Smith institute
for Industrial Mathematics and System Engineering

Dynamic Reserve Setting: improving setting of reserve requirements through machine learning

Version **1.0.1**

P20-050

Authors	Alex Evans
	Kieran Kalair
	Rachael Warrington
	Francis Woodhouse

Internal reviewers	Tim Boxer
	Pablo Levi

Executive Summary

In the Dynamic Reserve Setting (DRS) project, we have developed a set of proof-of-concept machine learning models (DRS PoC) to recommend reserve levels based on dynamic data such as wind speed forecasts. The problem is described by National Grid ESO[1] in the following way: “Currently ESO sets reserve levels that vary according to electricity demand seen at different times of the day and week – with levels informed by historical generation and forecasting errors and adjusted by forecast renewable generation output. These reserve levels could potentially be optimised to take better account of the effect on the system of forecast weather conditions, by linking generation and forecasting errors to weather driven effects or other variables, and buying reserve in day-ahead timescales.” The DRS PoC, on day-ahead timescales, uses available data to recommend reserve levels that meet a 1 in 365 risk appetite, and does so in a way that is explainable.

Results: Value to NGESO

Comparing with NGESO’s current approach (looking at positive reserve at a 4-hour lead time), we find that the DRS PoC gives fewer shortfalls that are, both on average and when considering the maximum, smaller.

Along with the recommended reserve levels, the DRS PoC generates a new and innovative explainability report to show, in a visual way, the impact of different features – things like the wind speed – on each reserve recommendation. This explainability report provides a future-proof approach through moving away from the pre-defined pots of reserve that are currently used to provide explainability, and through using a model-agnostic approach that allows the underlying reserve setting models to be changed without the explainability approach needing to be changed.

Approach

Data processing is the foundation of DRS, with significant effort applied to go from data from multiple NGESO systems to one cleaned, processed, database ready for model building. Future work on the same dataset could use this database as a starting point, getting straight into producing additional insight from the data. The data processing

[1] See https://www.nationalgrideso.com/news/national-grid-eso-and-smith-institute-begin-industry-pioneering-dynamic-reserve-setting-drs

involves calculating the total reserve error that the reserve setting models are trained on. This total reserve error combines upwards (for positive reserve) or downwards (for negative reserve) reserve error (URE/DRE), wind error, interconnector error, reserve for response error, and demand error. We have extended the URE definition and created a DRE definition, providing a solid foundation for model building.

The reserve setting models use gradient boosted trees: a machine learning approach that has been repeatedly shown to deliver best-in-class performance when modelling tabular, multi-source data, as is the case here. Through selecting features of the data with the greatest predictive power and tuning each model's hyperparameters, we build models for both positive and negative reserve that perform well when compared with a constant baseline.

Recommendations

We recommend that NGESO productionise the DRS PoC and explore additional valuable innovation opportunities.

First, we recommend that NGESO productionise the models to take advantage of the value that the DRS PoC has demonstrated. We expect that this work will involve building robust data pipelines that connect to existing NGESO databases, further testing and, where appropriate, automation of the code needed to train and run the models, creating an API for accessing model outputs, and creating a user interface for the DRS PoC. Further work could also build on and enhance the approach, for example through exploring open data sources that could enhance the DRS PoC's predictive power.

Likewise, we recommend that NGESO consider exploring two opportunities for innovation: using short-notice time series models to enhance the predictive power of the reserve setting models, and an alternative approach to calculating URE that will overcome the limitations of both the approach currently used by NGESO as well as the extension of that approach that is used in this DRS project, and will provide a way of calculating URE that is resilient to future changes in the electricity system.

Contents

Introduction

In the Dynamic Reserve Setting (DRS) project, Smith Institute have developed a set of proof-of-concept machine learning models that recommend reserve levels based on dynamic data such as wind speed forecasts. This proves the concept that the current reserve setting approach can be improved by using dynamic reserve setting to reduce under-holding and improve transparency with better explainability. Productionising the models and integrating them into NGESO's systems would improve on the current reserve setting process, which is done twice a year, to provide dynamic day-ahead reserve setting. This would enable NGESO to buy reserve at day-ahead time scales and to set reserve levels more accurately.

To balance the grid, ENCC draw on various forecasts to aid their decision making, and, like all forecasts, these are subject to errors. Reserve is needed to insure against those forecast errors: if national demand is much lower than forecast, for example, then reserve is needed to fill the gap. The reserve levels should be set neither too low – this could threaten system security – nor too high, as this could lead to high costs and emissions.

NGESO therefore need models that use available data to predict the forecast errors and recommend a reserve level that meets NGESO's risk appetite. The output from these models needs to be made available to ENCC and, importantly, to be something that they can trust. To build trust in the models, they must be explainable: not only do ENCC need to see the recommended reserve level, but they also need to see what has led to that recommendation. The scope of the DRS project was to develop a proof-of-concept of such a set of models, using a snapshot of 3 years' worth of historical data extracted from NGESO systems. Productionisation of the models, and their integration into NGESO systems, is beyond the scope of the current DRS project.

In this report, we describe the proof-of-concept development, and report on its performance. While productionisation was outside the scope of the current project, we discuss the steps we see as necessary for productionisation, and the main challenges that we foresee. As the project has progressed, we have come across various opportunities for future innovation and improvement that, while beyond the scope of the current project, we recommend exploring in future work. Overall, we hope that this report provides the reader with a clear sense of the value delivered by this project, as well as recommendations of future innovation ideas to explore.

Proof of Concept development

PoC Aims and Use

Our proof-of-concept implementation (DRS PoC) aims to give a set of dynamic reserve setting models that use available data to recommend a reserve level that meets a 1 in 365 risk appetite[2], and to do so in a way that is explainable.

The data that is available to use as an input to the models depends on the timescales on which NGESO will run the DRS PoC. These timescales are as follows: reserve recommendations covering 5am on the day of interest (D0) to 5am on the next day (D1) must be generated by 11am the day before (D-1). We therefore take care to only use data sources that are available before that 11am D-1 deadline, so we do not use, for example, forecasts of wind speed that are generated only one hour ahead of the time of interest. The work needed to read in and process these data sources, getting them into a model-ready format, can be seen in the Data Processing section below.

We create reserve recommendations for each lead time from 1 to 24 hours. To do this, we first calculate the total error (the quantity that the reserve recommendation is aiming to exceed for all but 1 in 365 settlement periods) for each of those lead times, taking the difference between the forecast at the relevant lead time and the actuals. The details of the error calculations are set out in the Defining Errors section later in this report. After calculating these errors, we use them to train 24 models, one for each lead time. The approach to building these models is set out in the Model Building section.

When NGESO run the DRS PoC, this will run each of the 24 models to produce reserve recommendations for each settlement period between 5am on D0 and 5am on D1 and will save them to an output file. Along with these reserve recommendations, we also output an explainability report to give ENCC insight into what has led to the recommendation. We describe the explainability report in more detail later in this report.

Data Processing

Before any models could be built, a large amount of data processing was required, providing a solid foundation for all future parts of the project. It became clear early into the

[2] On a settlement period basis. This risk appetite can be changed.

project that this constituted a significant amount of work, due to both the amount of data that was required for the project, and also the range of different data quality issues that were observed as work progressed. The fundamental goal of the data processing work was to take the raw data files provided and insert the data they held into a MySQL database, addressing any issues with the data before insertion into the database. Details of the data held in the database are given in Table 1.

Table 1: Details of data and errors held in the DRS database. Note that we are aware certain properties cannot be directly measured but are instead inferred. The term 'measured' is used for consistency across datasets.

Data	Details
BMU details	IDs & fuel types
BMU level measurements and forecasts	Forecast and actual values of (where appropriate) PN, NDZ, MEL, MIL, SEL, SIL, CL, MO & BOAs
BMU ramping details	Analysis identifying when BMUs are ramping to or from sync and desync events
National grid trading details	Trades made by the NGESO trading team that influence unit behaviours and cause deviations from forecasts
National level measurements and forecasts	Measured national demand, embedded wind, PV and interconnector flow values, as well as predictions of national demand from both PEF and BMRA. The PEF and BMRA forecasts were blended to create a hybrid demand forecast
Weather measurements and forecasts	Wind speeds, air temperatures, humidity, and various other descriptions of the weather at each time considered
Weather station details	IDs & locations

Wind BMU specific forecasts	Forecast wind BMU generation values
Computed Error	**Details**
Downwards reserve error	Computed by combining BMU details, measurements, forecasts, ramping details, and trading data
Interconnector error	Computed by combining BMU details, measurements, forecasts, and trading data
National demand error	Computed by combining national level measurements and forecasts
Reserve for response error	Computed by combining national level measurements and forecasts
Upwards reserve error	Computed by combining BMU details, measurements, forecasts, ramping details, and trading data
Wind unit error	Computed by combining BMU measurements and wind specific BMU forecasts

In addition to the data detailed in Table 1, we collate various features from the processed data and build a single table holding the features our models can be trained on. Doing this ensures the models produced can be re-trained on all, or subsets of, the data used in this project, with minimal additional effort in the future.

In total, we processed data from January 2018 until September 2021. This alone presented considerable computational demand, as we held data at a 30-minute resolution. In addition, a substantial amount of work was required as data quality issues were identified during processing of the data, some of which required input (and in some cases further data sets) from NGESO to resolve. The specific data quality issues identified and resolved during this project are:

- Temporal discrepancies between data-points. Some were given at the level of settlement periods, others with date-time stamps, and checks were completed to ensure all datasets were correctly aligned in time. Similarly, some data was given with GMT timestamps, and others with local timestamps, meaning alignment was necessary.
- Addressing missing values. Specifically for national demand forecasts, some time-periods did not have any available forecasts at lead times above 18 hours. When this was identified, the longest lead time forecast was used to fill the missing values.
- Interpolation of forecasts onto half-hour granularity for those given at lower resolutions
- Mapping forecasts onto an integer grid of hourly lead-times. Some forecasts, for example those relating to weather, were not given at integer lead times, or at a one-hour granularity. To ensure all data was available at the same resolution, in this case we carried the most recent forecast known to shorter lead times.
- Refining the list of BMUs that contribute to error calculations. It became clear that determining what set of BMUs to consider in the URE, DRE, and wind error calculations was not a simple task, and there were several discussions between SI and NGESO before settling on a pre-defined list provided by NGESO.
- Determining a national demand forecast to use for the times before the PEF forecasts were created
- Identification of BMU fuel types when missing in raw data
- Identification of data files that were missing in some data-dumps we received

Further, additional processing steps were required to implement updates to various error calculations. These were:

- Inclusion of NGESO trading data into error calculations, ensuring that when NGESO's actions instructed units to deviate from their forecasts, they did not contribute to the errors. This required loading and processing of trading data, and incorporation of it into the error computation logic.
- Exclusion of wind units that were subject to a BOA at a given time when computing wind unit errors.
- Detection of periods of time when units were ramping from a sync event, or to a desync event, and inclusion of these in the URE & DRE logic. This was done by inspecting sequential data-points for units in time, which differs from NGESO's existing approach that infers such behaviour without considering sequential information.

- Generating a hybrid national demand forecast that used PEF forecasts (when available) for lead times less than 4 hours, and BMRA data otherwise.

While building this database, and resolving the data quality issues identified, took a significant amount of time, both the final database produced and what we learned from the data processing are highly valuable going forward. First, the database itself (which is part of the project deliverables) enables easy examination of the various errors, features, and unit contributions throughout the time-period studied. Additionally, further models can be built and evaluated over subsets of this time-period with minimal effort. Finally, having identified and worked with NGESO to resolve each of the data quality issues mentioned, future work involving the data sources can apply the knowledge we gained, spending less time manipulating and cleaning data, and more producing additional insight from the data.

