iqast Intelligent Forecasting

Forecasting with Artificial Neural Networks for sparse Industry Data - an empirical evaluation

aric

IEEE Computational Intelligence Society

The International Neural Network Society (INNS)

Logistik-Initiative Hamburg

BVL Bundesvereinigung Logistik

GOR

IIF International Institute of Forecasters

AI.HAMBURG

Dr. Sven F. Crone
Assistant Professor Lancaster University
Research Centre for Marketing Analytics & Forecasting
CEO & Founder of iqast

Agenda

1. BDF project: next gen AI models
2. BDF Empirical Evaluation
    1. Experiment Design 47 algos / 600 series
    2. Parameter Sensitivity of DeepNet algorithms
    3. Other Landscapes (Tesa, Würth, Henkel, Hapag.Lloyd, Bosch, etc).
    4. Possible Root Causes
3. Conclusion & Discussion

- the elephant in the room?

Deep NeuralNets are not accurate on monthly sc industry data … yet (but it's 1/6th of GGDP)!

- But … you forgot to compare to the latest fuzzy particle swarm optimized TFT!
- But … you didn't tune parameters of the DeepNets like for your MLPs!"
- But … you didn't use my error metric!

> There are two kinds of fools: one says, "This is old, therefore it is good"; the other says, "This is new, therefore it is better."
>
> *William Ralph Inge (1931)*

KEY FACTS

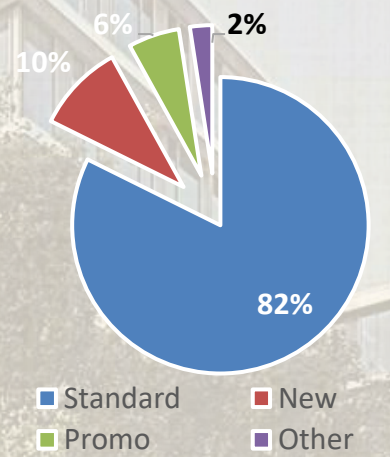
170
Global Affiliates

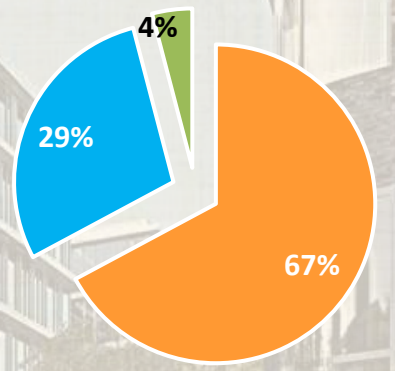
DAX
Listed

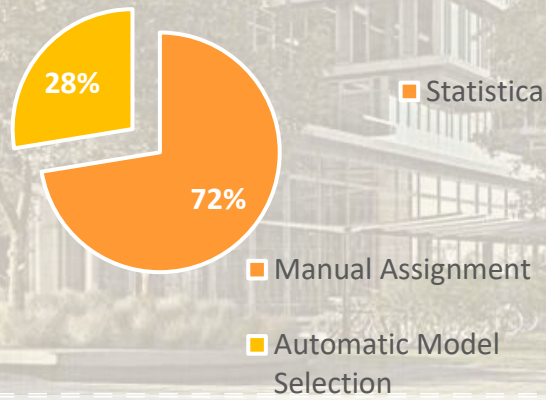
8,799
Mio. Euro Turnover 2022

HOW DO WE PLAN TODAY

HOW DO TYPICAL TIME SERIES IN BDF LOOK LIKE







Active FC Items ~ 50.000

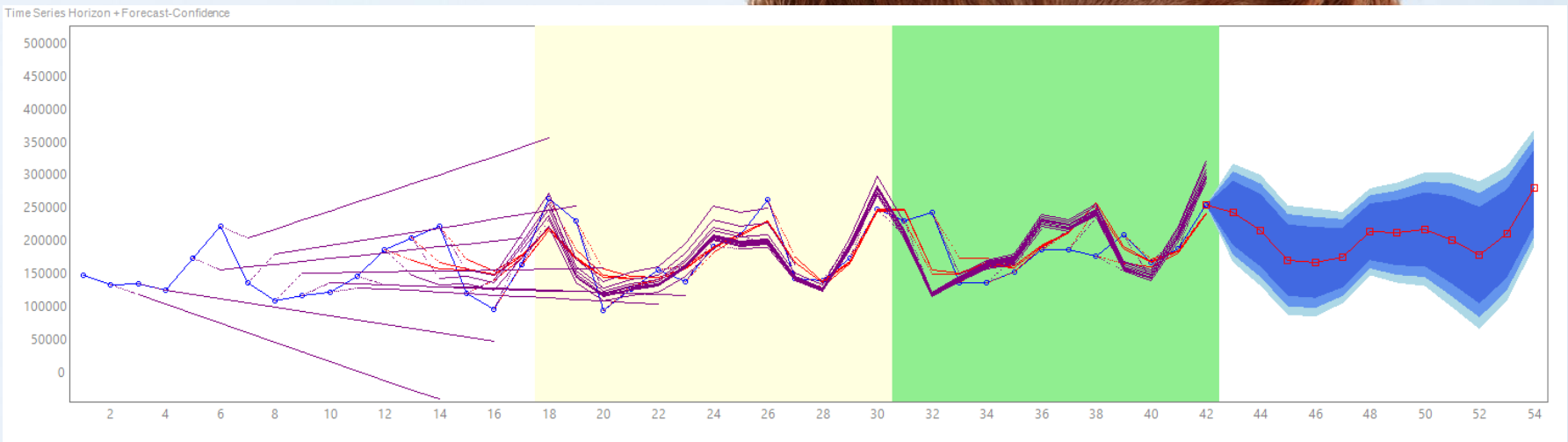
No. of Planners ~ 150

Avg. FC Items per planner ~ 350

Max. FC Items per planner ~ 1.200

AI in FMCG Forecasting @ Beiersdorf

iqast Intelligent Forecasting



[Project Costs:
Software €35k
Installation 5pd = 5k
Modelling 5pd = €5k
Training 5pd = €5k
= €50k external
+ 20pd internal
= ROI in months

Median sMAPE	Test
Seasonal Linear Regression (35B)	18.20
SLR + Planner Judgment	12.08
MLP AR + Selection	9.00
Improvement	-9.02
Improvement in %	-49.8%

NNET achieves super-human forecast accuracy

iqast SAP APO

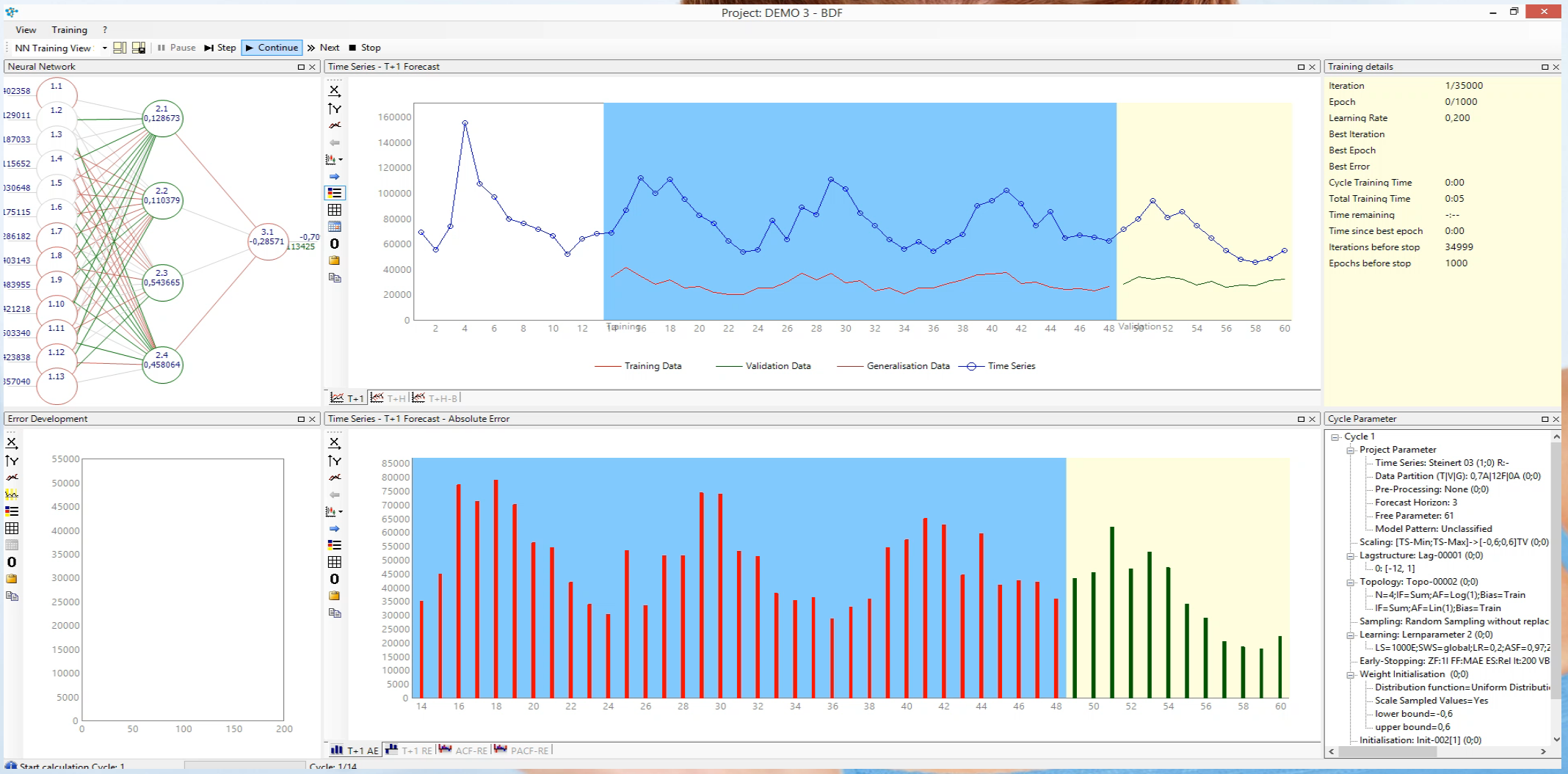
sMAPE test	SAP error	iqast error	Δ error	Δ error %	# items
Canada	40,7	33,8	-6,9	-16,9%	47
Germany	55,4	51,7	-3,7	-6.8%	155
France	43,7	42,6	-1,2	-2.7%	262
Greece	50,9	49,4	-1,7	-3.3%	196
Italy	42,7	39,9	-2,8	-6.5%	175
Netherlands	41,0	38,9	-2,1	-5.1%	154
Poland	55,2	47,1	-8,1	-14.7%	78
South Africa	37,3	35,9	-1,4	-3.7%	36
Average				-7.5%	

7.5% improvement overall, but should be used selectively by local demand planner's choice

→ Fully automatic, self-training AI forecasts productive since 2014 (9 years ago)!

AI in FMCG Forecasting @ BDF Beiersdorf

iqast Intelligent Forecasting

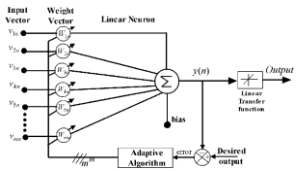


Un-Impressive? It's 10 years old! (IBF Dubai 2015?) → BDF wanted next generation of AI

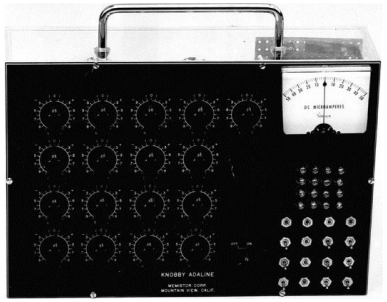
Evolution of Neural Net algorithms

1950s
0th Generation AI

T vs J problem



Adaline, Perceptron

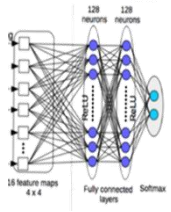


→ inventing the field of Nnets, military applications, e.g. submarine sonar detection

Science in Action with Widrow
https://www.youtube.com/watch?v=IEFRtz68m-8

1990s
1st Generation AI

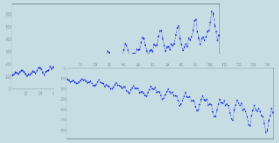
OCR

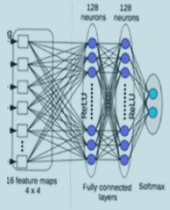


Multilayer Perceptrons MLP
Radial Basis Function Nets
Generalised Regression Nets
Elman & Jordan Nets

→ inventing new architectures
1st applications on time series (in finance, of course!)

2010
2nd Generation AI





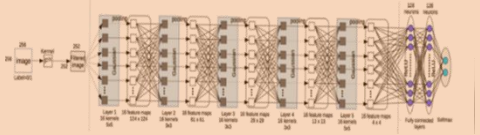
Ensembles of MLPs
Auto feature generation & feature selection
- Seasonality, Trends
- bank holidays, outliers

→Customising MLP NNs tailored to individual data using manual & automatic input feature engineering
→focus on Methodology!

