International Journal of Forecasting 36 (2020) 167–177

Contents lists available at ScienceDirect

International Journal of Forecasting

journal homepage: www.elsevier.com/locate/ijforecast

Discussion

Criteria for classifying forecasting methods

Tim Januschowski *, Jan Gasthaus, Yuyang Wang, David Salinas, Valentin Flunkert, Michael Bohlke-Schneider, Laurent Callot

Amazon Research, Germany

A B S T R A C T

Classifying forecasting methods as being either of a "machine learning" or "statistical" nature has become commonplace in parts of the forecasting literature and community, as exemplified by the M4 competition and the conclusion drawn by the organizers. We argue that this distinction does not stem from fundamental differences in the methods assigned to either class. Instead, this distinction is probably of a tribal nature, which limits the insights into the appropriateness and effectiveness of different forecasting methods. We provide alternative characteristics of forecasting methods which, in our view, allow to draw meaningful conclusions. Further, we discuss areas of forecasting which could benefit most from cross-pollination between the ML and the statistics communities.

1. Introduction

This article discusses the spectrum of "statistical" and "machine learning" (ML) methods, and the boundaries and intersections between them *in the context of forecasting*.[1] We argue that using the names "ML" and "statistical" to denote certain groups of techniques is unfortunate at best, as they imply a more profound, qualitative distinction than the one they are actually used to denote in practice.[2] In fact, implying a distinction using these general, connotation-laden terms can be misleading, as they incentivize sweeping statements such as "[...] the accuracy of ML models is below that of statistical ones [...]" (Makridakis et al., 2018b).

We argue that no meaningful grouping congruent with the original meaning of these terms can be established, and that distinctions along other dimensions are more useful for drawing valid conclusions in practice. We discuss how these dimensions allow to differentiate the plethora of forecasting methods and contrast how they are approached by the statistics and ML *communities*. In the process, we attempt to clear up and correct certain misconceptions, such as *ML methods do not handle uncertainty* (Makridakis et al., 2018b).

While we make general claims in this paper, our aim is to focus the discussion on forecasting, not to contrast statistics and ML in general. The relationship between the fields of statistics and ML is complicated, partially due to the clash of "two cultures", as put forth by Breiman (2001b). We are not oblivious to the observation that there are (partially) distinct ML and statistics *communities* in forecasting. It is our firm belief that these communities have much to learn from each other.[3] We argue that it is more constructive to seek common ground than it is to introduce artificial boundaries.

* Corresponding author.
E-mail addresses: tjnsch@amazon.com (T. Januschowski), gasthaus@amazon.com (J. Gasthaus), yuyawang@amazon.com (Y. Wang), dsalina@amazon.com (D. Salinas), flunkert@amazon.com (V. Flunkert), bohlkem@amazon.com (M. Bohlke-Schneider), lcallot@amazon.com (L. Callot).

[1] We use quotes around "statistical" and "ML" here to emphasize that, in our view, these are labels given to certain sets of techniques by some authors, not adjectives describing attributes of the methods.

[2] To be concrete, the extensive, operational definition of "statistical methods" as used e.g. in Makridakis, Spiliotis, and Assimakopoulos (2018a, 2018b, 2020) seems to be "variants of exponential smoothing and ARIMA methods", while "ML methods" is either used as an umbrella term to denote everything else or as a synonym for neural networks and random forests.

[3] And indeed all other communities that consider forecasting problems such as Econometrics, Data Science, and Systems, just to name a few.

https://doi.org/10.1016/j.ijforecast.2019.05.008
0169-2070/

2. Connotations of the terms "Statistics" and "ML"

Several authors have attempted to draw clear demarcation lines between statistics and ML.[4] These attempts can be thought-provoking, but fall short of establishing clear distinctions and are mostly of limited scientific value. Breiman (2001b) is an exception, as he proposes a distinction based on scientific culture rather than on a classification of methods. He distinguishes between a culture primarily concerned with in-sample fit (which he refers to as data modeling, explanation, or classical statistics) and one which is primarily concerned with out-of-sample fit (algorithmic modeling, prediction, or ML).

Breiman's article identifies a number of dimensions which tend to be used to classify methods as belonging to ML or statistics, such as: theoretical guarantees (statistics) vs. practical performance (ML) and mathematical elegance (statistics) vs. computational feasibility (ML). However, neither of these dimensions aligns with how these terms are used to distinguish forecasting methods.

Forecasting, as a field and discipline, draws from the qualities associated with both terms "statistics" and "ML", making use of methods derived from formal probabilistic theory with statistical guarantees (Durbin & Koopman, 2012; Hyndman, Koehler, Ord, & Snyder, 2008) as well as purely algorithmic approaches such as Croston's method (Croston, 1972; Hyndman & Shenstone, 2005). Forecasting is at its core an extrapolation problem,[5] where the performance of a model is evaluated using out-of-sample accuracy measures (Gneiting & Raftery, 2007; Hyndman & Koehler, 2006; Kolassa & Schuetz, 2007) or tests (Diebold & Mariano, 2002) rather than in-sample metrics. Finally, we remark that forecasting has long embraced what Donoho (2017) identifies as the secret sauce of ML: focus on predictive modelling and the Common Task Framework (CTF).[6] An instance of CTF consists of a publicly available benchmark data set, a number of competitors working on the same (predictive) task and an objective referee. Through the M competitions (Makridakis et al., 1982, 1993; Makridakis & Hibon, 2000; Makridakis et al., 2020), forecasting has had a long lasting tradition in CTF.

In Sections 3 and 4, we discuss several dimensions along which forecasting methods can be classified. Table 1 provides an overview. The dimensions we provide are by no means meant to be comprehensive, but should be regarded as a starting point for discussion. Notable omissions are frequentist vs. Bayesian approaches, multivariate vs. univariate models or the assumptions underlying the models (such as Gaussianity or IID properties).[7]

Table 1
Summary of all dimensions discussed.

Category	Dimension	Section
Objective	Global vs. Local Methods	3.1
	Probabilistic vs. Point Forecasts	3.2
	Computational Complexity	3.3
	Linearity & Convexity	3.4
Subjective	Data-driven vs. Model-driven	4.1
	Ensemble vs. Single Models	4.2
	Discriminative vs. Generative	4.3
	Statistical Guarantees	4.4
	Explanatory/Interpretable vs. Predictive	4.5

The dimensions in Section 3 are mathematical properties of the models, which is why we refer to them as *objective* dimensions. In Section 4 we consider a set of dimensions that are of a methodological, or indeed cultural nature in the sense of Breiman (2001b), and we refer to them as *subjective* dimensions. The dividing line between objective and subjective dimensions is blurry, and we mainly provide this distinction in the hope of adding some additional structure to the exposition. Section 5 synthesizes the discussion by identifying areas where we believe the interaction of both communities could be most fruitful.

3. *Objective* dimensions for classifying forecasting methods

This section considers a set of *objective* dimensions along which forecasting methods can be classified. We contrast how the statistics and ML communities tend to address them and highlight commonalities and complementarities.