Defining errors

Part of the data processing work described above involves computing various errors, and here we describe our extensions to NGESO's existing error definitions, as well as showing how the errors combine to give a total error. Our reserve recommendations are based on this total error: the models are trained such that the reserve recommendations produced are higher than the total error 99.7% of the time, i.e. for all but 1 in 365 settlement periods[3]. Getting a reliable reserve recommendation, therefore, depends on calculation of an appropriate measure of total error.

To get the total error, we combine several types of error: Upwards Reserve Error (URE) or Downwards Reserve Error (DRE) for non-wind-non-interconnector units, wind error, interconnector error, reserve for response error, and demand error, as is done in the approach currently used by NGESO (although we do not subtract off free headroom, and instead look at the whole total error.).

For positive reserve, we want to cover cases when, due to forecast errors, we have either:

- Less headroom from non-wind-non-interconnector units than forecast
- Less generation/more demand from wind units and interconnectors than forecast
- Higher national demand than forecast

[3] This risk appetite can be changed by NGESO as needed.

Our total positive reserve requirement is then be based on the 99.7th percentile of:

For negative reserve, we want to cover cases when, due to forecast errors, we have either:

- Less footroom from non-wind-non-interconnector units than forecast
- More generation/less demand from wind units and interconnectors than forecast
- Lower national demand than forecast

For negative reserve, therefore, we use the (100-99.7)th percentile of:

We calculate the errors for each lead time L and each settlement period of interest, allowing us to create reserve recommendations for each of those lead times and settlement periods.

Our extended error definitions, the details of which can be seen in Appendix A, have built on those currently used by NGESO in the following ways:

- We exclude unit, settlement period pairs where a trade has been initiated by NGESO, so that deliberate decisions to trade are not being counted as errors. This mainly affects the interconnector error, but could also affect the URE/DRE and wind error if there are trades initiated by NGESO on those types of unit.
- We extend the definition for URE to include demand-side and bidirectional units.
- We take the definition for URE and adapt it to give a formulation of DRE (for each of generation-side, demand-side, and bidirectional units), which is needed for calculating negative reserve.

With the data processing and error definitions providing a firm foundation to build on, we can progress to model building.

Model Building

As the primary class of predictive model, we chose gradient boosted trees. Boosting uses ensembles of predictive models to assemble one powerful predictor from the sum of many smaller, weaker ones, trained to work together to capture different features within the data. Applying boosting to decision trees has been repeatedly shown to deliver best-in-class performance when modelling tabular, multi-source data as is the case here. Though this power comes at the expense of non-parametric black-box modelling, gradient boosted trees can be made more transparent through explainability techniques as we have deployed in this project.

To build the models, we used the LightGBM framework. LightGBM has proven itself to be a significant competitor to the established favourite XGBoost, having similar performance while being more computationally efficient in training. It provides a number of hyperparameters to control the model structure, but not so much structural flexibility as to be overwhelming to fine-tune. It also has the advantage of natively supporting fitting to a given statistical quantile, giving further computational efficiency compared to other models where a custom loss function would need to be manually specified.

As part of model building, for each model that makes up the DRS PoC, we employed two important processes: feature selection, and hyperparameter tuning. The combination of these meant we were able to balance bias and variance to yield models that perform well on unseen data.

Feature selection involves determining the features of the data which provide the greatest predictive power. Simply selecting all available features can reduce model performance, especially through overfitting where the model is too heavily attuned to the precise data it has been trained on and therefore is not as accurate when used to make predictions using fresh data. To determine which features to add, we considered the correlations between a new feature and the residual difference of the target value and the current model prediction. The next feature trialled would then be selected based on the strength of this correlation and the simplicity of the feature. A simpler feature is more understandable to a human so these were favoured as were features which had types not yet present in the model (for example, wind/PV related features). Trialling a new feature involved training the model using cross-validation with the new feature added and comparing the resulting performance on the data not used for training to the model performance on the same data before this latest feature was added. A new feature was kept if it showed an improvement in model performance.

To find a good set of values for the hyperparameters of the model, we used a combination of automated and manual selection. The initial automated process involved passing the selected model features to a Tree-structured Parzen Estimator algorithm using the Python package Optuna. A search space of hyperparameter values is passed to this algorithm which then iteratively selects values for the hyperparameters and fits a new model using these hyperparameters. The hyperparameter selection for each iteration is guided by the performance of the model during previous iterations, making this more efficient than a simple grid search or Monte Carlo sampling approach. The manual process consisted of fine tuning hyperparameters related to when the model ceases improving its fit to the training data, further reducing the chances of overfitting.

There are alternative model classes we could have adopted for this regression problem. One obvious class with similar flexibility to LightGBM, albeit an entirely different paradigm, is neural networks. Neural networks are more cumbersome to design and train, and their behaviour under extrapolation can be more unpredictable than that of gradient boosted trees. Nevertheless, to compare raw performance of neural networks against our chosen LightGBM approach, we give some case studies below for example lead times and quantiles. These rely on an auto-ML approach to tune the neural network architecture and a custom loss function for quantile loss instead of the usual mean squared error, achieving comparable performance but at an unjustifiable computational burden.

Trading flexibility for control, other approaches could focus on linear regression models. These were used in the first, feasibility demonstration phase of the DRS project to provide a clean first pass when understanding the data and refining our approach. The significant nonlinear interactions inherent in the data available here mean that a powerful linear regression model would demand significant feature engineering to achieve performance comparable to strong non-parametric models. Therefore, we do not pursue such models beyond their use in the first phase of the DRS project.

Explainability

We understand that for the new dynamic reserve setting approach to be adopted for use by NGESO, the outputs of the models must be clear, explainable and understandable by control room engineers. To ensure this is the case, when our models are run, we produce an ‘explainability report’ alongside the reserve recommendations detailing the contributions of each feature to the final model output. As an example, this report might detail how many MWs of the output are due to the forecast wind speed, or the time of day.

The existing methodology to attribute a particular number of MWs of reserve requirement to a particular source is to first define some number of 'pots' (e.g.: wind, PV, regulating) and to place into each of these pots various amounts of reserve requirement. We aimed to improve upon this by removing the need to pre-define pots and instead to attribute a particular number of MWs of reserve requirement to each *feature* the model incorporates. We also do so in a model agnostic way, meaning if the reserve setting models were to change in the future, the explainability methodology we have devised remains valid. We achieve this by using a methodology called Shapley Additive Regression Values (SHAP)[4]. This is a game theoretic approach to explaining the output of a machine learning model, based on the values of the input features.

Before giving an example of the use of SHAP, we first note a few terms that are required to understand the subsequent plots explaining the model outputs:

- A 'baseline model output' that can be used as a reference value for explanations. This can be thought of as a surrogate for the average of the model outputs over the data it is trained on. In every plot of SHAP values explaining a prediction, the baseline value will be marked, with changes in the model output judged relative to this.
- A linear decomposition of a model output. Suppose a model had features A and B. By linear decomposition of the output, we mean the output can be written as:

  Output = baseline + (Feature A's contribution) + (Feature B's contribution)

  Here, we breakdown the output of the model (that is a single numerical value) into the sum of contributions from each feature, and the baseline model output. In doing this, we can say for a single prediction, this output differs from the baseline by some amount of MWs, and we can inspect how many of these MWs are due to each individual feature in our model taking the specific value they do for the given input. Note that this does *not* mean our model is linear, rather that the SHAP values we compute from the model can be added to the baseline to reach the model output.

[4] Technical details of SHAP can be found in: S. M. Lundberg, S. Lee. (2017) "A unified approach to interpreting model predictions". Advances in Neural Information Processing Systems 30 (NIPS 2017).

The most intuitive understanding of such an approach can be gained by inspecting one of the plots we generate and place in our explainability reports, shown in Figure 1.

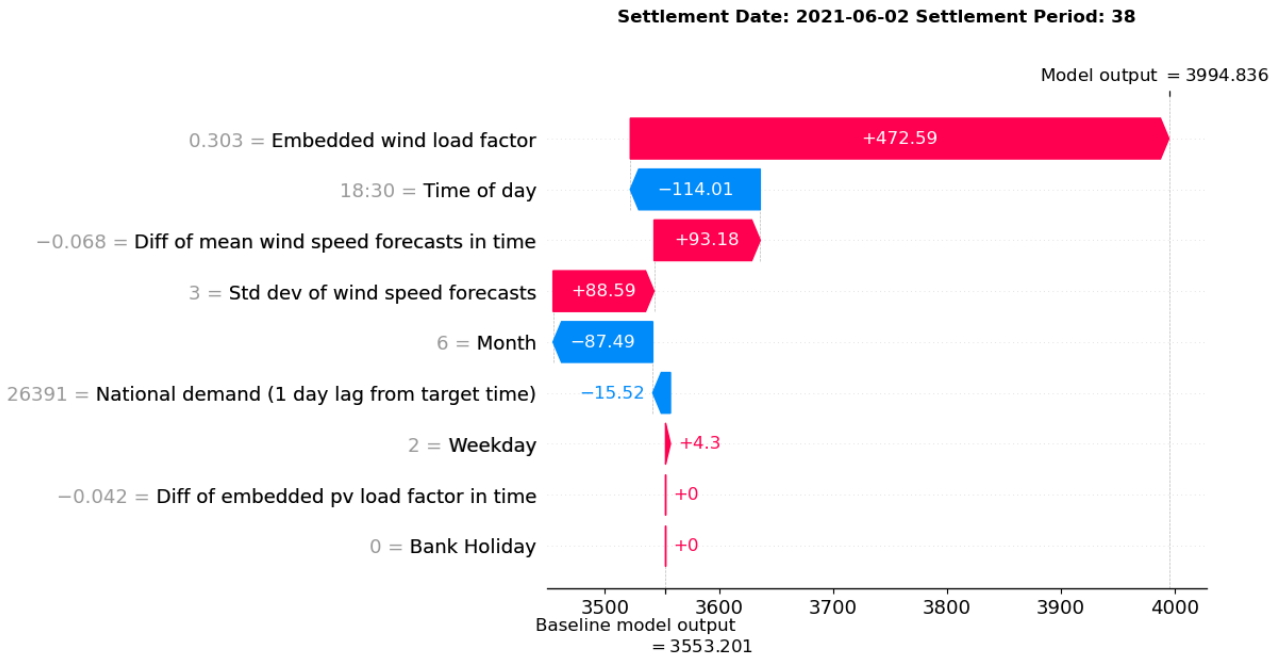


Figure 1: SHAP explanation plot for 2021-06-02, settlement period 38

In Figure 1, we see the explanations SHAP provides for a *single* prediction the (4-hour lead time, 99.7% quantile, total positive reserve error) model generates. In this example, we have a baseline model output of 3553.201 MW. It is from this baseline that we judge how much each feature increases or decreases the model output in this example. Increases in model output are shown in red, whereas decreases are shown in blue. Each feature is listed along the y-axis of the graph, and the final model output, after including each feature's contribution, is marked at 3994.836 MW. We would read such a plot as follows:

- The most influential feature in this example is the embedded wind load factor, which takes a value of 0.303. This leads to an increase in the model output of 472.59 MW compared to the baseline.
- The second most influential feature in this example is time of day, which takes the value 18:30, and decreases the model output by 114.01 MW compared to the baseline.

- The third most influential feature in this example is the difference in mean wind speed forecasts in time (measuring the expected changes in wind speed), which takes the value -0.068, and increases the model output by 93.18 MW
- The fourth most influential feature in this example is the standard deviation of wind speed forecasts (measuring the variability in the forecasts across the country) which takes the value 3 and increases the model output by 88.59 MW.
- After, we see the next most influential features are the month, national demand (at a lagged time of 1 day), weekday, the differences in embedded PV load factor in time, and finally the bank holiday indicator. Note that for weekday, 0 corresponds to Monday, 1 to Tuesday and so on.