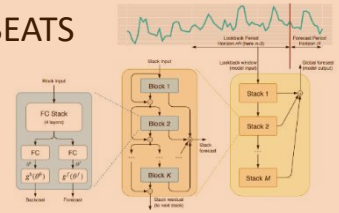
2015
3rd Generation AI

images
speech
text

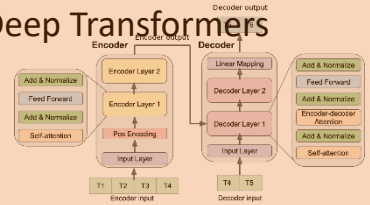
Convolutional Neural Nets



N-BEATS



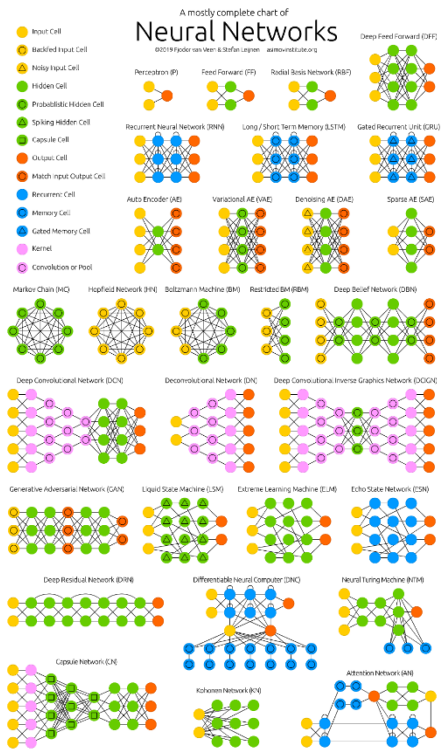
Deep Transformers



→ inventing new architectures with automatic feature engineering (e.g. for vision, different for speech etc.)
→ still ***narrow*** not universal AI

202?
4th Generation

Kolmogorov-Arnold?
Global zero-shot?

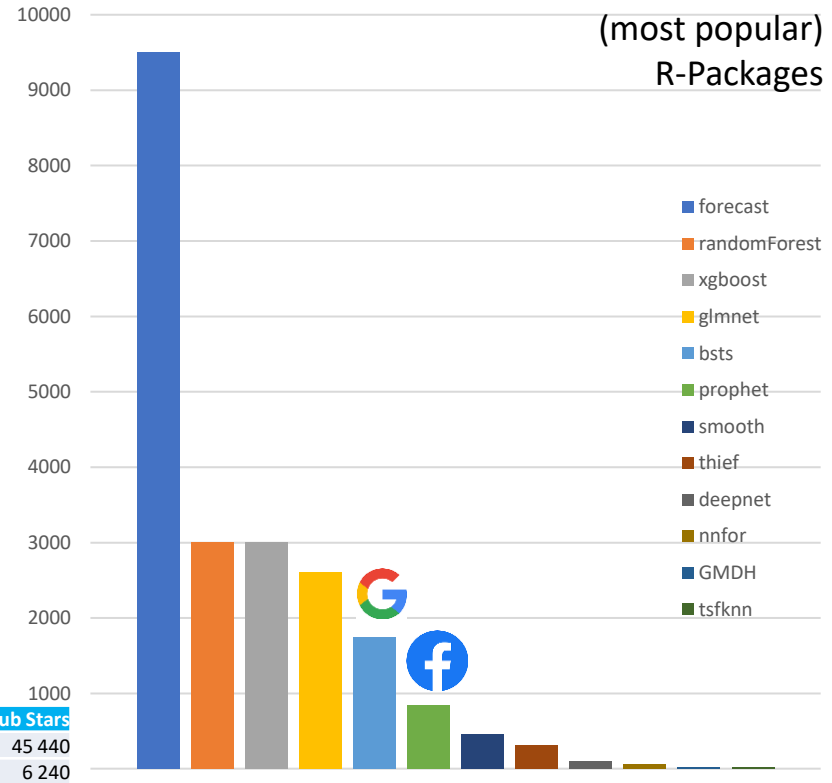


→ Benefits & costs of 2nd vs 3rd vs 4th generation Neural Networks?

the 2nd Challenge
More Forecast Algorithms >> time series data

Commercial software packages

Function	Description	Package	Author	Downloads
ets()	Exponential Smoothing	forecast	Rob J. Hyndman/CRAN	11.000
stlm()	Forecasting with STL Decomposition	forecast	Rob J. Hyndman/CRAN	11.000
tbats()	TBATS model	forecast	Rob J. Hyndman/CRAN	11.000
thetaf()	Theta Method	forecast	Rob J. Hyndman/CRAN	11.000
baggedModel()	Forecasting using a bagged model	forecast	Rob J. Hyndman/CRAN	11.000
splinef()	Cubic Spline Forecast	forecast	Rob J. Hyndman/CRAN	11.000
auto.arima()	Autoregressive Integraged Moving Average	forecast	Rob J. Hyndman/CRAN	11.000
xgbar()	Extreme Gradient Boosting	forecastxgb	Peter Ellis/GitHub (xgboost)	4.000
cv.glmnet()	Cross-Validation for Lasso/Elastic-Net Regularized Generalized Linear Model	glmnet	Jerome Friedman/CRAN	3.500
randomForest()	Random Forest for Regression	randomForest	Andy Liaw and M. Wiener/CRAN	3.200
bsts()	Bayesian Structural Time Series	bsts	Steven L. Scott/CRAN	1.600
prophet()	Additive Model with non-linear trend, seasonality and holiday effects	prophet	Sean Taylor /CRAN	500
es()	Exponential Smoothing	smooth	Ivan Svetunkov/CRAN	400
auto.ssarima()	Autoregressive Integraged Moving Average	smooth	Ivan Svetunkov/CRAN	400
thief()	Temporal Hierachical Forecasting	thief	Rob J. Hyndman/CRAN	300
nn.train	Deep Multilayer Perceptron	deepnet	Xiao Rong/CRAN	100
SaeDnnTrain()	Deep MLP with weights initialized by Stacked AutoEncoder	deepnet	Xiao Rong/CRAN	100
nnetar()	MLP	forecast	Rob J. Hyndman/CRAN	100
DbnDnnTrain	Deep neural network with weights initialized by Deep Belief Net	deepnet	Xiao Rong/CRAN	100
mlp()	MLP	nnfor	Nikolaos Kourentzes/CRAN	75
knn_forecasting	K-Nearest-Neighbor	tsfknn	Francisco Martinez/CRAN	30
fcast	Short Term Forecasting via GMDH-Type ANN	GMDH	Osman Dag/CRAN	30
elm()	Extreme Learning Machine feedforward Neural Network	nnfor	Nikolaos Kourentzes/CRAN	75

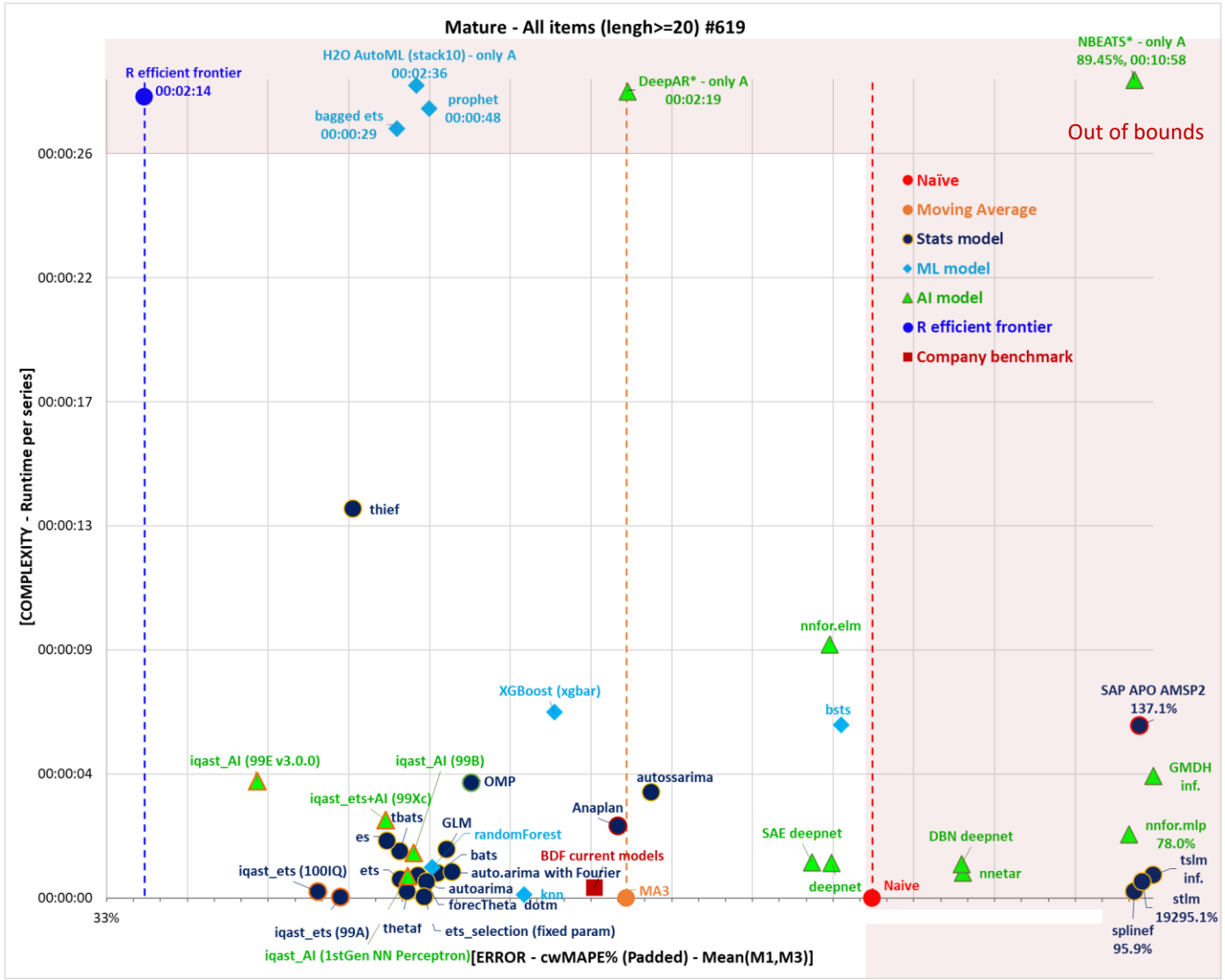


Average Daily Downloads (2020-09-30 until 2021-09-30)
no regressors / standard settings used (as per documentation)

Project Name	Description	daily download	GitHub Stars
glm-sklearn	scikit-learn compatible wrapper around the GLM module in statsmodels	151.295	45 440
statsmodels	Contains a submodule for classical time series models and hypothesis tests	132.710	6 240
Arrow	A sensible, human-friendly approach to creating, manipulating, formatting and converting dates, times, and timestamps	107.706	7 360
tensorflow_probab.	Bayesian Structural Time Series model in Tensorflow Probability	22.544	3 280
pyramid	port of R's auto.arima method to Python	12.609	3 540
Featuretools	Time series feature extraction, with possible conditionality on other variables	8.944	5 500
tsfresh	Extracts and filters features from time series, allowing supervised classificators and regressor to be applied to time series data	8.128	5 570
HMMLearn	Hidden Markov Models with scikit-learn compatible API	5.939	2 230
ruptures	Provides methods to find change points in time series such as shifts in the mean or scale of the signal	2.943	681
tslearn	Direct time series classifiers and regressors	2.731	1 570
sktime	A scikit-learn compatible library for learning with time series data including time series classification/regression forecasting	2.258	3 860
ta	Calculate technical indicators for financial time series	2.134	2 050
stumpy	Calculates matrix profile for time series subsequence all-pairs-similarity-search	2.012	1 760
prophet	Time series forecasting for time series data that has multiple seasonality with linear or non-linear growth	1.893	12 560
ta-lib	Calculate technical indicators for financial time series (python wrapper around TA-Lib)	1.738	5 040
Traces	A library for unevenly-spaced time series analysis	1.115	428
GluonTS	Gluon Time Series (GluonTS) is the Gluon toolkit for probabilistic time series modeling, focusing on deep learning-based models.	1.085	1 850
PyFlux	Classical time series forecasting models	662	1 870

BDF Austria Forecasting Landscape
Mean(M1,M3) – runtime per ts

Data partition: Tra-Rest/Val-12/Gen-6



*DeepAR and NBEATS use modeltime package default setting

- Stats methods (es, ets, arima) most perform competitively, but few (stlm, tslm) are poor
- thief (ets) outperforms most stats models
- APS systems using ets show large variety in errors, with SAP APO model selection poorest

- ML methods do not outperform stats models, random forrest >> xgboost, knn >> bsts
- BaggedETS shows no improvement on ets, but at long runtimes, as prophet & H2O

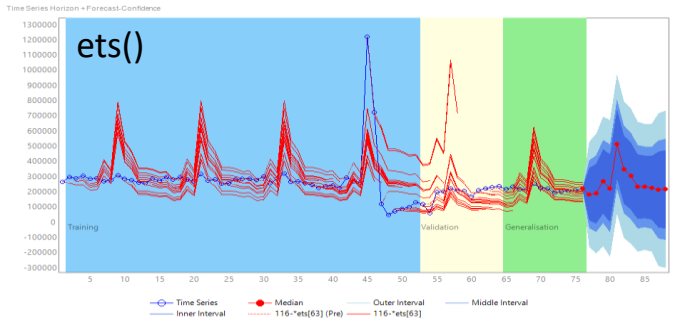
- AI methods in 2nd (nnfor, nnetar) & 3rd gen (DBD deepnet etc) do not outperform the Naïve, none outperform MOVAV3, incl NBEATS
- iqast customized 2nd gen MLPs outperform all AI, ML and Stats contenders

