Commodity Forecasting – a comparison of traditional and machine learning models  
Callum Barr  
*School of Accounting, Finance and Economics, University of Waikato*

# Motivation and research design

Exports play a strong role in the economies of Australian and New Zealand, contributing 26.8% and 24.1% of their respective GDPs (World Bank, 2025). Within these exports, commodity prices are crucial drivers due to their significant share in export markets. For example, dairy products accounts for 29% of in New Zealand exports, while mineral fuels make up for 35% of Australia’s exports (Trading Economics, 2025).

Accurate forecasting of commodity prices is essential for a range of organisations and financial market participants. Fonterra, for example, relies on accurate forecasts to set farmgate milk prices, which enables farmers to plan their operations effectively. Similarly, RBNZ uses such forecasts to inform monetary policy, while speculators and traders use them to manage risk and maximise their profits.

Traditionally, time series forecasting has relied on statistical models such as autoregressive (AR), autoregressive integrated moving average (ARIMA), and exponential smoothing. Recently, however, machine learning (ML) models have entered the landscape as alternatives, offering the ability to better capture complex patterns in data. The approach to do so differs from traditional statistical models which are typically built on assumptions about the relationship between dependent and independent variables. In contrast, ML models do not rely on such assumptions. Instead, they focus on identifying a combination of variables that minimizes forecast error (output) (Jung, Patnam, & Ter-Martirosyan, 2018). ML models have two components, the algorithm finds the relationship between the inputs and outputs, while the learning method identifies the best fit for the input variables based off the data, (Jung, Patnam, & Ter-Martirosyan, 2018) discuss this in greater detail in their paper.

The increasing use of machine learning models is evidenced in the Makridakis (M) Competitions, a well-known series of forecasting challenges. In the latest iteration, machine learning forecast models were commonplace (Makridakis S. , et al., 2024), particularly when compared to earlier competitions (Makridakis, et al., 1982).

Despite the growing popularity of ML models, some studies find ML models perform better compared to traditional methods (e.g. (Makridakis, Spiliotis, & Assimakopoulos, 2022)), while others do not (e.g. (Makridakis S. , et al., 2024)). Given this uncertainty in literature, this research aims to clarify some of the debate. Specifically, we evaluate the out-of-sample accuracy of various traditional and machine learning forecasting models for 11 commodity indices.

# Background

In an article, (Makridakis, Spiliotis, & Assimakopoulos, 2018) examined the claims that new ML models outperform traditional methods in forecasting. They found three common limitations in studies: conclusions were often based on limited datasets, only short-term forecasting horizons were considered, and naïve benchmarks were not used for comparison. These practices exaggerated the performance of ML models without providing sufficient proof. Using data from the M3 competition, their study found that the six most accurate models were statistical, with a simple Naïve 2 model (its forecast is equal to the last observed value incremented by a drift term) outperforming half of the ML methods. They concluded that, on the datasets analysed, traditional statistical methods were more accurate than ML models. Despite these findings, the authors remained optimistic about ML's potential, anticipating performance improvements as the field evolves.

Of the M competitions, the last three, M4 (Makridakis, Spiliotis, & Assimakopoulos, 2020), M5 (Makridakis, Spiliotis, & Assimakopoulos, 2022) and M6 (Makridakis S. , et al., 2024) are the most relevant to this study. The M4 competition found that hybrid approaches, that combined statistical methods and ML methods were more accurate than pure statistical or ML methods. The winning model (Smyl, 2020) used such an approach by combining exponential smoothing with recurrent neural network (RNN). Pure ML models performed relatively poorly, with only one of five performing better than the Naïve 2 benchmark.

The M5 competition marked the turning point for ML models, where for the first time pure ML models were the top performers. This competition tested prediction of sales for Walmart, with the winning submission using an equal weighted combination of LightGBM models. Second place used forecasts produced from N-BEATS and LightGBM model, while other models like DeepAR and N-BEATs showed potential. The competition showed ML models can be a more accurate tool in forecasting retail sales, allowing businesses to improve decision making.

The latest competition, M6, focused on forecasting financial markets, requiring participants to forecast 100 publicly traded assets over 12 submission dates throughout the year. This live forecasting format allowed competitors to adjust their models and use exogenous variables. To assess the usefulness of forecasts participants were also asked to provide investment positions in each of the assets, this aspect reveals if the forecasts were utilized to make good investment decisions. Financial forecasting proved to be difficult due to high volatility and the range of factors that influence them. This was reflected in the results where, only three teams outperformed the bench mark. Of the top 5% performers, four used traditional methods while three used ML based approaches. Given the similarities of commodity and financial markets where both are traded and have multiple influencing factors, M6 is particularly relevant to this research.

The Monash Time Series Forecasting Repository (Godahewa, Bergmeir, Webb, Hyndman, & Montero-Manso, 2021) is another significant contributor to forecasting research, offering a platform to compare forecasting models (Godahewa, Bergmeir, Webb, Hyndman, & Montero-Manso, 2025). This repository consists of 30 data sets with varying frequencies and missing values, creating 58 time series variations. It aims to fill the gap (Makridakis, Spiliotis, & Assimakopoulos, 2018) discussed, regarding the lack of comprehensive benchmark testing of new ML models. To provide a base line benchmark they tested 13 models across the data sets and published the results. Six of these were traditional models such as ARIMA and Exponential smoothing (ETS), while the remaining seven were global forecast models like N-Beats and DeepAR. This provides a robust benchmark for researchers to test their new models on and compare the performance to existing models. More models were added in November 2024, as outlined in a report available on the website (Godahewa, Bergmeir, Webb, Hyndman, & Montero-Manso, 2025). Of the new global models N-HiTS emerged as strong performer, producing the lowest mean absolute scaled error (MASE) seven times across the different datasets.

