Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2021.DOI

Two-Step Meta-Learning for Time-Series Forecasting Ensemble

Evaldas Vaiciukynas[1], Paulius Danenas[2], Vilius Kontrimas[3], and Rimantas Butleris[2]
[1]Department of Information Systems, Faculty of Informatics, Kaunas University of Technology, Kaunas, Lithuania
[2]Centre of Information Systems Design Technologies, Faculty of Informatics, Kaunas University of Technology, Kaunas, Lithuania
[3]Private limited liability company "Rivile", Vilnius, Lithuania

Corresponding author: Evaldas Vaiciukynas (e-mail: evaldas.vaiciukynas@ktu.lt).

This study contains research results from European Union funded project No. J05-LVPA-K-04-0004 "Artificial intelligence and statistical methods based time series forecasting and management" where a grant was administered by Lithuanian business promotion agency.

ABSTRACT Amounts of historical data collected increase and business intelligence applicability with automatic forecasting of time series are in high demand. While no single time series modeling method is universal to all types of dynamics, forecasting using an ensemble of several methods is often seen as a compromise. Instead of fixing ensemble diversity and size, we propose to predict these aspects adaptively using meta-learning. Meta-learning here considers two separate random forest regression models, built on 390 time-series features, to rank 22 univariate forecasting methods and recommend ensemble size. The forecasting ensemble is consequently formed from methods ranked as the best, and forecasts are pooled using either simple or weighted average (with a weight corresponding to reciprocal rank). The proposed approach was tested on 12561 micro-economic time-series (expanded to 38633 for various forecasting horizons) of M4 competition where meta-learning outperformed Theta and Comb benchmarks by relative forecasting errors for all data types and horizons. Best overall results were achieved by weighted pooling with a symmetric mean absolute percentage error of 9.21% versus 11.05% obtained using the Theta method.

INDEX TERMS business intelligence, univariate time-series model, forecasting ensemble, meta-learning, random forest, M4 competition

I. INTRODUCTION

Forecasting key performance indicators and any essential dynamics for an organization should be a high-priority business intelligence task. The aim is to envisage indicator values into the future, based on historical observations. An accurate forecast mitigates uncertainties about the future outlook and can reduce errors in decisions and planning, directly influencing the achievability of goals and contributing to risk management. Forecasting should be an integral part of the decision-making activities in management [1] since the strategic success of the organization depends upon the practical relation between accuracy of forecast and flexibility of resource allocation plan [2]. It is expected that increasing amounts of historical data records, which constitute valuable resources for the forecasting task, will facilitate accurate forecasting and boost these forecasts' importance. Although [3], while researching main determinants of forecasting accuracy, found that increasing time-series length has a small positive effect on forecasting accuracy, which contradicts insights from the machine learning and deep learning fields.

Almost 40 years ago worldwide, Makridakis forecasting competitions had started (organized in 1982, 1993, 2000, 2018, and 2020) with a goal to benchmark progress in forecasting techniques and derive scientific insights in time-series forecasting. During these events, teams of participants compete to obtain forecasts for ever-increasing amounts of time-series from diverse fields. Results are summarized into recommendations on the usefulness of various time-series models or their ensembles. In the recent M4 competition [4] the leading forecasting techniques (12 from 17 most accurate ones) featured model ensembles that pool forecasts of several, mainly statistical, models. The best solution was submitted by Uber Technologies, where the hybrid technique combined a statistical forecasting model with neural network architecture. The next most successful submission [5] featured an ensemble of statistical models where weights were thoroughly tuned, and the machine learning model learned these weight recommendations for later prediction. Insights after an older M3 competition [6] were that forecasts from univariate time-series models almost always (except for annual data) are more accurate than forecasts from multivariate time-series models with external variables (i.e., macroeconomic indicators). Comparison between univariate approaches revealed that more complex models do not guarantee higher accuracy.

Proceeding from results of Makridakis competitions [4], [6] and numerous academic researches [7]–[11] it can be concluded that an ensemble of univariate time-series models often outperforms the best member of the ensemble with respect to forecasting accuracy. The success of forecasting ensemble lies in the diversity of its members [12], which contributes to robustness against concept drift [13] and enhances algorithmic stability [14]. Besides ensemble diversity, the individual accuracy of its members is also of utmost importance [15]. A simple arithmetic average with all members weighted equally often outperforms more complex strategies for combining forecasts from several models. Advanced strategies seek to find optimal weights, for example, based on the model's Akaike information criterion [11] or model's in-sample forecasting errors [16] on the last dynamics of time-series in question. In practice, to avoid corrupting the final forecast by a single inaccurate model, variants of robust average or simply median are recommended in forecast pooling [9]. Choice of weights for ensemble members often relies upon in-sample forecasting errors. However, when the approximate ranking of the model pool is available instead of exact errors, the weight could be derived from the model's reciprocal rank [17].

There exists no forecasting method that performs best on all types of time-series [18]. However, efforts to create more universal approaches seek to adjust ensemble size and choose potential members or weights adaptively based on dynamics we try to extrapolate into the future. The argument that the relative accuracy of forecasting methods depends upon the properties of the time-series and information on dynamics at hand can be exploited to choose a suitable model is an old one [19].

Our research explores an adaptive construction of a forecasting ensemble consisting of various statistical and a few machine learning methods with a meta-learning approach. Meta-learners here seek to rank a pool of methods and recommend ensemble size based on historical time-series data characteristics. Recommendations of introduced forecasting assistants are based on training regression meta-models through forecasting experiments on a diverse set of real-world examples - micro-economic time-series from M4 competition. Experiments compare introduced forecasting ensemble based on recommendations from assistants with the best benchmark methods from M4 competition - Theta and Comb, which were outperformed only by 17 out of 49 submissions in M4 competition [4], [20].

II. RELATED WORK

The usefulness of statistics summarizing the data available in a time-series in predicting the relative accuracy of different forecasting methods was explored in [21] where they created regression models to predict the expected error of the forecasting method for the time-series at hand. More similar research to our idea of forecasting assistants, after early expert system with rules derived by human analysts in [22], are forecasting techniques based on meta-learning [15], [18], [23] and recommendation rules [24], [25]. In general, meta-learner after the induction phase is capable of indicating which learning method is the most appropriate to a given problem [26]. The meta-learning concept for time-series forecasting uses a machine learning model (i.e., decision tree or ensemble of trees) and trains it on a set of features, – various characteristics of time-series – to recommend the most suitable univariate time-series model. It was also found that meta-learning is effective even when the meta-learner is trained on time-series from one domain and tested on time-series from another [27], suggesting machine learning universal capability more widely known as transfer learning. FFORMS (Feature-based FORecast Model Selection) [23] idea was implemented in R package *seer* besides participation in M4 competition. However, due to mediocre performance, it was further developed into adaptive weight-producing forecasting ensemble FFORMA (Feature-based FORecast Model Averaging) [5], [28], available in R package *M4metalearning*, achieving second place in M4 competition. Three novel approaches for forecasting method recommendation, where the meta-learning task was based on classification or regression or both, were evaluated in [18] with recommendation considering explicitly a machine learning-based regressor method instead of a statistical one. Meta-learning approach of weigh-producing nature featuring double-channel convolutional neural network was introduced in [29]. It outperformed FFORMA and other variants of meta-learning strategies in retail sales forecasting. However, only 2 out of 9 strategies explored were univariate. In contrast, others, including the M0 winner, required influential factors (such as price, promotions, seasonality, and calendar events) with historical values and values along the forecasting horizon.

The main difference of FFORMA and M0 approaches over FFORMS and other forecasting method recommendation systems referenced above is that weights for a pool of models in an ensemble are recommended instead of a single best model. A detailed overview of historical meta-learning approaches can be found in [29]. Building upon the success of FFORMA, we propose to simplify meta-learning by decomposing it into two separate regression tasks, where A1 assistant ranks the pool of potential time-series models and A2 assistant recommends ensemble size to cap the ranked list. Such simplification avoids the tedious process of weight tuning and could diminish overfitting risk due to overtuning.