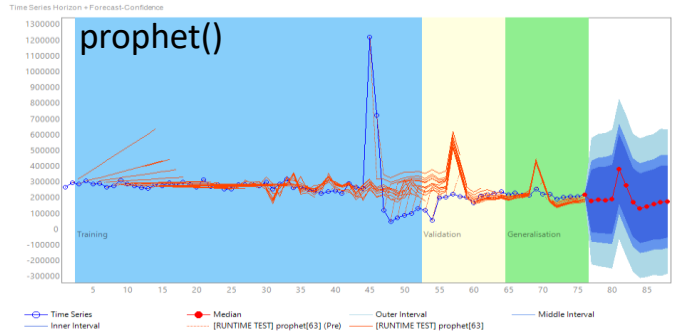
Robust Experimental design

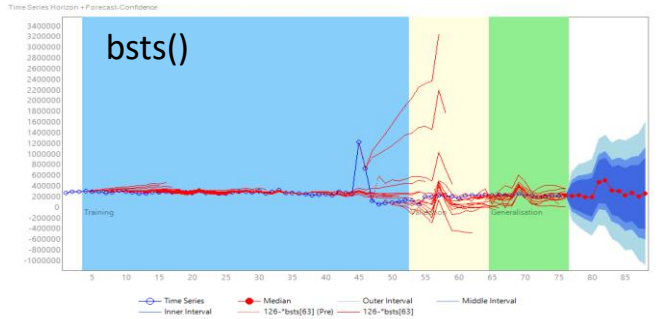
- 619 mature time series (<18 obs) across ABC XYZ excl. new and intermittent data
- 6, 9 and 12 test set with multiple time origins, fixed horizons (depending on company project 12 or 6 months) – here 12 shown
- company forecast horizon M0, M1, M2, mostly M3 (i.e. t+1, … t+4)
- Multiple error metrics sMAPE, MASE, RMSsE, wMAPE & cwMAPE
- Padding of missing runs with Naïve 0.4% - 23% (note: enhances fcst accuracy of poor methods if 30% is replaced by Naïve)
- Evaluation with & without parameter re-estimation at each origin

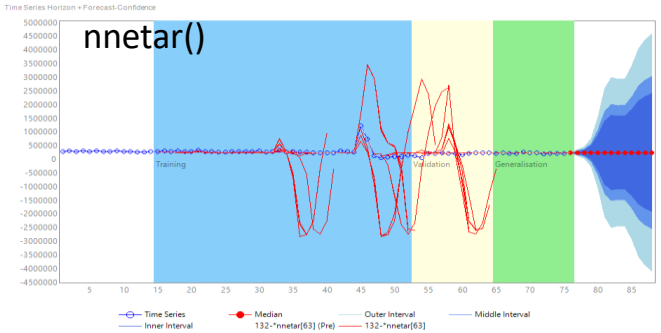
Statistics
Machine Learning ML
Artificial Intelligence AI

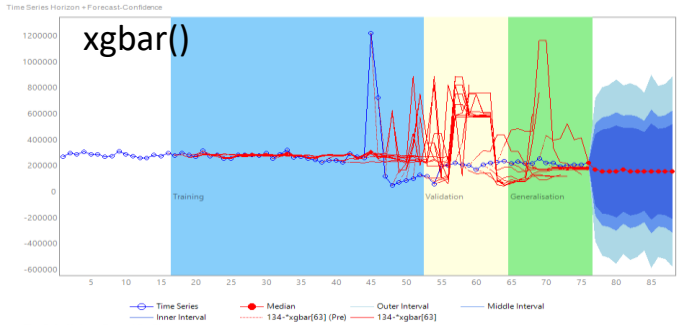
Monthly Data Set
Example #2 Level

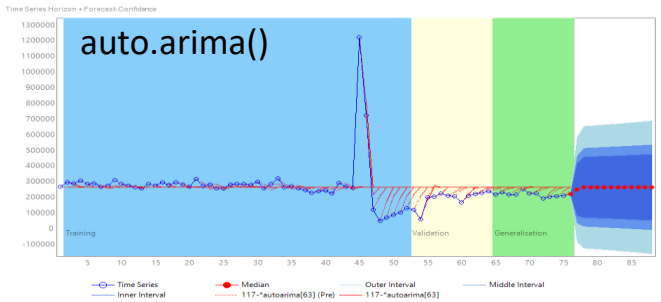


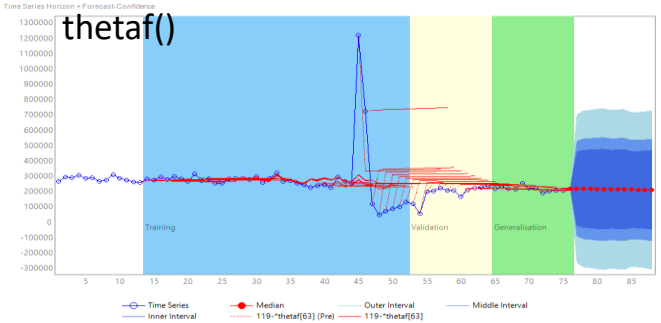




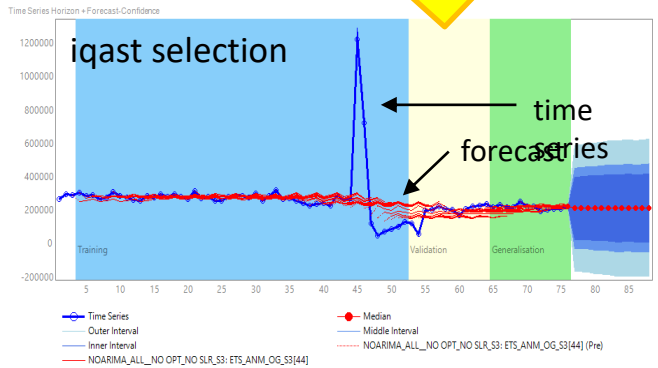








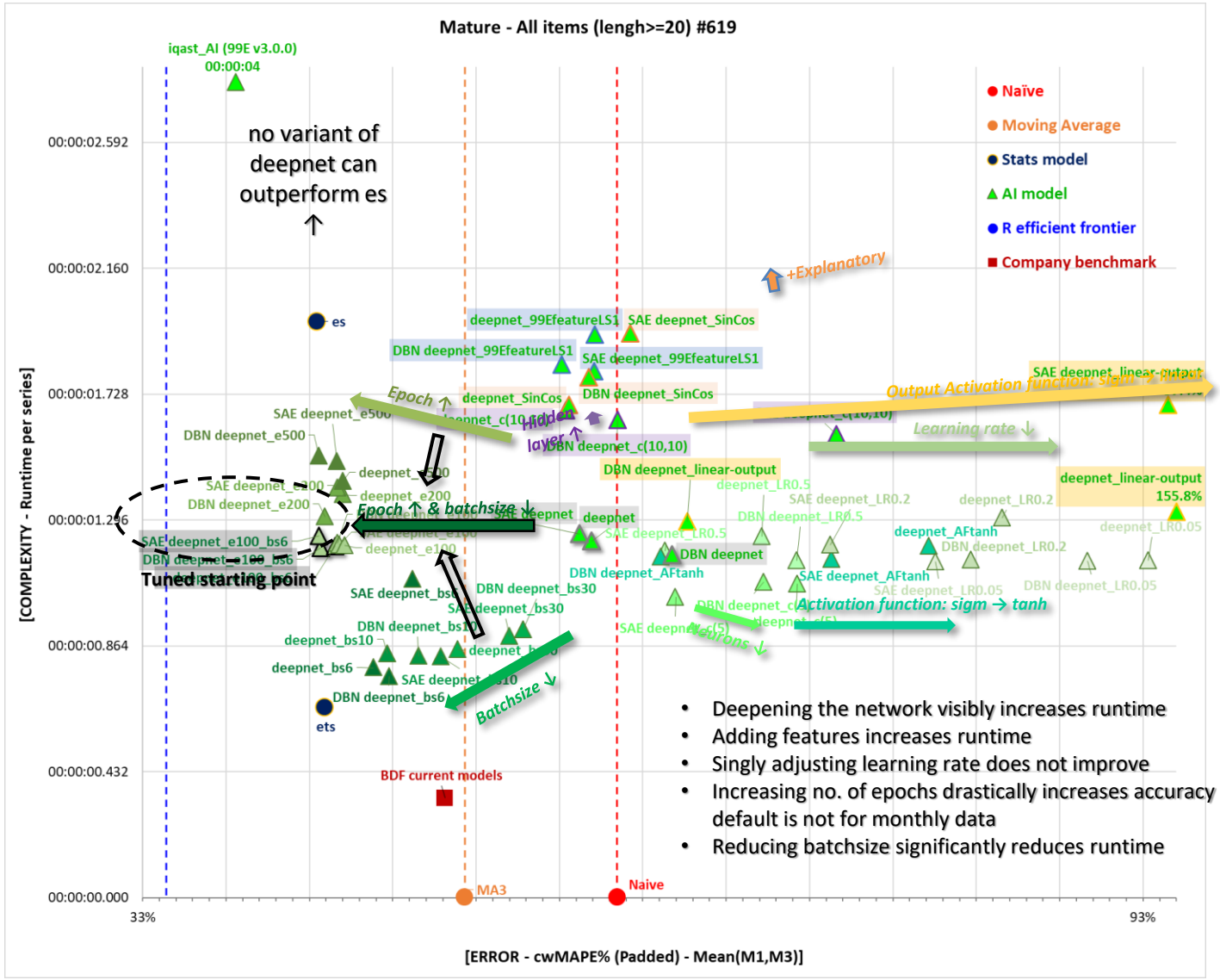
Adding Outlier Features leads to more stable forecast



→ Many AIML methods show very poor forecasting behavior …

→ confused by single outlier on simple series !?! Adopt methodology!

BDF Austria Landscape
deepnet models experiment

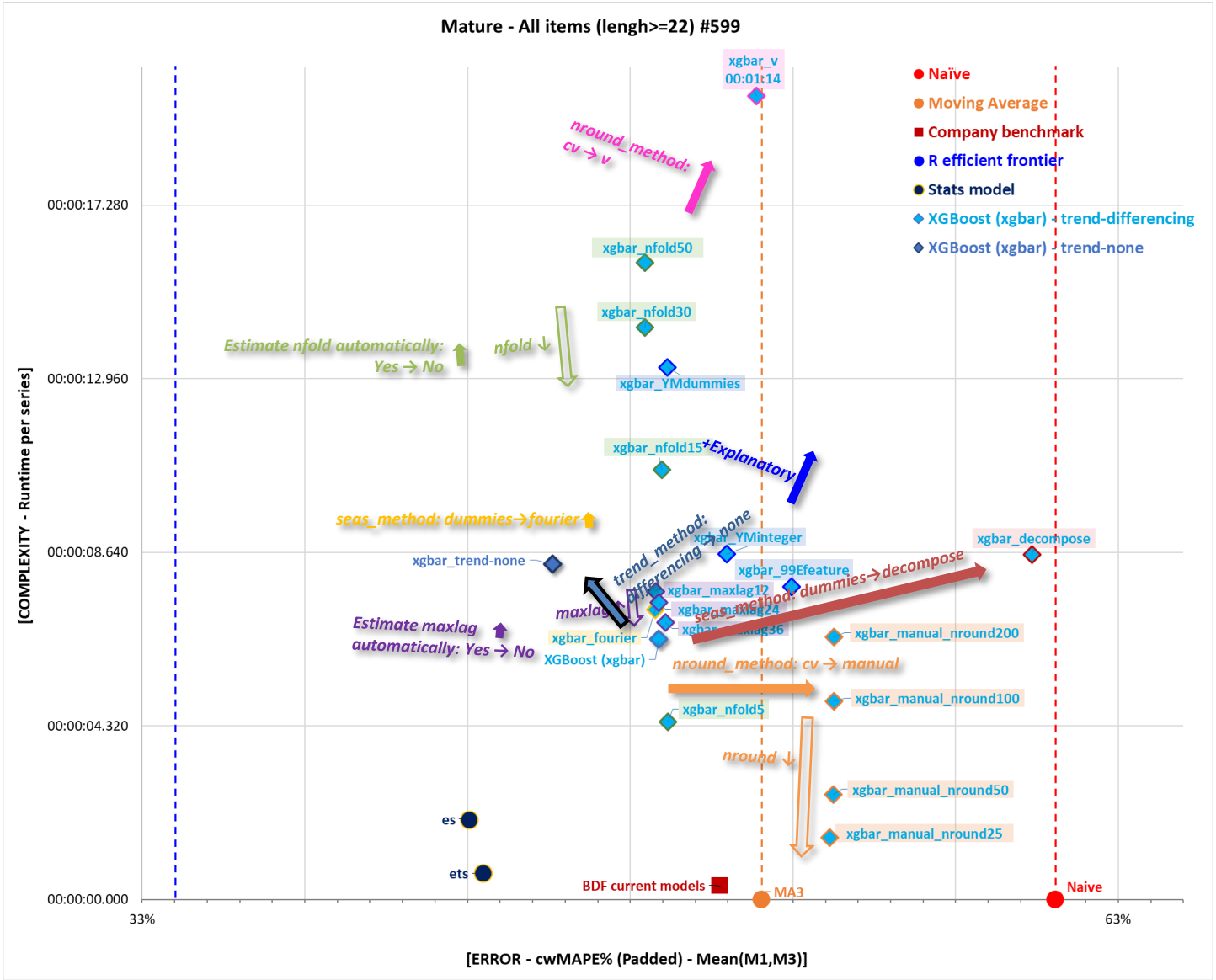


Postfix	Parameter changes
(None)	Default
_LR0.5	Learning rate: 0.8 → 0.5
_LR0.2	Learning rate: 0.8 → 0.2
_LR0.05	Learning rate: 0.8 → 0.05
_linear-output	Output activation function: sigm → linear (for deepnet and DBN)
_sigm-output	Output activation function: linear → sigm (for SAE)
_Aftanh	Activation function: sigm → tanh
_c(5)	No. of neurons: 10 → 5
_c(10,10)	No. of hidden layer: 1 → 2
_SinCos	Explanatory series added: SinCos12_6_3
_99EfeatureLS1	Explanatory series added: iqast features & lag structure
_bs30	bs: 100 → 30
_bs10	bs: 100 → 10
_bs6	bs: 100 → 6
_e100	Epoch: 3 → 100
_e200	Epoch: 3 → 200
_e500	Epoch: 3 → 500
_e100_bs6	Epoch: 3 → 100 & bs: 100 → 6