3.1. Global and local methods

Consider a forecasting problem in which we aim at predicting a large number of similar time series, for instance, predicting demand for similar products. To tackle such problems, we can distinguish between two extremes: methods that estimate model parameters independently for each time series (here referred to as *local* methods) and methods that estimate model parameters jointly from all available time series (here referred to as *global* methods) (Januschowski, Gasthaus, Wang, Rangapuram, & Callot, 2018).[8]

There are hybrids between these extremes (Ben Taieb, Taylor, & Hyndman, 2017; Geweke, 1977; Maddix, Wang,

[4] Or try to establish clear distinctions between the these two and one or more of the following: Artificial Intelligence, Data Mining, Econometrics, Data Science, Pattern Matching, Predictive Analytics. See e.g. https://towardsdatascience.com/no-machine-learning-is-not-just-glorified-statistics-26d3952234e3http://andrewgelman.com/2008/12/machine_learnin/ http://brenocon.com/blog/2008/12/statistics-vs-machine-learning-fight/ https://www.svds.com/machine-learning-vs-statistics/ .

[5] There are of course in-sample fit concerns such as (Kolassa, 2011), but these are not the main focus of the discipline.

[6] https://www.simonsfoundation.org/lecture/reproducible-research-and-the-common-task-method/.

[7] We focus on dimensions which capture parts of the tension between "statistical" and "ML" forecasting methods. Other classifications (e.g., Armstrong, 1985; Chambers, Mullick, & Smith, 1971) have provided further useful dimensions in more general contexts.

[8] Note that historically, local methods have also referred to methods such as local linear methods e.g., (Bottou & Vapnik, 1992; Weigend & Gershenfeld, 1993) and Bontempi, Ben Taieb, and Le Borgne (2013) and references therein. This is an unfortunate overlap in terminology, but the difference should be clear from the context.

& Smola, 2018; Stock & Watson, 2002; Wang et al., 2019), where a part of the parameters are estimated globally and a part locally.

Note that this distinction is about how the parameters of the model are estimated, and does not imply a particular (in-)dependence structure between the time series. In particular, one can estimate a global model and still assume independence between forecasts for different time series (e.g. for reasons of computational efficiency). As such, this distinction is complementary to the distinction between univariate and multivariate forecasting methods, where multivariate typically implies an explicitly modeled dependency structure between the time series (e.g., explicitly modelling the co-variance structure).

While many "statistical" methods are local methods, global methods have long been used in both the statistics and ML communities. One key difference (and source of some confusion) is that models such as NNs can be used both in global and local settings without modifications. Recently, NNs have been used primarily as global models (Laptev, Yosinsk, Li Erran, & Smyl, 2017; Salinas, Flunkert, Gasthaus, & Januschowski, 2019; Wen, Torkkola, & Narayanaswamy, 2017), thereby surpassing earlier, mixed results where NNs where mainly used as local models (Zhang, Patuwo, & Hu, 1998). In local settings, only models with a small number of free parameters can typically be fitted reliably as training data is limited. On the other hand, global NN models with millions of parameters can be successfully trained as long as a large-enough training set of related time series is available.

3.2. Uncertainty quantification and distributional forecasts

Forecasting methods can be classified into probabilistic and point forecasting methods. While point forecasts just provide a single, best prediction (relative to some error metric) (Gneiting, 2011a, 2011b), probabilistic forecasting methods quantify the *predictive uncertainty*, allowing this uncertainty to be taken account when making decisions based on the forecast. Uncertainty in forecasts (predictive uncertainty) has multiple sources, including: (a) uncertainty about the model's parameter values (e.g. the slope of a trend); (b) uncertainty about the model's structure (e.g. whether to include in linear or quadratic trend); (c) residual variation, i.e. randomness not explained by the model (either due to inherent randomness of the underlying phenomenon, or due to simplifying assumptions made by the model); (d) uncertain model input data (e.g., imprecisely measured input data used for training or the need to forecast causal drivers for prediction); and, (e) uncertainty about the stability of the model or model drift, where the causal drivers of the forecast or their relation to the target may change over time. Residual variation is often treated explicitly in forecasting models in the form of explicit noise terms. Parameter and model structure uncertainty are sometimes also taken into account, by either Bayesian (e.g. averaging model predictions with respect to the posterior distribution over parameters and/or models), or frequentist approaches (e.g. model ensembles and bootstrap sampling).

For a time series $z_1, z_2, \ldots, z_T$, the predictive uncertainty is fully characterized by the predictive distribution

$$P(z_{T+1}, z_{T+2}, \ldots, z_{T+h} | z_1, z_2, \ldots, z_T), \quad (1)$$

but probabilistic forecasting methods differ in how they allow the user to query this (typically intractable) joint distribution. Typical ways include pointwise predictive intervals (i.e. access to quantiles of the marginal predictive distribution for each forecast horizon), as well as Monte Carlo sample paths from the joint predictive distribution. Another, commonly-used option is to assume a parametric form of the distribution (e.g., (1) is a negative binomial distribution) and return the parameters.

Marking probabilistic forecasting methods as "statistical" and point forecasting methods as "ML" is implied in Makridakis et al. (2018b).[9] However, there is ample evidence that probabilistic modeling and handling various forms of uncertainty has been at the heart of much of ML for at least the last decade, exemplified by most introductory text books taking a probabilistic perspective (Barber, 2012; Bishop, 2007; Koller & Friedman, 2009; Murphy, 2012). Modern ML methods handle uncertainty, e.g. by postulating generative models whose parameters are estimated through maximum likelihood estimation (Salinas et al., 2019), or by estimating the quantile function directly (Gasthaus et al., 2019; Wen et al., 2017) similarly to the approach of Koenker (2005) in econometrics. Estimating model and parameter uncertainty using Bayesian approaches is a well developed area in the ML literature as well; Barber (Barber, 2012) provides an introductory overview and for recent contributions see Blundell, Cornebise, Kavukcuoglu, and Wierstra (2015), Gal (2016) and Kingma and Welling (2013).

Furthermore, we note that the results of the M4 competition have shown that prediction intervals obtained by methods from the ML community have proven to be highly accurate (Makridakis et al., 2020) even though they may lack a theoretical underpinning, thereby directly contradicting (Makridakis et al., 2018b).

3.3. Computational complexity and costs

As forecasting is becoming more wide-spread and is applied to increasingly large data sets, it is natural to consider a notion of computational cost in addition to predictive accuracy when comparing forecasting methods, and to consider the trade-off. Depending on the application, the size of the downstream impact through improvements in forecast accuracy by a computationally more demanding method may be outweighed by the increased computational cost. Different applications may place constraints on the maximum time that can be spent on generating a forecast for a single time series (e.g. in (near) real-time settings) or impose a limit on the time available to produce forecasts for the entire body of time series. If there's an application-specific absolute time

[9] The authors write that "at present, the issue of uncertainty has not been included in the research agenda of the ML field, leaving a huge vacuum that must be filled as estimating the uncertainty in future predictions is as important as the forecasts themselves".

limit, parallelizability becomes an important factor. So, for a data set consisting of many items that need forecasting, it may be both interesting to measure the maximum time to forecast a single item (this would be the bottleneck in a highly-parallelizable scenario) as well as measuring the cumulative time to produce all forecasts.