The explainability plots generated for any model output will always be ordered such that the most influential feature is at the top, the second below that and so on, with the least influential at the bottom. Since the model is non-linear, we may see significantly different orderings and contributions from other predictions we explain, however we can always use this breakdown to attribute how many MWs a particular feature contributes to the model output, given it takes the specified value.

The final explainability report produced each time the models are run includes a table of the MW values of reserve to hold at each target time, a plot of contingency reserve at each target time, and one plot as shown in Figure 1 per prediction made. Contingency reserve is defined as the difference in reserve recommendations between some lead-time and the recommendations at a 4-hour lead time. We generate a plot of this for 4, 6, 12, 18 and 24 hour lead times. Finally, recall that we make predictions for lead times from 1 to 24 hours and predict from 5am on some day to 4:30 on the next day. This means each lead time, time-point pair has an associated SHAP plot that explains contributions to the output from each feature.

Value to NGESO: Proof of Concept performance

In this section we discuss the performance of the dynamic reserve setting models. The models perform well compared to a reference constant model when evaluated on data not used to train either set of models. Further investigation is required to establish the correct quantiles to predict in order to match NGESO’s risk appetite. Comparison to historic

NGESO predictions looks similarly positive though with the strong caveat that the NGESO predictions were made against a different target to those considered in our work. A summary of the main improvements the DRS model[5] offers over the existing NGESO approach are as follows:

- The DRS model has fewer shortfalls in reserve. Actual reserve requirements exceeded the predictions made by the DRS model 0.43% of the time, compared to 2.66% for the existing NGESO approach.
- When the actual reserve requirement exceeded the predictions (a shortfall), the DRS model was closer to the true value than the existing approach. The average and maximum shortfalls for the DRS model were 397 MW and 1490 MW respectively. For the NGESO approach, these values were 774 MW and 4406 MW.
- The performance of the DRS model appeared consistent across different subsets of time, whereas the existing approach experienced more shortfalls as years progressed from 2018 to 2021.
- The DRS model offers a higher resolution of explainability than the existing approach, even though the DRS model is more complex. This is because the DRS model attributes a particular number of MWs to each input feature, whereas the existing approach relies on predefined ‘pots’ of reserve.

Detailed comparisons between the DRS results and the historic NGESO approach are discussed further towards the end of this section, with details on how these comparisons were done and the caveats that should be considered when interpreting them.

General performance of the DRS PoC

Figure 2 shows predictions from the models targeting the 99% and 99.7% quantiles of positive reserve error against the actual positive reserve error for one week of data drawn from the test set. The sinusoidal nature of the predictions is due to the influence of time of day with deviations from this driven by the other features of the models. The features used in the DRS PoC models are listed in Appendix B: Model features.

[5] The 4-hour-lead-time, 99.7% quantile positive reserve model, which is the one most appropriate for comparison with the NGESO current approach.

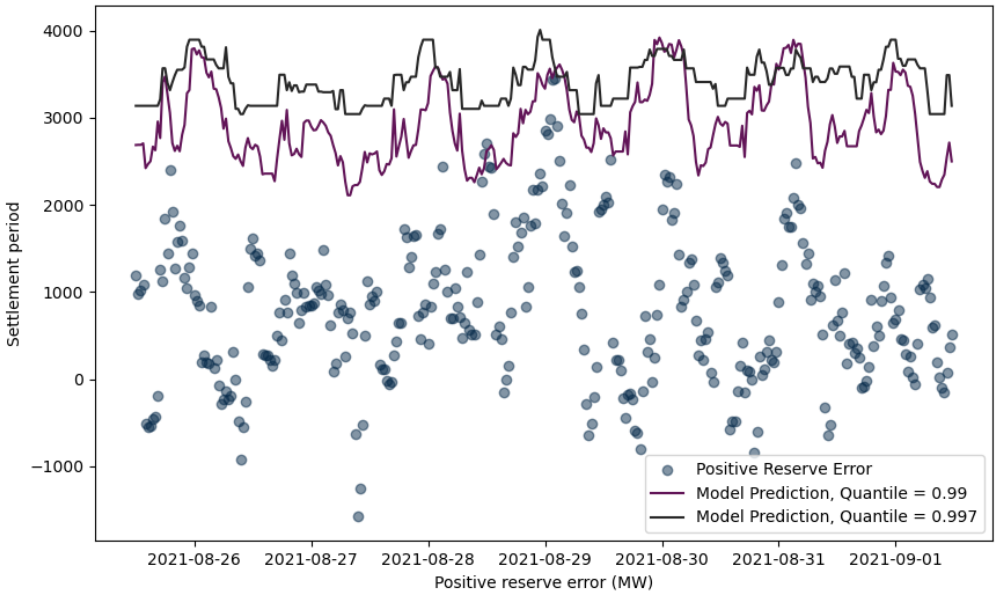


Figure 2: Predictions vs actuals (blue circles) for positive reserve error with predictions targeting the 99% quantile (purple) and 99.7% quantile (black).

To assess the general performance of the DRS PoC we compare predictions from the models to target actuals for a test set of data not used for training the models. To allow for comparison, we define a reference model as a constant estimate of reserve error. We discuss here the performance of our models against this reference.

We use several aggregated metrics to assess the model performance:

- Quantile loss – an appropriate measure of goodness of fit in this context, analogous to the use of root mean squared error for standard linear regression models.
- Average prediction – the average prediction of the model. Decreases in this metric represent decreases in cost to hold reserve to meet this requirement.
- Exceedance fraction – the fraction of times the target actual is more extreme than the model prediction. Decreases in this metric represent reduced occurrences of insufficient reserve.

In the following analysis we focus on the model predictions for positive reserve at the 99.7% quantile. Similar results were observed for the remaining quantiles (99% for positive reserve, and 1% and 0.3% for negative reserve) used to train models in this work. See

Appendix C for more details. Comparisons to the reference model are averaged across 20 randomised partitions of the data into training and test sets[6].

Figure 3 and Figure 4 respectively plot the quantile loss and average prediction of the proof-of-concept models and reference models for each lead time. Performance against these metrics is consistently better than the baseline at all lead times with greatest improvement over the reference models for longer lead times.

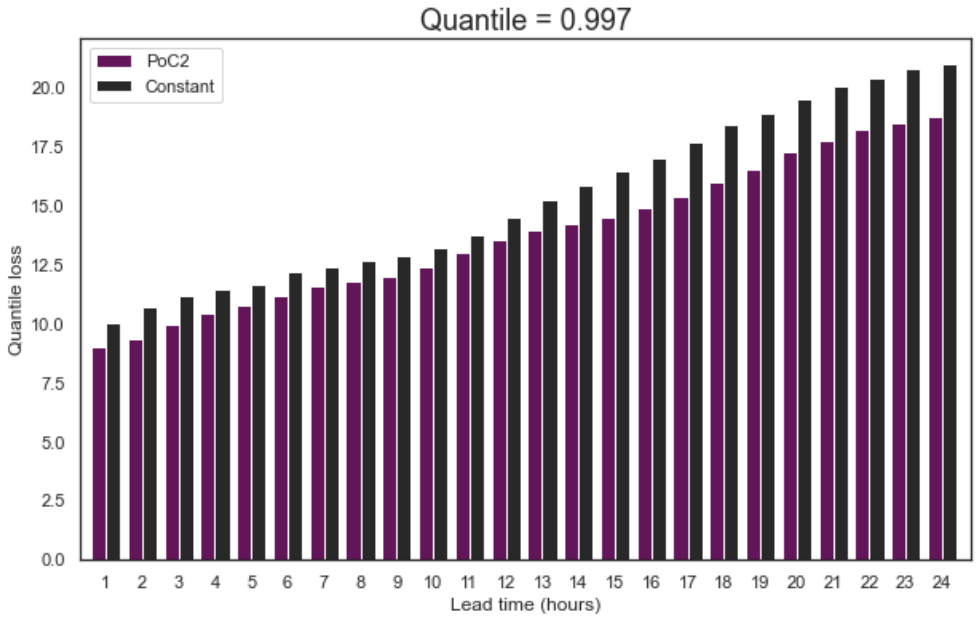


Figure 3: Quantile loss across lead times for positive reserve predictions at the 99.7% quantile for the DRS PoC (purple) and reference model (dark grey).

[6] For each of the 20 models that we average over to get the results described, the model is trained only on the training set for that model, with the performance evaluation made on the test set for that model. For another of the 20 models, data from the first model's test set may appear in its training set, but individual models are never tested on data that they are also trained on.

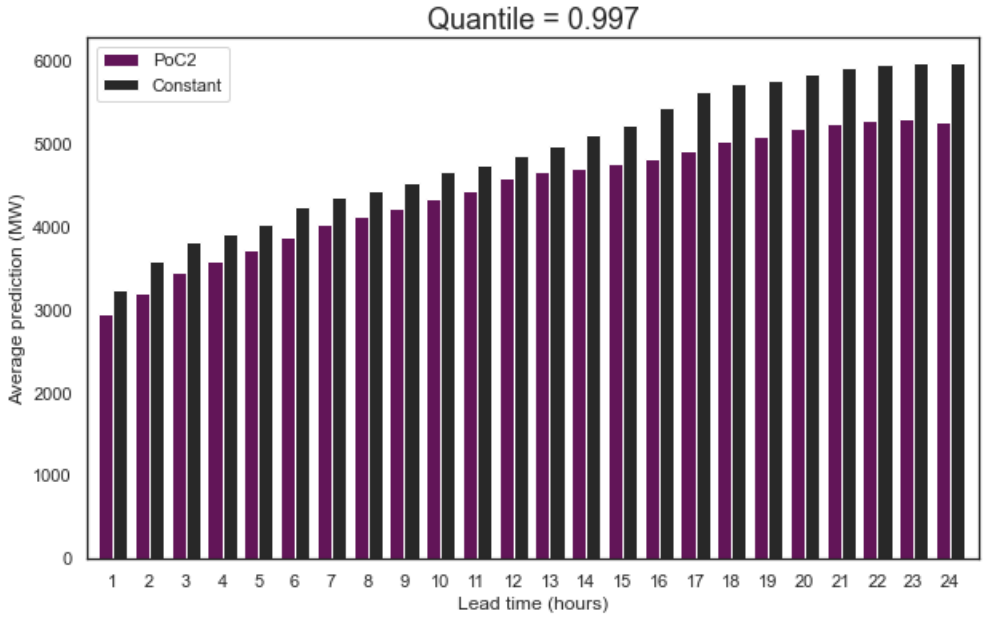


Figure 4: Average model prediction across lead times for positive reserve predictions at the 99.7% quantile for the DRS PoC (purple) and reference model (dark grey).

Figure 5 plots the fraction of settlement periods when the model prediction was exceeded by the true reserve error which are instances where the models are recommending a level of reserve which would have been insufficient. For a target quantile of 99.7% we expect this exceedance fraction to be around 0.3%. As shown in Figure 5, the models perform slightly worse than the reference models against this metric. However, the analysis shown later on in this report shows that the DRS PoC has a reduction in underholding when compared with the NGESO current approach. Further work is needed by NGESO to calibrate the quantile used in the DRS PoC to their risk appetite, which is based on events - which can cover multiple settlement periods - rather than on the settlement period granularity used in the DRS PoC. The DRS PoC has been developed in such a way that gives NGESO the flexibility to change the quantile to something other than the 99.7% that we have focussed on in this report.

