FORESIGHT

The International Journal of Applied Forecasting

ARTICLES *by* FORECASTERS *for* FORECASTERS: Q3:2023

Special Feature: Pitfalls in Forecast Evaluation

Join the *Foresight* readership by becoming a member of the International Institute of Forecasters
forecasters.org/foresight/

Common Pitfalls and Better Practices in Forecast Evaluation for Data Scientists

CHRISTOPH BERGMEIR

PREVIEW *Nowadays, forecasting is often performed by data scientists with no specialized forecasting training. Such forecasters may be unaware of many pitfalls in forecast evaluation, leading to the improper evaluation we find in numerous papers published in the machine learning literature. Here, Christoph Bergmeir explores forecast evaluation pitfalls and offers better practices to avoid them.*

INTRODUCTION

In recent years there has been a shift in the field of forecasting, with machine learning (ML) finally delivering on its decade-old promise of outperforming statistical methods. Early indications of the shift occurred in competitions on Kaggle and the M4, then undeniably in the M5 competition (Makridakis, Spiliotis, and Assimakopoulos, 2022). Apart from these changes in forecasting research, we see another shift in industry where companies may assign practitioners with quite generic data science skills to undertake forecasting tasks in addition to their regular data analysis tasks. This group of practitioners is the primary audience for this article, but I hope that more experienced forecasters will also find value in it.

For new practitioners, the challenge of evaluating forecasts can be surprising – even when dealing with just point forecasts, which are the focus of this article. The challenge is greater when evaluating probabilistic forecasting (but we won't address that here).

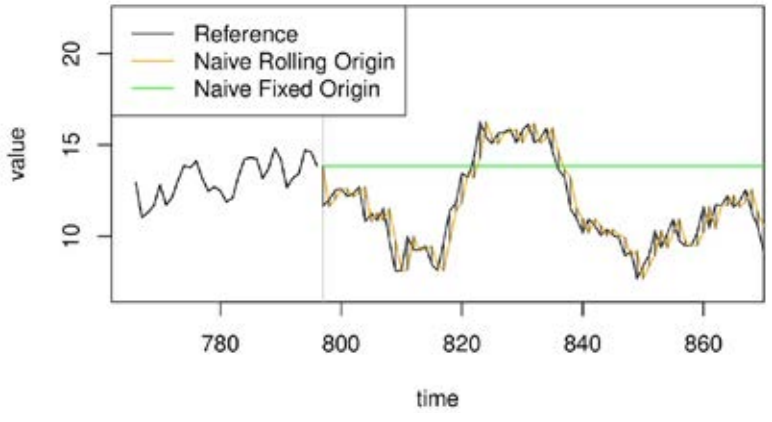
In the ML task of regression, evaluation is relatively straightforward and typically limited to the calculation of the Root Mean Squared Error (RMSE) or Mean Absolute Error (MAE) measures. K-fold cross-validation is the standard data partitioning method used for validation and, subsequently, model and feature selection, as well as hyper-parameter tuning. In contrast, in forecasting, more than 40 different accuracy measures (e.g., percentage errors, scaled errors, and relative errors) have been proposed in the literature. The selection of the most appropriate measure would effectively depend on the distribution of the data, the targets in the form of different horizons or multiple series of potentially different scales, and the objective of the forecast itself. Accordingly, various cross-validation schemes for data partitioning are available in forecasting settings.

Many ML researchers and practitioners are not aware of these intricacies, thus frequently make poor evaluation choices in forecasting papers and presumably in forecasting practice. Along with two colleagues, I wrote a comprehensive paper on the topic (Hewamalage, Ackermann, and Bergmeir, 2023), where we have identified six common pitfalls we've found in

Key Points

- Forecasting practitioners, in both industry and academia, are often data scientists who have a general statistics or machine learning background, but little specialized training in forecasting. This article is intended for this type of forecaster.
- Data scientists who lack forecasting training may be unaware of the intricacies of forecast evaluation, which leads to problems in their evaluation setup. There are many examples of this in the machine learning literature.
- The six common pitfalls of forecast evaluation in this context are
  - reliance on forecast plots
  - assumption that a forecast needs to be a realistic scenario
  - datasets too small/irrelevant
  - data leakage
  - not using adequate benchmarks
  - wrongly used or ad hoc evaluation measures.
- Better practices in forecast evaluation to avoid these pitfalls include
  - being aware of the complexities of forecast evaluation
  - relying on error measures instead of plots (especially when doing rolling origin)
  - using as large a dataset for evaluation as you can along with cross-validation schemes
  - paying attention to possible data leakage
  - implementing simple and appropriate benchmarks
  - using one of the many established error measures that is right for your business context.

