Improving the Forecast Accuracy of Protected Data Using Time Series Features

Matthew J. Schneider
Associate Professor
mjs624@drexel.edu

August 22nd, 2023, University of Moratuwa
Dept. of Transport Management
and Logistics Engineering

Joint work with:
Cameron D. Bale
Jinwook Lee

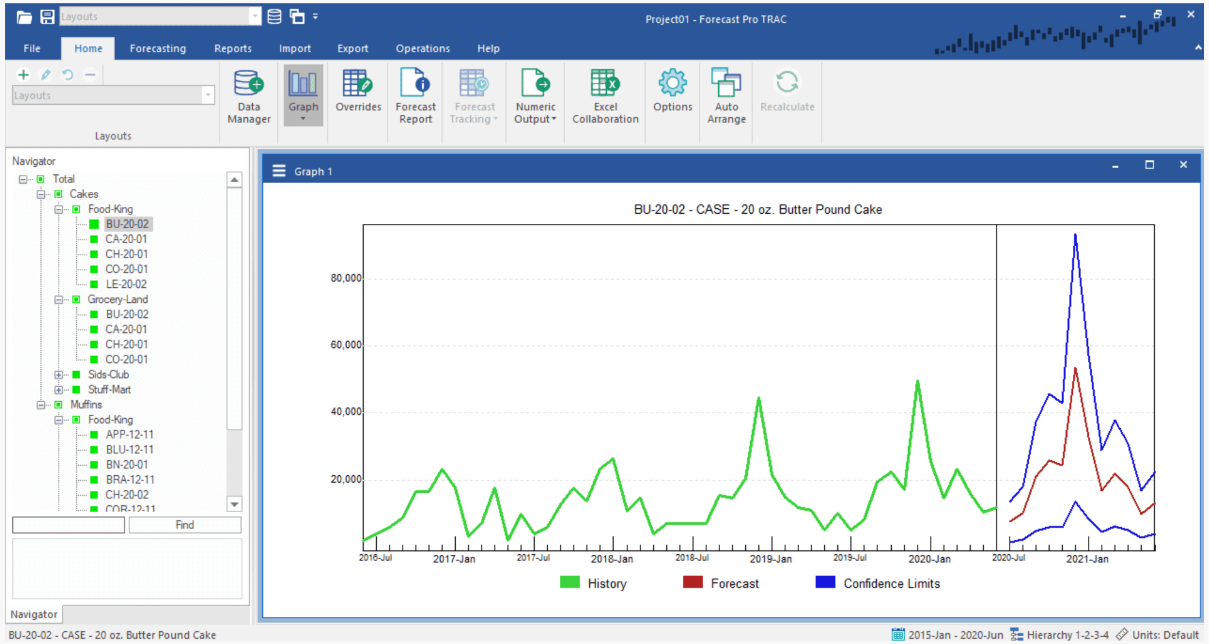
University of Moratuwa
TRANSPORT MANAGEMENT AND LOGISTICS ENGINEERING (TMLE)

DREXEL UNIVERSITY
LeBow
College of Business

Data sharing causes privacy breaches

Image Credit: [forecast Pro]

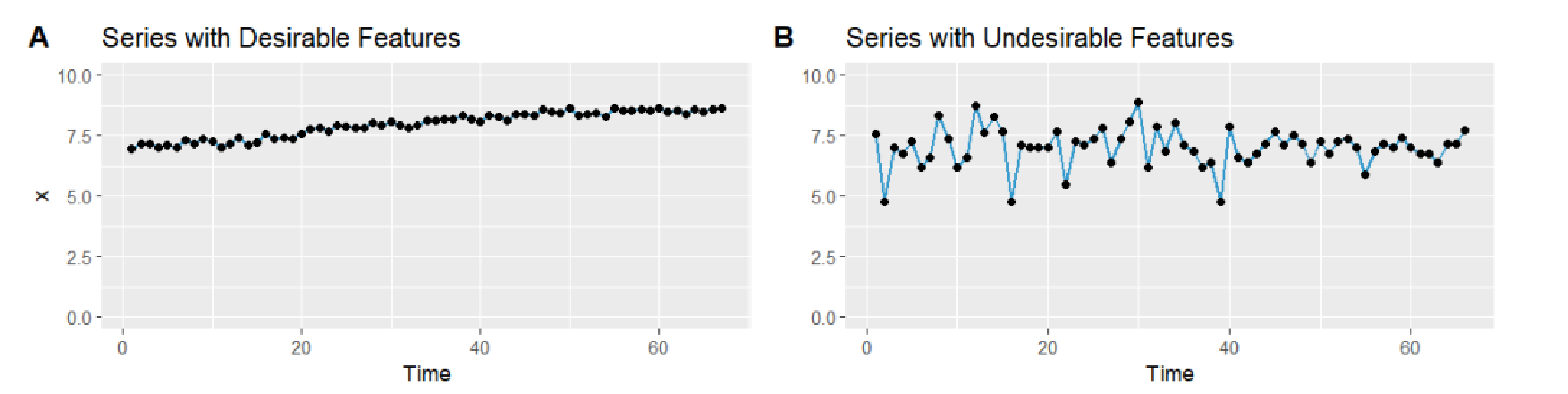
- Party A (*e.g.*, CPG companies) generates data for millions of products every day and shares data with Party B



- Party B (*e.g.*, third party forecasting company, retailer, or team of data scientists) produces daily or weekly forecasts

Time series values can be identifiable, sensitive, and/or missing

Can we swap (replace) time series values and maintain forecast accuracy?