	Mean(M1,M3) cost wMAPE			Runtime per series		
Postfix	deepnet	SAE deepnet	DBN deepnet	deepnet	SAE deepnet	DBN deepnet
(None)				00:00:01.223	00:00:01.247	00:00:01.176
_LR0.5				00:00:01.241	00:00:01.191	00:00:01.157
_LR0.2				00:00:01.304	00:00:01.210	00:00:01.160
_LR0.05				00:00:01.158	00:00:01.153	00:00:01.155
_c(5)				00:00:01.078	00:00:01.031	00:00:01.082
_c(10,10)				00:00:01.674	00:00:01.591	00:00:01.640
_Aftanh				00:00:01.207	00:00:01.162	00:00:01.170
_e100				00:00:01.208	00:00:01.212	00:00:01.216
_e200				00:00:01.384	00:00:01.407	00:00:01.309
_e500				00:00:01.430	00:00:01.498	00:00:01.517
_bs30				00:00:00.851	00:00:00.897	00:00:00.919
_bs10				00:00:00.838	00:00:00.827	00:00:00.829
_bs6				00:00:00.790	00:00:01.094	00:00:00.759
_SinCos				00:00:01.691	00:00:01.937	00:00:01.787
_99EfeatureLS1				00:00:01.932	00:00:01.806	00:00:01.830
_linear-output				00:00:01.325	00:00:01.691	00:00:01.292
_e100_bs6				00:00:01.207	00:00:01.242	00:00:01.200

As default settings are not based on monthly data, results of varying parameters from default are not representative
→ tune for a better starting point for more meaningful orthogonal experiment – increase epoch to 100 and reduce batchsize to 6
→ only experiment further on DBN deepnet considering similar performance of all 3 deepnets

BDF Austria Landscape
xgboost stage 1 experiment



xgbar default settings: xreg=NULL, Estimate_maxlag=Yes, nrounds=100, nrounds_method=cv, Estimate_nfold=Yes, nfold=30, lambda=1, seas_method=dummies, Estimate_K=Yes, K=5 (if seas_method=fourier), trend_method=differencing

Parameters	No. of options
xreg	2
Estimate maxlag	2
Maxlag	/
Nrounds	/
nrounds_method	3
Estimate n fold	2
Nfold	/
Lambda	/
seas_method	3
Estimate K	2
K	/
trend_method	2
Total metaparameter combinations	**288**

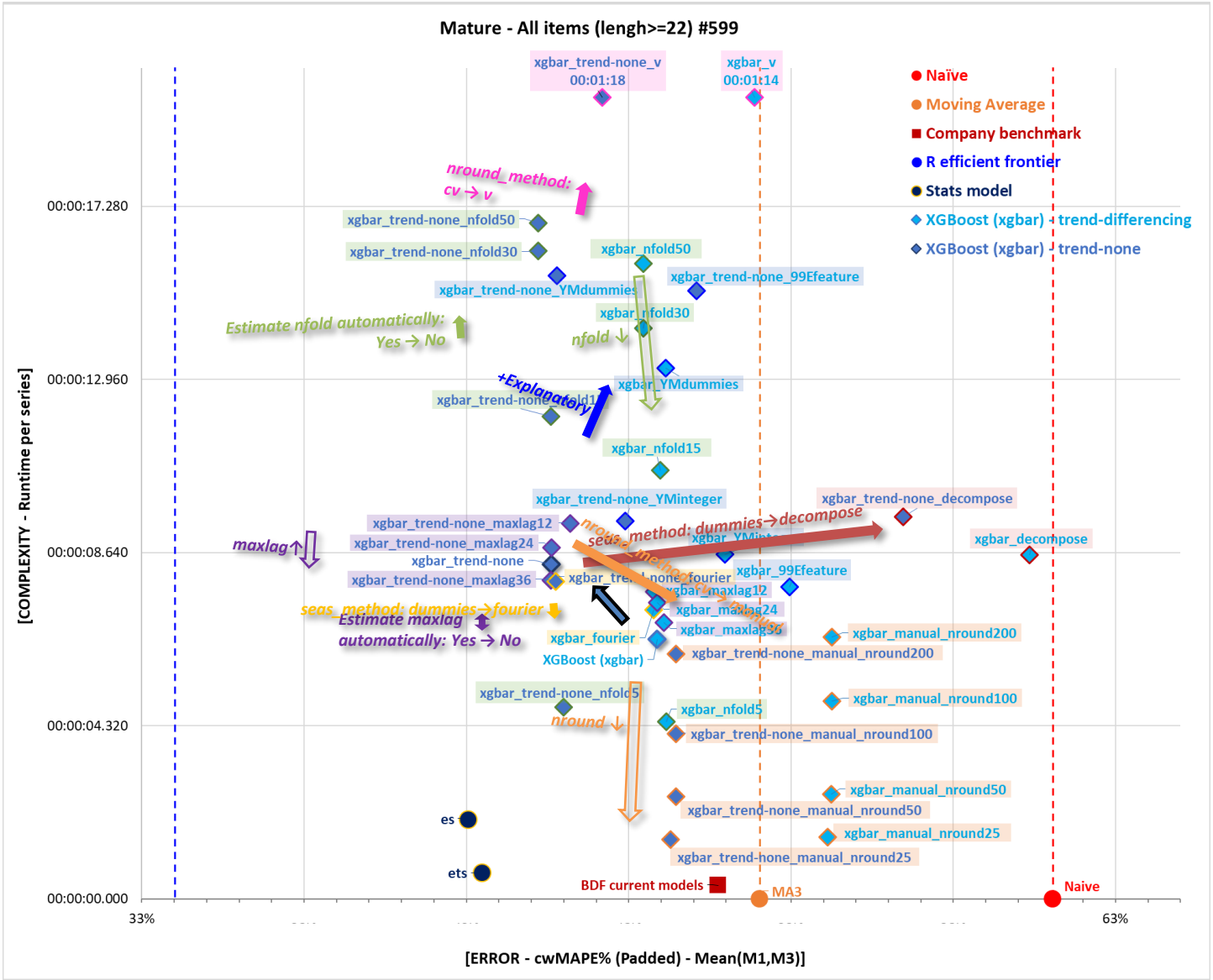
Varying only parameters with defined options already result in significant amount of possible combinations

Postfix	Parameter changes
(None)	Default
_decompose	seas_method: dummies→decompose
_manual_nround200	nround_method: cv→manual & nround: 100→200
_manual_nround100	nround_method: cv→manual (default nround = 100)
_manual_nround50	nround_method: cv→manual & nround: 100→50
_manual_nround25	nround_method: cv→manual & nround: 100→25
_99Efeature	Explanatory series added: iqast features
_Yminteger	Explanatory series added: year month integer dummies
_YMdummies	Explanatory series added: year month binary dummies
_v	nround_method: cv→v
_maxlag36	Estimate maxlag automatically: Yes→No & maxlag: 24→36
_maxlag24	Estimate maxlag automatically: Yes→No (default maxlag = 24)
_maxlag12	Estimate maxlag automatically: Yes→No & maxlag: 24→12
_fourier	seas_method: dummies→fourier
_nfold5	Estimate nfold automatically: Yes→No & nfold: 30→5
_nfold15	Estimate nfold automatically: Yes→No & nfold: 30→15
_nfold30	Estimate nfold automatically: Yes→No (default nfold = 30)
_nfold50	Estimate nfold automatically: Yes→No & nfold: 30→50
_trend-none	trend_method: differencing→none

Postfix	Runtime per series	cost wMAPE	Error diff to default
(None)	00:00:06.469		0.00%
_decompose	00:00:08.577		11.46%
_manual_nround200	00:00:06.531		5.37%
_manual_nround100	00:00:04.929		5.37%
_manual_nround50	00:00:02.608		5.36%
_manual_nround25	00:00:01.540		5.24%
_99Efeature	00:00:07.785		4.08%
_Yminteger	00:00:08.605		2.08%
_YMdummies	00:00:13.246		0.26%
_v	00:01:13.546		3.00%
_maxlag36	00:00:06.900		0.19%
_maxlag24	00:00:07.393		0.00%
_maxlag12	00:00:07.660		-0.10%
_fourier	00:00:07.219		-0.12%
_nfold5	00:00:04.426		0.28%
_nfold15	00:00:10.689		0.10%
_nfold30	00:00:14.232		-0.43%
_nfold50	00:00:15.854		-0.43%
_trend-none	00:00:08.348		-3.27%

Varying single parameter barely improves on xgboost, except for disabling differencing for trend

BDF Austria Landscape
xgboost stage 2 experiment



xgbar default settings: xreg=NULL, Estimate_maxlag=Yes, nrounds=100, nrounds_method=cv, Estimate_nfold=Yes, nfold=30, lambda=1, seas_method=dummies, Estimate_K=Yes, K=5 (if seas_method=fourier), trend_method=differencing

Parameters	No. of options
xreg	2
Estimate maxlag	2
Maxlag	/
Nrounds	/
nrounds_method	3
Estimate n fold	2
Nfold	/
Lambda	/
seas_method	3
Estimate K	2
K	/
trend_method	2
Total metaparameter combinations	**288**

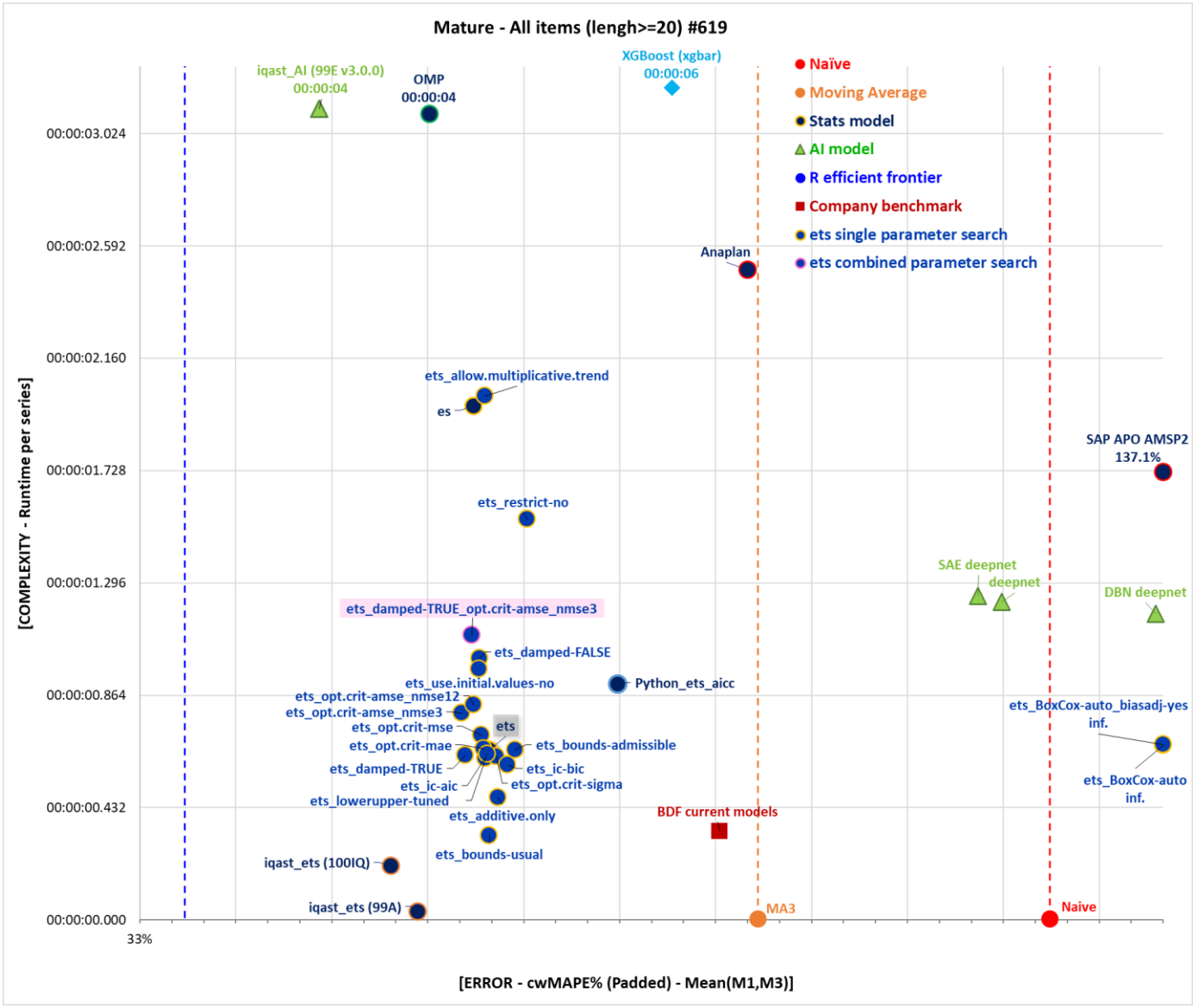
Varying only parameters with defined options already result in significant amount of possible combinations