For forecasting tasks where forecasts need be produced periodically (e.g. retail demand forecasts produced daily or weekly as input to an automatic ordering system), the computational cost of a forecasting method can be broken down into three components: (1) the computational cost of the experimental phase of finding an adequate model and model hyperparameters, (2) the computational cost of *training* the forecasting model, and (3) the computational cost of producing forecasts from a trained model (this is commonly referred to as *inference* in the deep learning literature). Note that in popular implementations of local models, (2) and (3) are conflated as the parameters are re-estimated for each forecast, while for global models, (2) and (3) are separate steps. This allows the cost of training (2) to be amortized over multiple forecasts (3), e.g. by re-training a model only once per month but using it daily to update the forecasts. There are some further subtleties to consider here, as both local and global methods may benefit from warm-starting using previously obtained parameters/states, and steps (2) and (3) benefit from parallelization in different ways. Prediction is typically embarrasingly parallel (i.e., parallelizable over the data), while training only achieves sub-linear speedup (primarily for global models; training for local models is embarrasingly parallel).

While some deep learning-based forecasting methods may require lengthy training steps (though even these are nowadays on the order of hours, not days or weeks, on data sets with millions of time series), they are typically comparable in running time to local "statistical" methods when making forecasts. For example, one of the most widely used "statistical" forecasting algorithm implementations is the `ets` from (Hyndman & Khandakar, 2008) due to its robustness and efficiency.[10] This robustness is achieved by running several models and keeping the one with the best Akaike Information Criterion. On a large weekly retail demand dataset (Seeger, Salinas, & Flunkert, 2016), `ets` takes around 30 ms to produce a forecast for a single time series (i.e. a full model selection/training/prediction cycle). The recurrent neural network model (RNN) of Salinas et al. (2019) requires about one hour for training and can then generate forecasts in around 100 ms per time series on a commodity laptop with no GPU. While `ets` is indeed faster, it is noteworthy that the time differential between the two methods is smaller than could be expect given the difference in model complexity. In addition, training time can be amortized when forecasts are made periodically without re-training. In a production setting, we have seen that more than 80% of the compute costs are incurred during inference and only 20% during training.

[10] In particular, with the default settings `model = "ZZZ"` which uses model selection. We use version 8.5 of the Forecast package.

This is very much in contrast with the experimental stage of model building where most of the time is spend on training, as that is what is needed to improve models.

The M4 competition and in particular (Makridakis et al., 2020) has taken a first step to take computation time into account.

In addition to computational costs, other costs may need to be considered as well: A forecasting method that requires supervision and tuning by an expert human operator may ultimately be more expensive than a computationally demanding but fully automated method.

3.4. Linearity & convexity

Classifying methods into linear and non-linear classes is common place in most mathematical disciplines, as it is typically easier to establish formal results for the class of linear methods than it is for non-linear methods.

In the forecasting setting, there are multiple aspects of a model which can be (non-)linear, e.g. the forecast can be a linear function of the past observations (as in linear autoregressive models), of the parameters, of the covariates (as in linear regression), or of time (i.e. models with linear trend); the underlying dynamical system can have linear dynamics; or the underlying optimization problem can have a linear objective function and/or constraints.

While some "statistical" methods are linear in one or more of the above senses, and many "ML" methods are non-linear, conflating linear with "statistical" and non-linear with "ML" methods as in Makridakis et al. (2018b) is an overgeneralization, as there exist "statistical" methods that are not linear in any meaningful way (e.g. exponential smoothing with damped or multiplicative trend), and there also exist "ML" methods that are linear (e.g. linear support vector regression (Smola & Schölkopf, 2004) or matrix factorization (Yu, Rao, & Dhillon, 2016)).

Similar to linearity, convexity of the underlying objective function of a learning/forecasting algorithm is an attribute that is often considered, as convex optimization problems are generally tractable. Postulating a convex loss function is a straight-forward way of ensuring that a practical optimization procedure (e.g. gradient descent) can identify a global optimum instead of just a local one. A non-convex loss function on the other hand makes identifying a global optimum practically impossible in most cases, making it challenging to establish theoretical guarantees for such methods. However, the statistics and ML communities have found non-convex models to be effective for many applications (Goodfellow, Bengio, & Courville, 2016; Hieber et al., 2018; van den Oord et al., 2016), including forecasting.

Forecasting methods widely used in the statistics community—such as variants of the generalized autoregressive conditional heteroskedasticity model—require solving non-convex optimization problems. The parameter identification problem, which occurs when several parametrizations of a model are observationally equivalent, is a core issue in the field of econometrics (Fisher, 1966) and stems from loss functions that have more than one global optimum and are therefore not strictly-convex.

In the field of ML, the regain in popularity of NNs triggered much discussion on the topic of (non-)convex loss functions. The skepticism towards models with non-convex loss functions (which includes almost all NN-based models) was overcome by their outstanding predictive performance in practice, helped by refinements in (mainly stochastic) gradient descent methods. Recent research in the deep learing community has to some extent validated these practical results by showing that for sufficintly large NNs almost all local minima are very similar to the global minimum (see e.g. Choromanska, Henaff, Mathieu, Arous, & LeCun, 2015).

4. *Subjective* dimensions for classifying forecasting methods

In this section we turn to subjective dimensions, dimensions of a methodological or cultural nature. While they are naturally less well-defined than objective dimensions, considering them is helpful since we believe that it is along these dimensions that the principal differences between the "ML" and "statistics" approaches to forecasting can be found.

4.1. Data-driven and model-driven methods

Methods that are generally considered as belonging to the realm of ML, such as NNs or random forest (Breiman, 2001a), are mostly data driven, in the sense that such methods memorize patterns effectively and do not make strong structural assumptions (such as "the trend has to be linear" or "the change from one point to the next must be smooth"). Data-driven models are flexible at the cost of being data-hungry, in the sense that they typically have a large number of parameters and require a sufficient amount of data to tune those parameters.

In Fig. 1, we show an example where the recurrent neural network model from (Salinas et al., 2019) is trained to predict a heteroskedastic noise with an oscillating variance amplitude.[11] From this example, one can see that a complex non-linear time series pattern can be retrieved only from the data. Clearly, being able to infer complex behavior only from data comes with the downside risk of over-fitting. Regularization schemes such as Dropout (Baldi & Sadowski, 2013) can help alleviate such effets. While it has been shown for instance that state of the art architectures can fit very large corpora of images almost perfectly with random labels (Zhang, Bengio, Hardt, Recht, & Vinyals, 2016), the case of time series may be a bit different. Indeed, recurrent neural networks can not memorize more than 5 bit of information per parameter (Collins, Sohl-Dickstein, & Sussillo, 2016). In this context, determining the amount of data (or regularization) required to avoid over-fitting in the case of forecasting remains an open problem.