Figure 1. Example of a fixed- and rolling-origin Naïve forecast



published papers that forecasting practitioners need to look out for, and that I come across regularly when serving as a reviewer for many ML outlets. I briefly present these pitfalls here and suggest some practices to avoid them.

Pitfall 1: Reliance on forecast plots

An important distinction in forecast evaluation is between fixed- and rolling-origin forecasting. In fixed-origin forecasting, the last known observation from where we forecast (the origin) is fixed, so that all observations in the test set are effectively forecasts with different horizons (1-step-ahead, 2-step-ahead, etc.). In rolling origin, the origin is moved forward through the test set, using data from the test set successively as input for the forecasting method (for inference with or without retraining). In the latter setup, 1-step-ahead forecasts are usually considered.

These two situations are very different, as illustrated by **Figure 1**, which shows a time series with a Naïve forecast (a no-change forecast that uses the last known observation as the forecast), in fixed- and rolling-origin setups. We see that the rolling-origin Naïve forecast follows the time series closely, whereas the fixed-origin Naïve forecast is constant throughout the test set. Note however that errors are calculated vertically (visualized by the vertical lines in the plot), and while the actuals and the rolling-origin forecasts appear close, they are close mostly horizontally, not vertically.

We would expect a forecasting method that adds any value to at least outperform Naïve, but determining from a plot like **Figure 2** whether an ML method outperforms rolling-origin Naïve is not straightforward. We would need to closely monitor the vertical differences of both the Random Forest and the Naïve method and compute their sum mentally to make any conclusion, but ultimately we would have to resort to some accuracy measure to quantitatively assess the quality of the ML forecasts.

We can see that rolling-origin Naïve follows the original series closely, and visually

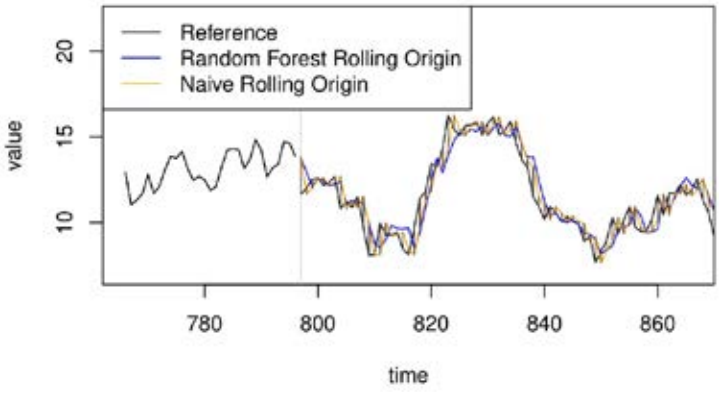
determining whether the Random Forest forecast is better than Naïve is difficult. Overall, plots that show forecasts versus actuals should be mostly used for sanity checking, e.g., to verify that a seasonality or trend is modeled as intended. To assess whether one method is better than another, these plots need to be used with caution, especially (1) in 1-step-ahead rolling-origin setups, (2) when methods have systematic differences (e.g., in the presence of special event dates), and (3) in situations where multiple series are being forecast.

Pitfall 2: Assumption that a forecast needs to be a realistic scenario

As the future holds inherent uncertainty, probabilistic forecasts in the form of a distribution can become particularly useful. However, if we focus on point forecasts, an important question then becomes which summary statistic (e.g., the median or the mean) of said forecast distribution our model should estimate. The answer needs to be based on the final objective of the forecasting exercise and with the business context in mind – the business metrics and accuracy measures need to be aligned. The aim of the process will be to build a model that maximizes accuracy with respect to the chosen accuracy measure. Thus, the statistical nature of a forecast may be fundamentally different from the actuals, and we cannot expect both to have similar characteristics.