Postfix	Parameter changes
(None)	Default
_decompose	seas_method: dummies→decompose
_manual_nround200	nround_method: cv→manual & nround: 100→200
_manual_nround100	nround_method: cv→manual (default nround = 100)
_manual_nround50	nround_method: cv→manual & nround: 100→50
_manual_nround25	nround_method: cv→manual & nround: 100→25
_99Efeature	Explanatory series added: iqast features
_Yminteger	Explanatory series added: year month integer dummies
_YMdummies	Explanatory series added: year month binary dummies
_v	nround_method: cv→v
_maxlag36	Estimate maxlag automatically: Yes→No & maxlag: 24→36
_maxlag24	Estimate maxlag automatically: Yes→No (default maxlag = 24)
_maxlag12	Estimate maxlag automatically: Yes→No & maxlag: 24→12
_fourier	seas_method: dummies→fourier
_nfold5	Estimate nfold automatically: Yes→No & nfold: 30→5
_nfold15	Estimate nfold automatically: Yes→No & nfold: 30→15
_nfold30	Estimate nfold automatically: Yes→No (default nfold = 30)
_nfold50	Estimate nfold automatically: Yes→No & nfold: 30→50
_trend-none	trend_method: differencing→none

Postfix	Runtime per series cost	wMAPE Error diff to _trend-none
(None)	00:00:06.469	0.00%
_trend-none	00:00:08.348	-3.27%
_trend-none_decompose	00:00:09.539	10.85%
_trend-none_manual_nround200	00:00:06.110	3.85%
_trend-none_manual_nround100	00:00:04.118	3.85%
_trend-none_manual_nround50	00:00:02.547	3.84%
_trend-none_manual_nround25	00:00:01.489	3.68%
_trend-none_99Efeature	00:00:15.173	4.46%
_trend-none_YMinteger	00:00:09.426	2.28%
_trend-none_YMdummies	00:00:15.557	0.17%
_trend-none_v	00:01:17.829	1.56%
_trend-none_maxlag12	00:00:09.365	0.57%
_trend-none_maxlag24	00:00:08.769	0.00%
_trend-none_maxlag36	00:00:07.948	-0.01%
_trend-none_fourier	00:00:07.927	0.13%
_trend-none_nfold5	00:00:04.785	0.38%
_trend-none_nfold15	00:00:12.035	-0.02%
_trend-none_nfold30	00:00:16.163	-0.41%
_trend-none_nfold50	00:00:16.861	-0.41%

From disabling differencing for trend, varying single parameter hardly improves xgboost further

BDF Austria Landscape
ets experiment



ets default settings: season=12, damped=Null (auto), additive.only=FALSE, lambda=NULL, biasadj = FALSE, lower = c(rep(1e-04, 3), 0.8), upper = c(rep(0.9999, 3), 0.98), opt.crit=lik, nmse=3, bounds=both, ic=aicc, restrict=TRUE, allow.multiplicative.trend=FALSE, use.initial.values=FALSE

R ets parameters	No. of options
Season=12	1
Model=ZZZ	1
damped	3
Alpha	/
Beta	/
Gamma	/
Phi	/
additive.only	2
lambda	2
biasadj	2
lower	/
upper	/
opt.crit	5
nmse	/
bounds	3
ic	3
restrict	2
allow.multiplicative.trend	2
use.initial.values	2
Total metaparameter combinations	**8640**

Experimented parameter search variations

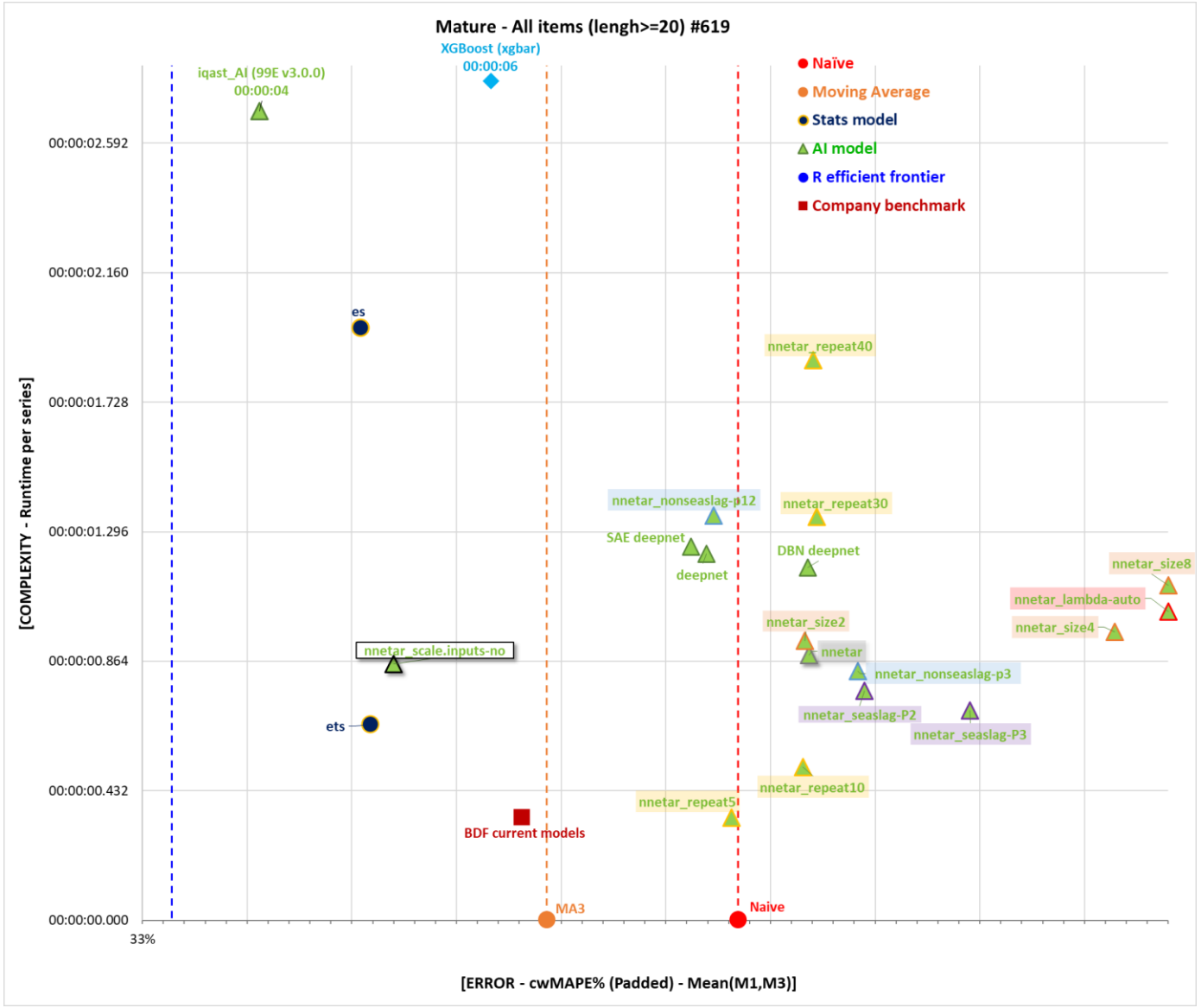
Postfix	Parameter changes
(None)	Default
_damped-TRUE	use damped trend: auto→TRUE
_damped-FALSE	use damped trend: auto→FALSE
_additive.only	consider additive model only: No→Yes
_BoxCox-auto	lambda: NULL→auto
_BoxCox-auto_biasadj-yes	lambda: NULL→auto & biasadj: No→Yes
_lowerupper-tuned	use tuned lower and upper parameter bounds
_opt.crit-amse_nmse3	optimisation criterion - lik→amse & nmse=3
_opt.crit-amse_nmse12	optimisation criterion - lik→amse & nmse=12
_opt.crit-mse	optimisation criterion - lik→mse
_opt.crit-mae	optimisation criterion - lik→mae
_opt.crit-sigma	optimisation criterion - lik→sigma (std of residuals)
_bounds-usual	both→usual (between specified lower and upper bounds)
_bounds-admissible	both→admissible (the admissible space)
_ic-aic	information criterion: aicc→aic
_ic-bic	information criterion: aicc→bic
_restrict-no	allow models with infinite variance: No → Yes
_allow.multiplicative.trend	allow multiplicative trend model: No → Yes
_use.initial.values-no	re-estimate initial values: Yes→No
_damped-TRUE_opt.crit-amse_nmse3	use damped trend: auto→TRUE & optimisation criterion - lik→amse & nmse=3 (default)

Postfix	Runtime per series	Mean(M1,M3) cost wMAPE	Diff to default
(None)	00:00:00.653		0.00%
_damped-TRUE	00:00:00.633		-0.74%
_damped-FALSE	00:00:01.006		-0.29%
_additive.only	00:00:00.471		0.29%
_BoxCox-auto	00:00:00.672		inf
_BoxCox-auto_biasadj-yes	00:00:00.674		inf
_lowerupper-tuned	00:00:00.623		-0.11%
_opt.crit-amse_nmse3	00:00:00.795		-0.85%
_opt.crit-amse_nmse12	00:00:00.828		-0.48%
_opt.crit-mse	00:00:00.710		-0.24%
_opt.crit-mae	00:00:00.659		-0.17%
_opt.crit-sigma	00:00:00.627		0.23%
_bounds-usual	00:00:00.325		0.00%
_bounds-admissible	00:00:00.653		0.82%
_ic-aic	00:00:00.637		-0.05%
_ic-bic	00:00:00.597		0.58%
_restrict-no	00:00:01.541		1.20%
_allow.multiplicative.trend	00:00:02.014		-0.13%
_use.initial.values-no	00:00:00.965		-0.31%
_damped-TRUE_opt.crit-amse_nmse3	00:00:01.096		-0.55%

Changing metaparameters can only marginally improve ets (<1%)