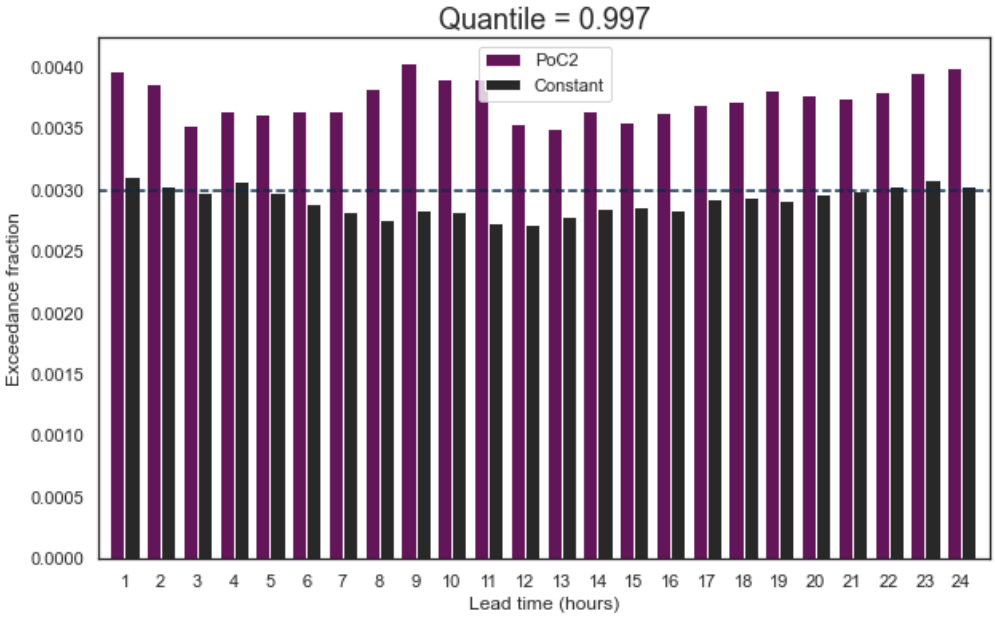


Figure 5: Fraction of settlement periods when the true reserve error exceeded the model prediction across lead times for positive reserve predictions at the 99.7% quantile for the DRS PoC (purple) and reference model (dark grey).

We have also performed comparison with an alternative choice of model where a neural network was constructed. Table 2 shows the performance of each type of model when predicting positive reserve error for one choice of partition of the data into training and test data. This table highlights two features of the model performance. The first is that there is a trade-off between the metrics. For example, the gradient boosting model outperforms the neural network model with regards to quantile loss at lead time 4 and quantile 99.7 but performs worse for the same lead time and quantile when measured against average prediction and exceedance percentage. The second observation is that performance is comparable between the two model types. We decided to proceed with the gradient boosting model approach as it requires far less computing resource to train and is more explicitly constructed than a neural network equivalent.

		Quantile loss		Average prediction		Exceedance %	
Lead time	**Quantile**	NN	GB	NN	GB	NN	GB
4	**99.7**	12.1	10.7	3855	3958	0.37	0.41
4	**99**	31.6	31.0	3338	2929	0.95	1.30
24	**99.7**	19.7	20.2	4868	5146	0.77	0.71
24	**99**	52.9	55.6	4334	3958	1.25	1.56

Table 2: Performance comparison between neural network (NN) models and gradient boosted (GB) models when predicting positive reserve error.

Comparison with historic NGESO approach

A summary of the main conclusions drawn from comparing the existing NGESO approach to the DRS model[7] are as follows. First, the DRS model had fewer instances where the model predictions fell below the actual reserve requirements, compared to the existing approach. Second, when predictions did fall below the actual reserve requirements, the DRS model was closer to the true requirement (both in terms of average and maximum shortfall) than the existing approach. Third, the performance of the existing approach appears to deteriorate as time progresses (from 2018 to 2021). This is not true for the DRS model. These conclusions must be considered in conjunction with the complexities that arose when comparing the two approaches and are detailed in the remainder of this section.

A final way to assess the performance of our model is to compare the outputs, where possible, to the existing approach used by NGESO. We were only given results for positive reserve by NGESO, and only for underholding, so do not give any comment on negative reserve performance or overholding in this context. It should be noted from the outset that

[7] The 4-hour-lead-time, 99.7% quantile positive reserve model, which is the one most appropriate for comparison with the NGESO current approach.

this comparison is not direct, as both the modelling approach and the target variable have changed. The inclusion of additional complexity in URE and other error terms means we cannot simply compare a prediction at a given time from the NGESO model with the DRS model and say which is better. Instead, we can repeat the analysis performed by NGESO on their existing approach for the DRS model and compare the behaviours we see.

Throughout this section, the results for the NGESO approach are taken from materials provided to us by NGESO. The results for the DRS model are computed using a model for total positive reserve, trained for the 0.997 quantile. Results are only computed for data-points in our test set. We also use the term shortfall to denote instances where a model's prediction was below the actual required reserve value, and the size of a shortfall (in MWs) as the difference between the actual reserve required and the model prediction.

The first comparison we make is to ask, what percentage of predictions made by each approach provide sufficient reserve when compared to the actual amount of reserve that was required? This is exactly the one minus the exceedance fraction as discussed previously in this report. This is shown in Figure 6. From this, we see that the existing approach recommends sufficient reserve for 97.34% of times, whereas the DRS model does so for 99.57% of times. Clearly, the DRS model will have fewer shortfalls in reserve as a result.

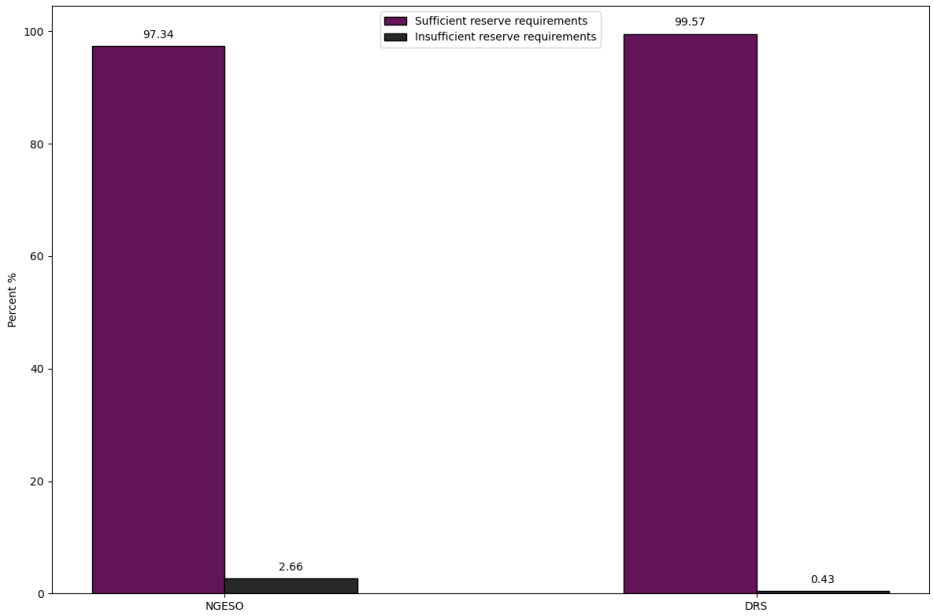


Figure 6: Comparison of sufficient and insufficient reserve recommendations by the NGESO approach and DRS model

A second comparison to make is to question whether the models show variation in performance over different periods of time. This analysis is presented in Figure 7, from which we see the percentage shortfalls for the existing approach appear to increase as time progresses. This is not true for the DRS model, which also shows a lower percentage of shortfalls across all time-periods analysed. It must be noted that when training the model, we created a training set by randomly selecting 75% of days as ‘training days’ and the remaining 25% as ‘test days’. Therefore, the DRS model will have seen some data-points that occur across the entire time range studied, whereas this is unlikely to be true for the NGESO approach. The main conclusion from this analysis is therefore that the DRS model does not show significant variability in performance across the different time-periods shown in Figure 7.

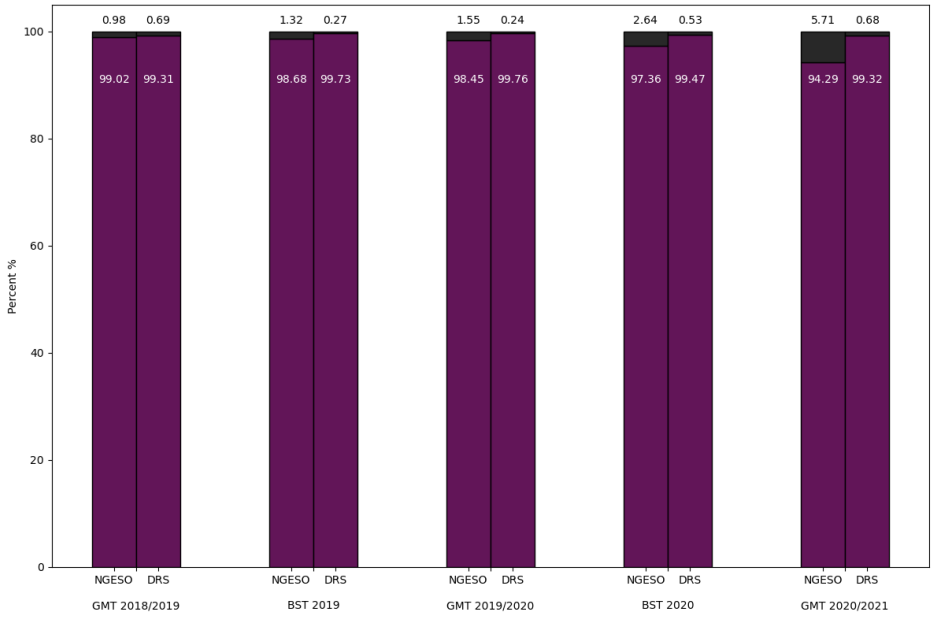


Figure 7: Comparison of sufficient and insufficient reserve recommendations by the NGESO approach and DRS model, split by GMT and BST time-periods across years. As in Figure 6, purple denotes sufficient reserve and graphite denotes insufficient.

As well as breaking down performance into clock change periods, we can also consider for which days of the week our model experiences the highest number of shortfalls. This is done in Figure 8. We do not show the NGESO results, as exact values could not be read off the provided slides, however we note that the NGESO approach showed the most significant number of shortfalls on Sundays. In comparison, we see from Figure 8 that the DRS model experiences the highest number on Tuesdays and Thursdays, and in fact has the lowest number on Sundays. It should be noted that, since the number of shortfalls the

DRS model experiences is small, there may be some variability in these results if more test data was attained and the analysis repeated.