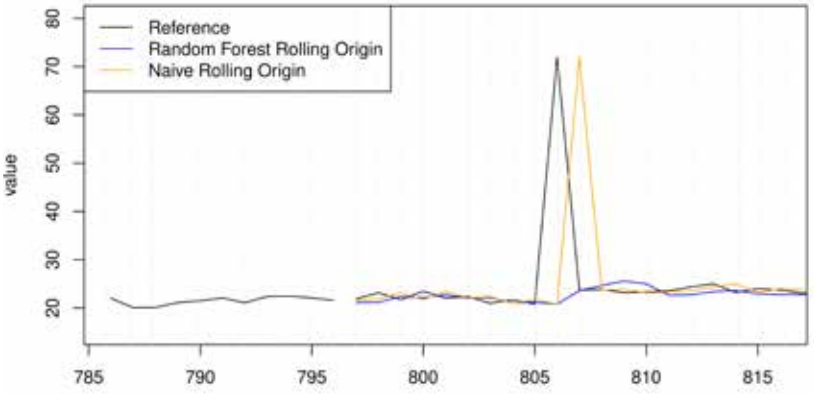
As an example, let us reconsider Figure 1. The series shown is a random walk, generated as the cumulative sum of i.i.d. normally distributed random numbers, with mean zero and variance one. The fixed-origin Naïve forecast shown in the figure is mathematically the best forecast we can achieve for this series (under a measure like RMSE), as the observations are i.i.d., that is to say not predictable beyond their mean, which is zero. However, from the plot we see that in a fixed-origin setup, the Naïve forecast is a constant. It is an important part of the work of a forecaster to explain to stakeholders less familiar with forecasting that, while it is clear that the series will not just magically turn into a constant value in the future, and as such this forecast is not a realistic scenario, it is still the best forecast in the sense that it adequately captures the forecast distribution and minimizes the forecast error. (See Stephan Kolassa's OpEd on flatline forecasts in this issue.)

Figure 2. Rolling-origin Naïve forecast versus Random Forest forecast



Another example is shown in **Figure 3**, where the actuals have a prominent spike in the forecasting period. We see that the Naïve forecast will follow the spike and therefore predict one, but at the wrong position. This leads to the spike contributing twice to the error, once as a false negative and once as a false positive. In contrast, a method that doesn't predict a spike at all but focuses on the mean of the series will lead to a smaller error (e.g., in terms of RMSE), but may be deemed unrealistic due to not predicting any spikes.

Figure 3. Forecasting with an unpredictable spike. A method that predicts the spike but at the wrong position will lead to double the error of a method that doesn't predict any spike.



In ML settings, the tendency of the model to predict such spikes would effectively depend on the loss used for training the model. An L1 loss (least absolute deviations) would result in a model that does not predict spikes at all, while an L2 loss (least squared errors) would result in predictions that scale the magnitude of a spike with how certainly a spike will happen at a certain time. Either way, the magnitude of the predicted spikes (if any) will be different from that observed in the data. Therefore, the forecasts will be unrealistic, despite minimizing the forecast error.

Pitfall 3: Datasets too small / irrelevant

This pitfall is more an academic problem, and usually not relevant for practitioners in their daily business as they would work with datasets relevant to them. However, it is an important point to keep in mind when assessing the quality of newly published research or adopting new research in industry. To quote Paul Goodwin (2001), "If the name of a method contains more words than the number of observations that were used to test it, then it's wise to put any plans to adopt the method on hold." My suggestion would be to ask yourself the following questions when reading academic papers:

Can we assume that the authors could have had more data readily available, if they had wanted to? A classic example in this context is stock market prediction. There are many papers published that present new methods for stock market prediction, testing on a handful of time series. Share-price data are easily available for hundreds or thousands of stocks. So do those authors explain why they chose the series they chose, and not others?

Do the authors care about their application? Is their method tailored to their application, or do they just claim it is? If their application is to forecast a time series in container shipping, do they convincingly explain what makes their series so different to others that they need to implement, for example, a neuro-fuzzy system fitted with a meta-heuristic inspired by migration patterns of a rare ant species? Why not just use a Gradient-Boosted Tree instead? Is there an ablation study? If their algorithm is essentially modeling autocorrelation, trend, and seasonality on standard time series, why do they not evaluate it on a way broader dataset, such as the M4 dataset? Even if they win on their particular dataset against standard methods, it is likely a spurious result of little to no relevance to the broader forecasting community.

Pitfall 4: Data leakage

Data leakage (using information about the future that would not be known at the time the prediction was made) is always a risk in any ML task. However, in forecasting it is more difficult to avoid due to the self-supervised nature of the task, where data is used both as targets and inputs during training and testing. In particular, in a rolling-origin setup, data travels routinely from the test set to the training set. It becomes difficult to completely separate the code bases for training and testing without greater implementation complexity and computational cost.