III. METHODOLOGY

Assistants A1 and A2 use time-series features for modeling and meta-learning target attribute, which corresponds to a rank of a specific forecasting technique for A1 and a recommended ensemble size for A2. Each time-series used in meta-learning preparation phase is split into train/test parts. Features are calculated on the training part, while forecasts using a pool of univariate forecasting models are obtained on the testing part. Forecasting errors on the testing part are estimated, and forecasting models are ranked for A1; meanwhile, all possible ensembles are evaluated for A2. After the training phase of A1 and A2 models, their usefulness in helping with time-series forecasting is evaluated using M4-micro dataset in the testing phase. Methodology pipeline is illustrated by Fig. 1. Additionally, variable importance from A1 and A2 models is reported in performed experiments.

A. TIME-SERIES FEATURES

Feature engineering for assistant models consisted of pooling various known time-series characteristics into a collection of 130 features (see Table 1). Almost half of all features were previously introduced as state-of-the-art time-series features, available in R packages *catch22* [30], consisting of carefully selected 22 features, and *tsfeatures* [31], consisting of 42 features also used in FFORMA framework [5], [28].

Seeking to extend characteristics of time-series we considered calculating features not only on the original data (*orig*), but also on the results of 2 transformations (*diff* and *log*):

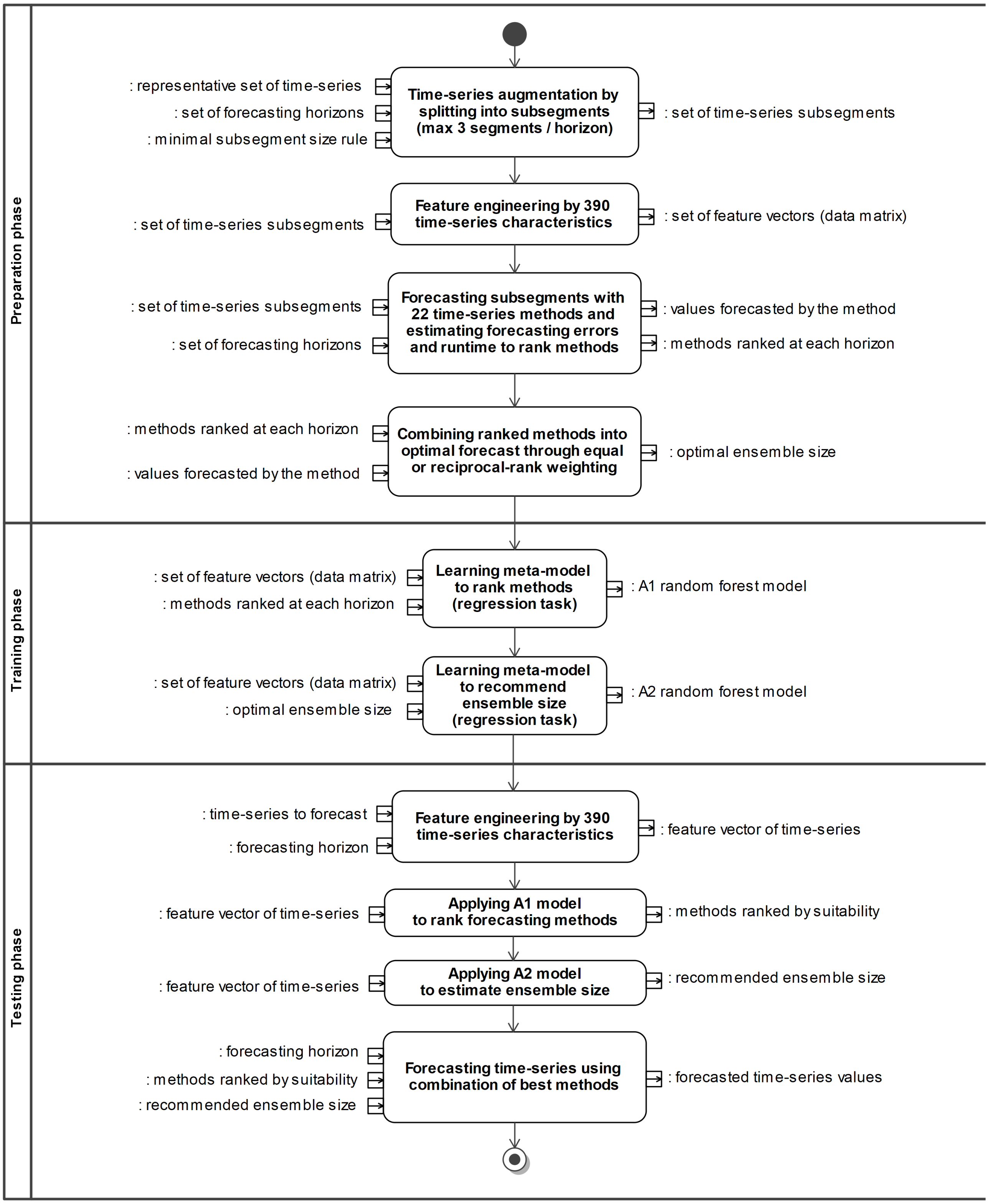
Preparation phase
: representative set of time-series
: set of forecasting horizons
: minimal subsegment size rule
Time-series augmentation by splitting into subsegments (max 3 segments / horizon)
: set of time-series subsegments
: set of time-series subsegments
Feature engineering by 390 time-series characteristics
: set of feature vectors (data matrix)
: set of time-series subsegments
: set of forecasting horizons
Forecasting subsegments with 22 time-series methods and estimating forecasting errors and runtime to rank methods
: values forecasted by the method
: methods ranked at each horizon
: methods ranked at each horizon
: values forecasted by the method
Combining ranked methods into optimal forecast through equal or reciprocal-rank weighting
: optimal ensemble size
Training phase
: set of feature vectors (data matrix)
: methods ranked at each horizon
Learning meta-model to rank methods (regression task)
: A1 random forest model
: set of feature vectors (data matrix)
: optimal ensemble size
Learning meta-model to recommend ensemble size (regression task)
: A2 random forest model
Testing phase
: time-series to forecast
: forecasting horizon
Feature engineering by 390 time-series characteristics
: feature vector of time-series
: feature vector of time-series
Applying A1 model to rank forecasting methods
: methods ranked by suitability
: feature vector of time-series
Applying A2 model to estimate ensemble size
: recommended ensemble size
: forecasting horizon
: methods ranked by suitability
: recommended ensemble size
Forecasting time-series using combination of best methods
: forecasted time-series values

FIGURE 1: Methodology outline by unified modeling language activity diagram: input (*left hand side*) and output (*right hand side*) artefacts are listed besides each action block and all process is split into 3 phases. To adhere to machine learning standards, we assure that due to performed cross-validation a set of time-series in the preparation and training phases do not mach time-series in the testing phase.

TABLE 1: Overview of time-series features for meta-learning: 130 time-series characteristics in total.