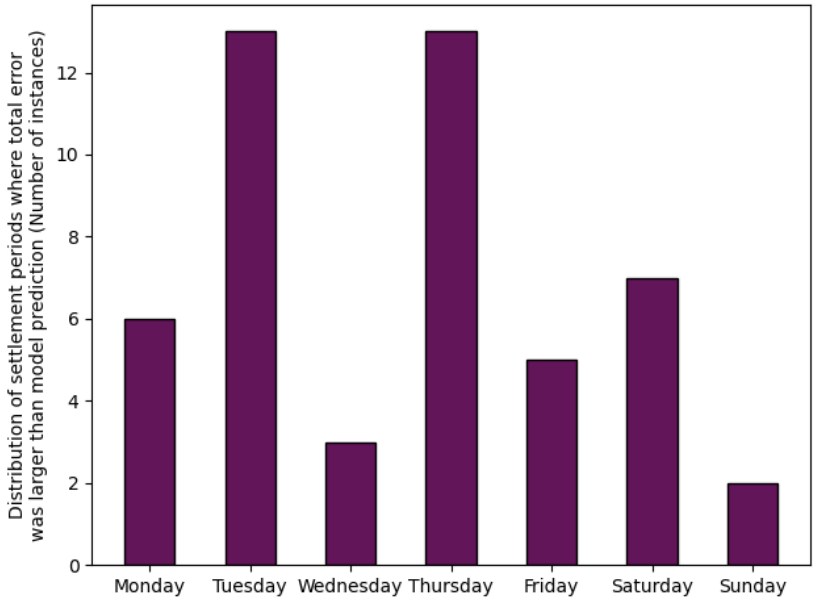


Figure 8: Distribution of shortfalls across days of the week for the DRS model

In the analysis provided to us by NGESO, there was an investigation of the size of shortfalls, that is the number of MWs the model prediction fell short of the actual reserve requirement, when the prediction was lower than the true requirement. Large shortfalls correspond to significant underprediction of the required amount of reserve. We compare the average and maximum values for shortfall sizes in Table 3.

Table 3: Comparison of the size of shortfalls experienced for the NGESO approach and DRS model

	Mean shortfall size (MW)	Max shortfall size (MW)
NGESO	744	4406
DRS	397	1490

It is clear from Table 3 that the DRS model has significantly smaller average and maximum shortfalls than the existing approach. The difference in average shortfall

between the two approaches is 347 MW, and the difference between the largest shortfalls observed is 2196 MW.

While we have observed that there are both a smaller percentage of shortfalls for the DRS model compared to the existing approach, and these shortfalls are smaller in terms of MWs, we also verify that the DRS model is not simply recommending vastly larger reserve values to reach this performance. Analysis of the average prediction of the two approaches is given in Table 4, computed for the NGESO approach using historic predictions and positive errors provided to us during the project.

Table 4: Comparison of average predictions for the NGESO approach and DRS model

Data subset	Average prediction (MW)	
	DRS	NGESO
Overall	3546	3536
Monday	3501	3478
Tuesday	3564	3537
Wednesday	3524	3466
Thursday	3524	3613
Friday	3560	3574
Saturday	3689	3572
Sunday	3482	3521

We observe that the average predictions the DRS model and NGESO approach make are not significantly different in absolute or relative terms, meaning the gains observed throughout this section are not simply due to the DRS model outputting larger predictions.

The analysis shared by NGESO highlighted 20 periods where the existing approach experienced the largest shortfall. Of these 20 periods, half were given in BST time periods, and half were given in GMT time periods. We found that 10 of these periods lay in our test dataset. Across these days, the existing approach experienced shortfalls of several thousand MWs, however we found that the DRS model only experienced 1 shortfall in this period, of 16 MWs. It must be noted however that, due to the changes in error definitions developed during this project, the ‘actual’ values in this case for the two approaches were significantly different. It is therefore appropriate for us to isolate what we find to be the 10 days with the most severe shortfalls in the DRS work, and show these to understand how our model behaves in the extremes. Such analysis is shown in Figure 9.

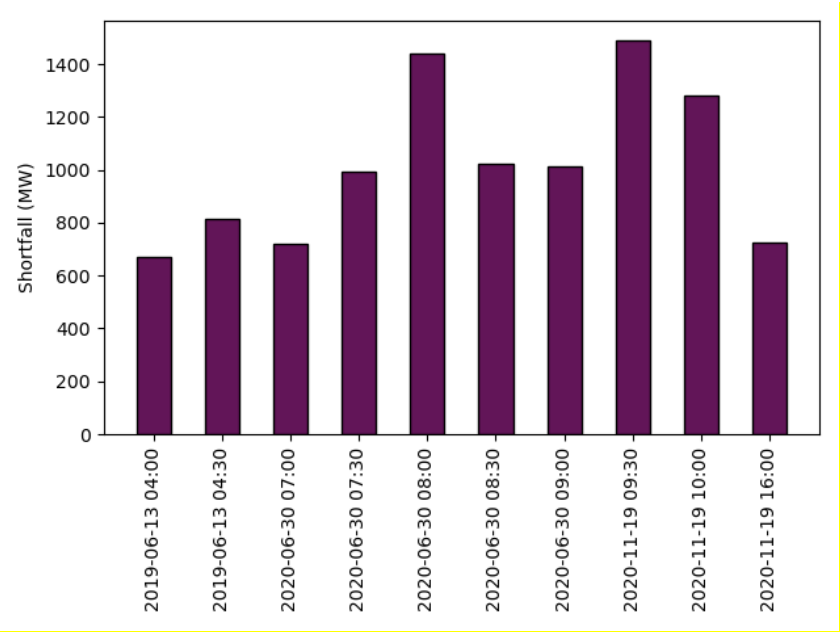


Figure 9: The 10 largest shortfalls observed by the DRS model

Inspecting Figure 9, we see that the 10th most extreme shortfall we observed from the DRS model was just above 600 MW, showing that that size of the *extreme* shortfalls the DRS model experiences are comparable to the *average* shortfall the existing approach experiences. Many of the time-points shown are also adjacent in time, showing a clear temporal dependence; alluding to the fact that there may be much to gain by considering very short-term (30 minute) reserve prediction. This is discussed further at the end of this report.

Throughout this section, we have compared the DRS model to the existing approach used by NGESO and seen that the DRS model has both fewer shortfalls, smaller shortfalls (in terms of MWs) and does not recommend significantly larger reserve values. This must be considered in conjunction with the modifications made to the error definitions. However, these results do indicate that the overall achievement of the DRS project compared to the existing approach is to generate more reliable reserve recommendations, with smaller shortfalls while offering even more transparency in the model predictions than the existing method.

Recommendations

In this section, we set out our recommendations both for directly building on the DRS PoC and for future innovation: using short-notice time series models, and an alternative approach to calculating URE.

Productionisation

In the DRS project, we have created a proof-of-concept implementation of a set of machine-learning-based dynamic reserve setting models. For these models to be used in the control room, it needs to be productionised, to make it robust and appropriate for that use case. We expect that the main steps involved in this productionisation are:

- Building robust, production-ready data pipelines that bring data from existing NGESO databases like NED, process it to make it model-ready, and then store that processed data in a DRS-specific database. This will involve not only moving from reading in csv files to connecting to existing NGESO databases (which we expect to be a significant challenge), but also creating code that is robust to data issues e.g. missing data.
- Productionising, and automating where appropriate, the code needed to train and run the models. We expect this to include automating the running of scripts to pull data from existing NGESO databases as well as putting appropriate systems in place to handle failures of those scripts – if the script failed because the data was unavailable, for example, we may wish to have it re-run at a later date.
- Ensuring that the model outputs are accessible by the relevant NGESO systems, which we expect to be by creating an API.
- Creating an interface that allows users to run the models, and to see the results.

In addition to these steps to productionise what has been implemented in the DRS PoC, further work could be done to build on and enhance the approach:

- Incorporate trading schedules into calculation of interconnector errors. Currently, a unit is assumed to contribute zero to the error in a given settlement period if NGESO trade on it during that settlement period.
- Dynamically find a BMU list, rather than using the predefined list (provided by NGESO) that the DRS PoC relies on for filtering. We believe this would be a challenging problem to solve.

- Explore (open) data sources beyond those provided for the DRS project, such as wind forecasts in Germany, to see whether they would be useful features for the models.

Future innovation and improvement

In this section, we set out two recommendations for future innovation: using short-notice time series models, and an alternative approach to calculating URE.

We fundamentally formulated the modelling challenge as a regression problem: to predict the forecast error at a settlement period, given a set of system variables. This is driven by the operational need to have reserve levels known with significant notice for a range of lead times. However, in preliminary investigation, we found that a very strong predictor for a settlement period's forecast error is the error of the preceding period; in other words, the forecast error is highly correlated in time, and therefore so is the reserve required. This suggests that a two-tiered approach, with a long-notice forecast then corrected by a short-notice time series model, could add more predictive power. This would need careful system engineering to ensure the necessary live streaming data were available for the short-term forecasting, and that the short-term adjustments are compatible with the reduced flexibility in reserve holding at short notice.

Another potential area for innovation is the approach to calculating URE. The current URE formulation's multiple cases and sharp jumps between those cases lead to sensitivity, with a small change in an input having a large effect on URE. As well as this sensitivity, the current formulation makes assumptions that, given the variety of new types of units that are operating now and may operate in the future, might no longer be valid. The current formulation assumes, for example, that units with an NDZ of 20 or less can be counted as potentially available to contribute in this settlement period. With the variety of current and future types of units, we expect that some units might have a longer NDZ but be able to contribute meaningfully very quickly after that NDZ period, while others may take so long to ramp up that, even if their NDZ is less than 20, they wouldn't make a meaningful contribution. We recommend replacing this assumption with a more nuanced consideration of ramp rates. The current formulation also considers units individually without taking account of the properties of the wider system, such as group constraints, so could be overestimating the capacity that is actually available in the system.

URE is essentially the difference between the capacity expected to be available over a settlement period and the actual capacity available, and this could be calculated using the cost bands run mode that is currently in development (at the pre-release stage) as part of ongoing work on the Modernized Dispatch Algorithm (MDA). This run mode of the MDA respects group constraints, power ranges and ramp limits, but disregards national generation requirement. Instead, the optimiser is first run with the objective to maximise generation (whilst satisfying the listed constraints). The optimiser is then run again with the objective to minimise generation. This then tells us how much headroom and footroom one has in the considered dispatch window. The run mode further allows users to specify limits on the bids and offers that will be accepted, revealing the headroom and footroom available subject to economic considerations. This is currently considered on the dispatch time-scale but may be suitable for adaptation to the problem of URE calculation.

A simplified version of the problem solved by the cost bands run mode might also be sufficient for the purposes of reserve setting. We could, for example, assume that group constraints are static and take a single, representative value for each constraint, rather than using the full dynamic data. This would have the benefit of not relying on SORT data, which may be hard to access. We suggest formulating a simplified approach and comparing the results it generates with the results produced using the cost bands run mode of the MDA, to determine whether the simplified version of the problem would be suitable for use in reserve setting.

Conclusions

In this report, we have set out the development of, the performance of, and the value brought by the DRS PoC. Our machine learning models, driven by dynamic data inputs, perform well when compared with a constant reference model. The 4-hour-lead-time, 99.7% positive reserve model[8] also gives fewer shortfalls, that are, both on average and when considering the maximum, smaller than the current approach. In addition to this, we provide explainability in a future-proof way that doesn't rely on the current approach's pre-defined pots of reserve. Our enhanced definition of URE, while suffering from some of the same limitations as NGESO's current definition, builds on and extends that current

[8] The model most appropriate for comparison with NGESO's current approach.

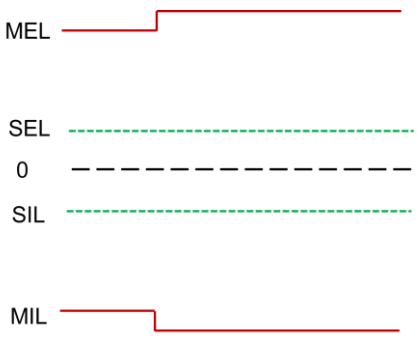
definition. We also transfer knowledge from the URE definition to formulate a DRE definition, which then forms the basis of models for setting negative reserve. Our database of processed data provides a solid foundation both for this work, and also for future work relying on the same data. Having this database in place means that in that future work, less time would be needed for manipulating and cleaning data, leaving more time for producing additional insight from the data. From data inputs all the way through to explainability outputs, the DRS PoC brings value.