A limitation of past forecast error studies is the absence of Australian and New Zealand commodity datasets in the analysis. Additionally, models such as TimeGPT and NeuralProphet, have not been extensively evaluated in prior forecasting competitions or the Monash Forecast Repository.

# Data and econometric methods

## Data

The commodity price indices were sourced from ANZ and RBA respectively. The ANZ commodity price index (ANZ Research, 2025) is a monthly index that captures the movements in the prices New Zealand receives for its main export categories. This study used the index calculated in New Zealand dollar terms. The index is heavily weighted towards dairy (44%) and meat, skin and wool (12%) (ANZ Research, 2025), with additional sub-indexes for meat, skins and wool; Dairy Products; Horticultural Products; Forestry Products; and Aluminium. As well as analysing models’ ability to forecast the overall ANZ commodity price, we also evaluate the models’ ability to forecast the aforementioned sub-indices.

The RBA Index of Commodity Prices (RBA, 2025) is a monthly index of Australia’s export prices in Australian dollars, and is comprised of 21 commodities (RBA, 2024). Bulk commodities like iron ore account for 60% of the index, followed by other resources such as LNG (28%) and rural commodities (RBA, 2024). As with the ANZ commodity price index, we evaluate the models’ ability to forecast the overall index, as well as the following sub-indices: rural component, non-rural component, base metals and bulk commodities[[1]](#footnote-1). For comparability, the sample of commodity prices we use from both the RBA and the ANZ spans January 1993 to December 2024.

## Method

The models were evaluated through 200 prediction tests using mean absolute error (MAE) and root mean squared error (RMSE) as accuracy metrics. The data was initially split into a training set of 268 months and test set of 200 months. A rolling window approach was used to conduct the test, after making predictions for a specified forecast horizon (h), the training window was expanded by one month, the models were re-fitted, and new forecasts were tested on the updated test set [[2]](#footnote-2). This process was repeated 200 times, where the forecast errors (MAE and RMSE) were calculated for each forecast. The errors were then averaged to derive the mean forecast error of the index. Forecast horizons of one and four months were used to assess the robustness of the across different time frames.

To compare the forecasting performance of each model, the mean absolute error (MAE) and root mean squared error (RMSE) were calculated for each forecast. The total average was then calculated over the 200 forecast tests. These are scale dependant errors so they cannot be compared across the RBA and ANZ indexes due to them being in Australian and New Zealand dollars.

Where t is the number of observations in the training sample, n is maximum number of months ahead forecast accuracy is being assessed over; this is set at 1 (one month forecast ahead) and 4 (four month forecast ahead) in my case, is forecast and is the observed value (Hyndman & Athanasopoulos, 2021).

This methodology reveals which model, on average, is best at predicting the RBA and ANZ commodity indexes. Due to time constraints, exogenous variables were not included in the models.

## Statistical Models

The statistical models used in this study range from simple benchmarks to more advanced techniques. Benchmark models provide a point of comparison for evaluating the performance of more complex methods. If an advanced model cannot outperform these benchmarks, it may be unnecessarily complicated.

The Average Method is the simplest statistical model. Its forecast values are the mean of the historic data, which is described in (Hyndman & Athanasopoulos, 2018). Similarly, the Naïve Method assumes the forecast values are equal to the last observed value in the training data (Hyndman & Athanasopoulos, 2018). These methods provide a simple approach to forecasting datasets without trends or seasonality.

Building on the Naïve Method is the Drift model (also called Naïve 2), it allows the forecast to change over time by incrementing the last observed value by a drift term. This drift term is equal to the average historical change which makes it useful for modelling trends in data (Hyndman & Athanasopoulos, 2018).

The moving average (MA(k)) model computes the average of the last k observations, where k is the number of periods (months) included in the calculation. A higher value of k makes the forecast less sensitive to short-term fluctuations (Baysan, 2022).

More complicated smoothing techniques include Simple Exponential Smoothing (SES), it forecasts based on weighted averages of past observations. To prioritise more recent observed values the weights decrease exponentially as observations get older. The SES only uses the level component of the equation (Hyndman & Athanasopoulos, 2021) and is suitable for data that exhibits no trend or seasonal pattern. For data with trends a Double exponential smoothing (DES) model is more suitable. It includes a trend equation in addition to the level equation allowing forecasts to account for upward and downward trends (Hyndman & Athanasopoulos, 2021). To account for seasonality, Triple Exponential Smoothing (TES) includes an additional component called the seasonal equation. TES allows for additive seasonality, which is suitable for constant seasonal effects, or multiplicative, which adjusts proportionally to the level of the time series (Hyndman & Athanasopoulos, 2021).