BDF Austria Landscape

nnetar experiment

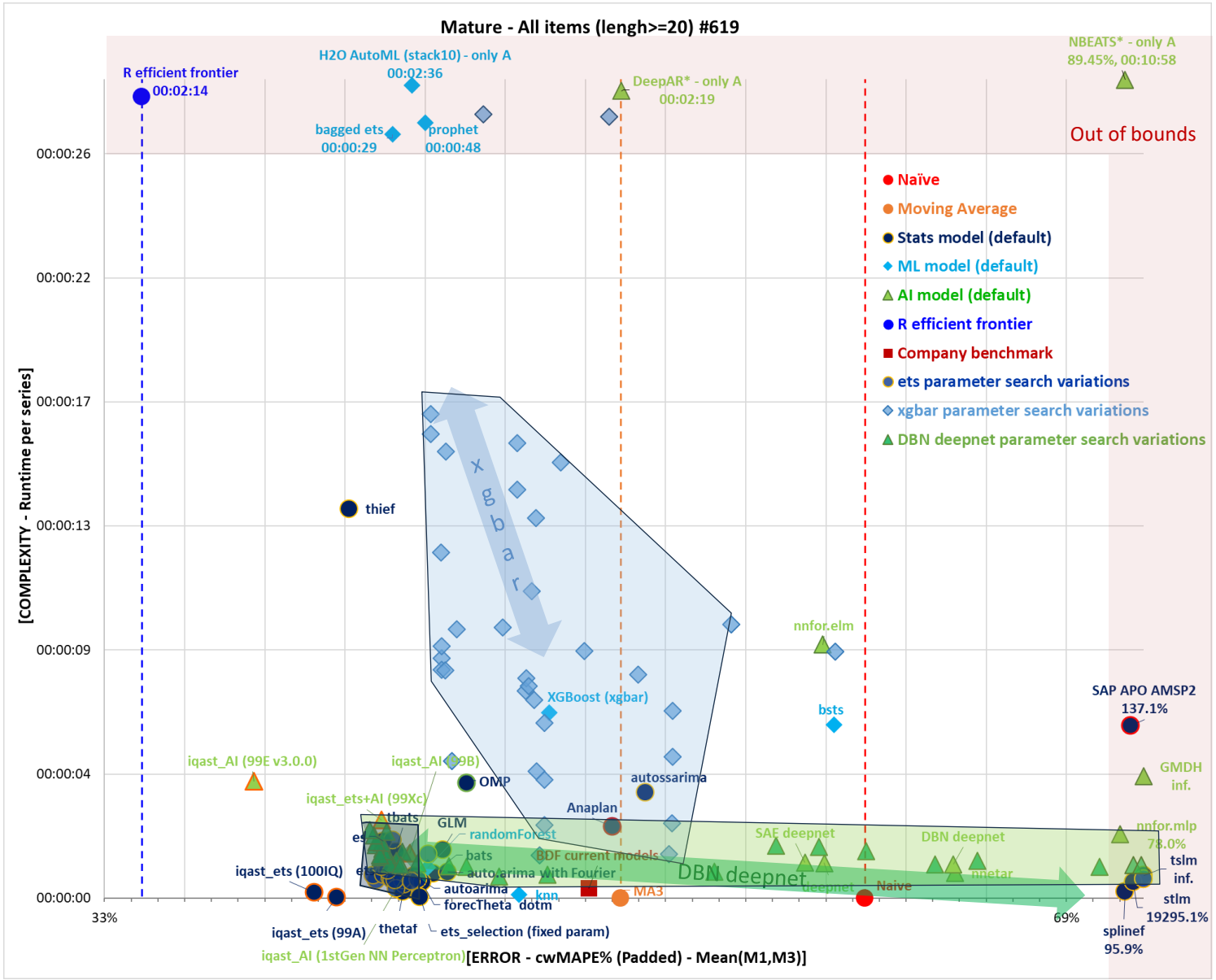


Postfix	Parameter changes
(None)	Default
_lambda-auto	BoxCox transformation: NULL→auto
_nonseaslag-p3	Estimate p: Yes→No & p=3
_nonseaslag-p12	Estimate p: Yes→No & p=12
_seaslag-P2	Estimate P: Yes→No & P=2
_seaslag-P3	Estimate P: Yes→No & P=3
_size2	No. of nodes in the hidden layer = 2
_size4	No. of nodes in the hidden layer = 4
_size8	No. of nodes in the hidden layer = 8
_repeat10	No. of networks: 20→10
_repeat30	No. of networks: 20→30
_repeat40	No. of networks: 20→40
_repeat5	No. of networks: 20→5
_scale.inputs-no	**scale.inputs: Yes→No**

Postfix	Runtime per series	cost	wMAPE Diff to default
(None)	00:00:00.885		0.00%
_nonseaslag-p3	00:00:00.831		2.33%
_nonseaslag-p12	00:00:01.349		-4.55%
_seaslag-P2	00:00:00.766		2.64%
_seaslag-P3	00:00:00.701		7.69%
_size2	00:00:00.933		-0.18%
_size4	00:00:00.962		14.61%
_size8	00:00:01.116		36.40%
_repeat10	00:00:00.511		-0.27%
_repeat30	00:00:01.346		0.36%
_repeat40	00:00:01.868		0.21%
_repeat5	00:00:00.343		-3.70%
_scale.inputs-no	00:00:00.855		-19.84%
_lambda-auto	00:00:01.030	inf	inf

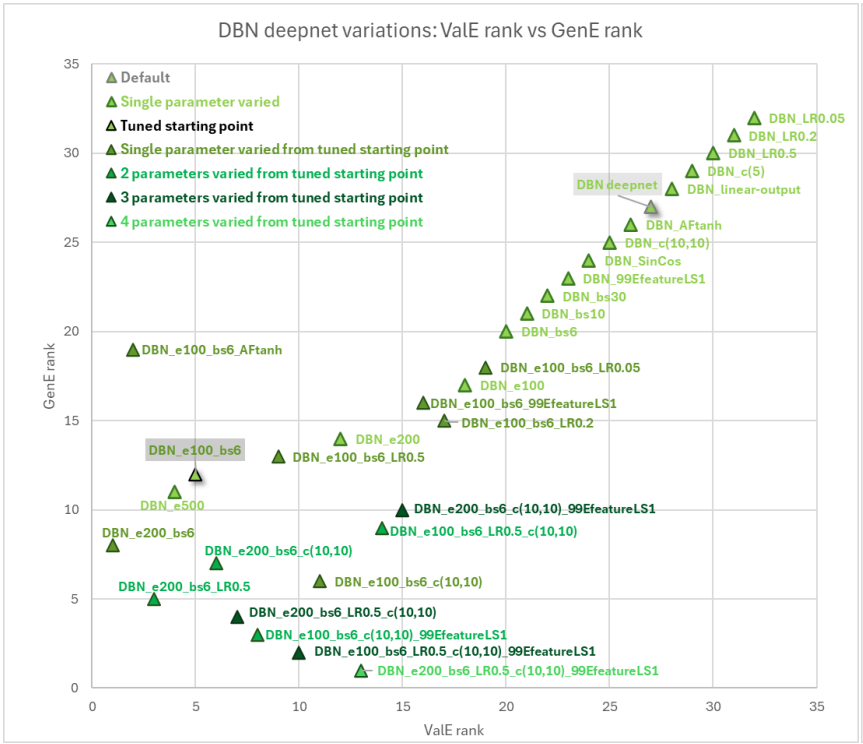
BDF Austria Landscape

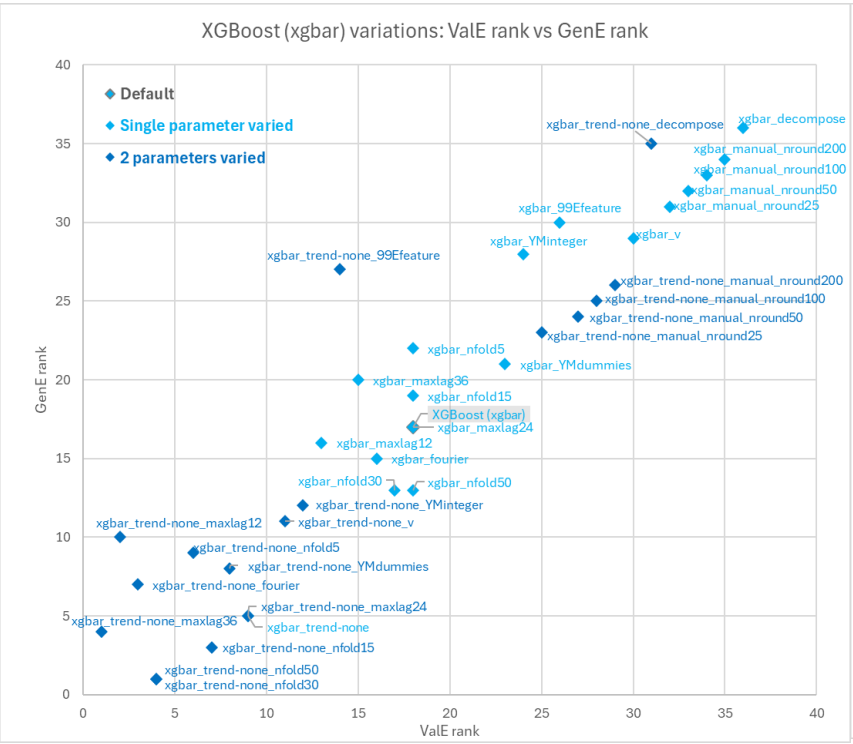
R models parameter search experiments

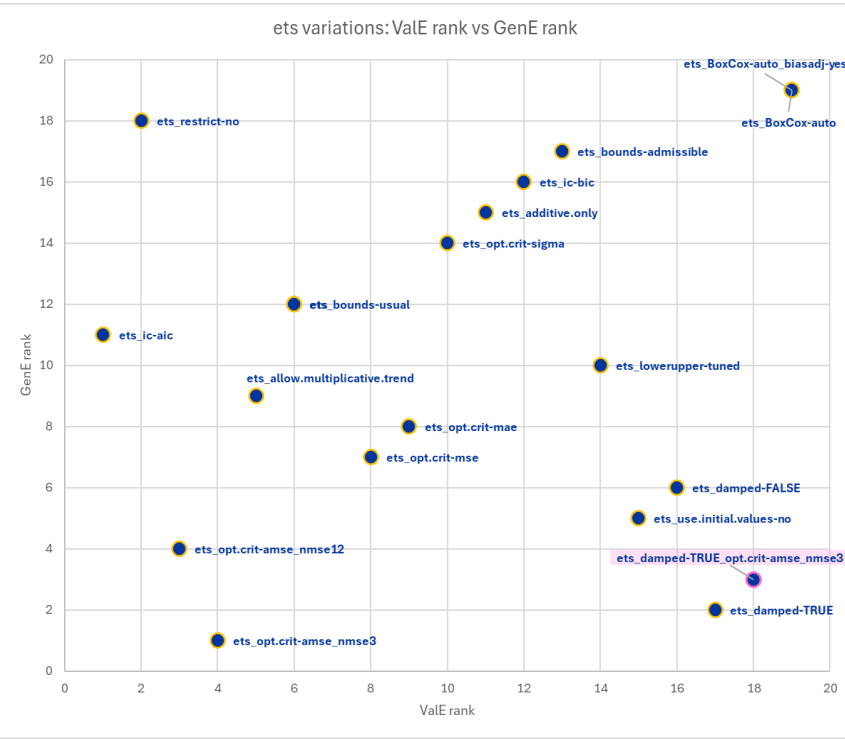


- ets shows smallest variance of error when tuning meta parameters
- xgbar is more reactive in computational performance than accuracy when varying meta parameters, slight improvement might require significant compromise in runtime
- DBN deepnet shows the biggest range of variations in accuracy when varying parameters, best variant can slightly outperform ets where the margin of improvement gets minimal

BDF Austria DBN landscape
Rank on ValE vs GenE



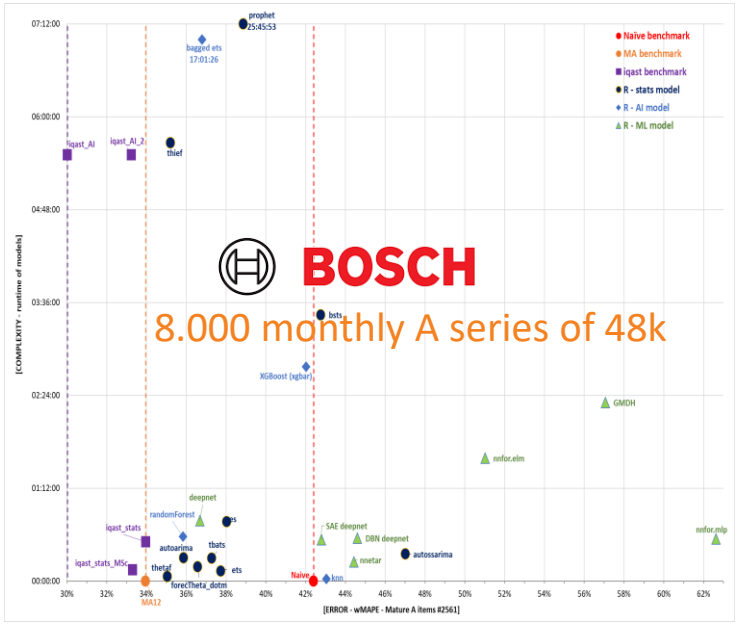


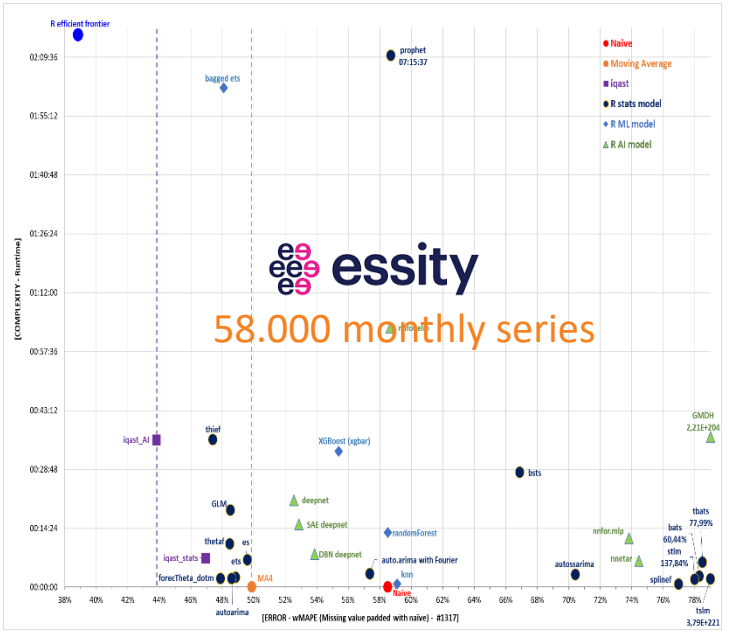


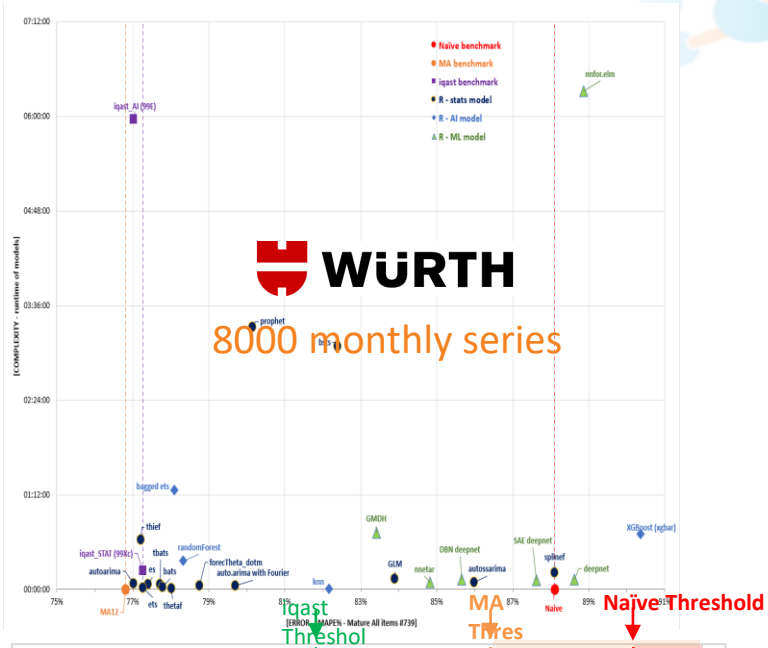
→ Performance is consistent by GenE or ValE when varying single parameter from default