In general ML, it is common knowledge that we should not calculate normalization parameters such as the mean and variance over the full dataset before partitioning into training and test sets. Rather, these values must be calculated over the training set alone. However, in forecasting, operations that smooth or (seasonally) decompose the data, such as STL (Seasonal-Trend decomposition using LOESS) or EMD (Empirical Mode Decomposition), should also not be employed before splitting the data. These operations process the complete series – not just the data up to the current observation which is what statistical forecasting models like ETS and TBATS (Trigonometric seasonality, Box-Cox transformation, ARMA errors, Trend, and Seasonal components) would do. Thus, those operations need to be performed separately over input windows for them to be applicable to forecasting.

Data leakage can also come in more subtle forms. For example, we could have an unaligned dataset like the M3 or M4 competition datasets, where some series'

training sets share the time stamps of other series' test sets. Let us now assume that there might be big external shocks (recent examples being a pandemic or a war) that influence all the series. If we now train global models that learn across multiple series, they may have an advantage and be able to predict the future of some series better than would be possible in practice, i.e., when we didn't have knowledge of such global events. Another example involves situations where an unintended forecast horizon is considered. For instance, we may use a daily average in one of our input features; however, we may need to provide forecasts by 5 p.m. every day. Thus we cannot use daily means that run from midnight to midnight, as in the real world we will have available only data until 5 p.m. from that day.

Although most of these issues will eventually become evident when deploying the forecasting solution into production, they will still have a negative impact on the development cost and lead to worse forecasts than expected based on previous tests. Furthermore, in academia, where there may not be a deployment into production, undetected situations of data leakage may result in wrong conclusions.

Pitfall 5: Not using adequate benchmarks

Forecasting has traditionally been a research field where simple methods can perform just as well as more complicated ones. While this notion has evolved with global ML models, it is still critical to use simple benchmarks to continuously monitor forecasting performance and measure the forecast value added.

This is particularly true in applications involving, for instance, the prediction of asset prices or wind power production, where due to the stochastic nature of the series, relatively simple methods could be proved quite competitive. A notable example is a series of some recent deep learning papers that all use for evaluation a daily exchange rate dataset, where the task is to forecast up to 720 days ahead (which already seems quite impossible). The authors typically compare against other deep learning architectures, but never against a simple Naïve forecast, which my colleagues and I found to be more accurate.

Nevertheless, benchmarks should not be limited to the Naïve forecast, but selected based on the forecasting task at hand. For example, sophisticated global models can be compared to simpler, linear global models (Bandara and colleagues, 2022). Similarly, models that account for multiple seasonalities can be compared against benchmarks that are capable of modeling only a single seasonality (like ETS or even seasonal Naive).

Pitfall 6: Wrongly used or ad hoc evaluation measures

Evaluation measures are a notoriously difficult field in forecasting, which has led to a considerable body of literature. The most straightforward measures to define are on the same scale as the time series, such as RMSE and MAE. These work well until we want to calculate an overall forecast error across series on different scales. In that case, we need a scale-free measure that uses a normalizing factor. For researchers and practitioners outside of the field of forecasting, it usually comes as a surprise that after decades of research we still have not found a normalizing factor that works universally for any given series.

Besides the most common measures such as MAPE and sMAPE with their well-documented asymmetries and shortcomings when the series contain zeros or small values, there are over 40 accuracy measures that all have problems under certain conditions. One problem that research in ML often does not appreciate is that time series can have vastly different characteristics, thus making implicit assumptions that are not explicitly stated. For example, if we work on wind or solar power production forecasting of a single wind or solar farm, it is reasonable to normalize forecast errors based on the maximum of the series. But this choice will be wrong for series that are characterized by trends or shifts, as is the case with Bitcoin prices, for instance. Unfortunately, each error measure has its own advantages and disadvantages, while inventing ad hoc

measures with largely unexplored statistical properties is not a good solution.

More generally, the problem of identifying an appropriate accuracy measure lies in the different types of non-stationarities and non-normalities that series may have. In this context, I want to quote from the statistical jokes page that Rob Hyndman hosts (***https://robjhyndman.com/hyndsight/statistical-jokes/***): "Classification of mathematical problems as linear and nonlinear is like classification of the Universe as bananas and non-bananas." As such, linearity is a well-defined concept, while non-linearity is just "everything else." Similar is the concept of stationarity vs non-stationarity. There are many different types of non-stationarities that we may encounter in our time series, such as stochastic or deterministic trends, seasonalities, structural breaks, level shifts, and others. Consequently, we will need to deal with them in different ways in our modeling and evaluation efforts. For example, as financial time series usually exhibit random walk behavior, their non-stationarity can be addressed with differencing. But non-stationary series with exponential trend will not be properly adjusted through differencing, thus requiring a different type of processing.