An alternative to smoothing methods is the Autoregressive Model (AR), which predicts future values from past values (lags) of itself. An AR of order p, denoted as AR (p), uses p lagged terms to predict the future value (Hyndman & Athanasopoulos, 2021). For this study AR (1) was used.

The Autoregressive Integrated Moving Average (ARIMA) combines AR and MA models. It is denoted as ARIMA (p, d, q), where p refers to the order of the AR process, d represents the degrees of differencing to make the time series stationary and q specifies the order of the moving average component (Hyndman & Athanasopoulos, 2021). An ARMA model was also used, it is similar to an ARIMA but is not differenced, and was set to ARMA (1,1). ARIMA can be extended to account for seasonality through the Seasonal ARIMA model (SARIMA), which incorporates seasonal autoregressive and moving average components. It is denoted as SARIMA (p, d, q) (P, D, Q) [m], where m is the seasonal frequency (e.g. 12 for monthly data with yearly seasonality) (Hyndman & Athanasopoulos, 2021). This study used ARIMA (1,1,1), as well as ARIMA and SARIMA models where lag lengths were selected using the auto.arima function in R. This function tests different p and q values through a variation of the Hyndman- Khandakar algorithm until the minimum AICc is achieved (Hyndman & Athanasopoulos, 2024).

## ML Models

There are many potential machine learning models that can be applied for forecasting. We selected four machine learning models that are relatively new and well known. Specifically:

Prophet, developed by Facebook (Taylor & Lethan, 2017), is an open-source forecasting model that can handle features commonly seen in business time series data. It is user friendly, allowing users to optimise the model without having to understand the underlying model. The model decomposes the series into components of trend, seasonality and holidays. Their method is said to have numerous advantages over ARIMA models, in particular, its flexibility in adapting to structural changes using piecewise trends [[3]](#footnote-3).

N-BEATS (Oreshkin, Carpov, Chapados, & Bengio, 2020) is a deep neural architecture designed with backward and forward residual links and a deep stack of connected layers. Its versatile and efficient as it can be applied to a range of datasets without modification and is fast to train. N-BEATS has been tested on M3 and M4 datasets which included financial, industry, demographic, macro, micro and other time series, where it outperformed statistical benchmarks and other pure ML models (Oreshkin, Carpov, Chapados, & Bengio, 2020) [[4]](#footnote-4) .

NeuralProphet (Triebe, et al., 2021) builds upon the foundation of Facebook Prophet. It is considered a hybrid framework that is based on PyTorch and trained with deep learning methods. The author aims to introduce local context, which Prophet lacks, by incorporating autoregression and covariate models, configured as neural networks or linear regression (Triebe, et al., 2021). This aimed to improve short term forecasting. In the tests of the model by (Triebe, 2025) it was found to outperform Prophet in the short to medium forecast horizon but this came with a drawback of computational time, where on average it was four times slower [[5]](#footnote-5).

N-HiTS develops on the N-BEATS model, seeking to improve long horizon forecasting (Challu, et al., 2022). The authors (Challu, et al., 2022) identify volatility of predictions and computational complexity as issues when modelling long horizon forecasts. They address this “by incorporating novel hierarchical interpolation and multi-rate data sampling techniques” (Challu, et al., 2022). They tested the model on datasets with varying frequencies (10 minutes, 15 minutes, hourly, daily and weekly) across time series such as exchange rates, weather and traffic. Forecast horizons ranged from 96 to 720, and N-HiTS consistently performed better than other ML and ARIMA models (Challu, et al., 2022) [[6]](#footnote-6).

TimeGPT, a generative pretrained transformer for time series, is a recently published model introduced by (Garza, Challu, & Mergenthaler-Canseco, 2024). Testing it over 300,000 data sets across a range of industries showed it performed well against other models like N-HITS, as well as statistical benchmarks like exponential smoothing models. Of all the models used it scored best in monthly (12 month forecast horizon) and weekly (1 week forecast horizon) forecasts, but performed poorly in daily and hourly forecasts (Garza, Challu, & Mergenthaler-Canseco, 2024). A limitation of their analysis was not including Prophet or ARIMA tests due computational times of training them [[7]](#footnote-7).

## ML Model Optimisation

Prior to running the tests, ML models were optimized by adjusting hyperparameters. Due to time constraints, this optimization was limited to 50 forecast tests conducted on the ANZ Commodity Index with a forecast horizon of four months. The selected hyperparameter combination for each model that minimized the average MAE and RMSE during these tests was then applied across all series in the main analysis.

The documentation provided by (Prophet, 2025) listed hyperparameters that could be tuned to reduce error. Initially Prophet was run in default settings, yielding an MAE of 20.91 and RMSE of 21.98. The first parameter tuned was changepoint scale which controls how much the trend changes at trend points, and is described as being the most impactful. Trailing values between 0.1 and 0.5 revealed 0.2 as the optimal value where the MAE was 20.55 and RMSE was 21.48. Next seasonality prior scale, which controls the flexibility of the seasonal patterns, was lowered from the default of 10 to 5 but no improvement was seen, so the default setting was retained. Similarly holidays prior scale was adjusted but reverted back to default after no improvements were observed. Changing the seasonality mode from the default additive to multiplicative, increased the error so was reverted back to default. Finally tuning change point range to 0.9 from the default 0.8 improved the MAE and RMSE to 18.87 and 19.59 respectively, increasing further resulted in less accuracy. This left a final model of changepoint prior scale of 0.2 and a changepoint range of 0.9.