To make the most of the value that this DRS PoC brings, we recommend that NGESO take steps to productionise it. In parallel with this, we recommend that NGESO explores short-notice time series models to enhance and the predictive power of the reserve setting models, as well as an alternative, innovative approach to calculating URE that will overcome the limitations of the current approach and provide a way of calculating upwards reserve error that is resilient to future changes in the electricity system.

Appendix A: Detailed definitions of errors

Upwards Reserve Error (URE)

For non-wind, non-interconnector units, URE at lead time L is the difference in how much we can increase the unit's generation (or decrease its demand) to between lead time L and real time. For generation-side and bidirectional units that we expect to be providing non-zero generation (or that can start providing this is less than 20 minutes, so could contribute this settlement period), this difference is normally the Maximum Export Limit (MEL) at lead time L minus MEL at real time (below we present various special cases that differ from this). For demand-side units that we expect to have non-zero demand, this difference is the Stable Import Limit (SIL) at lead time L minus SIL at real time. To get the total URE, we sum the URE contribution from each unit.



There are various special cases, where the level we could increase a unit's generation to (or decrease its demand to) is not equal to MEL (or SIL):

- For generation side units where there is a step change downwards in MEL, there may be times while the unit is ramping down to its new value when metered output (MO) is higher than MEL.
- For generation side units that are undergenerating by more than 10%, we assume that something has gone wrong, and the maximum they could provide at real time was MO. This assumption was created with large coal plants in mind, and might not be so relevant to newer types of units, such as aggregators and batteries.
- For generation side units that are undergenerating, but not by as much as 10%, we assume that the maximum they could provide at real time is reduced by the amount they were undergenerating by.
- For demand side units where there is a step change in SIL, taking it further away from zero, there may be times while the unit is ramping to its new value when metered output is closer to zero than SIL.

With these special cases in mind, the URE for non-wind, non-interconnector units is as follows:

If (FPN_MW != 0 or CL != 0 or PN_L !=0 or (FPN_MW = 0 and NDZ <= 20)) and no trades initiated by NGESO for this unit and settlement period (at any lead time):

If the unit is due to be either exporting or importing at a non-zero level, or could start up sufficiently quickly. Note that we assume all variables can be evaluated, but in reality there may be missing values that impact the results. Currently we use the default MySQL behaviour when missing values are encountered.

If MEL_L > 0 or MEL_RT > 0 or MO > 0:

If we expect, either at lead time L or at real time, that its generation could be set to some non-zero value. This will cover generation-side and bidirectional units, which means that, for bidirectional units, only their MEL loss, and not their SIL loss is considered – future work could include an extension to also cover SIL loss. Note that if a "broken meter" tag, identifying units where metering should be ignored so that they aren't misidentified as e.g. bidirectional, could be included here if available, but has not been included in the DRS PoC.

If MO < SEL_RT and MO > 0:

If the unit is operating below SEL and is generating (so we exclude bidirectional units that could generate, but are currently importing). Note that in the DRS PoC, we compare with SEL_RT + 0.25 rather than SEL, and 0.25 rather than 0, to avoid numerical precision issues.

If RT_CCL[t] – RT_CCL[t-1] > epsilon (where epsilon is a small number, used in place of zero, to avoid numerical precision issues):

If the unit is ramping up from a sync event, we look only at underdelivery rather than also at MEL loss. Note that we currently determine if a BMU is ramping from a sync event or to a desync even based on changes in its RT_CCL through time. If a BMU has RT_CCL at or above SEL, then we do not consider this to be ramping due to a sync event – when a BMU has reached SEL, we assume it is synchronised. There is an edge case where RT_CCL has reached SEL but MO lies below SEL. In this case, the DRS PoC assumes the unit's contribution would be MEL_L – MAX(MO, MEL_RT), but this could be changed in future work, if desired.

RT_CCL – MO

Else if RT_CCL[t] – RT_CCL[t-1] < - epsilon:

If the unit is ramping down to a desync event, or is not ramping, then we look at MEL loss, taking into account the metered value taking time to ramp down following a MEL redeclaration.

MEL_L – MAX(MO, MEL_RT)

Else:

The unit is operating between 0 and SEL, but isn't ramping.

MEL_L – MAX(MO, MEL_RT)

Else if RT_CCL > 0.1 and (RT_CCL – MO) / RT_CCL > 0.1:

If the unit is undergenerating by more than 10%. Note that we compare RT_CCL with 0.1, rather than 0, to avoid numerical precision issues where RT_CCL should be zero, but is showing up as a very small nonzero number. The issues will still occur, but around RT_CCL=0.1, which doesn't appear as much in the data. Note also that numerical precision may have an effect when the value of (RT_CCL – MO)/RT_CCL is extremely close to 0.1. In the DRS PoC implementation, we use the default behaviour of MySQL.

MEL_L – MO

Else if RT_CCL > 0 and (RT_CCL – MO) > 0:

If it is undergenerating, but not by as much as 10%

MEL_L – (MEL_RT – (RT_CCL – MO))

Else:

If it is a generation-side or bidirectional unit that isn't undergenerating

MEL_L – MAX(MO, MEL_RT)

Else if FPN_MW < 0 or CL < 0:

If it's a pure demand-side unit. Note that considering the value of MIL could also determine this, although that has not been used in the DRS PoC.

If MO > SIL_RT:

If the (demand-side) unit is operating between 0 and SIL

If RT_DCCL[t] – RT_DCCL[t-1] < - epsilon

If the unit is ramping from a sync event, then include just the underdelivery part

RT_DCCL – MO

Else if RT_DCCL[t] – RT_DCCL[t-1] > epsilon:

If the unit is ramping to a desync event

SIL_L – MAX(MO, SIL_RT)

Else:

If the unit is operating between SIL and 0, but isn't ramping

SIL_L – MAX(MO, SIL_RT)

Else:

For demand-side units that have reached SIL

SIL_L – MAX(MO, SIL_RT)

Else:

All other cases, which will include pure demand-side units that can start up within 20 minutes (which we want to exclude, because those demand-side units starting up wouldn't help create headroom)

0

Else:

Units that we either don't expect to be exporting/importing, or cannot start up within 20 minutes, contribute nothing to the URE.

0

To get the total URE, we sum the URE for each unit. The formulation above extends the formulation used in Phase 1 (which aimed to replicate the current formulation of URE) to include demand-side and bidirectional units.

Definition of terms

- FPN_MW: the MW value of the Final Physical Notification for the specified settlement period
- CL: Committed level
- PN_L: the MW value of the Physical Notification at lead time L for the specified settlement period
- NDZ: Notice to deviate from zero (minutes)
- MEL_L: MW power Maximum Export Level at lead time L ahead of the specified settlement period
- MEL_RT: MW power Maximum Export Level at real time for the specified settlement period
- MO: Metered output
- SEL_RT: the MW value of the Stable Export Level at real time, for the specified settlement period
- RT_CCL: Real time capped committed level, minimum of MEL_RT and FPN_MW + BOA_MW. When considering ramping, we look at RT_CCL for the specified settlement period, denoted by RT_CCL[t], and for the settlement period immediately before it, denoted by RT_CCL[t-1]. Where simply RT_CCL is written, this refers to the specified settlement period.
- BOA_MW: MW value of the BOA for the specified settlement period
- SIL_L: MW power Stable Import Level at lead time L ahead of the specified settlement period (this should be a value that is less than or equal to zero)
- SIL_RT: MW power Stable Import Level at real time, for the specified settlement period (this should be a value that is less than or equal to zero)
- RT_DCCL: Equivalent of RT_CCL, but looking at MIL rather than MEL. RT_DCCL = MAX(MIL_RT, FPN_MW + BOA_MW). We've chosen to refer to it as RT_DCCL (downwards CCL), but is there a term that NGESO use for this? As with RT_CCL, when considering ramping, we look at RT_DCCL for the specified settlement period, denoted by RT_DCCL[t], and for the settlement period immediately before it, denoted by RT_DCCL[t-1]. Where simply RT_DCCL is written, this refers to the specified settlement period.

Downwards Reserve Error (DRE)

Extending what was done in Phase 1, where we considered only URE for positive reserve, we can transfer this to negative reserve, and create a formulation for Downwards Reserve Error (DRE). For non-wind, non-interconnector units, we consider the difference in how

much we can decrease the unit's generation (or increase its demand) to between lead time L and real time. DRE is a mirrored version of URE, considering MIL and SEL rather than MEL and SIL. As part of the DRE formulation, we define the mirrored version of CCL, which we denote DCCL ("downwards CCL") and define as RT_DCCL = MAX(MIL_RT, FPN_MW + BOA_MW). The DRE formulation is as follows:

If (FPN_MW != 0 or CL != 0 or PN_L!=0 or (FPN_MW = 0 and NDZ <= 20)) and no trades initiated by NGESO for this unit and settlement period (at any lead time):

If the unit is due to be either exporting or importing at a non-zero level, or could start up sufficiently quickly. Note that, as with URE, in the DRS PoC, any missing data is handled in the MySQL default way.

If MIL_L < 0 or MIL_RT < 0 or MO < 0:

If we expect, either at lead time L or at real time, that its demand could be set to some non-zero value. This will cover demand-side and bidirectional units, which means that, for bi-directional units, only their MIL loss, and not their SEL loss is considered – future work could include an extension to also cover SEL loss.

If MO > SIL_RT and MO < 0 :

If the unit is operating below SIL and is importing (so we exclude bidirectional units that could import, but are currently exporting)

If RT_DCCL[t] – RT_DCCL[t-1] < -epsilon

If the unit is ramping from a sync event. Note that, as with URE, we currently determine if a BMU is ramping from a sync event or to a desync even based on changes in its RT_CCL through time. There is an edge case where RT_CCL has reached SIL but MO lies above SIL. In this case, the DRS PoC assumes the unit's contribution would be MIL_L – MAX(MO, MIL_RT), but this could be changed in future work, if desired.

RT_DCCL – MO

Else if RT_DCCL[t] – RT_DCCL[t-1] > epsilon:

If the unit is ramping to a desync event

MIL_L – MIN(MO, MIL_RT)

Else:

If the unit is operating between SIL and 0 but is not ramping

MIL_L – MIN(MO, MIL_RT)

Else if RT_DCCL < 0 and (RT_DCCL – MO) / RT_DCCL > 0.1:

If the unit is underimporting by more than 10%

MIL_L – MO

Else if RT_DCCL < 0 and (RT_DCCL – MO) < 0:

If it is underimporting, but not by as much as 10%

MIL_L – (MIL_RT – (RT_DCCL – MO))

Else:

If it is a demand-side or bidirectional unit that isn't undergenerating

MIL_L – MIN(MO, MIL_RT)

Else if FPN_MW > 0 or CL > 0:

If it's a pure generation-side unit

If MO < SEL_RT:

If it's operating between 0 and SEL

If RT_CCL[t] – RT_CCL[t-1] > epsilon:

If it's ramping up from a sync event

RT_CCL - MO

Else if RT_CCL[t] – RT_CCL[t-1] < -epsilon:

If it's ramping down to a desync event

SEL_L – MIN(MO, SEL_RT)

Else:

If it's operating between 0 and SEL but not ramping

SEL_L – MIN(MO, SEL_RT)

Else:

If it is operating above SEL

SEL_L – MIN(MO, SEL_RT)

Else:

All other cases, which will include pure demand-side units that can start up within 20 minutes (which we want to exclude, because those demand-side units starting up wouldn't help create headroom)

0

Else:

Units that we either don't expect to be exporting/importing, or cannot start up within 20 minutes, contribute nothing to the URE.

0

To get the total DRE, we sum the DRE for each unit.