Some suggestions and better practices

To address some of the aforementioned pitfalls, I will now discuss some guidelines and better practices around data partitioning and error measures that my colleagues and I identified in our 2023 article.

Data Partitioning: Make full use of your dataset

While fixed-origin evaluations are the simplest to implement, they make very limited use of the data. In contrast, rolling-origin evaluations can lead to more stable results. This type of evaluation, also called prequential evaluation or time series cross-validation (usually not using all possible origins but skipping over some of them), is an analogy to k-fold cross-validation versus leave-one-out cross-validation in general ML tasks.

However, it is a common misconception that the temporal order of a time series always needs to be respected when partitioning data, in the sense that our training dataset always needs to be prior to the test dataset in time. While for a stationary time series by definition it won't make any difference if we train on data in the past to predict on data in the future or vice versa, also non-stationary prediction tasks can benefit from cross-validation schemes that do not respect the order in time, when used adequately. To address dependency in the data, these schemes usually choose training and test sets not randomly but in blocks. Consequently, these schemes, called blocked cross-validation in the past (see, e.g., Bergmeir, Hyndman, and Koo, 2018) and reinvented under the name of purged cross-validation, can offer powerful evaluation solutions to practitioners. This was the case with the winning method of the M5 uncertainty competition (Lainder and Wolfinger, 2022). Especially for small datasets, blocked cross-validation can have substantial advantages over schemes that always respect the temporal order, as they exploit all available data, both for training and testing. There are also situations, like when using purely autoregressive models with no covariates and no internal state, in which even randomized cross-validation is applicable, as long as models lead to uncorrelated residuals.

Error measures: Use the right one for your task

As outlined under Pitfall 2, forecasting needs to be driven by the business context, and evaluation measures need to be selected accordingly. For example, series could be in monetary value (such as dollars or euros), or physical dimensions (such as liters or square feet), and so on. When series are using a common unit of measure, large values would be more important than series with small values, and a scaled measure like RMSE or MAE can be used accordingly. If on the other hand the series should contribute with a different weighting (e.g., all series get the same weight) to the error measure, we need to normalize, which brings various difficulties.

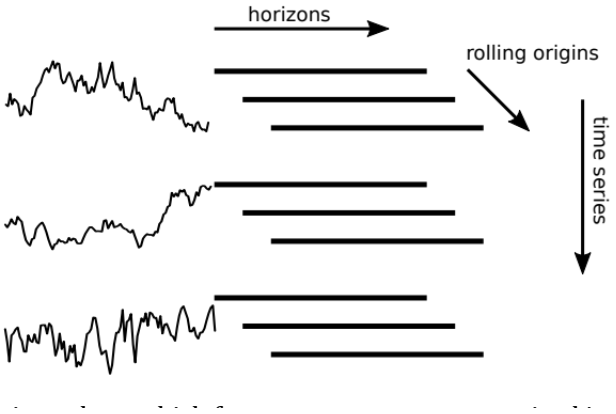
In general, error measures would summarize along three dimensions, namely over different horizons, over different forecast origins (in a rolling-origin setup), and over different time series, as illustrated in **Figure 4**. One problem that many error measures address is that each of these dimensions can be small (e.g., only one horizon, only one origin, only one time series), and then we can be in the situation of dividing by zero, if a single value in the series or a single forecast is zero. Also, the scale of the data can change considerably in all three dimensions. This is not surprising for the time series dimension, with series on different scales. But, as discussed in Pitfall 6, the scale can also change drastically across horizons and forecast origins.

Depending on the forecasting setup, the size of these three dimensions, and scale changes along the dimensions, normalizing factors can be calculated. For example, if our horizon is long and no strong trends are present in the series, summing over the actuals in the test set can give us a good factor by which we normalize per series, to then be able to deal with series on different scales. In contrast, if we only use a horizon of one and a single forecast origin, this is not an option.

To address this problem of a limited size of the test set, scaled error measures, such as the MASE, have been proposed in the literature, which normalize by dividing by the performance of the Naïve forecast over the training set. While this solves many problems that other error measures have, scaled errors may still work poorly when the characteristics of the training and test set differ vastly, being also difficult to interpret.