NeuralProphet was optimised using its hyperparameter tuning guide (NeuralProphet, 2025). The baseline model, with 12 lags and a forecast horizon of 4, achieved a MAE of 10.36 and RMSE of 11.94. NeuralProphet finds optimal learn rate, epochs and loss function through the data, so these were left as default. AR layers can be adjusted but increase computational time. (NeuralProphet, 2025) notes that often default achieves good performance, thus the parameter was unchanged. Similarly to the Prophet model, change point range was adjusted where 0.9 was found to be the optimal value, producing MAE equal to 10.15 and RMSE of 11.57. Seasonality related parameters were left as default which allowed the model to find the best combinations automatically. Increasing the lags to 24 months made the error worse, thus leaving a final model of 12 lags, a changepoint range of 0.9, and forecast horizons of 1 or 4 months.

TimeGPT was optimised following an article written by Nixtla (Nixtla, 2025). The initial model required inputs of forecast horizon (4), frequency (month start) and fine tune loss set as MAE, these were left the same throughout the optimisation process and resulted in an initial MAE of 11.99 and RMSE of 13.53. Increasing finetune steps from the default to 50 improved the MAE and RMSE to 10.45 and 11.79, respectively. Another parameter adjusted by (Nixtla, 2025) was finetune depth, unfortunately this was not available in R so was left at the default setting. Other adjustments used such as exogenous variables were not considered as that was out of the scope of study. A long horizon model was an option but the documentation recommends only using this when the forecast horizon is larger than one cycle of the time series (1 year in this case), so was not applicable. The final model included 50 finetune steps, monthly frequency, and MAE as the loss function with forecast horizon being one or four months.

For N-BEATS, hyperparameter tuning was guided by (Filho, 2023). The baseline model, with a forecast horizon of 4, 12 input lags, and 500 max steps, produced a MAE of 11.96 and RMSE of 13.58. Adjusting shared weights to be true, batch size to 256 and number of blocks to [1,1,1] had no effect, so were left out. Setting MLP units to [512, 512, 512] (Godahewa, Bergmeir, Webb, Hyndman, & Montero-Manso, 2025) improved the MAE to 11.55, unfortunate adding n-polynomials did not improve on this so was taken out. Max steps was increased to 1000 but that mode the errors worse, lowering it to 300 minimised the MAE which was now 10.69. Setting the learn rate to 0.001 marginally improved MAE to 10.64, which was lowered further by adding harmonics (n\_harmonics = 8) where the MAE was now 10.55. Increasing the input size to 24 reduced the MAE to 9.83. The final model included 24 input lags, MLP units of [512, 512, 512], 8 harmonics, 300 max steps and a learning rate of 0.001.

N-HiTS began with forecast horizon of 4, input size of 24 and max steps of 300, achieving an MAE of 9.23. Increasing max steps to 400 improved MAE to 8.9, but setting the learn rate to 0.001 did not improve upon this. Adding MLP units and adjusting n\_blocks worsened the results, while introducing a pooling kernel size of [1, 1, 1] lowered the MAE to 8.94. Additional parameters, such as frequency down sampling, reduced accuracy and were omitted. The final model retained an input size of 24, max steps of 300, a learning rate of 0.001, and a pooling kernel size of [1, 1, 1].

It is likely that there are combinations of parameters that would tune the models better but time constraints did not allow for this. To tune hyper parameters more thoroughly application of Bayesian optimisation through a library such as (Optuna, 2025) in Python is recommended. This automates the tuning process by applying combinations of hyperparameters and finding which results in the lowest error.

# Results and interpretation

## ANZ indices results

For the overall ANZ commodity index the best model was dependant on the forecast horizon. When this was one month the autoARIMA performed the best, with an MAE of 4.91. This strong performance of autoARIMA, however, did not carry over to the longer forecast horizon of four months ahead. As the forecast horizon increased to four months, the MAE rose to 8.95, with the TES additive model performing the best in this category. N-HiTS was the most accurate ML model with MAE of 5.04 for the one-month forecast, and 9.08 for the four-month horizon. The Prophet model performed poorly across both time horizons, with MAE values significantly higher than those of the other models. Specifically, the Prophet model had an MAE of 14.15 for the one-month forecast and 16.57 for the four-month forecast, outperforming only the simple average model. NeuralProphet, an enhanced version of Facebook’s Prophet, demonstrated superior performance in comparison to its predecessor, achieving lower MAE values of 5.16 for a one-month horizon and 10.26 for a four-month horizon).

Across all commodity indices, similar trends were observed (Table 1).

**ANZ meat, skin and wool:** For a one-month horizon the auto ARIMA was the best model (MAE = 5.11), followed by NeuralProphet at 5.12. When forecasting four months ahead, ARIMA (1,1,1) performed best with an MAE of 9.69, while N-HiTS, the most accurate ML model achieved a higher MAE of 9.81.

**ANZ dairy products:** ARMA (1,1) was the best model for one month and four-month horizons with MAE of 8.1 and 15.17 respectively. N-BEATS, was the most accurate ML model for one month with an MAE of 8.53, while TimeGPT produced the best result for the longer forecast at an MAE of 16.25.