Feature set name	**Size**	**R package(-s) used // detailed list of feature names, mainly corresponding to function names in R package(-s)**
catch22	22	*catch22* // DN_HistogramMode_5, DN_HistogramMode_10, CO_f1ecac, CO_FirstMin_ac, CO_HistogramAMI_even_2_5, CO_trev_1_num, MD_hrv_classic_pnn40, SB_BinaryStats_mean_longstretch1, SB_TransitionMatrix_3ac_sumdiagcov, PD_PeriodicityWang_th0_01, CO_Embed2_Dist_tau_d_expfit_meandiff, IN_AutoMutualInfoStats_40_gaussian_fmmi, FC_LocalSimple_mean1_tauresrat, DN_OutlierInclude_p_001_mdrmd, DN_OutlierInclude_n_001_mdrmd, SP_Summaries_welch_rect_area_5_1, SB_BinaryStats_diff_longstretch0, SB_MotifThree_quantile_hh, SC_FluctAnal_2_rsrangefit_50_1_logi_prop_r1, SC_FluctAnal_2_dfa_50_1_2_logi_prop_r1, SP_Summaries_welch_rect_centroid, FC_LocalSimple_mean3_stderr
tsfeats	13	*tsfeatures* // stability, lumpiness, crossing.points.fraction, flat.spots.fraction, nonlinearity, ur.kpss, ur.pp, arch.lm, ACF1, ACF10.SS, ACF.seas, PACF10.SS, PACF.seas
stlfeats	12	*tsfeatures* // nperiods, seasonal_period, trend, spike, linearity, curvature, e_acf1, e_acf10, seasonal_strength, peak, trough, lambda
hctsa	13	*tsfeatures* // embed2_incircle_1, embed2_incircle_2, ac_9, firstmin_ac, trev_num, motiftwo_entro3, walker_propcross, std1st_der, boot_stationarity_fixed, boot_stationarity_ac2, histogram_mode_10, outlierinclude_mdrmd, first_acf_zero_crossing
hpfeats: heterogeneity, portmanteau	6	*tsfeatures*, *WeightedPortTest* // arch_acf, garch_acf, arch_r2, garch_r2, lag1.Ljung.Box, lagF.Ljung.Box
snfeats: stationarity, normality	10	*tseries*, *stats*, *nortest* // ADF, KPSS.Level, KPSS.Trend, PP, ShapiroWilk, Lilliefors, AndersonDarling, Pearson, CramerVonMises, ShapiroFrancia
ksmfeats: kurtosis, skewness, misc	11	*PerformanceAnalytics* // kurtosis.fisher, kurtosis.sample, skewness.fisher, skewness.sample, skewness.variability, skewness.volatility, skewness.kurtosis.ratio, misc.smoothing.index, misc.Kelly.ratio, misc.drowpdown.average.depth, misc.drowpdown.average.length
Hurst	17	*PerformanceAnalytics*, *longmemo*, *tsfeatures*, *liftLRD*, *pracma*, *fractal* // PerformanceAnalytics, Whittle, HaslettRaftery, lifting, pracma.Hs, pracma.Hrs, pracma.He, pracma.Hal, pracma.Ht, fractal.spectral.lag.window, fractal.spectral.wosa, fractal.spectral.multitaper, fractal.block.aggabs, fractal.block.higuchi, fractal.ACVF.beta, fractal.ACVF.alpha, fractal.ACVF.HG
fractality	7	*fractaldim* // HallWood, DCT, wavelet, variogram, madogram, rodogram, periodogram
entropy	9	*TSEntropies*, *ForeCA* // TSE.approximate, TSE.fast.sample, TSE.fast.approx, spectral.smoothF.wosa, spectral.smoothF.direct, spectral.smoothF.multitaper, spectral.smoothT.wosa, spectral.smoothT.direct, spectral.smoothT.multitaper
anomaly	10	*pracma*, *anomalize* // fraction.TukeyMAD, twitter.iqr.fraction, twitter.iqr.infraction.pos, twitter.iqr.fraction.pos, twitter.iqr.abs.last.pos, twitter.iqr.rel.last.pos, twitter.iqr.infraction.neg, twitter.iqr.fraction.neg, twitter.iqr.abs.last.neg, twitter.iqr.rel.last.neg

- *diff* - first differences help to improve/achieve stationarity;
- *log* - logarithmic transform has variance stabilizing properties.

Note that *log* transformation here is not applied column-wise to the table with extracted features, but to the original time-series, which seeks to achieve a variance-stabilizing effect on the dynamics at hand. Calculating 130 features on 3 variants of time-series (*orig*, *diff* and *log*) results in a final set of 390 features, which are later used for building assistant meta-learners.

B. FORECASTING MODELS

A representative pool of 22 univariate time-series forecasting models was selected (see Table 2). The diversity of models to consider as a potential ensemble member varies from simple, such as the seasonal naive and linear trend, to complex BATS and Prophet models. However, most of them are statistical, except for machine learning approaches NNAR and xgb. Model implementations from 6 R packages were used, where parameters when creating a model on time-series training part were chosen automatically if model implementation had that capability.

C. FORECASTING ERRORS

After fitting the univariate time-series model on the training part, forecasting can be performed for a required number of steps ahead, i.e., forecasting horizon. Comparing forecasted values with ground truth allows evaluating how accurate the forecast was, and forecasting errors are used for this purpose. We estimate three absolute and three relative forecasting errors.

Absolute forecasting errors considered:

- RMSE - root mean squared error;
- MAE - mean absolute error;
- MDAE - median absolute error.

Relative forecasting errors considered:

- SMAPE - symmetric mean absolute percentage error;
- MAAPE - mean arctangent absolute percentage error [45];
- MASE - mean absolute scaled error [46].

The final ranking of forecasting models was constructed by averaging individual rankings, obtained for each error type separately. After averaging out individual rankings, a faster model was given priority in the final ranking in case of ties. Incorporating absolute and relative errors into the final ranking allows us to sort out models more comprehensively without less bias towards a single type of error. Relative forecasting errors were also reported in experiments to compare the introduced approach to benchmark methods (Theta and Comb).

D. META-LEARNER MODEL

Meta-learner for our experiments was random forest (RF) [47] regression machine learning model. RF is an ensemble of many (*ntrees* in total) CART (classification and regression tree) instances. Each CART is built on an independent bootstrap sample of the original dataset while selecting from a random subset (of size *mtry*) of features at each tree node. Fast RF implementation in R package *ranger* [48] was chosen, which, conveniently for the specifics of our assistants, allows us to always include some variables as candidates for a binary node split besides *mtry* randomly selected ones. Time-series features were left for random selection, but a few critical meta-information features were set to *always.split.variables* parameter. Meta-information features were forecasting horizon length and data type (daily, weekly or monthly). Additionally, A1 assistant included model name

TABLE 2: A selected pool of 22 base models for univariate time-series forecasting. Most models are statistical, except for NNAR and xgb, which are based on machine learning. The horizontal line separates a few simple models from the remaining complex ones.

Model	R package::function	Description
SNaive	forecast::snaive	Seasonal naïve method
LinTrend	forecast::tslm	Linear trend
LinTrendSeason	forecast::tslm	Linear trend with seasonal dummies
QuadTrend	forecast::tslm	Quadratic trend
QuadTrendSeason	forecast::tslm	Quadratic trend with seasonal dummies
TSB	tsintermittent::tsb	Teunter-Syntetos-Babai method (based upon Croston for intermittent demand) with optimized parameters [32]
ARIMA	forecast::auto.arima	Autoregressive integrated moving average [33]
SARIMA	forecast::auto.arima	Seasonal autoregressive integrated moving average [33]
ETS	forecast::ets	Family of exponential smoothing state space models [34], [35]
HoltWinters	stats::HoltWinters	Holt-Winters filtering with additive seasonality [36]
Theta	forecast::thetaf	Theta method - simple exponential smoothing with drif [37]
STL-ARIMA	forecast::stlm	ARIMA model on seasonal decomposition of time-series [38]
STL-ETS	forecast::stlm	ETS model on seasonal decomposition of time-series [38]
StructTS	stats::StructTS	Basic stuctural model - local trend with seasonality [39]
BATS	forecast::tbats	Exponential smoothing with Box-Cox transform, ARMA errors, trend and complex seasonality [40]
Prophet	prophet::prophet	Decomposable time-series and generalized additive model with non-linear trends [41]
NNAR	forecast::nnetar	Neural network with a hidden layer and lagged inputs [42]
xgb-none	forecastxgb::xgbar	Extreme gradient boosting model with lagged inputs [43]
xgb-decompose	forecastxgb::xgbar	Extreme gradient boosting model with lagged inputs and decomposition-based seasonal adjustment [43]
thief-ARIMA	thief::thief	Temporal hierarchical approach with ARIMA at each level [44]
thief-ETS	thief::thief	Temporal hierarchical approach with ETS at each level [44]
thief-Theta	thief::thief	Temporal hierarchical approach with Theta at each level [44]

(first column in Table 2) and three dummy indicators on model capabilities such as seasonality, complexity, and decomposition.