Another discussion currently happening in the forecasting space is whether it is reasonable to use error measures that are different from the loss used during training. Kolassa (2020) has argued that one and only one error measure should be used for evaluation, so that the loss function can be tailored to produce the best forecasts for this error measure. Others have argued that oftentimes many

Figure 4. Dimensions along which forecast errors are summarized into forecast error measures.



metrics show comparable results so that the choice of only one measure is less critical (Koutsandreas and colleagues, 2021). Our paper (Hewamalage, Ackermann, and Bergmeir, 2023) discussed over 40 different error measures and presented a diagram showing in which situations what measures should be used and which ones should be avoided. Our standpoint is that while the loss will determine under which measure a forecast performs well, other measures can be used additionally for sanity checking and stability/generalizability considerations. Equally important, measuring forecast bias in addition to accuracy is important and can often give valuable insights.

CONCLUSIONS

The field of forecasting has changed considerably over recent years – a development called the "forecasting spring" by Makridakis and colleagues (2022) – with global cross-series modeling, series with higher frequencies, and richer covariates and metadata available. I have argued in this paper that the profile of forecasting practitioners has changed as well, with forecasting increasingly performed by data scientists with backgrounds in statistics and ML but little particular forecasting experience. For this target audience, these are my main suggestions:

1. Be aware of the challenges present in forecast evaluation – it is not straightforward.

2. Rely on error measures and use forecasting plots just for a sanity check as they may be misleading, especially with rolling-origin evaluation.
3. Use as large a dataset as you can get – don't rely on anecdotal evidence. If lack of data, use a cross-validation scheme that doesn't respect the temporal order.
4. Be wary of data leakage.
5. Invest time into implementing simple yet suitable benchmarks.
6. None of the accuracy measures work for any and all time series. Choose one that works well with the characteristics of your series and that can capture what is relevant for your business context. MAPE is not the best solution, but inventing your own is even less so.

Christoph Bergmeir is a María Zambrano Senior Fellow in the Department of Computer Science and Artificial Intelligence at University of Granada, Spain. Before this he was a Visiting Research Data Scientist at Meta Inc. in the U.S., and a Senior Lecturer at Monash University, Australia. Christoph holds a PhD in computer science from the University of Granada, and an MSc degree in computer science from the University of Ulm, Germany. He has worked in forecasting for capacity planning, sustainable energy, and supply chain, and has over 5,000 citations, an h-index of 29, and has received more than $2.7 million in external research funding. Four of his publications on time series forecasting have been Clarivate Web of Science Highly Cited Papers (top 1% of their research field).

christoph.bergmeir@gmail.com

REFERENCES

Bandara, K., Hewamalage, H., Godahewa, R. & Gamakumara, P. (2022). A Fast and Scalable Ensemble of Global Models with Long Memory and Data Partitioning for the M5 Forecasting Competition, *International Journal of Forecasting,* 38(4), 1400–1404.

Bergmeir, C., Hyndman, R.J. & Koo, B. (2018). A Note on the Validity of Cross-Validation for Evaluating Autoregressive Time Series Prediction, *Computational Statistics and Data Analysis,* 120, 70–83. ***https://doi.org/10.1016/j.csda.2017.11.003***.

Goodwin, P. (2011). High on Complexity, Low on Evidence: Are Advanced Forecasting Methods Always as Good as They Seem? *Foresight*, Issue 23, 10-12.

Hewamalage, H., Ackermann, K. & Bergmeir, C. (2023). Forecast Evaluation for Data Scientists: Common Pitfalls and Best Practices, *Data Mining and Knowledge Discovery,* 37(2), 788–832.

Kolassa, S. (2023). All Hail the Flatline Forecast!, *Foresight*, Issue 70, 62-63.

Kolassa, S. (2020). Why the "best" point forecast depends on the error or accuracy measure, *International Journal of Forecasting,* 36 (1), 208-211.

Koutsandreas, D., Spiliotis, E., Petropoulos, F. & Assimakopoulos, V. (2021). On the Selection of Forecasting Accuracy Measures, *Journal of the Operational Research Society,* 73(5), 1–18.

Lainder, A.D. & Wolfinger, R. (2022). Forecasting with Gradient Boosted Trees: Augmentation, Tuning, and Cross-Validation Strategies: Winning Solution to the M5 Uncertainty Competition, *International Journal of Forecasting,* 38(4), 1426–33.

Makridakis, S., Spiliotis, E. & Assimakopoulos, V. (2022). M5 Accuracy Competition: Results, Findings, and Conclusions, *International Journal of Forecasting,* 38(4), 1346–64.

Zeng, A., Chen, M., Zhang, L. & Xu, Q. (2022). Are Transformers Effective for Time Series Forecasting? *arXiv Preprint arXiv:2205.13504*.