**ANZ horticulture:** Interestingly the simple Naïve 2 model, outperformed all others with MAE of 2.69 for the one-month forecast. N-BEATS came close with MAE equal to 2.7, which was the best of ML models. The longer horizon changed the results, DES was best overall with an MAE of 4.6, while TimeGPT produced the best longer-range ML forecast with its MAE of 4.72, narrowly beating N-HiTS.

**ANZ forestry:** For one- and four-month horizons, SARIMA was the most accurate with an MAEs of 3.18 and 5.36. Among ML models, N-BEATS was the best performer for the one-month forecast (MAE = 3.25), while TimeGPT provided the best result for the four-month forecast with an MAE of 5.7

**ANZ aluminium:** For the first time an ML model was the most accurate, NBEATS achieved this with a MAE of 5.19 when forecasting one month. It appears complex models struggled as the Naïve 1 and 2 bench marks scored second best where the MAE was 5.21. For the four-month forecast, all complex models struggled and were not able to beat Naive 1’s score of 8.68. Here, TimeGPT was the best ML model with MAE of 9.44, narrowly beating NeuralProphet’s 9.45.

Table . ANZ index MAE and RMSE, note table had to be split and stacked. Lowest errors are in bold.



*Note: moving average order 3 (MA3), moving average order 6 (MA6), moving average order 12 (MA12), simple exponential smoothing (SES), double exponential smoothing (DES), triple exponential smoothing additive (TESAdd), triple exponential smoothing multiplicative (TESMulti), autoregressive order 1 (AR(1)), autoregressive moving average (ARMA(1,1), autoregressive integrated moving average (ARIMIA(1,1,1)). AutoARIMA is the ARIMA model that the R package finds is best, similarly with AutoSARIMA but this includes seasonality.*

## RBA index results

In the RBA total index (Table 2), the errors were low, particularly for the one-month horizon where the ARIMA (1,1,1) was the most accurate, with an MAE of 1.61. N-BEATS, the best ML model, was slightly higher at 1.9. In contrast, the Prophet model performed poorly, with an MAE of 8.14, outperforming only the simple average model. As expected, MAE increased with the longer horizon. ARIMA (1,1,1) remained the best but error approximately doubled to 3.53. The same occurred with the best ML model, however it was TimeGPT that was best with its MAE of 4.25. Prophet continued to produce high error rates, with an MAE of 8.92.

**RBA rural:** ARMA (1,1) was the most accurate model with average MAE of 1.76 for a one-month forecast. N-HiTS was the best ML model with its prediction error of 1.82. For a four-month prediction ARIMA (1,1,1) was best producing an MAE of 3.07, while TimeGPT had an MAE of 3.15, the best of ML models.

**RBA non-rural:** ARIMA (1,1,1) was the best for both horizons achieving MAE of 1.74 and 3.83. For the one month forecast N-BEATS and N-HiTS were equally the best ML models, scoring an MAE of 2.08. TimeGPT was better in the four-month horizon, where its MAE was 4.8.

**RBA base metals:** The most accurate model was an auto SARIMA, producing MAE of 2.17. N-BEATS had a MAE of 2.19 which was the best of the ML models. For the four-month horizon, the Naïve 2 model outperformed all models with an MAE of 3.88. TimeGPT’s MAE of 3.91 was the best accuracy of all ML models.

**RBA bulk:** For one-month prediction ARIMA (1,1,1) was the best with an MAE of 2.49. N-BEATS was the best ML model where its MAE was equal to 2.84. With the longer forecast horizon ARIMA (1,1,1) continued to be the best, although the MAE rose to 5.67. TimeGPT was the only ML model to achieve MAE under 7, with a score of 6.76.

Table . RBA index MAE and RMSE, note table had to be split and stacked. Lowest errors are in bold.



*Note: moving average order 3 (MA3), moving average order 6 (MA6), moving average order 12 (MA12), simple exponential smoothing (SES), double exponential smoothing (DES), triple exponential smoothing additive (TESAdd), triple exponential smoothing multiplicative (TESMulti), autoregressive order 1 (AR(1)), autoregressive moving average (ARMA(1,1), autoregressive integrated moving average (ARIMIA(1,1,1)). AutoARIMA is the ARIMA model that the R package finds is best, similarly with AutoSARIMA but this includes seasonality.*

## One month forecast horizon results

To create a summary statistic to compare the forecasting models across all indices, we take the average one month forecast MAE across the indices.

For the ANZ indices (Table 4), ARIMA models were the top performers, with ARIMA (1,1,1) achieving the best MAE of 4.95. This model, which includes one lag, first-order differencing, and one moving average component, performed similarly to ARMA (1,1) and AutoARIMA after rounding. Among ML models, N-BEATS was the most accurate (MAE of 5.01), followed by N-HiTS (MAE of 5.08), which narrowly outperformed the naïve 2 benchmark. As expected, the simple average benchmark performed poorly due to the trending nature of the data. Prophet also struggled, managing to only outperform the simple average, while NeuralProphet showed significant improvement over Prophet, demonstrating the benefit of its updated framework.