[11] More precisely, this artificial dataset consists of one time series $z_t = \varepsilon_t \sin(t)$ with $\varepsilon_t \sim \mathcal{N}(0, 1)$. We train with N=1M observations with the values z_t with $t < N$ and plot the 80% prediction interval forecasted for the 150 times units following $t = N$, e.g. the last point seen in the training. Note that the frequency of the noise is inferred only from the data.

Table 2
Comparison between a pure RNN and a hybrid RNN method for the M4 dataset.

Metric	DeepAR	Smyl
sMAPE	0.1192	0.1137
MASE	1.500	1.54
owa	0.837	0.821
MSIS	12.07	12.23

At the other end of the spectrum are models generally considered as belonging to the realm of statistics, such as ARIMA models and GLMs, that are parsimoniously parameterized and therefore need little data to be accurately fitted. Model-driven approaches are more rigid in the sense that they can only model a limited set of patterns defined by the assumptions made when specifying the model. If these assumptions are correct, these models are very data-efficient as they need little data for their parameters to be accurately estimated. Counter-intuitively, systematically misspecified models may still lead to better predictions than correctly specified models (Kolassa, 2016).

In our experience, data-driven models tend to be a good choice when used as global models employed for operational forecasting problems (Januschowski & Kolassa, 2019), where they are trained on a large number of time series. For these problems, they are able to extract complex patterns with little intervention from the researcher. Model-driven approaches on the other hand often need careful feature engineering and specification to be able to adequately capture the regularities in the data, and can be a very efficient choice when dealing with a sufficiently small number of series for which the researcher is able to specify an appropriate model.

Smyl's winning solution to the M4 competition (Smyl, Ranganathan, & Pasqua, 2020) shows that models in which data-driven and model-driven methods interact can perform extremely well, and more research in understanding when and how such beneficial interactions occur is needed. For example, in Smyl's approach, a natural next experiment could be an ablation study to tease out the effect of the data-driven and the model-driven parts of the solution on the overall accuracy.

Table 2 contains the results of a data-driven method (Salinas et al., 2019) as implemented in Januschowski et al. (2018) with default hyper-parameters. The method uses a recurrent NN but no exponential smoothing equation. Its accuracy is close to the winning solution in the M4 competition.

4.2. Ensembles, model combinations, and single models

Section 3.4 introduced a dimension which allows to distinguish simple from complex models. Another dimension along which we can differentiate between simple and complex models is to single out ensemble methods as exemplified in the M4 competition.

The M4 competition has a single entry for both combination and single models, but does mark a method as a *combination* method (Table 4 in Makridakis et al.

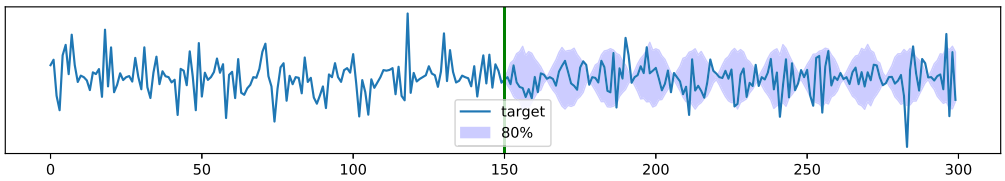

Fig. 1. Plot of the 80% prediction interval for the 150 units following the last point seen in a training set consisting of observations of an heteroskedastic cyclic noise $z_t = \varepsilon_t \sin(t)$ with $\varepsilon_t \sim \mathcal{N}(0, 1)$.

(2020)). The extremes on this distinction are clear. One the one extreme are methods such as (Montero-Manso, Athanasopoulos, Hyndman, & Talagala, 2020) where a gradient-boosted tree (Chen & Guestrin, 2016) is used to combine multiple models. On the other extreme is a fully specified model with fixed hyper-parameters where only the primary parameters are estimated.

In between both extremes, there's considerable ambiguity. Consider a method that automatically selects the best model structure in the class of ARIMA models (e.g. `auto.arima`). This method does result in a single model being used to produce a forecast, but to select this particular model multiple models were estimated and evaluated on some goodness of fit or out-of-sample accuracy criterion. The same ambiguity exists, for example, for GLMs that rely on automated feature selection techniques or for any model that makes use of hyper-parameter optimization procedures (Jenatton, Archambeau, González, & Seeger, 2017), mixtures of experts (Bishop, 2007), or drop-out regularized NNs (Baldi & Sadowski, 2013).

To us, it remains unclear where on the continuum of single models vs. combination models the above examples would fall. Despite its ambiguity, the value of this dimension comes from its separation of concerns. It allows to isolate the assessment of forecasting models from combination techniques and it allows for studies comparing end-to-end learnt approaches (such as NNs) against staged approaches and ensembles.

4.3. Discriminative & generative models

We can in general distinguish between discriminative and generative models in statistics and ML and hence, also in forecasting. For a time series $z_1, z_2, \ldots, z_T$ which we want to forecast, a generative forecasting model aims at modelling $P(z_1, z_2, \ldots, z_T, \ldots, z_{T+h})$.

In contrast, a discriminative forecasting method directly models this predictive distribution

$$P(z_{T+1}, z_{T+2}, \ldots, z_{T+h} | z_1, \ldots, z_T).$$

Often, generative models decompose $P(z_1, z_2, \ldots, z_T, \ldots, z_{T+h})$ into telescoping conditional distributions

$$P(z_1, z_2, \ldots, z_{T+h}) = P(z_1)P(z_2|z_1)P(z_3|z_1, z_2)\cdots \cdot P(z_{T+h}|z_1, \ldots, z_{T+h-1}).$$

Using Bayes rule, we can then obtain the predictive distribution for making forecasts for periods $T+1, \ldots, T+h$, conditioned on the past

$$P(z_{T+1}, z_{T+2}, \ldots, z_{T+h} | z_1, z_2, \ldots, z_T).$$

Note that while this distinction seems rigorous at first for forecasting models, there are models that can be used both as discriminative and as generative models, hence we decided to place this distinction in the subjective category.

Different NN architectures for forecasting can be compared along this dimension. Sutskever, Vinyals, and Le (2014) provide a NN for discriminative, neural sequence models while concurrent research in the ML community has also provided generative models based on NNs (e.g., Fraccaro, Sønderby, Paquet, & Winther, 2016; Krishnan, Shalit, & Sontag, 2015; Rangapuram et al., 2018).

Understanding the differences between discriminative and generative models can trigger relevant research. For example, discriminative models may be more accurate in certain settings at the expense of model limitations (for example, models such as (Sutskever et al., 2014) must be re-trained if h changes). Finally, and to the best of our knowledge, there are no general theoretical results establishing the superiority of one approach over the other in a given setting, with the exception of Ng and Jordan (2002) for logistic regression.