RF size *ntrees* was fixed at 256, as recommended in literature [49], [50]. Classical RF should be composed of unpruned CART, allowing growing trees to maximal possible depth, corresponding to *min.node.size*=1 setting, but in our case *min.node.size* parameter was tuned together with *mtry* using Bayesian optimization in R package *tuneRanger* with 21 warm-up and 9 tuning iterations. The minimization objective for A1 assistant was the out-of-bag root mean squared logarithmic error - a variant of RMSE penalizing errors at lower values and achieving that prediction of best-ranked cases is more precise than of worse-ranked candidates. The minimization objective for the A2 assistant was a simple out-of-bag RMSE metric. Both being RF regression models, A1 could be nick-named as a "ranker" whereas A2 as a "capper" due to different tasks they are dedicated to. A1 ranks 22 forecasting models to find the best candidates for the time-series at hand, while A2 tries to propose an optimal size of forecasting ensemble, i.e., a number of best-ranked models to choose for forecast pooling.

Two variants of forecast pooling, namely, Simple (arithmetic average) and Weighted (weighted average), were evaluated for meta-learning. A1 assistant was identical in both variants, but A2 was constructed separately after evaluation of cumulative pooling of forecasts from the best A1-ranked univariate time-series models, where pooling was done either with equal weights or weights derived from reciprocal rank [17].

IV. M4-MICRO DATASET

We excluded two monthly cases (ID=19700 and ID=19505) from an original M4 subset of 12563 micro-economic time-series due to the lack of dynamics. Among 12561 selected cases level of aggregation was as follows: 1476 daily, 112 weekly, and 10973 monthly. All selected cases were pre-processed by segmenting them properly into train/test splits at several forecasting horizons. Forecasting horizons with a varying number of steps ahead were considered: 15, 30, 90, 180, 365, and 730 days for daily data; 4, 13, 26, 52, and 104 weeks for weekly data; 6, 12, 24, 60, and 120 months for monthly data.

Concerned with specifics of time-series representation space [51] and to avoid potential concept drift situation if meta-learners are built using random sub-spaces, we split the dataset by performing a stratified 2-fold cross-validation (2-fold CV) carefully. Stratification is done here by an efficient balanced sampling [52] on 3D t-SNE [53] projection of time-series features, which allows splitting the time-series dataset into two equally-sized parts where each part covers the overall representation space of the initial dataset. The result of such stratification is visualized in Fig. 2 with resulting CV folds depicted after 2D t-SNE projection.

Time-series expansion by segmenting data into several train/test hold-out splits was done as follows. Initially, we expand a set of time-series from 12561 to 38633 (see initial expansion in Table 3) to be able to test various forecasting horizons. Then we increase the amount of time-series from 38633 to 92846 (see final expansion in Table 3) by considering additional splitting time-series in half to be able to train meta-learners on more data. Initial expansion was carried out to benchmark various forecasting horizons, and only recent data was split-off for testing. In contrast, final expansion was considered as a way of data augmentation to harvest more training data for meta-learners. Besides initial expansion, extra two splits were done where possible - hold-out on older and newer halves of time-series. Following the recommendation of [54] for using an out-of-sample hold-out split in multiple testing periods, we consider 80/20 as a sufficient train/test ratio. After leaving out the last observations for testing (based on the forecasting horizon's length), if the amount of the training data drops down below 80% we refuse to segment time-series.

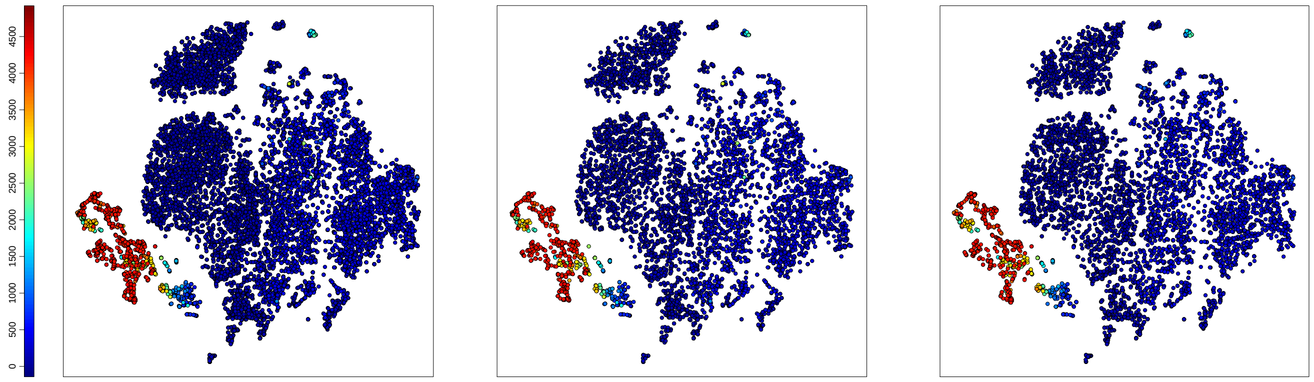

FIGURE 2: Visualization of M4-micro dataset by 2D t-SNE projection of time-series features: full sample of 12561 time-series (*left*) and result after balanced sampling into 2 cross-validation folds, containing 6281 (*center*) and 6280 (*right*) time-series. Color of the point denotes the length of time-series. Note that to avoid any informational leakage splitting into 2 cross-validation folds was done for the original dataset before proceeding with expansions.

TABLE 3: Expanding M4 micro dataset for benchmarking and augmentation. The expansion was performed by segmenting each time-series into various train/test splits, fulfilling the 80/20 heuristic. Heuristic ensures that the amount of data available for time-series model training is at least four times larger than for testing. The amount of data for estimating forecasting error of the meta-learner solution is defined by the forecasting horizon. Full sample for benchmarking at the smallest forecasting horizon consists of 12561 time-series, visualized on the left of Fig. 2.

Horizon	**Initial expansion for benchmarking**			**Final expansion for augmentation**		
	Full sample	**CV fold 1**	**CV fold 2**	**Full sample**	**CV fold 1**	**CV fold 2**
15 days	1476	736	740	4374	2178	2196
30 days	1443	720	723	4201	2100	2101
90 days	1335	667	668	3699	1853	1846
180 days	1181	593	588	3313	1661	1652
365 days	1047	524	523	2693	1346	1347
730 days	780	389	391	780	389	391
$\sum$	7262	3629	3633	19060	9527	9533
4 weeks	112	58	54	206	106	100
13 weeks	112	58	54	206	106	100
26 weeks	47	24	23	141	72	69
52 weeks	47	24	23	119	60	59
104 weeks	36	18	18	76	40	36
$\sum$	354	182	172	748	384	364
6 months	10973	5486	5487	32904	16452	16452
12 months	10973	5486	5487	22317	11166	11151
24 months	5574	2792	2782	14110	7055	7055
60 months	3432	1713	1719	3630	1813	1817
120 months	65	31	34	77	33	44
$\sum$	31017	15508	15509	73038	36519	36519
$\sum$	38633	19319	19314	92846	46430	46416

V. EXPERIMENTAL RESULTS

The forecasting experiment was performed using 2-fold CV by creating assistants on CV fold 1 of final expansion and testing success of assistant-recommended forecasting ensemble on CV fold 2 of initial expansion and vice versa. Besides forecasting using the proposed approach, benchmark methods Theta and Comb were used on initial expansion, and relative forecasting errors were calculated for comparison.

