	LAB_PRILNH3	0.229
P1_MX1_CUR_U	LAB_PRILNH3	0.224
FAN3_POWER	LAB_PRLNH3	0.212

As shown in Table 1, there was no significant relationships identified between specific input-output pairs. To proceed with the modelling exericse, addition input was sought from SMEs to identify any additional features that could be utilised to improve the peformance. The SMEs suggested that due to the nature and arrangement of equipment items into "banks", it may be necessary to describe these groups as single variables to be modelled (by summation) rather than modelling individually on their own (for example banks of Direct On-Line loads).

Table 2: Correlation Results from Summing Input and Output

Output Variable	Input Variable	Correlation
AAA_BBB_AB_MTR_POWER_sum	AAA_DDD_PP_PMP_FLOW_sum	0.612957
AAA_BBB_POW_TOT_POWER	AAA_DDD_PP_PMP_FLOW_sum	0.587953
AAA_POW_84EH_RCP_KW_sum1	AAA_DDD_PP_PMP_FLOW_sum	0.520822
AAA_POW_84EH_RCP_KW_sum2	AAA_DDD_PP_PMP_FLOW_sum	0.486567
AAA_DDD_PP_MTR_POWER_sum	AAA_DDD_PP_PMP_FLOW_sum	0.407763

The results in Table 2 show a significant improvement in the correlation values compared to the correlations shown in Table 1.

Model Training

As distinct from the Model Performance (R2), the first model trained was a linear regression model (line-of-best fit). Table 3 provides a summary of Performance Metric (R2) results of the regression models for all of the output variables across the testing data set. The results are summarised using standard statistical distribution metrics.

Table 3: Results of Linear Regression Model

	target_var_level	training_sample_size	testing_same_size	training_r2	testing_r2
count	86.000000	91.000000	91.000000	9.100000e+01	9.100000e+01
mean	3.186047	368068.340659	40486.505495	-3.024999e+22	-2.186651e+51
std	0.951922	33631.246658	3621.570554	7.888542e+22	6.108886e+51
min	1.000000	173925.000000	19459.000000	-6.312535e+23	-4.173889e+52
25%	3.000000	376044.500000	41256.000000	-2.511048e+22	-1.241836e+51
50%	3.000000	376106.000000	41259.000000	-3.480914e+21	-9.392376e+49
75%	4.000000	376163.000000	41267.000000	-3.954026e+20	-7.758452e+48
max	4.000000	380081.000000	42244.000000	9.911714e-01	5.656678e-01

The linear regression model does not sufficently capture the system complexity.

The second model trained was a random forest model (ensemble learning method for classification/regression) ^REF Table 4 provides a summary of results given by the model for all the output variables. The results are summarised using standard statistical distribution metrics.

Table 4: Results of Random Forest Model

	target_var_level	training_sample_size	testing_same_size	training_r2	testing_r2
count	86.000000	91.000000	91.000000	91.000000	91.000000
mean	3.186047	368068.340659	40486.505495	0.894201	-0.772288
std	0.951922	33631.246658	3621.570554	0.141761	6.148299
min	1.000000	173925.000000	19459.000000	0.254913	-51.418317
25%	3.000000	376044.500000	41256.000000	0.864801	-0.062699
50%	3.000000	376106.000000	41259.000000	0.955340	0.274962
75%	4.000000	376163.000000	41267.000000	0.981609	0.613376
max	4.000000	380081.000000	42244.000000	0.999658	0.989728

Significant non-linear relationships between input and output variables can be observed when using random forest (decision tree based) models.To improve the random forest models, input and output lags were utilised ranging from 60-115min intervals.

Table 5 below provides a summary of results given by the model for all the output variables.

Table 5: Results of Random Forest Model using Input/ Output Lag Variables

	target_var_level	training_sample_size	testing_same_size	training_r2	testing_r2
count	35.000000	40.000000	40.000000	40.000000	40.000000
mean	1.971429	361212.225000	39901.900000	0.889444	-0.868882
std	1.224402	45790.181108	4980.824089	0.140579	8.681933
min	-1.000000	173925.000000	19459.000000	0.412811	-53.996786
25%	1.000000	376033.500000	41256.000000	0.883200	0.607340
50%	2.000000	376097.500000	41263.000000	0.929019	0.710139
75%	3.000000	377079.000000	42152.000000	0.963472	0.766388
max	3.000000	380081.000000	42244.000000	0.999929	0.999702

Using input and output lagged values improves the R2 metrics. As we can see, random forest with lags (Table 5) results in a 260% improvement of the R2 metric when considering the 50th percentile results (Table 4 & 5). Further improvements were possible utilising the SME direction to aggregate banks of equipment into single variable representations (Table 6).

Table 6 below is a summary of results given by the model for all the output variables.

Table 6: Results of Random Forest Model Using Input/ Output Lag Variables and Sum of Flow Variables

	target_var_level	training_sample_size	testing_same_size	training_r2	testing_r2
count	97.000000	102.000000	102.000000	102.000000	102.000000
mean	2.711340	364177.676471	40095.009804	0.916911	0.185944
std	1.607008	40821.821773	4400.068394	0.121911	3.504270
min	-1.000000	173339.000000	19454.000000	0.365824	-33.001000
25%	2.000000	376038.750000	41256.000000	0.879888	0.625378
50%	3.000000	376106.000000	41259.000000	0.959802	0.760289
75%	4.000000	376181.750000	41267.000000	0.989068	0.916179
max	4.000000	380081.000000	42244.000000	0.999971	0.999682

The Random Forest Model with input/output lags and equipment aggregation exhibited the best R2 values of all the models considered.

Model Deployment

One model for each output variable was deployed using AWS utilising S3, SageMaker and Lambda to create a single multi-model endpoint. This presents a single model to the user/consumer to enable simplified access and scalability (serverless) to run simulations of varying complexity / time horizons.The UI/UX utilised BootStrap JS and Kibana.The architecture/procedure-call flow chart is shown diagramatically in Figure 3.

USE-CASE RESULTS

Use Case 1 relating to the prediction of asset utilisation above name plate was demonstrated by running the model with hypothetical input data sets describing future load scenarios.
Use Case 2 relating to the prediction of plant state 60 minutes in advance was tested using similar techniques to Use Case 1 except with input data likely to generate excursions beyond equipment name plate capacity.

OPPORTUNITIES FOR ENHANCEMENT

Future enhancement opportunities exist for example the exploration of RNNs (recurrent neural networks) as well as expansion of the UX/UI functionality developed around input data set definition and configuration - this would enable rapid scenario analysis.

CONCLUSION

The aims of the project were achieved and it was demonstrated that machine learning techniques have a significant role to play in the automated model building arena providing the foundation for powerful, value-driven Use-Cases. A relatively high degree of confidence was achieved (R^2 of 0.76) but areas for improvement were identified for future exploration.implementation.