4.4. Statistical guarantees & bounds

A reader familiar with the scientific literature published in statistical journals and ML venues will have been struck by the structural differences in the articles.

An article published in a statistics journal[12] proposing some new method (model, estimator, or test) typically starts by stating a series of assumptions on a data generating process. It then establishes the properties of the proposed method, often in the form of asymptotic results or finite sample bounds. The authors continue by reporting the results of an empirical study (typically small-scale and partially on synthetic data) aimed at validating the theoretical results and comparing to existing benchmarks, occasionally followed by a "real-world" application demonstrating the practical relevance of the proposed method.

A paper published in an ML venue[13] proposing a new procedure generally starts by describing that procedure and motivating its use. The authors then seek to contextualize the proposal within the existing literature before reporting results of experiments using the proposed

[12] We invite the reader to refer to articles from statistics journals cited in this paper.

[13] We invite the reader to refer to articles from ML journals or conferences cited in this paper.

method on several standard datasets and comparing it to benchmark procedures.

The description of articles in ML and statistics outlets in the two paragraphs above is somewhat caricatural and certainly is an over-generalization (there are application-focussed statistic papers, just as there are purely theoretical ML papers), but it does reflect a substantial methodological difference between both research communities. This methodological difference is, in our view, one of the main source of tension between the statistics and ML communities. While the statistics approach tends to favor providing theoretical guarantees based on assumptions that are not always testable (and routinely violated in practice), the complex models used in the ML community are usually not amenable to full theoretical analysis and the ML community therefore relies heavily on validation through empirical evaluation of out-of-sample performance. These purely experimental results can be hard to verify since the outcome can hinge on details of the experimental setup. The reproducibility concern has been recognized in the ML community (e.g., major ML outlets such as KDD and NeurIPS now "expect" source code). Even if the experiments are reproducible, the results are not necessarily generalizable to other datasets and problems. Furthermore, an over-reliance on standardized benchmarks may result in overfitting on these benchmarks.

4.5. Explanatory modeling & interpretability

Forecasting is primarily concerned with *predictive modelling* rather than *explanatory modelling* (Shmueli, 2010). Still, interpretability, e.g. defined as "the degree to which an observer can understand the cause of a decision" (Miller, 2017), of forecasting models often high on the list of desiderata despite the fuzzyness of the concept. "ML" methods are often seen as a black boxes, and therefore inherently not amenable to interpretation. Indeed, the complex, non-linear mappings implemented by NNs, for example, make interpretation challenging. This is an active research topic (Lapuschkin et al., 2019; Montavon, Samek, & Müller, 2018; Ribeiro, Singh, & Guestrin, 2016).

In contrast, because of their parsimonious parametrization linear models are often viewed as being straightforward to interpret. This simplicity is deceptive though. Consider adding to an estimated model a new covariate that is not orthogonal with those already included in the model. This will modify the partial correlations estimated by the model, including possibly the sign of the associated parameters. As a consequence, the interpretation of the model will change. The amount of structural insight on the data generating process that can be gained from looking at the partial correlations estimated by a linear model is limited and contingent.

In the particular case of forecasting, non-linear transformations of the raw input data (which are at the core of NNs) and feature engineering are often crucial for achieving high accuracy and the success of many classical methods. These transformations often hinge on complex (and sometimes manual) data pipelines that transform data, e.g. by de-seasonalization, de-trending, and other complex pre-processing steps (see e.g., Hyndman & Athanasopoulos, 2017; Mohammadipour, Boylan, & Syntetos, 2012). Even if we assume the core model to be interpretable, it is only so in light of its direct inputs which have been subject to difficult to interpret transformations (Böse et al., 2017). A classic example of this are cold start (e.g. new product) forecasts (Kahn & Chase, 2018), where many approaches require determining the similarity between items (via judgmental approaches, clustering or embeddings in modern ML parlay). Most of these are highly non-interpretable.

Causality. Interpreting partial correlations does not help understand "the cause of a decision" (Miller, 2017). Causal modeling is an active research topic in the statistics literature (Granger, 2001; Imbens & Rubin, 2015; Pearl, 2009b) as well as in the ML literature (Pearl, 2009a; Peters, Janzing, & Schölkopf, 2017). Research on causality at the intersection of traditional ML and statistics methods provides an example of fruitful cross-pollination between approaches traditionally classified as statistical (or econometric) and ML, see for example (Athey & Imbens, 2015; Chernozhukov et al., 2016; Wager & Athey, 2017), but this topic is beyond the scope of the present paper.

5. Opportunities & challenges

The previous sections aimed at showing that the methods developed by the statistics and ML communities share a lot of concepts but that these communities have developed divergent cultures and vocabularies, and emphasized different topics in their research agendas. We believe that they can learn and benefit from each others' strengths. The conclusion of this paper discusses what we view as the most promising opportunities.

Software frameworks. ML frameworks have reached a high degree of sophistication over the last years and extended their reach beyond pure deep learning. These frameworks allow for a quick turn-around in model development. This is enabled by many components coming together in these frameworks: fast computational primitives on tensors, auto-differentiation, implementations of sophisticated optimization algorithms, probabilistic tools such as distributions and sampling algorithms, allowing e.g. the implementation of MCMC frameworks (Anonymous, 2019; Bingham et al., 2018).

MXNet (Chen et al., 2015) for example even offers an R front-end, although most deep learning frameworks focus on Python. Many application-specific toolkits exist (e.g., for machine translation (Hieber et al., 2018), Gaussian Processes (Dai, Meissner, & Lawrence, 2018), computer vision, natural language processing[14]) and it is only a question of time until one for forecasting will be available.[15]

The convergence on an open-source ecosystem for forecasting based on these deep learning frameworks

[14] https://kdd18.mxnet.io/ https://github.com/dmlc/gluon-cv.

[15] https://eng.uber.com/m4-forecasting-competition/.

will allow for greater reproducibility of experiments, better model exchange, and help popularize these methods, much like the R `forecast` package has stimulated research.

Empirical rigor. The M-competitions are examples for the rigor with which the forecasting community approaches empirical experiments. Adopting this rigor in the empirical evaluations for forecasting methods which is also present in journal such as the IJF would be a large step forward for the ML community. Makridakis et al. (2018b) correctly points this out. The ML community working on forecasting should compare against the state of the art in the forecasting community and commonly accepted baselines such as the R `forecast` package (Hyndman & Khandakar, 2008), e.g., (Mukherjee et al., 2018; Wen et al., 2017). Conversely, a number of forecasting competitions are available on Kaggle.[16] Conclusions from these competition such as the success of ensemble and data-driven methods have foreshadowed the results of the M4 competition and could be taken more seriously in the academic literature.