For the RBA indices (Table 3), the results were consistent, with ARIMA (1,1,1) again delivering the best performance (MAE of 1.96). The Double Exponential Smoothing (DES) model also performed well, surpassing the naïve 2 benchmark with an MAE of 2.15 [[8]](#footnote-8). Among ML models, N-BEATS successfully beat the benchmark, but N-HiTS failed to replicate its performance from the ANZ indices, falling below both naïve 1 and 2 [[9]](#footnote-9).



Table . Average error of all RBA indexes (h = 1).



Table . Average error of all ANZ indexes (h = 1).

*Note: moving average order 3 (MA3), moving average order 6 (MA6), moving average order 12 (MA12), simple exponential smoothing (SES), double exponential smoothing (DES), triple exponential smoothing additive (TESAdd), triple exponential smoothing multiplicative (TESMulti), autoregressive order 1 (AR(1)), autoregressive moving average (ARMA(1,1), autoregressive integrated moving average (ARIMIA(1,1,1)). AutoARIMA is the ARIMA model that the R package finds is best, similarly with AutoSARIMA but this includes seasonality.*

The box and whisker plot summaries the average MAE of all the ANZ and RBA indices (Figure 1 and 2). To improve the plot’s readability, the simple average model was excluded as it performed significantly worse than the next poorest model, Prophet.

For ANZ indices, Prophet's mean MAE was notably higher, sitting outside the upper quartile of all other models (Figure 1). Alongside Prophet, MA12, MA6, and MA3 exhibited greater variability in MAE, suggesting inconsistent performance. In contrast, the remaining models had limited variability with similar spreads, indicating more stable performance across indices. Notably, no outliers were present in any of the plots.

On the RBA indices Prophets poor performance is more apparent, where its lower quartile is greater than all other models’ upper quartile. Its spread, alongside MA12 and MA6, is noticeably larger than other models, highlighting greater variability in forecast accuracy. In contrast, ARIMA (1,1,1) and AutoARIMA exhibit minimal spread, as shown by their very small whiskers, indicating consistently low forecast error. Other ML models, including N-BEATS, N-HiTS, and NeuralProphet, show slightly higher variation, while Naïve 1 and 2 deliver impressive consistency, despite their simplicity. Interestingly, higher-order moving average models perform worse, suggesting that recent data points are more relevant for accurate forecasting.

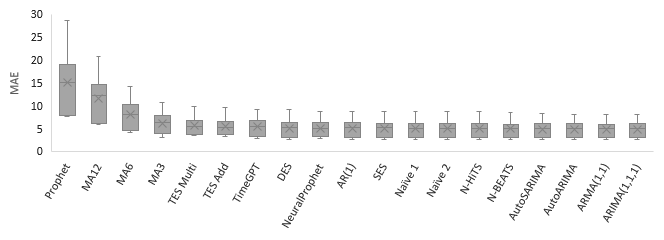


Figure . Box and whisker plot of ANZ average index errors (h = 1)

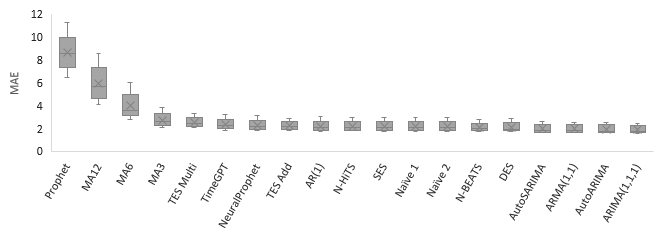


Figure . Box and whisker plot of RBA average index errors (h = 1).

## Four month forecast horizon results

The longer forecast horizon yielded results consistent with the 1-month horizon. Across ANZ indices (Figure 5), Prophet continued to perform poorly, surpassing only the simple average, with an average MAE of 16.49. At the upper end of performance, AutoARIMA was the only model to outperform the Naïve 1 benchmark, achieving an MAE of 8.84 compared to the benchmark’s 8.86. Among the ML models, TimeGPT performed best with an MAE of 9.37, followed closely by N-HiTS at 9.51.

Across the RBA indices, more models were able to outperform the Naïve 1 benchmark of 4.22, with all the models successful at beating the Naïve 1 model being traditional ARIMA variants. Among them, ARIMA (1,1,1) was the best, achieving an MAE of 4.00. TimeGPT emerged as the strongest ML model with an MAE of 4.57, though it still fell short of outperforming the Naïve 2 benchmark of 4.22. Interestingly, N-BEATS outperformed N-HiTS, which was unexpected given N-HiTS is designed for better performance over longer horizons. This suggests the forecast horizon may not have been long enough to fully leverage N-HiTS' strengths.



Table . Average error of all ANZ indices (h = 4).



Table . Average error of all RBA indices (h = 4).

*Note: moving average order 3 (MA3), moving average order 6 (MA6), moving average order 12 (MA12), simple exponential smoothing (SES), double exponential smoothing (DES), triple exponential smoothing additive (TESAdd), triple exponential smoothing multiplicative (TESMulti), autoregressive order 1 (AR(1)), autoregressive moving average (ARMA(1,1), autoregressive integrated moving average (ARIMIA(1,1,1)). AutoARIMA is the ARIMA model that the R package finds is best, similarly with AutoSARIMA but this includes seasonality.*

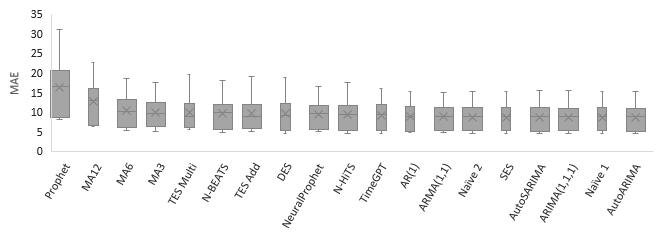


Figure . Box and whisker plot of ANZ average index errors (h = 4).