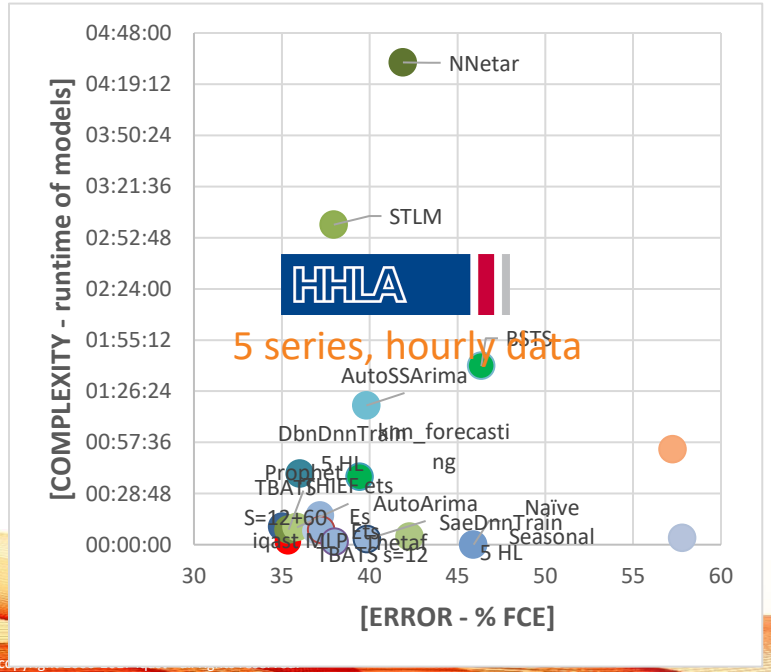
→ Error ranking is inconsistent when varying parameters from tuned starting point, but relative difference in error is too subtle to observe visually

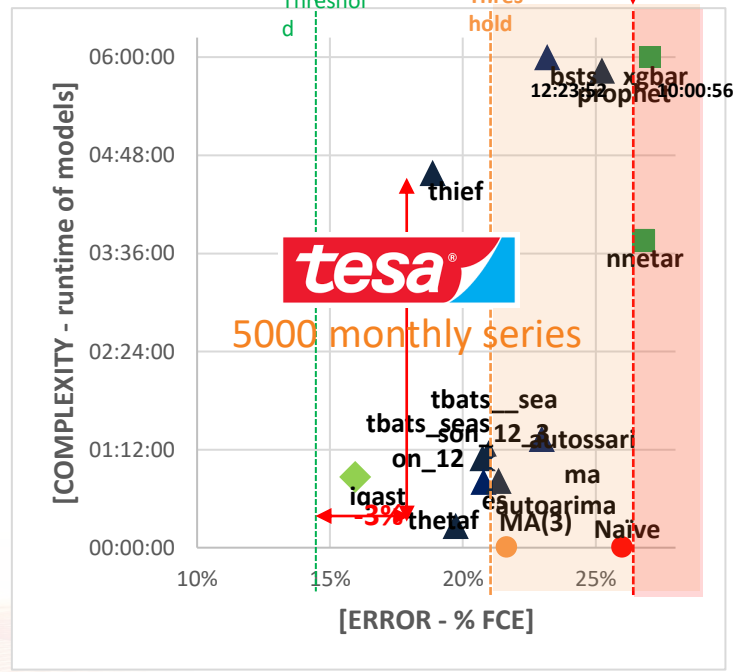
Replicated across industries











AI in practice … many projects fail!

Forbes

Oct 14, 2020, 07:00am EDT | 828 views

Why Do Most AI Projects Fail?

Prajit Datta Forbes Councils Member
Forbes Technology Council COUNCIL POST | Membership (fee-based)
Innovation

AI Research Scientist at AFRY, one of the Scandinavian consulting giants, overseeing the AI Global Competence Group.

TOWARDS AI HOME PUBLICATION EDITORIAL NEWSLETTER SUBSCRIBE ABOUT CONTACT

Top 10 reasons Why 87% of Machine learning projects fail?

Are Language Models Actually Useful for Time Series Forecasting?

Mingtian Tan
University of Virginia
wtd3gz@virginia.edu

Mike A. Merrill
University of Washington
mikeam@cs.washington.edu

Vinayak Gupta
University of Washington
vinayak@cs.washington.edu

Tim Althoff
University of Washington
althoff@cs.washington

Thomas Hartvigsen
University of Virginia
hartvigsen@virginia.edu

Abstract

Large language models (LLMs) are being applied to time series tasks, particularly time series forecasting. However, are language models actually useful for time series? After a series of ablation studies on three recent and popular LLM-based time series forecasting methods, we find that removing the LLM component or replacing it with a basic attention layer does not degrade the forecasting results—in most cases the results even improved. We also find that despite their significant computational cost, pretrained LLMs do no better than models trained from scratch, do not represent the sequential dependencies in time series, and do not assist in few-shot settings. Additionally, we explore time series encoders and reveal that patching and attention structures perform similarly to state-of-the-art LLM-based forecasters.[1]

1 Introduction

Time series analysis is a critical problem across many domains, including disease propagation forecasting [7], retail sales analysis [3], healthcare [23, 15] and finance [28]. A great deal of recent work in time series analysis (constituting repositories with more than 1200 total stars on GitHub) has focused on adapting pretrained large language models (LLMs) to classify, forecast, and detect anomalies in time series [13, 42, 19, 4, 5, 29, 12, 37, 14]. These papers posit that language models, being advanced models for sequential dependencies in text, may generalize to the sequential dependencies in time series data. This hypothesis is unsurprising given the popularity of language models in machine learning research writ large. So to what extent are language models *really* beneficial for traditional time series tasks?

Our main claim is simple but profound: **popular methods for adapting language models for time series forecasting perform the same or worse than basic ablations, yet require orders of magnitude more compute.** Derived from extensive ablations, these findings reveal a worrying trend in contemporary time series forecasting literature. Our goal is not to imply that language models will never be useful for time series. In fact, recent works point to many exciting and promising ways that language and time series interact, like time series reasoning [22] and social understanding [6]. Rather, we aim to highlight surprising findings that existing methods do very little to use the innate reasoning power of pretrained language models on established time series tasks.

We substantiate our claim by performing three ablations of three popular and recent LLM-based forecasting methods [42, 13, 19] using eight standard benchmark datasets from reference methods

[1]All code, environments, and data to reproduce our work are available in this anonymized repository: https://github.com/BennyTMT/TS_Models

Preprint. Under review.

cted that 87% of AI projects will never make it into production.

TABULAR DATA: DEEP LEARNING IS NOT ALL YOU NEE

Ravid Shwartz-Ziv
ravid.ziv@intel.com
IT AI Group, Intel

Amitai Armon
amitai.armon@intel.com
IT AI Group, Intel

November 24, 2021

ABSTRACT

A key element in solving real-life data science problems is selecting the types of models to use. Tree ensemble models (such as XGBoost) are usually recommended for classification and regression problems with tabular data. However, several deep learning models for tabular data have recently been proposed, claiming to outperform XGBoost for some use cases. This paper explores whether these deep models should be a recommended option for tabular data by rigorously comparing the new deep models to XGBoost on various datasets. In addition to systematically comparing their performance, we consider the tuning and computation they require. Our study shows that XGBoost outperforms these deep models across the datasets, including the datasets used in the papers that proposed the deep models. We also demonstrate that XGBoost requires much less tuning. On the positive side, we show that an ensemble of deep models and XGBoost performs better on these datasets than XGBoost alone.

Zillow, facing big losses, quits flipping houses and will lay off a quarter of its staff.

a and AI

When algorithms go bad: Zillow, Amazon, Facebook and the pitfalls of rampant automation

BY TODD BISHOP on November 13, 2021 at 8:59 am

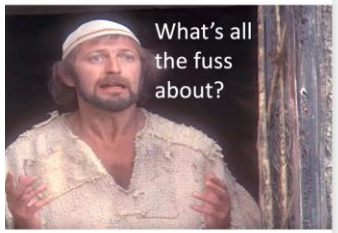
Share 72 Tweet Share Reddit Email

Become a GeekWire member

Listen to this episode

Subscribe: Apple Podcasts | Google Podcasts | Spotify | Stitcher | TuneIn | RSS

phet" Package Basching



Blog | 12 min

Is Facebook's "Prophet" the Time-Series Messiah, or Just a Very Naughty Boy?

Published on February 3, 2021

Peter Cotton, PhD, Founder

my subjective view:

20 years of forecasting projects (both academic & consultancy) but only slowly changing picture:

pioneers in forecasting with artificial intelligence

Has your company tried forecasting with AI/ML (Artificial Intelligence / Machine Learning) methods?

You can see how people vote. Learn more

No, not yet	55%
Yes, running a Pilot or PoC	26%
Yes, running in production	6%
No, stopped after some trials	13%

→ more projects fail (13%) than run in production (6%)
→ majority (55%) has not tried to use AI/ML in forecasting yet

Why did your project in forecasting with AI/ML (Artificial Intelligence/Machine Learning) fail?

You can see how people vote. Learn more

AI algorithms not accurate	11%
Lack of AI skills in team (HR)	6%
Lack of available data for AI	72%
Cost of AI software/systems	11%

18 votes • 2w left • Hide results

→ majority indicate lack of data → have your say!

What I want to share today: ❶ why projects fail? ❷ tell a success story

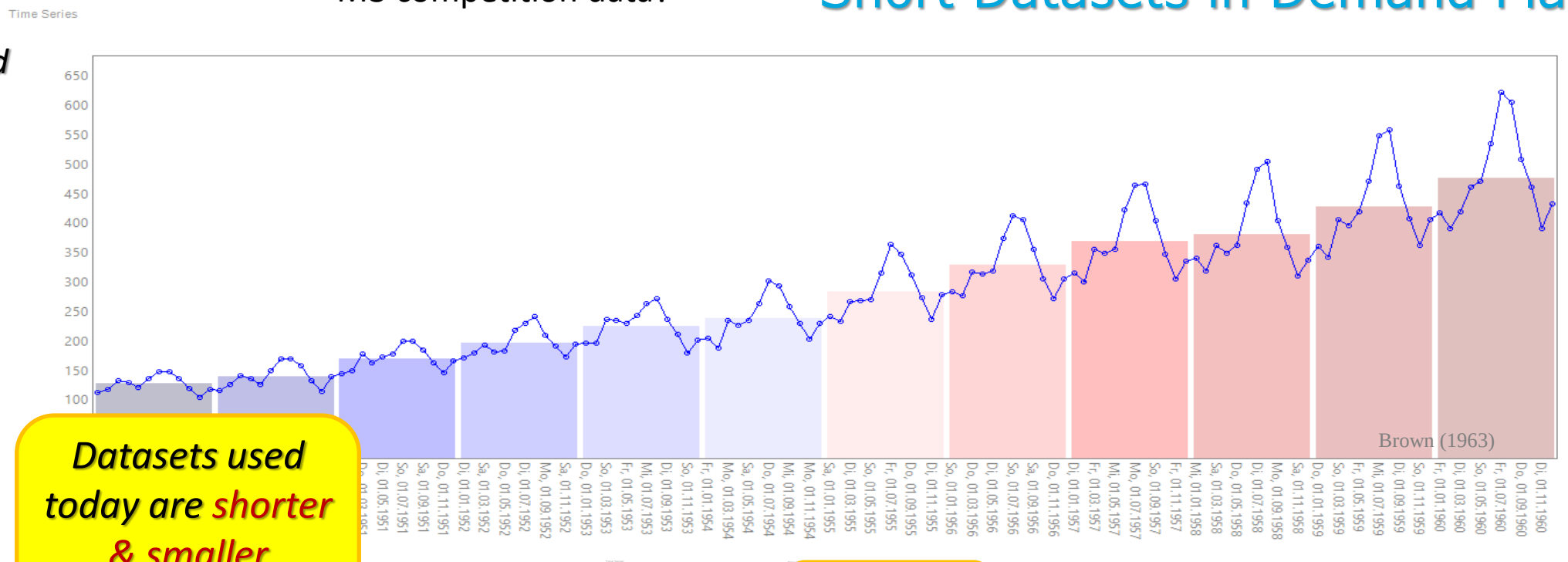
The New York Times

→ SOME DEEP NEURAL NETS FAILED THEE TESTS!!! ;-)

Short Datasets in Demand Planning

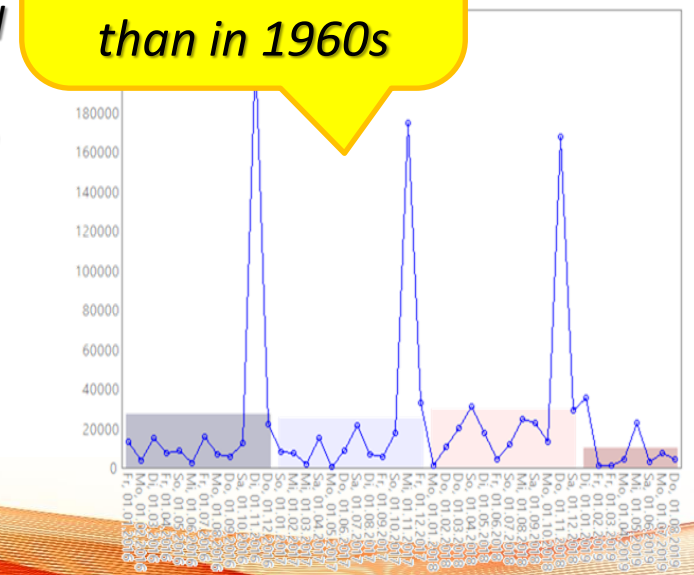
M3 competition data?