It is important to note that we should limit the conclusions based on empirical evidence to the data sets on which the methods are evaluated. The forecasting landscape is rich and contains many different forecasting tasks and data sets. If the benchmark dataset only covers certain domains within forecasting or focuses on a single forecasting task (e.g., only on cold-start forecasting), it is important to clearly define the intent of the experiment and to restrict the conclusions drawn from the results of an experiment to this context. For example, we expect data-driven methods to perform well on operational forecasting problems and model-based methods on strategic forecasting problems (Januschowski & Kolassa, 2019).

Practical considerations. Success in forecasting applications depends on many details that are often brushed under the carpet by the different communities to varying degrees. We'll provide a few examples in the following. Carefully crafting the training schemes of forecasting models, such as (Laptev et al., 2017; Salinas et al., 2019) is an important predictive performance boost. Designing co-variates and making them available in software packages,[17] handling them appropriately, in particular in the context of global models (Salinas et al., 2019), assembling pipelines of models (Böse et al., 2017; Mohammadipour et al., 2012), modern hyper-parameter optimization techniques (see e.g., (Jenatton et al., 2017)), model combination techniques (Section 4.2) or the reuse of already trained models and topics such as transfer and meta-learning are a selection of the many areas where the communities can learn from each other. The requirements of modern forecasting scenarios via high-dimensional, streaming or big data use cases (e.g., via internet of things applications) amplify the importance of such practical considerations.

[16] For example: https://www.kaggle.com/c/favorita-grocery-sales-forecasting/data https://www.kaggle.com/c/walmart-sales-forecasting/data https://www.kaggle.com/c/rossmann-store-sales/data https://www.kaggle.com/c/recruit-restaurant-visitor-forecasting https://www.kaggle.com/c/global-energy-forecasting-competition-2012-load-forecasting/data.

[17] For example https://tsfresh.readthedocs.io https://github.com/thiyangt/seer.

Theory. Section 4.4 describing the difference between papers published in statistics and in ML journals noted that theoretical guarantees are not considered necessary to a contribution in ML. This is an issue that is debated within the ML community with researchers pointing out that the lack of a theoretical explanation for the effectiveness of deep learning methods[18] puts continued progress in the field at risk.

Other researchers in the ML community argue that the lack of theoretical foundations is not a sufficient reason to reject a method that has proven its effectiveness empirically. This view can be supported by considering the history of the Lasso estimator. The original article proposing the Lasso (Tibshirani, 1996), published in a leading statistical journal, contains little theoretical results on the method but is mainly descriptive. After the Lasso proved to be an extremely useful estimator, a large body of research was dedicated to studying its properties in a wide range of settings, for times-series and forecasting see Basu and Michailidis (2015), Chan, Yau, and Zhang (2014), Kock and Callot (2015) among many others.

NN methods have proven their usefulness across a large number of applications, but the theoretical understanding is lagging. The gap is beginning to be closed though, see e.g. Farrell, Liang, and Misra (2018) for NNs and Athey, Tibshirani, and Wager (2016) for random forests. This is an area in which collaboration between the ML and statistics community could be beneficial to both.

6. Conclusion

While we believe that there is no added value in classifying methods according to being an ML or a statistical method, we do think that making this distinction points to a larger, important issue. The scientific communities working on forecasting, while working on the same problem, do not nearly interact as much as they should. The communities working on the forecasting problem have contributed relevant results and we therefore encourage all readers of this article to step outside their comfort zone.

We note that there are encouraging steps. For example, current research is concerned with the combination of probabilistic models with NNs (Fraccaro, Kamronn, Paquet, & Winther, 2017; Fraccaro et al., 2016; Januschowski et al., 2018; Krishnan et al., 2015; Krishnan, Shalit, & Sontag, 2017; Rangapuram et al., 2018). There are many more examples of such hybrids from methods from the ML and the statistical community where we can use gradient-boosted trees to select classical methods (see e.g., Montero-Manso et al. (2020) and runner-ups from the M4 competition), using ML approaches to improve multi-step ahead forecasting (Bontempi et al., 2013; Taieb & Hyndman, 2014) or pipelines of models which contain a combination of ML and statistical methods (Böse et al., 2017). Similar to (Makridakis et al., 2020), we expect much progress to come from this line of work and we look forward to the M5 competition.

[18] See A. Rahimi *NIPS 2017 Test-of-Time Award* acceptance speech: https://www.youtube.com/watch?v=ORHFOnaEzPc

Acknowledgments

We thank Stephan Kolassa for inspiring discussions on the topic.

References

Anonymous (2019). Modular deep probabilistic programming. In *International conference on learning representations*. (under review).

Armstrong, J. S. (1985). *Long-range forecasting: From crystal ball to computer* (2nd ed.). Wiley.

Athey, S., & Imbens, G. W. (2015). Machine learning for estimating heterogeneous causal effects. *Research papers*, Stanford University, Graduate School of Business.

Athey, S., Tibshirani, J., & Wager, S. (2016). Generalized random forests. arXiv preprint arXiv:1610.01271.

Baldi, P., & Sadowski, P. J. (2013). Understanding dropout. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems* (pp. 2814–2822). Curran Associates, Inc..

Barber, D. (2012). *Bayesian reasoning and machine learning*. New York, NY, USA: Cambridge University Press.

Basu, S., & Michailidis, G. (2015). Regularized estimation in sparse high-dimensional time series models. *The Annals of Statistics*, *43*(4), 1535–1567.

Ben Taieb, S., Taylor, J. W., & Hyndman, R. J. (2017). Coherent probabilistic forecasts for hierarchical time series. In *Proceedings of the 34th international conference on machine learning* (pp. 3348–3357).

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., et al. (2018). Pyro: deep universal probabilistic programming. *Journal of Machine Learning Research (JMLR)*, *20*, 973–978.

Bishop, C. M. (2007). Pattern recognition and machine learning, In *Information science and statistics*, (1st ed.). Springer.

Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight uncertainty in neural networks. arXiv preprint arXiv:1505.05424.

Bontempi, G., Ben Taieb, S., & Le Borgne, Y.-A. (2013). Machine learning strategies for time series forecasting. In *European business intelligence summer school* (pp. 62–77).

Böse, J.-H., Flunkert, V., Gasthaus, J., Januschowski, T., Lange, D., Salinas, D., et al. (2017). Probabilistic demand forecasting at scale. *Proceedings of the VLDB Endowment*, *10*(12), 1694–1705.

Bottou, L., & Vapnik, V. (1992). Local learning algorithms. *Neural Computation*, *4*(6), 888–900.

Breiman, L. (2001a). Random forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L. (2001b). Statistical modeling: the two cultures. *Statistical Science*, *16*(3), 199–231.

Chambers, J. C., Mullick, S. K., & Smith, D. D. (1971). How to choose the right forecasting technique. *Harvard Business Review*, *49*(4), 45.

Chan, N. H., Yau, C. Y., & Zhang, R.-M. (2014). Group LASSO for structural break time series. *Journal of the American Statistical Association*, *109*(506), 590–599.

Chen, T., & Guestrin, C. (2016). XGBoost: a scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 785–794). ACM.

Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., et al. (2015). MXNet: a flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint arXiv:1512.01274.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., & Newey, W. K. (2016). Double machine learning for treatment and causal parameters. *Technical Report*.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., & LeCun, Y. (2015). The loss surfaces of multilayer networks. In *Artificial intelligence and statistics* (pp. 192–204).

Collins, J., Sohl-Dickstein, J., & Sussillo, D. (2016). Capacity and trainability in recurrent neural networks. arXiv preprint arXiv:1611.09913.

Croston, J. D. (1972). Forecasting and stock control for intermittent demands. *The Journal of the Operational Research Society*, *23*(3), 289–303.

Dai, Z., Meissner, E., & Lawrence, N. D. (2018). MXFusion: a modular deep probabilistic programming library. In *NIPS Workshop MLOSS (machine learning open source software)*.

Diebold, F. X., & Mariano, R. S. (2002). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, *20*(1), 134–144.

Donoho, D. (2017). 50 years of data science. *Journal of Computational and Graphical Statistics*, *26*(4), 745–766.

Durbin, J., & Koopman, S. J. (2012). *Time series analysis by state space methods: Vol. 38*. OUP Oxford.

Farrell, M. H., Liang, T., & Misra, S. (2018). Deep neural networks for estimation and inference: application to causal effects and other semiparametric estimands. arXiv preprint arXiv:1809.09953.

Fisher, F. M. (1966). *The identification problem in econometrics*. McGraw-Hill.

Fraccaro, M., Kamronn, S., Paquet, U., & Winther, O. (2017). A disentangled recognition and nonlinear dynamics model for unsupervised learning. In *Advances in neural information processing systems: Vol. 30* (pp. 3604–3613).

Fraccaro, M., Sønderby, S. K., Paquet, U., & Winther, O. (2016). Sequential neural models with stochastic layers. In *Advances in neural information processing systems: Vol. 29* (pp. 2199–2207).

Gal, Y. (2016). Uncertainty in deep learning (Ph.D. thesis), University of Cambridge.

Gasthaus, J., Benidis, K., Flunkert, V., Salinas, D., Wang, Y., & Januschowski, T. (2019). Probabilistic forecasting with spline quantile function RNNs. In *Proceedings of AISTATS: Vol. 89* (pp. 1901–1910).

Geweke, J. (1977). The dynamic factor analysis of economic time series. In D. Aigner, & A. Goldberger (Eds.), *Latent variables in socio-economic models* (pp. 365–383). North-Holland.

Gneiting, T. (2011a). Making and evaluating point forecasts. *Journal of the American Statistical Association*, *106*(494), 746–762.

Gneiting, T. (2011b). Quantiles as optimal point forecasts. *International Journal of Forecasting*, *27*(2), 197–207.

Gneiting, T., & Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, *102*(477), 359–378.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Granger, C. W. (2001). *Essays in econometrics: collected papers of Clive WJ Granger: Vol. 32*. Cambridge University Press.

Hieber, F., Domhan, T., Denkowski, M., Vilar, D., Sokolov, A., Clifton, A., et al. (2018). The sockeye neural machine translation toolkit at AMTA 2018. In *Proceedings of the 13th conference of the association for machine translation in the americas: Vol. 1: Research Papers* (pp. 200–207).

Hyndman, R. J., & Athanasopoulos, G. (2017). *Forecasting: principles and practice*.

Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, *27*, 1–22.

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679–688.

Hyndman, R. J., Koehler, A. B., Ord, J. K., & Snyder, R. D. (2008). *Springer series in statistics*, *Forecasting with exponential smoothing: the state space approach*. Springer.

Hyndman, R. J., & Shenstone, L. (2005). Stochastic models underlying Croston's method for intermittent demand forecasting. *Journal of Forecasting*, *24*(6), 389–402.

Imbens, G. W., & Rubin, D. B. (2015). *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.

Januschowski, T., Gasthaus, J., Wang, Y., Rangapuram, S. S., & Callot, L. (2018). Deep learning for forecasting: current trends and challenges. *Foresight: The International Journal of Applied Forecasting*, *51*, 42–47.

Januschowski, T., & Kolassa, S. (2019). A classification of business forecasting problems. *Foresight: The International Journal of Applied Forecasting*, *52*, 36–43.

Jenatton, R., Archambeau, C., González, J., & Seeger, M. (2017). Bayesian optimization with tree-structured dependencies. In *Proceedings of the 34th international conference on machine learning* (pp. 1655–1664).

Kahn, K. B., & Chase, . C. W. (2018). The state of new-product forecasting. *Foresight: The International Journal of Applied Forecasting*, *51*, 24–31.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114.