Definition of terms

- MIL_L: MW power Maximum Import Level at lead time L ahead of the specified settlement period (this should be a number that is less than or equal to zero)
- MIL_RT: MW power Maximum Import Level at real time for the specified settlement period (this should be a number that is less than or equal to zero)
- SEL_L: MW power Stable Export Level at lead time L ahead of the specified settlement period

Other types of error

Wind error

For wind units, we take the error to be the difference between the forecast (FOL) wind and the metered output, unless there is an active BOA (in that case, we set the error to be zero, since we don't want actions initiated by NGESO to be counted as part of the error). We sum the contribution from each unit to give the total wind error.

Interconnector error

For interconnectors, we take the difference between the PN at lead time L and the final PN. Following discussions with NGESO, we use the final PN rather than the metered output because, for interconnectors, metered output varies considerably depending on where it along the interconnector it is measured. As with the wind, we want to exclude from this error calculation any cases where actions have been initiated by NGESO. To do this, we use trading data, excluding units where a trade has been initiated by NGESO for the settlement period of interest (at any lead time). We sum the contribution from each unit to give the total interconnector error.

Reserve for response error

For lead times of up to and including 4 hours, we calculate the reserve for response error using the expression provided to us by NGESO:

((1260 - (0.01 * national demand forecast at lead time L) - 500) / 0.68 / 0.6) - ((1260 - (0.01 * actual national demand) - 500) / 0.68 / 0.6).

Demand error

For the demand error, we simply take the difference between the actual national demand and the national demand forecast with a lead time L.

Appendix B: Model features

Here we list the data features used to construct the DRS PoC models. As part of the model construction process, features of the following types were considered:

- Temporal (e.g., time of day, day of week)
- Weather (e.g. wind speed, solar radiance)
- National demand
- Embedded wind and PV load factors
- Interconnector flows

with multiple mutations of these base features considered for each including lagging and differencing across settlement periods, days and weeks. Only features which improved the model performance were retained in the final PoC models.

For positive reserve error the features used for the PoC models were:

- Time of day
- Month of year
- Day of week
- Bank holiday indicator
- National demand lagged by 1 day
- Embedded wind load factor
- Embedded PV load factor differenced across consecutive settlement periods
- Standard deviation across weather stations of the forecast mean wind speed
- Mean across weather stations of the forecast mean wind speed differenced across consecutive settlement periods

For negative reserve error the features used for the PoC models were:

- Time of day
- Month of year
- Day of week
- Bank holiday indicator
- Embedded wind load factor
- Embedded wind load factor lagged by 1 day

- Embedded PV load factor

Appendix C: Further performance comparison

Here we show model performance for the models targeting positive reserve error at the 99% quantile and negative reserve error at the 1% and 0.3% quantiles. The results are similar to those shown above for the 99.7% quantile for positive reserve error.

Figure 10, Figure 11 and Figure 12 plot the quantile loss of the proof-of-concept models and reference models for each lead time for the 99%, 1% and 0.3% target quantiles respectively. As seen for the predictions of the 99.7% quantile, the DRS PoC models perform better than the reference models, especially at long lead times.

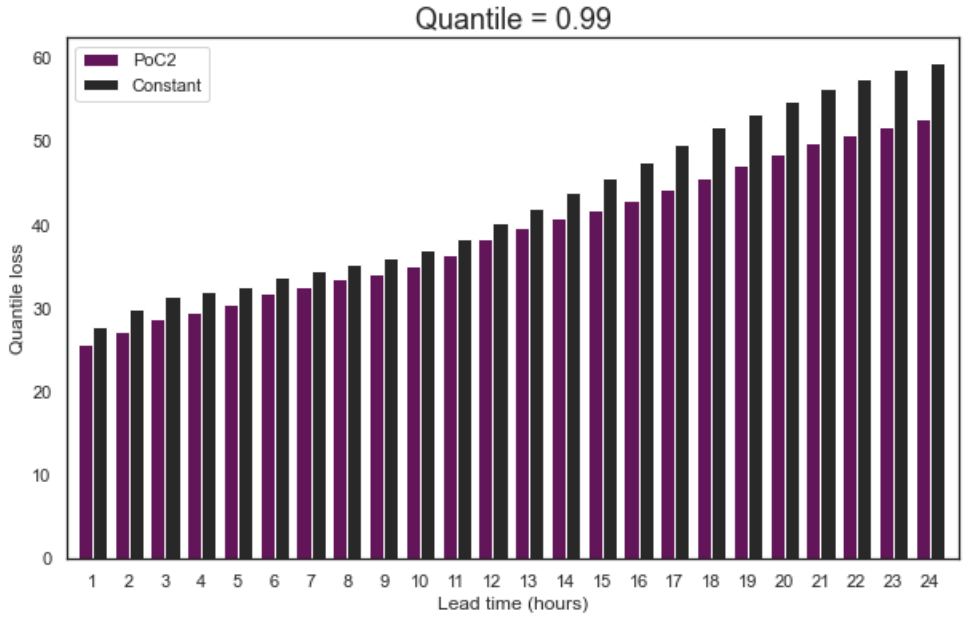


Figure 10: Quantile loss across lead times for positive reserve predictions at the 99% quantile for the DRS PoC models (purple) and reference model (dark grey).

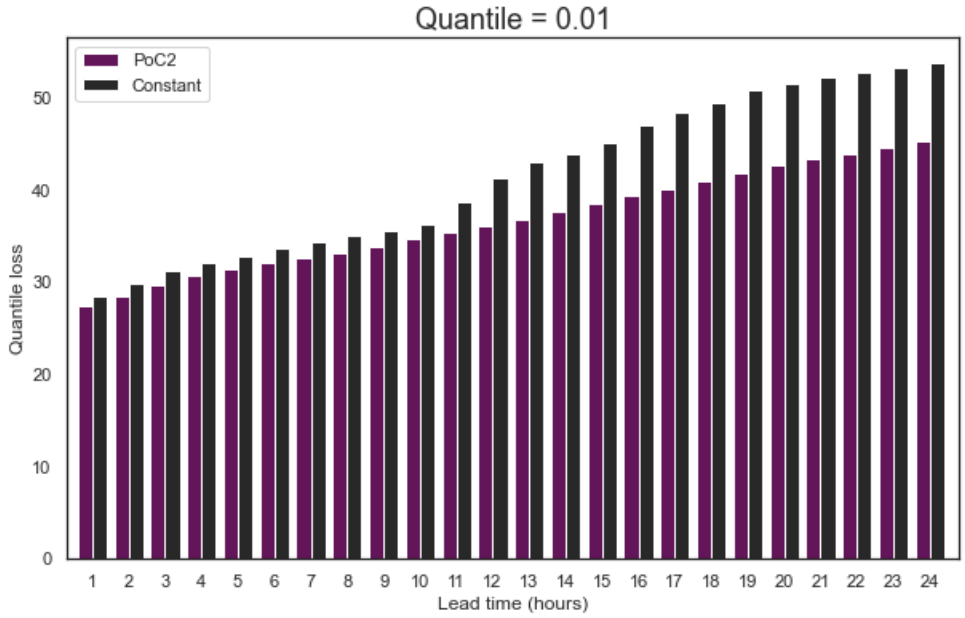


Figure 11: Quantile loss across lead times for negative reserve predictions at the 1% quantile for the DRS PoC (purple) and reference model (dark grey).

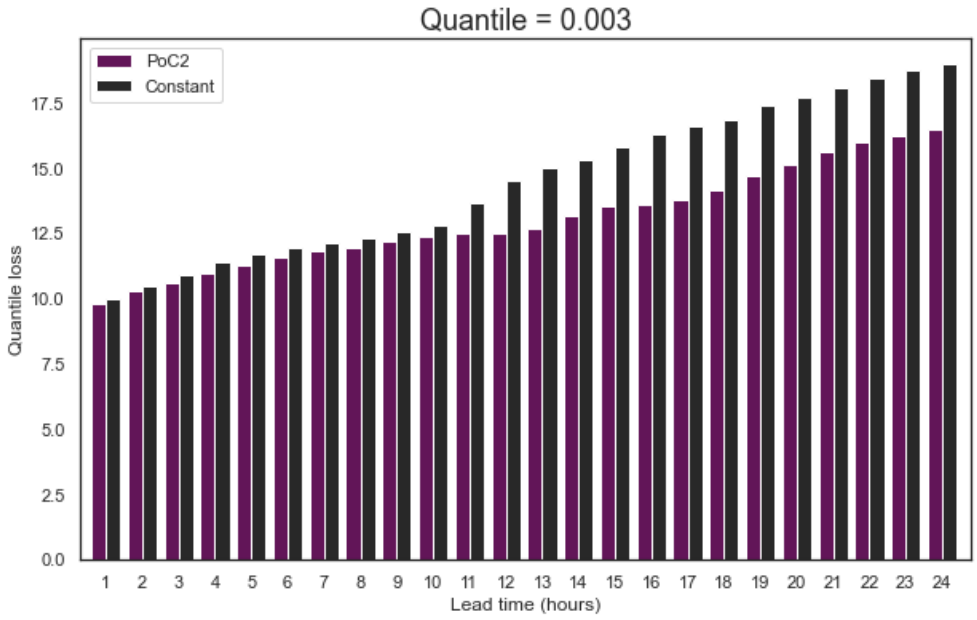


Figure 12: Quantile loss across lead times for negative reserve predictions at the 0.3% quantile for the DRS PoC (purple) and reference model (dark grey).

Figure 13, Figure 14 and Figure 15 plot the average model prediction of the proof-of-concept models and reference models for each lead time for the 99%, 1% and 0.3% target quantiles respectively. Again, we see that the proof-of-concept models perform better than the reference models, especially at long lead times.

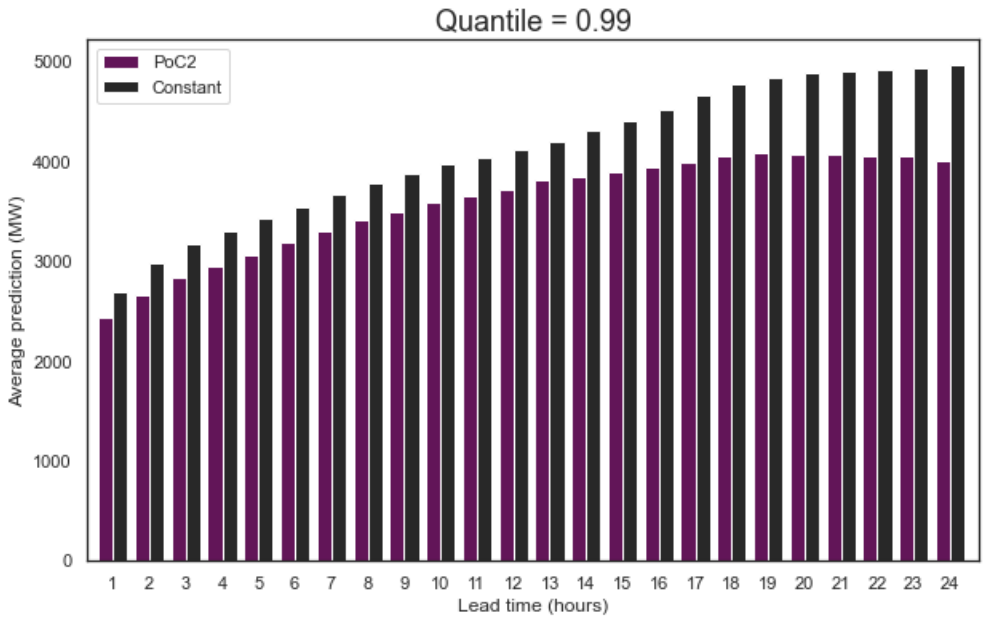


Figure 13: Average model prediction across lead times for positive reserve predictions at the 99% quantile for the DRS PoC (purple) and reference reference model (dark grey).