Existing anonymization methods destroy forecast accuracy!

- Usefulness of anonymized data is often an afterthought (Blanco-Justicia et al., 2022)

- Existing centralized anonymization methods destroy forecast accuracy!
  - Weak differential privacy ($\epsilon = 20$) (Gonçalves et al. 2021a)
  - 21% accuracy reduction *when half of data points are noised* (Luo et al. 2018)
  - Aggregation to achieve k-anonymity (Nin & Torra, 2009)

Focused on pre-defined privacy criteria

- Decentralized Methods (Gonçalves et al., 2021a,b,c; Sommer et al., 2021)
  - Secure multi-party computation and federated learning do not produce time series data
  - Data markets still pose privacy concerns over the market operator

Literature review on time series features

Time series features are useful for:

- **Classification** (Fulcher & Jones, 2014)
- **Clustering** (Bandara et al., 2018)
  - 18 features, RNNs accuracy improved 2 to 11%
- **Forecast accuracy prediction** (Makridakis et al., 2018; Spiliotis et al., 2020)
  - increasing frequency, kurtosis, linearity, and seasonal strength improved forecast accuracy
  - increasing skewness, self-similarity, and randomness degraded forecast accuracy
- **Model selection and forecast combination** (Montero-Manso et al., 2020; Qi et al., 2022; Talagala et al., 2022; Li et al., 2022; Kang et al., 2022)
  - forecasts using the strength of trend and seasonality for exponential smoothing model selection had lower errors than information-based selection methods for

We focus on using time series features to inform **time series value replacement** in the context of time series anonymization.

Contributions

Show that the most useful features for predicting forecast accuracy (Makridakis et al., 2018; Spiliotis et al., 2020) are not necessarily the most useful for swapping time series values.

Application findings: spectral entropy, hurst, and skewness are not most useful for improving forecast accuracy

Feature	**Description**	**Value Range**	**Selected (Literature)**	**Selected (k-nTS+)**
Spectral Entropy	Signal-to-noise ratio of the time series.	[0, 1]	X	
Hurst	Long-range dependence (self-similarity) of a time series.	[0, 1]	X	
Skewness	Symmetry of the distribution of time series values.	$(-\infty, \infty)$	X	
Kurtosis	Weight of the tails of the distribution of time series values.	$(-\infty, \infty)$	X	
Error ACF	First autocorrelation coefficient of the error component of the decomposed series.	[-1, 1]	X	

Application findings: spike, max level shift, and max variance shift are most useful for improving forecast accuracy

Feature	**Description**	**Value Range**	**Selected (Literature)**	**Selected (k-nTS+)**
Trend	Strength of the trend.	$[0, 1]$	X	X
Seasonality	Strength of the seasonality.	$[0, 1]$	X	
Mean	Mean of the time series.	$[0, \infty)$	X	X
Variance	Variance of the time series.	$[0, \infty)$	X	X
Spike	Variance of the leave-one-out variances of the remainder component of the decomposed series.	$[0, \infty)$		X
Max Variance Shift	Largest variance shift between two consecutive sliding windows.	$[0, \infty)$		X
Max Level Shift	Largest mean shift between two consecutive sliding windows.	$[0, \infty)$		X

Contributions

Show that the most useful features for predicting forecast accuracy (Makridakis et al., 2018; Spiliotis et al., 2020) are not necessarily the most useful for swapping time series values.