Kock, A. B., & Callot, L. (2015). Oracle inequalities for high dimensional vector autoregressions. *Journal of Econometrics*, *186*(2), 325–344.
Koenker, R. (2005). *Econometric Society Monographs*, *Quantile regression*. Cambridge University Press.
Kolassa, S. (2011). Combining exponential smoothing forecasts using Akaike weights. *International Journal of Forecasting*, *27*(2), 238–251.
Kolassa, S. (2016). Sometimes it's better to be simple than correct. *Foresight: The International Journal of Applied Forecasting*, *40*, 20–26.
Kolassa, S., & Schuetz, W. (2007). Advantages of the mad/mean ratio over the MAPE. *Foresight: The International Journal of Applied Forecasting*, *6*, 40–43.
Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: principles and techniques – adaptive computation and machine learning*. The MIT Press.
Krishnan, R. G., Shalit, U., & Sontag, D. (2015). Deep Kalman filters. arXiv preprint arXiv:1511.05121.
Krishnan, R. G., Shalit, U., & Sontag, D. (2017). Structured inference networks for nonlinear state space models.. In *AAAI* (pp. 2101–2109).
Laptev, N., Yosinsk, J., Li Erran, L., & Smyl, S. (2017). Time-series extreme event forecasting with neural networks at Uber. In *ICML time series workshop*.
Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., & Müller, K. (2019). Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, *10*, 1–8.
Maddix, D. C., Wang, Y., & Smola, A. (2018). Deep factors with Gaussian processes for forecasting. In *Proceedings of bayesian deep learning (NIPS workshop)* (pp. 1–5).
Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., et al. (1982). The accuracy of extrapolation (time series) methods: results of a forecasting competition. *Journal of Forecasting*, *1*(2), 111–153.
Makridakis, S., Chatfield, C., Hibon, M., Lawrence, M., Mills, T., Ord, K., et al. (1993). The M2-competition: a real-time judgmentally based forecasting study. *International Journal of Forecasting*, *9*(1), 5–22.
Makridakis, S., & Hibon, M. (2000). The M3-competition: results, conclusions and implications. *International Journal of Forecasting*, *16*(4), 451–476.
Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018a). The M4 competition: results, findings, conclusion and way forward. *International Journal of Forecasting*, *34*(4), 802–808.
Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018b). Statistical and machine learning forecasting methods: concerns and ways forward. *PLOS ONE*, *13*(3), 1–26.
Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 competition: 100,000 time series and 61 methods. *International Journal of Forecasting*, *36*(1), 54–74.
Miller, T. (2017). Explanation in artificial intelligence: insights from the social sciences. arXiv preprint arXiv:1706.07269.
Mohammadipour, M., Boylan, J., & Syntetos, A. (2012). The application of product-group seasonal indexes to individual products. *Foresight: The International Journal of Applied Forecasting*, *26*, 20–26.
Montavon, G., Samek, W., & Müller, K. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing* , *73*, 1–15.
Montero-Manso, P., Athanasopoulos, G., Hyndman, R. J., & Talagala, T. S. (2020). FFORMA: feature-based forecast model averaging. *International Journal of Forecasting*, *36*(1), 86–92.
Mukherjee, S., Shankar, D., Ghosh, A., Tathawadekar, N., Kompalli, P., Sarawagi, S., et al. (2018). ARMDN: Associative and recurrent mixture density networks for eretail demand forecasting. arXiv preprint arXiv:1803.03800.
Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. The MIT Press.
Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. In *Advances in neural information processing systems. Vol. 14* (pp. 841–848). MIT Press.
van den Oord, A., Dieleman, H., Zen, S., Simonyan, K., Vinyals, O., Graves, A., et al. (2016). WaveNet: a generative model for raw audio. In *The 9th ISCA speech synthesis workshop* (p. 125).
Pearl, J. (2009a). *Causality*. Cambridge University Press.
Pearl, J. (2009b). Causal inference in statistics: an overview. *Statistics Surveys*, *3*, 96–146.
Peters, J., Janzing, D., & Schölkopf, B. (2017). *Elements of causal inference: foundations and learning algorithms*. MIT Press.
Rangapuram, S. S., Seeger, M., Gasthaus, J., Stella, L., Wang, Y., & Januschowski, T. (2018). Deep state space models for time series forecasting. In *Advances in neural information processing systems: Vol. 31* (pp. 7785–7794).
Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135–1144).
Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2019). DeepAR: probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, (in press).
Seeger, M. W., Salinas, D., & Flunkert, V. (2016). Bayesian intermittent demand forecasting for large inventories. In *Advances in neural information processing systems: Vol. 29* (pp. 4646–4654).
Shmueli, G. (2010). To explain or to predict?. *Statistical Science*, *25*(3), 289–310.
Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*(3), 199–222.
Smyl, S., Ranganathan, J., & Pasqua, A. (2020). M4 Forecasting competition: introducing a new hybrid ES-RNN model. *International Journal of Forecasting*, *this issue*.
Stock, J. H., & Watson, M. W. (2002). Macroeconomic forecasting using diffusion indexes. *Journal of Business & Economic Statistics*, *20*(2), 147–162.
Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems: Vol. 27* (pp. 3104–3112).
Taieb, S. B., & Hyndman, R. J. (2014). Boosting multi-step autoregressive forecasts. In E. P. Xing, & T. Jebara (Eds.), *Proceedings of the 31st international conference on machine learning: Vol. 32* (1), (pp. 109–117). Bejing, China: PMLR.
Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *58*, 267–288.
Wager, S., & Athey, S. (2017). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, *113*, 1228–1242.
Wang, Y., Maddix, D. C., Gasthaus, J., Foster, D., Smola, A., & Januschowski, T. (2019). Deep factors for forecasting. In *Proceedings of the 36th international conference on machine learning* (in press).
Weigend, A. S., & Gershenfeld, N. A. (1993). *Time series prediction: forecasting the future and understanding the past*. Santa Fe Institute Series.
Wen, R. W., Torkkola, K., & Narayanaswamy, B. (2017). A multi-horizon quantile recurrent forecaster. In *NIPS time series workshop* (pp. 1–8).
Yu, H.-F., Rao, N., & Dhillon, I. S. (2016). Temporal regularized matrix factorization for high-dimensional time series prediction. In *Advances in neural information processing systems: Vol. 29* (pp. 847–855). Curran Associates, Inc..
Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530.
Zhang, G., Patuwo, B., & Hu, M. Y. (1998). Forecasting with artificial neural networks: the state of the art. *International Journal of Forecasting*, *14*(1), 35–62.

Tim Januschowski is a Machine Learning Science Manager in Amazon's AWS AI Labs. He has worked on forecasting since starting his professional career at SAP. At Amazon, he has produced end-to-end solutions for a wide variety of forecasting problems, from demand forecasting to server capacity forecasting. Tim's personal interests in forecasting span applications, system, algorithm and modeling aspects and the downstream mathematical programming problems. He studied Mathematics at TU Berlin, IMPA, Rio de Janeiro, and Zuse-Institute Berlin and holds a PhD from University College Cork.

Jan Gasthaus is a Senior Machine Learning Scientist in Amazon's AWS AI Labs, working mainly on time series forecasting and large-scale probabilistic machine learning. He is passionate about developing novel machine learning solutions for addressing challenging business

problems with scalable machine learning systems, all the way from scientific ideation to productization. Prior to joining Amazon, Jan obtained a BS in Cognitive Science from the University of Osnabrueck, a MS in Intelligent Systems from UCL, and pursued a PhD at the Gatsby Unit, UCL, focusing on Nonparametric Bayesian methods for sequence data.

Yuyang Wang is a Senior Machine Learning Scientist in Amazon AI Labs, working mainly on large-scale probabilistic machine learning with its application in Forecasting. He received his PhD in Computer Science from Tufts University, MA, US and he holds an MS from the Department of Computer Science at Tsinghua University, Beijing, China. His research interests span statistical machine learning, numerical linear algebra, and random matrix theory. In forecasting, Yuyang has worked on all aspects ranging from practical applications to theoretical foundations. algebra, and random matrix theory.

David Salinas is a senior Machine Learning scientist working at Amazon's AWS AI Labs. He has been focusing on various forecasting problems including development of new models or their productization. In particular, he worked on applying ISSM to intermittent demand and on leveraging neural networks for forecasting. Recently, he contributed to bringing a neural network forecasting model (DeepAR) as a service in AWS SageMaker. Before working at Amazon, he was working on Computational Topology and Geometry Processing.

Valentin Flunkert is a Senior Machine Learning Scientist in Amazon's AWS AI Labs, where he has developed new deep learning based forecasting methods and applied them to solve a range of business problems. Prior to joining Amazon he received his PhD in Theoretical Physics from TU Berlin and worked as a software engineer at SAP.

Michael Bohlke-Schneider is a Data Scientist in Amazon's AWS AI Labs. His research interests are machine learning, smart search algorithms, and data fusion. Michael studied Biology at Ruhr-Universitaet Bochum and MIT and received his PhD in Computer Science from TU Berlin.

Laurent Callot is a Senior Economist in the Intelligent Cloud Control Machine Learning team at Amazon. He received his PhD in time series econometrics at CREATES, Aarhus University. In his academic and professional careers he has worked in time series econometrics, forecasting, causal modeling, and statistical machine learning, all of which continue to be interests of his.