Results by SMAPE (see Table 4) demonstrate that both Simple and Weighted variants of meta-learning outperform Theta and Comb techniques. Weighted slightly outperformed Simple variant for more than half (10 out of 16) horizons and also overall (see the last row in Table 4).

Results by MAAPE (see Table 5) demonstrate that both Simple and Weighted variants of meta-learning outperform Theta and Comb techniques. Weighted slightly outperformed Simple variant for more than half (11 out of 16) horizons and also overall (see the last row in Table 5).

Results by MASE (see Table 6) demonstrate that both Simple and Weighted variants of meta-learning outperform Theta and Comb techniques. Weighted slightly outperformed Simple variant for more than half (12 out of 16) horizons and also overall (see the last row in Table 5).

Forecasting errors showed an expected and consistent tendency to increase together with increasing forecasting horizon length. Interestingly, MASE errors were the lowest overall for weekly whereas SMAPE and MAAPE for daily data. To summarize over all data types and horizons: among benchmark methods, Theta tends to outperform Comb slightly, and meta-learning approaches win over both benchmarks with the Weighted variant

TABLE 4: Forecasting results according to SMAPE forecasting error. $\sum$ denotes results over all horizons, the best result for each row is formatted in italic-bold.

Horizon	Theta	Comb	Simple	Weighted
15 days	2.507 ± 0.152	2.511 ± 0.146	**2.326 ± 0.145**	2.332 ± 0.153
30 days	3.354 ± 0.277	3.340 ± 0.282	3.233 ± 0.287	**3.229 ± 0.286**
90 days	5.587 ± 0.312	5.576 ± 0.318	**5.060 ± 0.310**	5.064 ± 0.310
180 days	8.296 ± 0.444	8.565 ± 0.452	7.298 ± 0.438	**7.282 ± 0.438**
365 days	17.258 ± 0.919	17.304 ± 0.895	**14.098 ± 0.829**	14.145 ± 0.828
730 days	18.270 ± 1.003	19.052 ± 1.042	16.396 ± 0.974	**16.395 ± 0.991**
$\sum$	8.003 ± 0.245	8.133 ± 0.248	**7.026 ± 0.225**	7.031 ± 0.226
4 weeks	9.483 ± 1.152	9.653 ± 1.198	7.941 ± 1.024	**7.867 ± 1.000**
13 weeks	9.365 ± 1.130	8.910 ± 1.047	**8.481 ± 1.076**	8.637 ± 1.222
26 weeks	8.895 ± 4.395	8.594 ± 4.404	8.184 ± 4.378	**8.092 ± 4.371**
52 weeks	13.340 ± 4.619	12.575 ± 4.036	**10.763 ± 3.519**	10.922 ± 3.784
104 weeks	17.452 ± 5.277	18.198 ± 6.037	16.273 ± 4.767	**16.008 ± 4.855**
$\sum$	10.690 ± 1.126	10.534 ± 1.127	**9.366 ± 1.010**	9.374 ± 1.041
6 months	12.150 ± 0.278	12.147 ± 0.286	9.158 ± 0.214	**9.096 ± 0.212**
12 months	12.183 ± 0.246	12.874 ± 0.269	10.479 ± 0.222	**10.420 ± 0.220**
24 months	9.544 ± 0.352	9.325 ± 0.353	**8.451 ± 0.327**	8.486 ± 0.333
60 months	12.724 ± 0.604	14.780 ± 0.740	11.617 ± 0.583	**11.449 ± 0.567**
120 months	15.051 ± 5.820	13.656 ± 4.244	10.798 ± 3.371	**10.650 ± 3.329**
$\sum$	11.763 ± 0.161	12.192 ± 0.174	9.774 ± 0.140	**9.719 ± 0.139**
$\sum$	11.047 ± 0.139	11.414 ± 0.149	9.254 ± 0.121	**9.210 ± 0.120**

TABLE 5: Forecasting results according to MAAPE forecasting error. $\sum$ denotes results over all horizons, the best result for each row is formatted in italic-bold.

Horizon	Theta	Comb	Simple	Weighted
15 days	2.520 ± 0.151	2.536 ± 0.153	2.335 ± 0.148	**2.335 ± 0.153**
30 days	3.312 ± 0.268	3.296 ± 0.269	3.185 ± 0.273	**3.183 ± 0.272**
90 days	5.538 ± 0.316	5.526 ± 0.318	**5.033 ± 0.317**	5.036 ± 0.317
180 days	8.047 ± 0.433	8.366 ± 0.444	7.161 ± 0.434	**7.142 ± 0.434**
365 days	15.431 ± 0.783	15.538 ± 0.772	**13.132 ± 0.766**	13.165 ± 0.765
730 days	17.696 ± 0.971	17.989 ± 0.975	16.306 ± 0.993	**16.286 ± 1.000**
$\sum$	7.622 ± 0.226	7.719 ± 0.228	6.842 ± 0.218	**6.842 ± 0.219**
4 weeks	8.750 ± 1.027	8.856 ± 1.061	7.570 ± 0.953	**7.514 ± 0.938**
13 weeks	9.475 ± 1.146	8.896 ± 1.057	**8.636 ± 1.142**	8.717 ± 1.207
26 weeks	8.471 ± 4.175	8.233 ± 4.190	7.945 ± 4.213	**7.849 ± 4.208**
52 weeks	12.084 ± 3.451	11.620 ± 3.374	**10.231 ± 3.223**	10.270 ± 3.320
104 weeks	15.683 ± 3.885	16.119 ± 4.218	15.251 ± 3.890	**14.979 ± 3.921**
$\sum$	10.090 ± 0.955	9.891 ± 0.962	9.091 ± 0.938	**9.064 ± 0.949**
6 months	11.059 ± 0.242	10.983 ± 0.243	8.892 ± 0.207	**8.840 ± 0.206**
12 months	12.239 ± 0.244	13.203 ± 0.273	10.648 ± 0.224	**10.593 ± 0.223**
24 months	9.119 ± 0.320	8.906 ± 0.319	**8.190 ± 0.306**	8.200 ± 0.307
60 months	12.500 ± 0.601	13.512 ± 0.601	11.330 ± 0.556	**11.197 ± 0.548**
120 months	11.883 ± 3.194	11.499 ± 2.941	9.389 ± 2.538	**9.341 ± 2.539**
$\sum$	11.289 ± 0.151	11.676 ± 0.158	9.658 ± 0.136	**9.607 ± 0.136**
$\sum$	10.589 ± 0.129	10.916 ± 0.135	9.123 ± 0.118	**9.082 ± 0.117**

as the best.

We have measured the prognostic usefulness of time-series characteristics for additional insights into meta-learners by calculating permutation-based variable importance for a random forest model built on a full M4-micro dataset. Variable importance was measured as a drop in mean squared error of the random forest regression model after permuting each feature, i.e., how strongly distorting ties between a feature in question and a target affects the model's accuracy. Feature sets had several features, so their corresponding individual importance estimates were averaged for a fair comparison and are reported in Fig. 3. From comparison it can be noticed that both transformations (*log* and *diff*) besides original time-series were useful, with the exception of *diff* transformation for *hctsa* and *catch22* features. Interestingly, the pre-processing of time-series by *log* transformation was highly useful for most feature sets. The best feature sets for meta-learning tasks were *entropy*, *tsfeats*, and *Hurst*. Among the least useful feature sets *anomaly* could be considered for removal.

Since we did not use the full M4 dataset (just 12.561% from the entire dataset of 100000 time-series) and tested much more forecasting horizons, it would be improper to compare SMAPE and MASE errors directly to M4 competition results. Nonetheless, the measure "% improvement of method over the benchmark" from Table 4 in [20] could lend itself for comparison with the winner of M4 competition - Smyl (Uber) reached an improvement of 9.4% in SMAPE and 7.7% in MASE over baseline Comb method. Our Weighted variant here demonstrated an improvement of 19.3% in SMAPE and 16.3% in MASE.