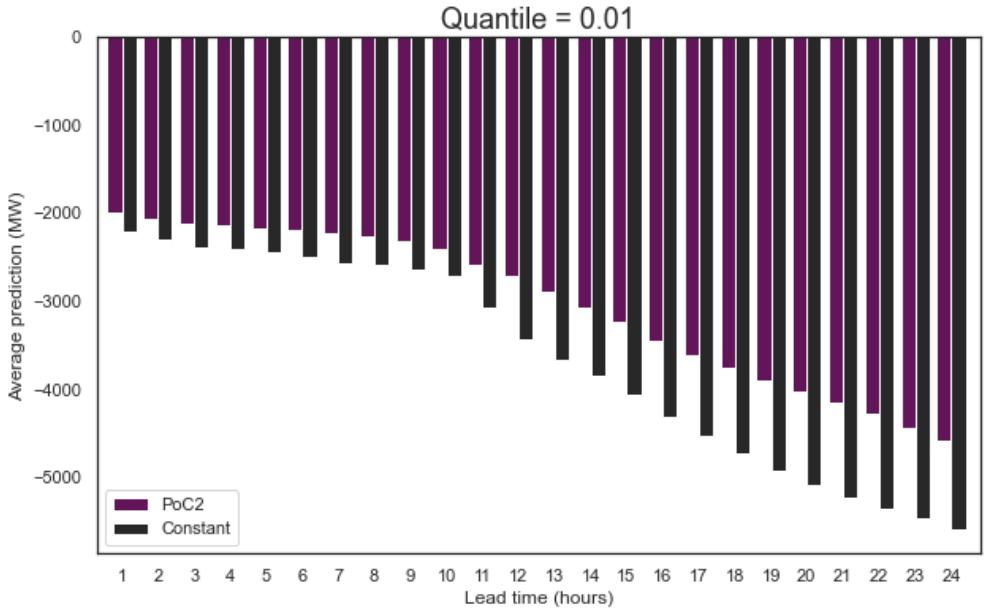


Figure 14: Average model prediction across lead times for negative reserve predictions at the 1% quantile for the DRS PoC (purple) and reference model (dark grey).

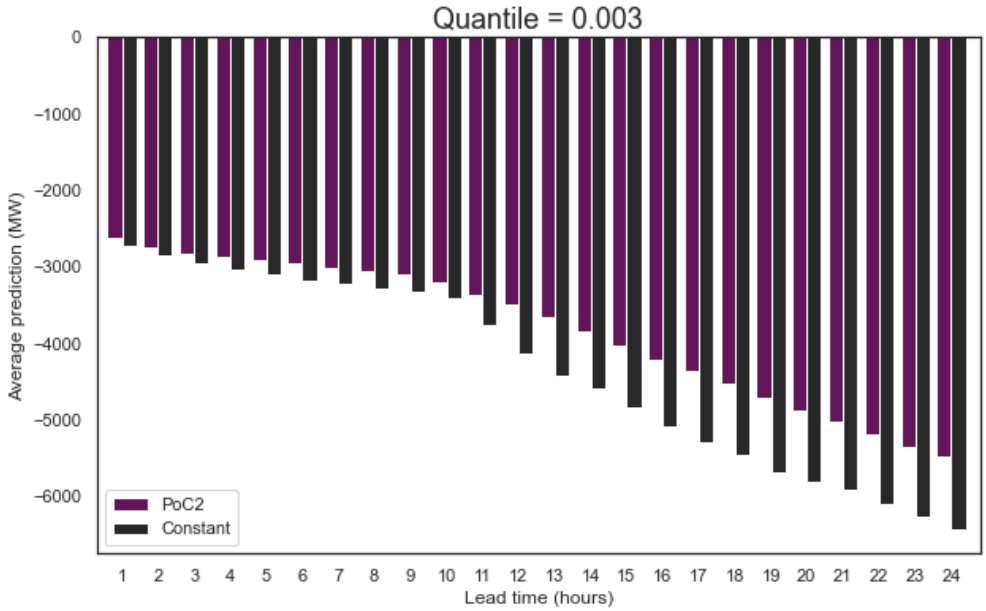


Figure 15: Average model prediction across lead times for negative reserve predictions at the 0.3% quantile for the DRS PoC (purple) and reference model (dark grey).

As was seen for the 99.7% positive reserve predictions, the exceedance fraction for the remaining three quantiles is higher for the DRS PoC models than for the reference models. Figure 16, Figure 17 and Figure 18 plot the fraction of settlement periods when the model prediction was exceeded by the true reserve error for target quantiles of 99%, 1% and 0.3% respectively. These are instances where the models are recommending a level of reserve which would have been insufficient. Again, this higher exceedance rate was deemed acceptable since it remains unclear what the level of exceedance should be to match NGESO's risk appetite for reserve setting.

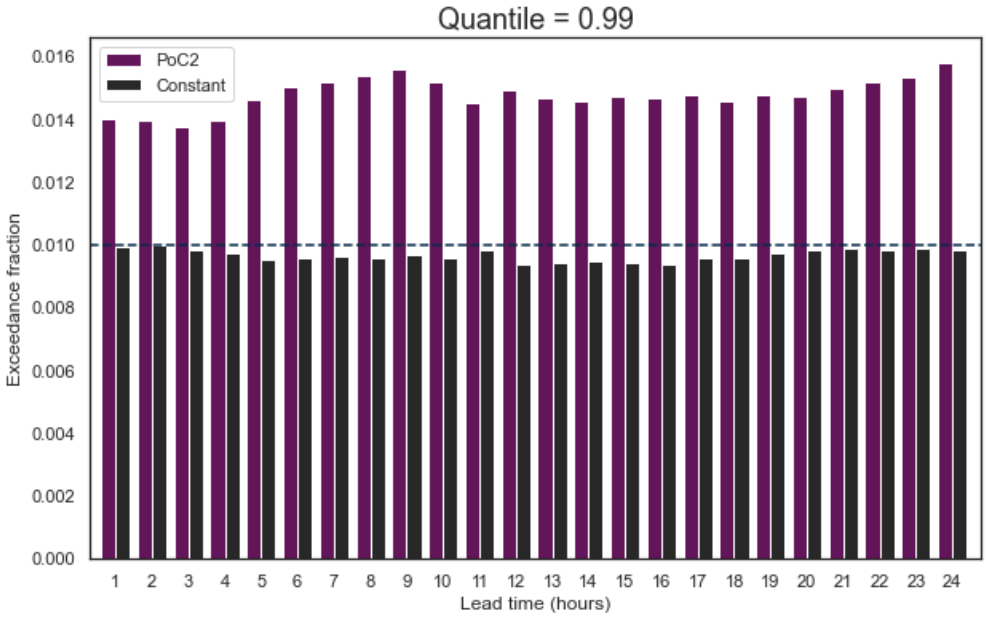


Figure 16: Fraction of settlement periods when the true reserve error exceeded the model prediction across lead times for positive reserve predictions at the 99.7% quantile for the DRS PoC (purple) and reference model (dark grey).

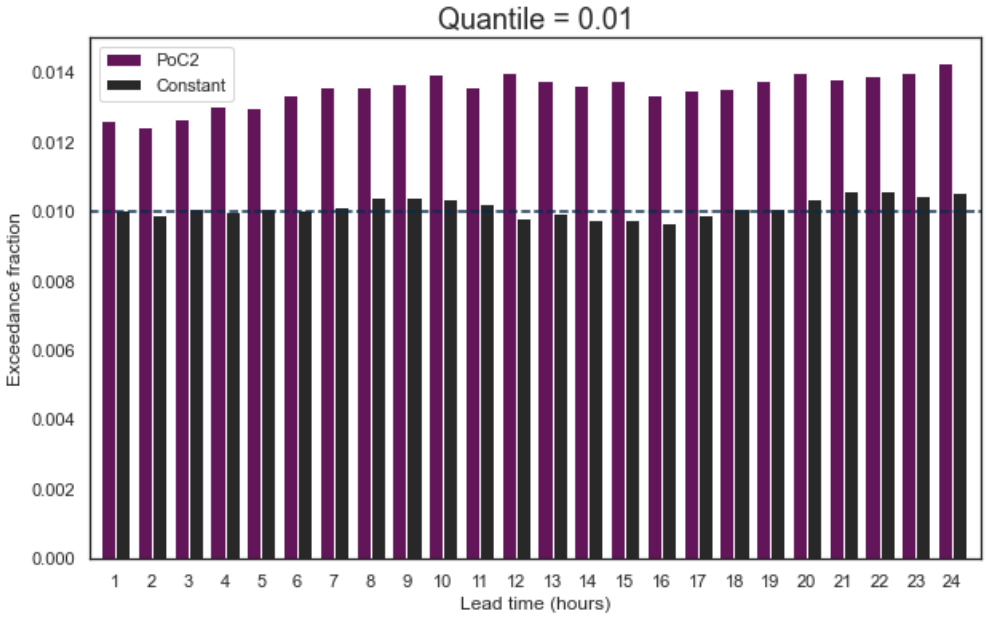


Figure 17: Fraction of settlement periods when the true reserve error exceeded the model prediction across lead times for negative reserve predictions at the 1% quantile for the DRS PoC (purple) and reference model (dark grey).

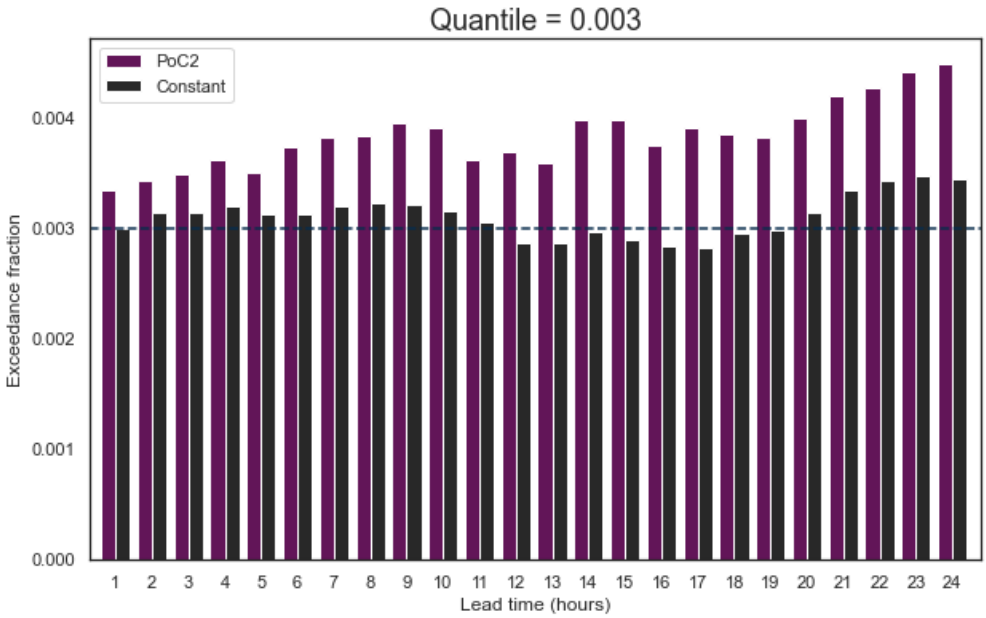


Figure 18: Fraction of settlement periods when the true reserve error exceeded the model prediction across lead times for negative reserve predictions at the 0.3% quantile for the DRS PoC (purple) and reference model (dark grey).