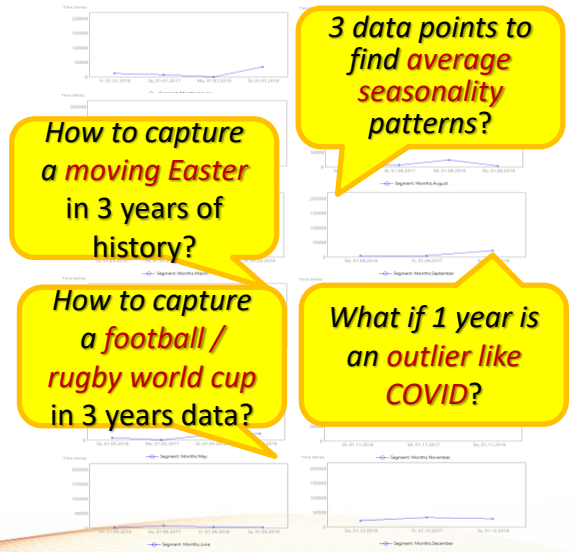
Dataset used to develop Exponential Smoothing & ARIMA algorithms



Datasets used today are shorter & smaller than in 1960s

Dataset used today in Supply Chain & Demand Forecasting





How to capture a moving Easter in 3 years of history?

3 data points to find average seasonality patterns?

How to capture a football / rugby world cup in 3 years data?

What if 1 year is an outlier like COVID?

Note: M3 Industry has average of 122 observations (>10 years)

2017 Survey on Demand Planning practices

→ 72% used 3 years of monthly data!

→ up to 20 internal & external data sourced used mostly judgmentally

→ Most data exchanged by email

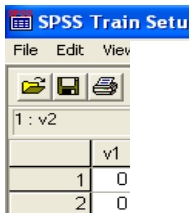
(Weller & Crone, 2017)

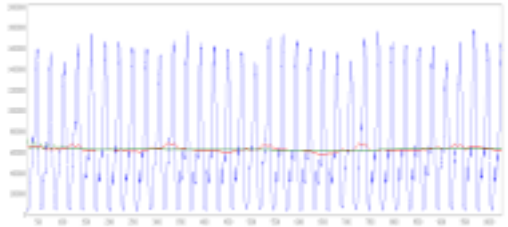
Sparse Data Stylized Facts

Low Frequency ← Time Series Frequency → High Frequency

NONE | Yearly | Quarterly | Monthly | Weekly | Daily | Hourly | Minutes

One / Many — Number of Variables

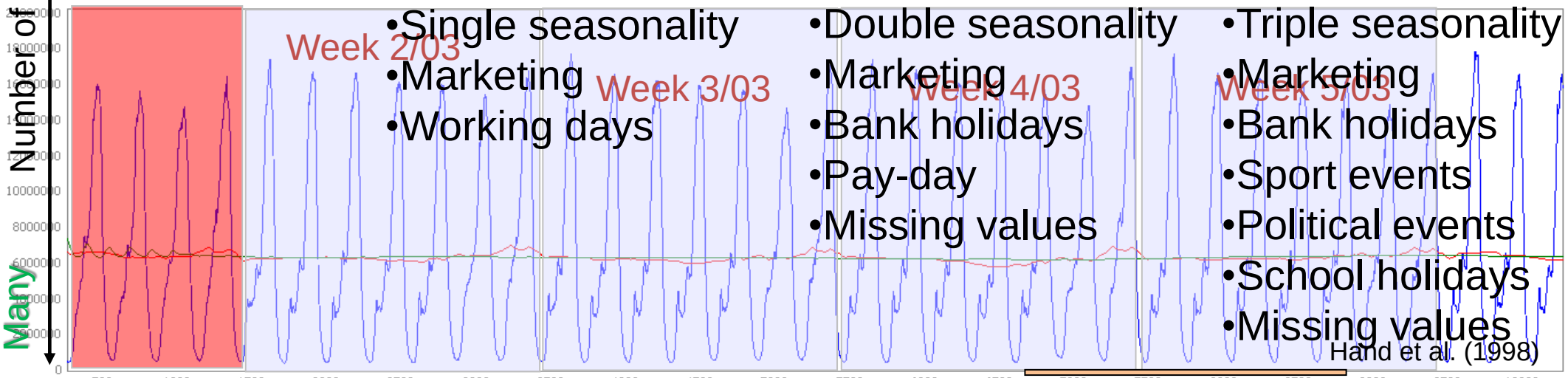




Dummy Variables:
- Single seasonality
- Marketing
- Working days

Dummy Variables:
- Double seasonality
- Marketing
- Bank holidays
- Pay-day
- Missing values

Dummy Variables:
- Triple seasonality
- Marketing
- Bank holidays
- Sport events
- Political events
- School holidays
- Missing values

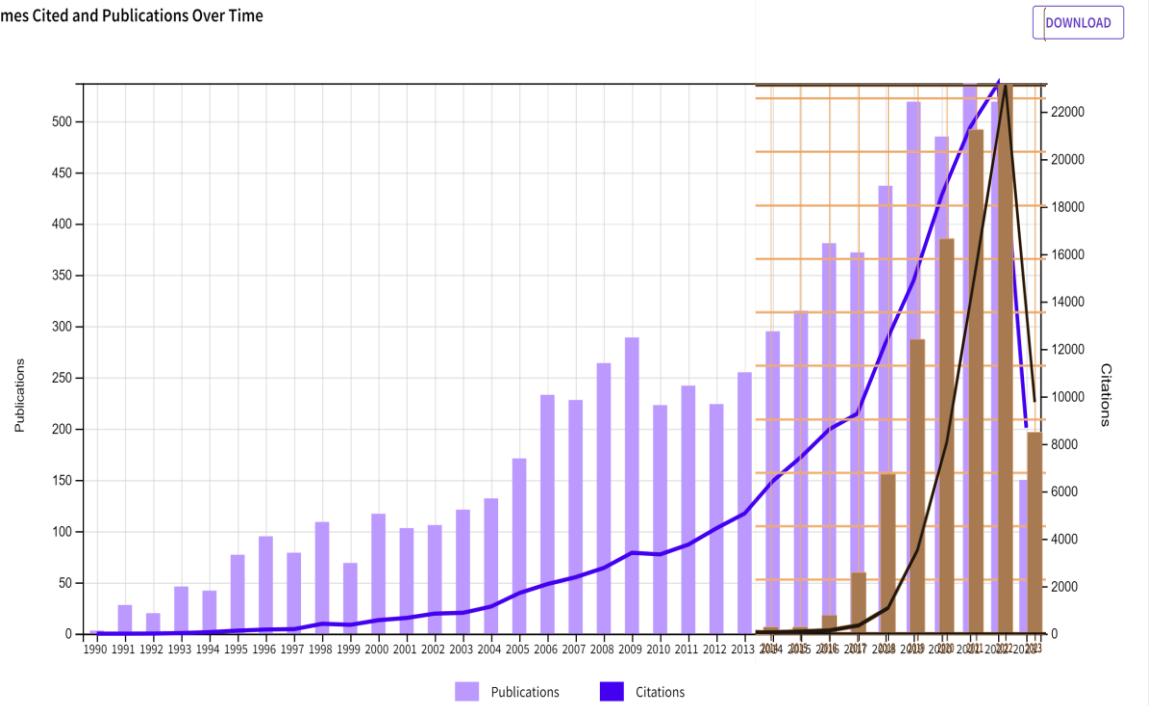


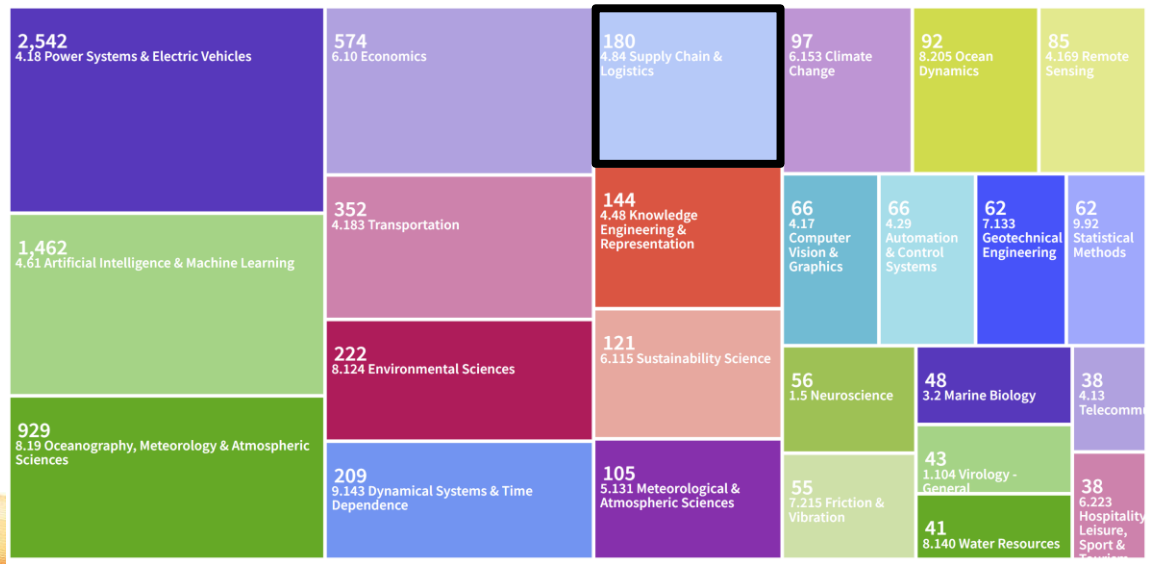
Hand et al. (1998)

New Year → Zoom in on first 10.000 observations (≈ 4 days ≈ 4 weeks)

Many binary dummy variables → data mining features

Deep Learning in Forecasting





Unprecedented academic interest in neural net forecasting

- 9032 publications on neural net forecasting (title)
  - 95k citing papers with 202k citations (22.45 average citations)
  - Accounts for 4% of 221k papers in neural nets
  - strong interest continues across disciplines driven by IEEE

- significant increase in Deep Neural Network papers from 2014 adding to consistent level of shallow (i.e. non-deep) neural papers
- Deep nets mostly in engineering & ???

Unprecedented non-academic interest

- Deep Nets exceed neural nets 3fold (ML exceeds it)
- Exceeds interest in forecasting and ETS ;-)
- (ETS still not very interesting)

→ Deep Learning dominates popular interest, and for 1st time matches academic interest in neural nets in forecasting!

((LSTM) or (Deep Net*) or (DeepAR) or (temporal fusion transformer*) or (convolutional neural) or (NBEATS)) NOT ((multilayer perceptro*) or (Echo state neural)) and (forecast* or (time serie*))

((neural net*) or (multilayer perceptro*) or (Echo state neural) or (general regresison neural) or (Extreme learning machine*)) NOT ((LSTM) or (Deep Net*) or (DeepAR) or (temporal fusion transformer*) or (convolutional neural) or (NBEATS)) and (forecast* or (time serie*))

Summary & Conclusions

- Summary & conclusion
  - on 10 of 11 company datasets all three (!) 3rd gen Deep Neural Net functions (in R) fail to outperform the Naïve
  - on 8 of 11 company datasets all 2nd generation Neural Nets fail to outperform the Naïve
  - on 11 of 11 company datasets iqast ai approach (see ISF2021 auto feature creation & permutation) outperforms stats & ai
  - root cause? 10 company datasets from industry are shorter than M3 industry dataset!
  - General additional insights
    - Deep and shallow nets are often faster than ML methods in computation time (e.g. random forrest) → extend search space
    - Some “newer” methods underperform others rathe rconsistently, e.g. mlp << nnetar, es << ets – need for individual tuning / customization
    - Theta & thief perform well overall
    - iqast stats models outperforms all other stats contenders
    - iqast ai models outperform all other ai/ml methods across all datasets
- Outlook
  - Need to improve data in length, annotated events and possibly switch frequency to better leverage data hungry methods
- Limitations of the study
  - No all python based sota algos Moira, TimeGPT,
  - No other commercial software packages yet
  - Sample selection bias in industries & coverage of industries (more MedTech & FMCG, less transport, none in manufacturing) → need to compile industry specific landscapes of multiple companies to generalize results