TABLE 6: Forecasting results according to MASE forecasting error. $\sum$ denotes results over all horizons, the best result for each row is formatted in italic-bold.

Horizon	Theta	Comb	Simple	Weighted
115 days	1.047 ± 0.050	1.056 ± 0.055	***0.963 ± 0.046***	0.964 ± 0.049
30 days	1.365 ± 0.076	1.362 ± 0.077	1.311 ± 0.081	***1.310 ± 0.081***
90 days	2.245 ± 0.106	2.239 ± 0.107	1.997 ± 0.102	***1.996 ± 0.101***
180 days	3.265 ± 0.200	3.334 ± 0.200	2.910 ± 0.202	***2.904 ± 0.202***
365 days	5.987 ± 0.309	6.132 ± 0.320	***5.053 ± 0.301***	5.060 ± 0.300
730 days	7.676 ± 0.521	7.798 ± 0.517	***6.747 ± 0.484***	6.761 ± 0.504
$\sum$	3.115 ± 0.098	3.161 ± 0.099	***2.750 ± 0.091***	2.751 ± 0.092
4 weeks	0.579 ± 0.077	0.592 ± 0.084	0.480 ± 0.070	***0.475 ± 0.068***
13 weeks	0.486 ± 0.052	0.476 ± 0.058	***0.426 ± 0.047***	0.432 ± 0.051
26 weeks	0.681 ± 0.324	0.680 ± 0.350	0.622 ± 0.347	***0.599 ± 0.340***
52 weeks	0.993 ± 0.299	0.967 ± 0.308	0.821 ± 0.282	***0.806 ± 0.286***
104 weeks	1.713 ± 0.731	1.689 ± 0.701	***1.564 ± 0.729***	1.587 ± 0.753
$\sum$	0.733 ± 0.103	0.728 ± 0.103	0.637 ± 0.101	***0.635 ± 0.103***
6 months	0.642 ± 0.011	0.637 ± 0.011	0.513 ± 0.010	***0.511 ± 0.010***
12 months	0.730 ± 0.013	0.750 ± 0.014	0.625 ± 0.012	***0.622 ± 0.012***
24 months	1.153 ± 0.029	1.118 ± 0.029	0.977 ± 0.027	***0.977 ± 0.027***
60 months	1.986 ± 0.099	2.428 ± 0.131	1.919 ± 0.221	***1.827 ± 0.130***
120 months	4.217 ± 1.153	4.025 ± 1.120	3.162 ± 0.982	***3.144 ± 0.989***
$\sum$	0.921 ± 0.015	0.968 ± 0.018	0.797 ± 0.026	***0.785 ± 0.017***
$\sum$	1.332 ± 0.023	1.378 ± 0.025	1.163 ± 0.028	***1.153 ± 0.023***

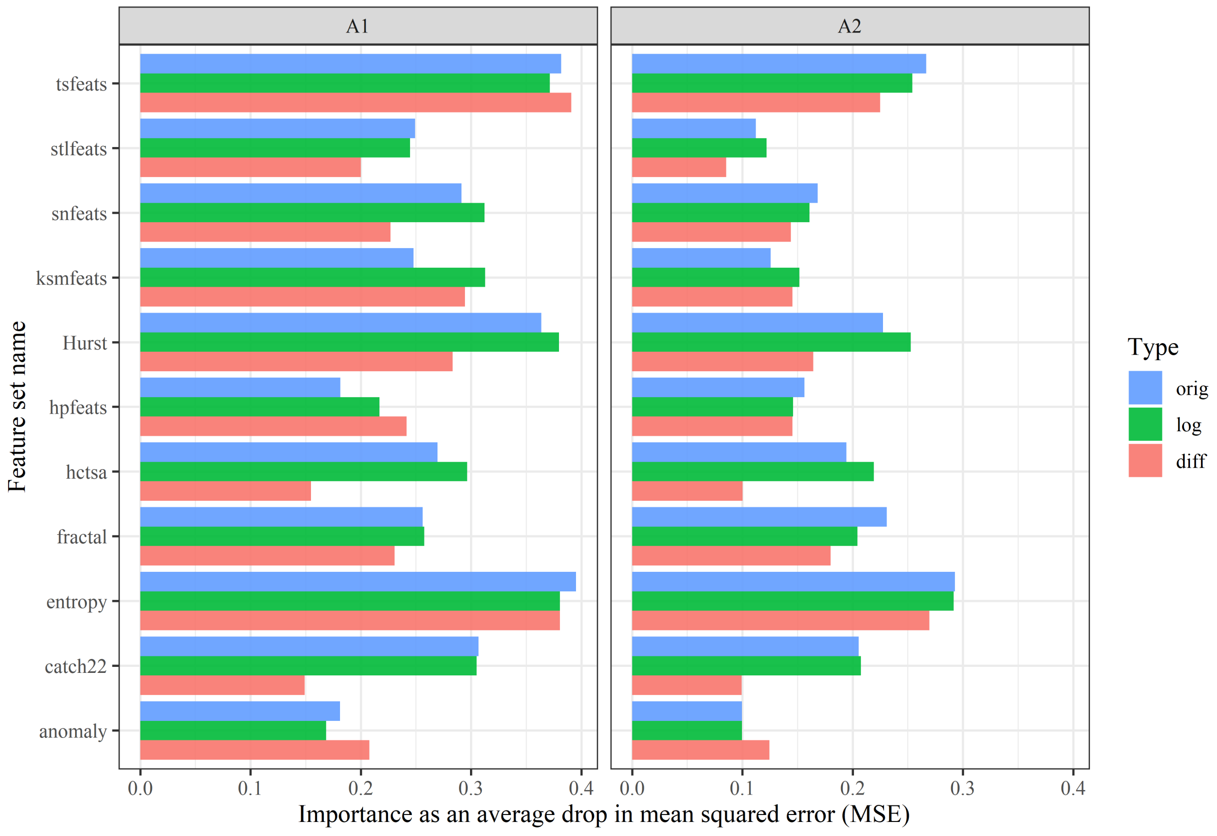

FIGURE 3: Permutation-based variable importance from meta-learner – a random forest regression model: A1 (*left*) and A2 (*right*) assistants.

VI. CONCLUSIONS

Extensive evaluation of the proposed meta-learning approach on micro-economic time-series from M4 competition demonstrated that meta-learning could outperform benchmark methods Theta and Comb. The best performance was achieved by pooling forecasts from assistant-recommended univariate time-series models using weighted average with weights corresponding to reciprocal rank. Lower forecasting errors were obtained using a weighted variant of forecasting ensemble over the Theta method: 9.21% versus 11.05% by SMAPE, 9.08% versus 10.59% by MAAPE, 1.15 versus 1.33 by MASE. The regression meta-learner model was more successful and had a better out-of-bag fit for A1 than for A2 assistant. All transformations were found to be useful for feature engineering and the most effective time-series characteristics for meta-learner were *entropy*, *tsfeats*, and *Hurst* feature sets. Considering a more extensive set of time-series data for meta-learning and exploring automatic feature engineering's usefulness using sequence-to-sequence auto-encoder would be exciting directions for further research.

DATA AVAILABILITY STATEMENT

This study was a re-analysis of existing data, which is publicly available in:

- R packages *M4comp* and *M4comp2018*
- Kaggle platform
- Zenodo platform:
  -- M4 Daily Dataset [55]
  -- M4 Weekly Dataset [56]
  -- M4 Monthly Dataset [57]

REFERENCES

[1] R. J. Hyndman, "Business forecasting methods," in *International Encyclopedia of Statistical Science*, M. Lovric, Ed. Springer, Berlin, Heidelberg, 2010, p. 185–187, doi: 10.1007/978-3-642-04898-2_156.