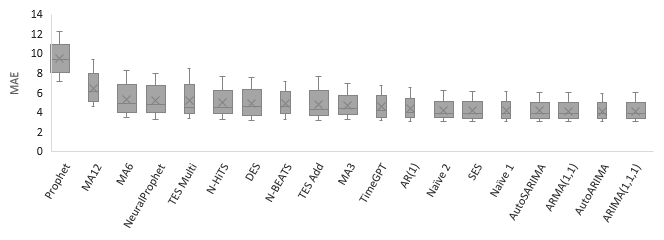


Figure . Box and whisker plot of RBA average index errors (h = 4).

For the reasons discussed earlier, the simple average, which performed the worst with an MAE of 43.36 (Table 5) on the ANZ indices and 27.37 (Table 6) on the RBA indices was excluded from the box and whisker plots.

Across the ANZ indices (Figure 3), all models exhibit longer upper whiskers, suggesting occasional high errors. Prophet shows the largest variation, with its mean lying outside the upper quartiles of all other models, reaffirming its poor performance. By contrast, the remaining models exhibit similar levels of variation, with no significant outliers present.

For the RBA indices (Figure 4), the whiskers are more symmetrical across all models. Prophet's lower quartile exceeds the upper quartile of all other models, indicating consistently poor performance relative to its peers. Furthermore, the variation across models is less consistent, with ARIMA models demonstrating a noticeably smaller spread. Combined with their lower average errors, this highlights the robustness of ARIMA models in this context.

# Conclusion

This study compared the performance of traditional forecasting models and machine learning (ML) models on ANZ and RBA commodity price indices over one-month and four-month forecast horizons.

The findings indicate that ARIMA variants consistently outperformed naïve benchmarks and delivered the best overall accuracy. In contrast, ML models struggled to consistently outperform the benchmarks, revealing no clear advantage. However, Prophet stood out as the worst-performing ML model, surpassing only the simple average, while NeuralProphet’s improved accuracy over Prophet was confirmed. A key observation was the substantial computational time required by ML models, which did not translate into better forecasting accuracy.

These results suggest that analysts forecasting ANZ and RBA commodity prices should prioritize ARIMA models for their robustness and accuracy. For future research, in-depth optimization of ML models is recommended, particularly using Bayesian methods to automate hyperparameter tuning across models and indexes (AWS, 2025). These automated methods test various parameter combinations to identify the optimal settings which would give them a fairer representation in the study. Exploring longer forecast horizons could also yield valuable insights, as accurate long-term predictions are highly beneficial. Additionally, incorporating exogenous variables, such as macroeconomic indicators or weather data, may enhance predictive performance.

# References

ANZ Research. (2025, 1). *ANZ Commodity Price Index*. Retrieved from ANZ: https://www.anz.co.nz/about-us/economic-markets-research/commodity-price-index/

ANZ Research. (2025). *ANZ Commodity Price Index.* ANZ.

AWS. (2025, 02 01). *What is Hyperparameter Tuning?* Retrieved from AWS: https://aws.amazon.com/what-is/hyperparameter-tuning/

Baysan. (2022, 03 01). *Introduction to Time Series Forecasting: Smoothing Methods*. Retrieved from Medium: https://medium.com/codex/introduction-to-time-series-forecasting-smoothing-methods-9a904c00d0fd

Challu, C., Olivares, K. G., Oreshkin, B. N., Garza, F., Mergenthaler-Canseco, M., & Dubrawski, A. (2022). N-HiTS: Neural Hierarchical Interpolation for Time Series Forecasting.

Filho, M. (2023, 06 2). *Forecastegy*. Retrieved from Multiple Time Series Forecasting With N-BEATS In Python: https://forecastegy.com/posts/multiple-time-series-forecasting-nbeats-python/#how-to-tune-the-n-beats-hyperparameters

Garza, A., Challu, C., & Mergenthaler-Canseco, M. (2024). TimeGPT-1.

Godahewa, R., Bergmeir, C., Webb, G. I., Hyndman, R. J., & Montero-Manso, P. (2021). *Monash Time Series Forecasting Archive.*

Godahewa, R., Bergmeir, C., Webb, G. I., Hyndman, R. J., & Montero-Manso, P. (2025, 01 29). *Monash Time Series Forecasting Repository*. Retrieved from Monash Time Series Forecasting Repository: https://forecastingdata.org/

Hyndman, R. J., & Athanasopoulos, G. (2024, 10 19). *ARIMA modelling in R*. Retrieved from Forecasting: Principles and Practice: https://otexts.com/fpp2/arima-r.html

Hyndman, R., & Athanasopoulos, G. (2018). *Forecasting: principles and practice, 2nd edition.* Melbourne, Australia: OTexts.