[2] J. G. Wacker and R. R. Lummus, "Sales forecasting for strategic resource planning," *International Journal of Operations & Production Management*, vol. 22, no. 9, p. 1014–1031, 2002, doi: 10.1108/01443570210440519.

[3] F. Petropoulos, S. Makridakis, V. Assimakopoulos, and K. Nikolopoulos, "'horses for courses' in demand forecasting," *European Journal of Operational Research*, vol. 237, no. 1, pp. 152–163, 2014, doi: 10.1016/j.ejor.2014.02.036.

[4] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The m4 competition: Results, findings, conclusion and way forward," *International Journal of Forecasting*, vol. 34, no. 4, pp. 802–808, 2018, doi: 10.1016/j.ijforecast.2018.06.001.

[5] P. Montero-Manso, G. Athanasopoulos, R. J. Hyndman, and T. S. Talagala, "FFORMA: Feature-based forecast model averaging," *International Journal of Forecasting*, vol. 36, no. 1, pp. 86–92, January 2020, doi: 10.1016/j.ijforecast.2019.02.011.

[6] S. Makridakis and M. Hibon, "The m3-competition: results, conclusions and implications," *International Journal of Forecasting*, vol. 16, no. 4, pp. 451–476, 2000, the M3- Competition, doi: 10.1016/S0169-2070(00)00057-1.

[7] R. T. Clemen, "Combining forecasts: A review and annotated bibliography," *International Journal of Forecasting*, vol. 5, no. 4, pp. 559 –583, 1989, doi: 10.1016/0169-2070(89)90012-5.

[8] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990, doi: 10.1109/34.58871.

[9] D. F. Hendry and M. P. Clements, "Pooling of forecasts," *The Econometrics Journal*, vol. 7, no. 1, pp. 1–31, 06 2004, doi: 10.1111/j.1368-423X.2004.00119.x.

[10] A. Timmermann, "Forecast combinations," in *Handbook of Economic Forecasting*, G. Elliott, C. W. J. Granger, and A. Timmermann, Eds. Elsevier, 2006, vol. 1, ch. 4, pp. 135–196, doi: 10.1016/S1574-0706(05)01004-9.

[11] S. Kolassa, "Combining exponential smoothing forecasts using Akaike weights," *International Journal of Forecasting*, vol. 27, no. 2, pp. 238–251, 2011, doi: 10.1016/j.ijforecast.2010.04.006.

[12] M. Oliveira and L. Torgo, "Ensembles for time series forecasting," in *Proceedings of Machine Learning Research*, D. Phung and H. Li, Eds., vol. 39. Nha Trang City, Vietnam: PMLR, November 26-28 2015, pp. 360–370. [Online]. Available: http://proceedings.mlr.press/v39/oliveira14.html

[13] W. Zang, P. Zhang, C. Zhou, and L. Guo, "Comparative study between incremental and ensemble learning on data streams: Case study," *Journal of Big Data*, vol. 1, no. 1, pp. 1–5, 2014, doi: 10.1186/2196-1115-1-5.

[14] H. Zou and Y. Yang, "Combining time series models for forecasting," *International Journal of Forecasting*, vol. 20, no. 1, pp. 69–84, 2004, doi: 10.1016/s0169-2070(03)00004-9.

[15] C. Lemke and B. Gabrys, "Meta-learning for time series forecasting and forecast combination," *Neurocomputing*, vol. 73, pp. 2006–2016, 2010, doi: 10.1016/j.neucom.2009.09.020.

[16] J. Smith and K. F. Wallis, "A simple explanation of the forecast combination puzzle," *Oxford Bulletin of Economics and Statistics*, vol. 71, no. 3, pp. 331–355, 2009, doi: 10.1111/j.1468-0084.2008.00541.x.

[17] M. Aiolfi and A. Timmermann, "Persistence in forecasting performance and conditional combination strategies," *Journal of Econometrics*, vol. 135, no. 1-2, p. 31–53, 2006, doi: 10.1016/j.jeconom.2005.07.015.

[18] A. Bauer, M. Züfle, J. Grohmann, N. Schmitt, N. Herbst, and S. Kounev, "An automated forecasting framework based on method recommendation for seasonal time series," in *Proceedings of the ACM/SPEC International Conference on Performance Engineering*. ACM, April 2020. doi: 10.1145/3358960.3379123.

[19] D. Reid, "A comparison of forecasting techniques on economic time series," *Forecasting in Action. Operational Research Society and the Society for Long Range Planning*, 1972.

[20] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The m4 competition: 100,000 time series and 61 forecasting methods," *International Journal of Forecasting*, vol. 36, no. 1, pp. 54–74, 2020, doi: 10.1016/j.ijforecast.2019.04.014.

[21] N. Meade, "Evidence for the selection of forecasting methods," *Journal of Forecasting*, vol. 19, no. 6, pp. 515–535, 2000, doi: 10.1002/1099-131X(200011)19:6<515::AID-FOR754>3.0.CO;2-7.

[22] F. Collopy and J. S. Armstrong, "Rule-based forecasting: Development and validation of an expert systems approach to combining time series extrapolations," *Management Science*, vol. 38, no. 10, pp. 1394–1414, 1992, doi: 10.1287/mnsc.38.10.1394.

[23] T. S. Talagala, R. J. Hyndman, and G. Athanasopoulos, "Meta-learning how to forecast time series," Department of Econometrics and Business Statistics, Monash Business School, Monash University, resreport 6/18, May 2018, working Paper Series.

[24] X. Wang, K. Smith-Miles, and R. Hyndman, "Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series," *Neurocomputing*, vol. 72, pp. 2581–2594, 2009, doi: 10.1016/j.neucom.2008.10.017.

[25] M. Zuefle, A. Bauer, V. Lesch, C. Krupitzer, N. Herbst, S. Kounev, and V. Curtef, "Autonomic forecasting method selection: Examination and ways ahead," in *2019 IEEE International Conference on Autonomic Computing (ICAC)*. IEEE, June 2019. doi: 10.1109/icac.2019.00028.

[26] L. Rokach, "Decomposition methodology for classification tasks: a meta decomposer framework," *Pattern Analysis and Applications*, vol. 9, p. 257–271, 2006, doi: 10.1007/s10044-006-0041-y.

[27] A. R. Ali, B. Gabrys, and M. Budka, "Cross-domain meta-learning for time-series forecasting," *Procedia Computer Science*, vol. 126, pp. 9–18, 2018, doi: 10.1016/j.procs.2018.07.204.

[28] P. Montero-Manso, G. Athanasopoulos, R. J. Hyndman, and T. S. Talagala, "Fforma: Feature-based forecast model averaging," Department of Econometrics and Business Statistics, Monash Business School, Monash University, resreport 19/18, November 2018, working Paper Series.

[29] S. Ma and R. Fildes, "Retail sales forecasting with meta-learning," *European Journal of Operational Research*, vol. 288, no. 1, pp. 111–128, 2021, doi: 10.1016/j.ejor.2020.05.038.

[30] C. H. Lubba, S. S. Sethi, P. Knaute, S. R. Schultz, B. D. Fulcher, and N. S. Jones, "catch22: CAnonical time-series CHaracteristics," *Data Mining and Knowledge Discovery*, vol. 33, no. 6, pp. 1821–1852, aug 2019, doi: 10.1007/s10618-019-00647-x.

[31] R. Hyndman, E. Wang, Y. Kang, T. Talagala, and Y. Yang, "tsfeatures: Time series feature extraction," 2019. [Online]. Available: https://github.com/robjhyndman/tsfeatures/

[32] N. Kourentzes, "On intermittent demand model optimisation and selection," *International Journal of Production Economics*, vol. 156, pp. 180–190, 2014, doi: 10.1016/j.ijpe.2014.06.007.