Hyndman, R., & Athanasopoulos, G. (2021). *Forecasting: principles and practice, 3rd edition.* Melbourne, Australia: OTexts.

Jung, J. K., Patnam, M., & Ter-Martirosyan, A. (2018). An algorithmic crystal ball: Forecasts-based on machine learning. *International Monetary Fund*.

Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., . . . Winkler, R. (1982). The accuracy of extrapolation (time series) methods: results of a forecasting competition. *Journal of Forecasting, 1*, 111-153.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 54-74.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting, Volume 38, Issue 4*, 1346-1364.

Makridakis, S., Spiliotis, E., Hollyman, R., Petropoulos, F., Swanson, N., & Gaba, A. (2024). The M6 forecasting competition: Bridging the gap between forecasting and investment decisions. *International Journal of Forecasting*.

Meta. (2025, 20 1). *Forecasting at scale.* Retrieved from Prophet: https://facebook.github.io/prophet/

NeuralProphet. (2025, 01 25). *Selecting the Hyperparameters*. Retrieved from Neural Prophet: https://neuralprophet.com/how-to-guides/feature-guides/hyperparameter-selection.html

Nixtla. (2025, 01 25). *Improve Forecast Accuracy with TimeGPT*. Retrieved from Nixtla: https://docs.nixtla.io/docs/tutorials-improve\_forecast\_accuracy\_with\_timegpt

Nixtla. (2025, 01 15). *NBEATS*. Retrieved from Nixtla: https://nixtlaverse.nixtla.io/neuralforecast/models.nbeats.html#usage-example

Nixtla. (2025, 01 15). *Nixtla*. Retrieved from NHITS: https://nixtlaverse.nixtla.io/neuralforecast/models.nhits.html

Optuna. (2025, 02 03). *Optimize Your Optimization*. Retrieved from Optuna: https://optuna.org/

Oreshkin, B. N., Carpov, V., Chapados, N., & Bengio, Y. (2020). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting.

Prophet. (2025, 01 25). *Diagnostics*. Retrieved from Prophet: https://facebook.github.io/prophet/docs/diagnostics.html#hyperparameter-tuning

Prophet. (2025, 01 15). *Documentation*. Retrieved from Prophet: https://facebook.github.io/prophet/docs/quick\_start.html#r-api

RBA. (2024, 04 02). *Weights for the Index of Commodity Prices*. Retrieved from Reserve Bank of Australia: https://www.rba.gov.au/statistics/frequency/commodity-prices/2024/weights-icp-20240402.html

RBA. (2025, 1). *Index of Commodity Prices*. Retrieved from Reserve Bank of Australia: https://www.rba.gov.au/statistics/frequency/commodity-prices/2024/

Robinson, T., & Wang, H. (2013). *Changes to the RBA Index of Commodity Prices: 2013.* RBA.

Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent. *International Journal of Forecasting, 36*, 75-85.

Taylor, S. J., & Lethan, B. (2017). *Forecasting at Scale.* PeerJ Preprints.

Trading Economics. (2025, 01 28). *Australia Exports By Category*. Retrieved from Trading Economics: https://tradingeconomics.com/australia/exports-by-category,exports-by-category

Trading Economics. (2025, 01 28). *New Zealand Exports By Category*. Retrieved from Trading Economics: https://tradingeconomics.com/new-zealand/exports-by-category

Triebe, O. (2025, 01 15). *Quick Start Guide*. Retrieved from NeuralProphet: https://neuralprophet.com/quickstart.html

Triebe, O., Hewamalage, H., Pilyugina, P., Laptev, N., Bergmeir, C., & Rajagopal, R. (2021). *NeuralProphet: Explainable Forecasting at Scale.*

World Bank. (2025, 01 28). *Exports of goods and services (% of GDP)*. Retrieved from World Bank: https://data.worldbank.org/indicator/NE.EXP.GNFS.ZS?name\_desc=false

# Appendix

The dataset, R and Python code are extensive so have not been included. These can be provided to you upon request.

1. Rural component includes (RBA, 2024): Wool, beef and veal, wheat, barley, canola, sugar, cotton, milk powder and lamb and mutton.

   Base metals include: Aluminium, lead, copper and zinc.

   Bulk commodities include: Iron ore, metallurgical coal and thermal coal.

   Other resources include: LNG, crude oil, alumina, gold, copper ore and lithium. [↑](#footnote-ref-1)
2. The observation added to the training dataset was dropped from the test dataset. [↑](#footnote-ref-2)
3. Prophet was implemented in R, where the authors documentation was followed (Prophet, 2025) [↑](#footnote-ref-3)
4. N-Beats was implemented in Python using Nixtla documentation (Nixtla, 2025). [↑](#footnote-ref-4)
5. NeuralProphet was used in Python following documentation published by the author (Triebe, 2025). [↑](#footnote-ref-5)
6. N-HiTS was applied in Python following Nixtla documentation (Nixtla, 2025). [↑](#footnote-ref-6)
7. TimeGPT was accessed through the nixtlar package on R using an API key. [↑](#footnote-ref-7)
8. Naïve 2 model’s forecast is equal to the last observed value incremented by a drift term. [↑](#footnote-ref-8)
9. Naïve 1 model uses the last observed value as the forecast. [↑](#footnote-ref-9)