[33] R. Hyndman and Y. Khandakar, "Automatic time series forecasting: The forecast package for r," *Journal of Statistical Software, Articles*, vol. 27, no. 3, pp. 1–22, 2008, doi: 10.18637/jss.v027.i03.

[34] R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose, "A state space framework for automatic forecasting using exponential smoothing methods," *International Journal of Forecasting*, vol. 18, no. 3, pp. 439–454, 2002, doi: 10.1016/S0169-2070(01)00110-8.

[35] R. Hyndman, A. Koehler, K. Ord, and R. Snyder, *Forecasting with Exponential Smoothing*. Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-71918-2.

[36] P. R. Winters, "Forecasting sales by exponentially weighted moving averages," *Management Science*, vol. 6, no. 3, pp. 324–342, 1960, doi: 10.1287/mnsc.6.3.324.

[37] V. Assimakopoulos and K. Nikolopoulos, "The theta model: a decomposition approach to forecasting," *International Journal of Forecasting*, vol. 16, no. 4, pp. 521–530, 2000, doi: 10.1016/S0169-2070(00)00066-2.

[38] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "STL: A seasonal-trend decomposition procedure based on loess (with discussion)," *Journal of Official Statistics*, vol. 6, pp. 3–73, 1990.

[39] J. Durbin and S. J. Koopman, *Time Series Analysis by State Space Methods*. Oxford University Press, may 2012. doi: 10.1093/acprof:oso/9780199641178.001.0001.

[40] A. M. D. Livera, R. J. Hyndman, and R. D. Snyder, "Forecasting time series with complex seasonal patterns using exponential smoothing," *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1513–1527, 2011, doi: 10.1198/jasa.2011.tm09771.

[41] S. J. Taylor and B. Letham, "Forecasting at scale," *PeerJ Preprints*, vol. 5:e3190v2, sep 2017, doi: 10.7287/peerj.preprints.3190v2.

[42] R. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 2nd ed. OTexts, 2018, ch. Neural network models, p. 11.3. [Online]. Available: https://otexts.com/fpp2/nnetar.html#nnetar

[43] P. Ellis, "Timeseries forecasting using extreme gradient boosting," Nov. 2016. [Online]. Available: http://freerangestats.info/blog/2016/11/06/forecastxgb

[44] G. Athanasopoulos, R. J. Hyndman, N. Kourentzes, and F. Petropoulos, "Forecasting with temporal hierarchies," *European Journal of Operational Research*, vol. 262, no. 1, pp. 60–74, 2017, doi: 10.1016/j.ejor.2017.02.046.

[45] S. Kim and H. Kim, "A new metric of absolute percentage error for intermittent demand forecasts," *International Journal of Forecasting*, vol. 32, no. 3, pp. 669–679, 2016, doi: 10.1016/j.ijforecast.2015.12.003.

[46] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006, doi: 10.1016/j.ijforecast.2006.03.001.

[47] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, October 2001, doi: 10.1023/A:1010933404324.

[48] M. Wright and A. Ziegler, "ranger: A fast implementation of random forests for high dimensional data in c++ and r," *Journal of Statistical Software, Articles*, vol. 77, no. 1, pp. 1–17, 2017, doi: 10.18637/jss.v077.i01.

[49] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, "How many trees in a random forest?" in *Machine Learning and Data Mining in Pattern Recognition*. Springer Berlin Heidelberg, 2012, pp. 154–168, doi: 10.1007/978-3-642-31537-4_13.

[50] P. Probst and A.-L. Boulesteix, "To tune or not to tune the number of trees in random forest," *Journal of Machine Learning Research*, vol. 18, no. 1, p. 6673–6690, Jan. 2017. [Online]. Available: https://www.jmlr.org/papers/v18/17-269.html

[51] E. Spiliotis, A. Kouloumos, V. Assimakopoulos, and S. Makridakis, "Are forecasting competitions data representative of the reality?" *International Journal of Forecasting*, vol. 36, no. 1, pp. 37–53, 2020, doi: 10.1016/j.ijforecast.2018.12.007.

[52] J.-C. Deville and Y. Tillé, "Efficient balanced sampling: The cube method," *Biometrika*, vol. 91, no. 4, pp. 893–912, 12 2004, doi: 10.1093/biomet/91.4.893.

[53] L. van der Maaten, "Accelerating t-SNE using tree-based algorithms," *Journal of Machine Learning Research*, vol. 15, no. 93, pp. 3221–3245, 2014. [Online]. Available: http://jmlr.org/papers/v15/vandermaaten14a.html

[54] V. Cerqueira, L. Torgo, and I. Mozetič, "Evaluating time series forecasting models: an empirical study on performance estimation methods," *Machine Learning*, 2020, doi: 10.1007/s10994-020-05910-7.

[55] R. Godahewa, C. Bergmeir, and G. Webb, "M4 daily dataset," 2020," doi: 10.5281/ZENODO.3898361.

[56] ——, "M4 weekly dataset," 2020," doi: 10.5281/ZENODO.3898376.

[57] ——, "M4 monthly dataset," 2020," doi: 10.5281/ZENODO.3898380.

EVALDAS VAICIUKYNAS received the M.Sc. degree in Informatics Engineering and the Ph.D. degree in Informatics from Kaunas University of Technology (KTU, Kaunas, Lithuania), in 2006 and 2013, respectively. He had a short-term post-doc internship at Halmstad University (Halmstad, Sweden) in 2016.

Currently, he lectures as a Professor in the Department of Information Systems at KTU. Lecturing experience also included bachelor and master courses at ISM University of Management and Economics (Vilnius, Lithuania) during 2012 – 2018. Research interests encompass statistical data analysis, signal and image processing, machine learning, time series forecasting and business intelligence.

Dr. Vaiciukynas also serves as a reviewer in several highly ranked scientific journals and participates in various research projects.

PAULIUS DANENAS obtained a PhD degree in Informatics from Vilnius University (Vilnius, Lithuania) in 2013.

Currently, he is a Scientific Researcher at the Centre of Information Systems Design Technologies, Kaunas University of Technology (Kaunas, Lithuania). Research interests cover topics in artificial intelligence, machine learning, natural language processing, data science, software engineering and model-driven development, as well as decision support systems (including business and financial domains).

Dr. Danenas is a co-author of multiple papers in highly-rated academic journals and proceedings of international conferences. He has served as a reviewer for a number of highly-ranked academic journals, including the ones published by Springer, Elsevier, Wiley, Taylor & Francis, IEEE and others.

VILIUS KONTRIMAS received M.Sc. in Informatics in 2003 from Kaunas University of Technology (KTU, Kaunas, Lithuania), M.Sc. in Management and Administration in 2005 from Vilnius Gediminas Technical University (VGTU, Vilnius, Lithuania), M.Sc. in Civil Engineering in 2008 from VGTU (Vilnius, Lithuania) and Ph.D. in Informatics in 2009 from KTU (Kaunas, Lithuania), expected Master of Management in 2021 from ISM University of Management and Economics (Vilnius, Lithuania) and BI Norwegian Business School (Oslo, Norway).

Currently, he is CEO and founder of technology company, chief scientist at leading enterprise resource planning company in Lithuania, board member at ISM University of Management and Economics. Previously Dr. Kontrimas has held C level positions in his businesses, ISM University of Management and Economics and leading DIY retailer in Baltic countries. Vilius research interests cover various topics in artificial intelligence and information systems.

RIMANTAS BUTLERIS , Ph.D., is the Head of the Center of Information Systems Design Technologies and a Full Professor at the Department of Information Systems, Kaunas University of Technology (Kaunas, Lithuania).

His main areas of research is requirements engineering, semantic technologies and business rules modeling. During his career, he has authored or co-authored more than one hundred seventy research papers and participated in more than twenty national and international research projects; in many of those he was a leading researcher.

• • •