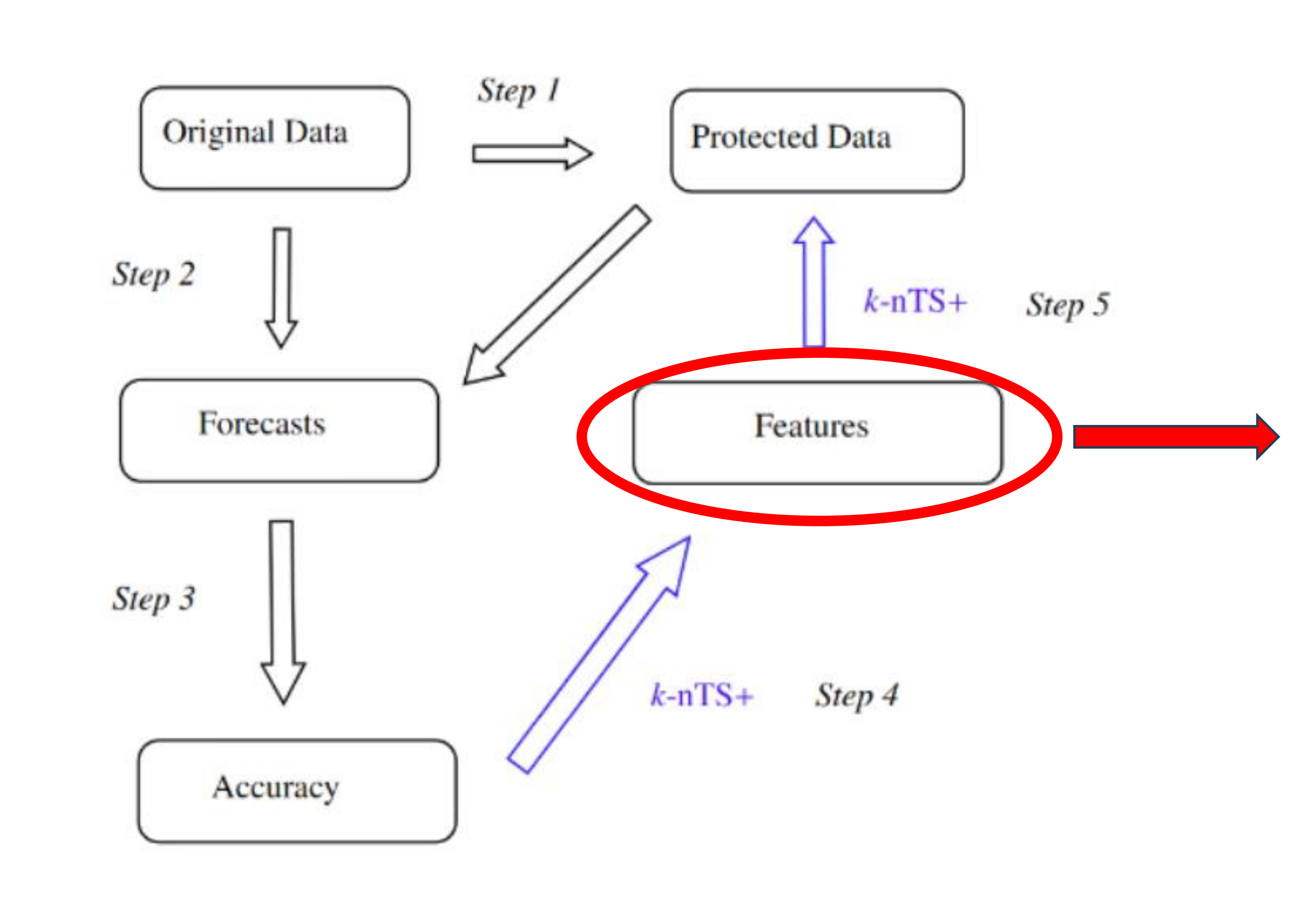
Use machine learning-based feature selection to improve forecast accuracy of swapped data by (14% worse than original) over manual feature selection (40% worse).

Enable organizations to share protected time series data with

(1) good forecast accuracy

(2) useful time series features

Swapped data are generated based on features that improve forecast accuracy



<u>Two-Step Feature Selection</u>

(1) RReliefF (Robnik-Sikonja & Kononenko, 2003)
(2) Recursive Feature Elimination (Gregorutti et al., 2017)

Features are identified as useful or not useful for swapping.

RReliefF: Features with a higher improvement weight vary across series with different forecast errors (desired)

Define an improvement weight for feature m as W_m calculated on the difference of two conditional probabilities.

Let π_m and π_ε denote the events that two nearest time series have different forecast values for feature m and different forecast errors, respectively.

$$W_m = p(\pi_m | \pi_\epsilon) - p(\pi_m | \pi_\epsilon^c)$$

Features with $W_m > 0$
have a higher probability
of varying across series with different forecast errors (desired)
than varying across series with similar forecast errors (not desired).

RReliefF: Features with a higher improvement weight vary across series with different forecast errors (desired)

Define an improvement weight for feature m as W_m calculated on the difference of two conditional probabilities.

Let π_m and π_ε denote the events that two nearest time series have different forecast values for feature m and different forecast errors, respectively.

$$W_m = p(\pi_m | \pi_\epsilon) - p(\pi_m | \pi_\epsilon^c)$$

We don't want features to change when the errors are the same! Noisy for no gain!

Features with $W_m > 0$
have a higher probability
of varying across series with different forecast errors (desired)
than varying across series with similar forecast errors (not desired).

Recursive Feature Elimination (RFE) handles "curse of dimensionality"

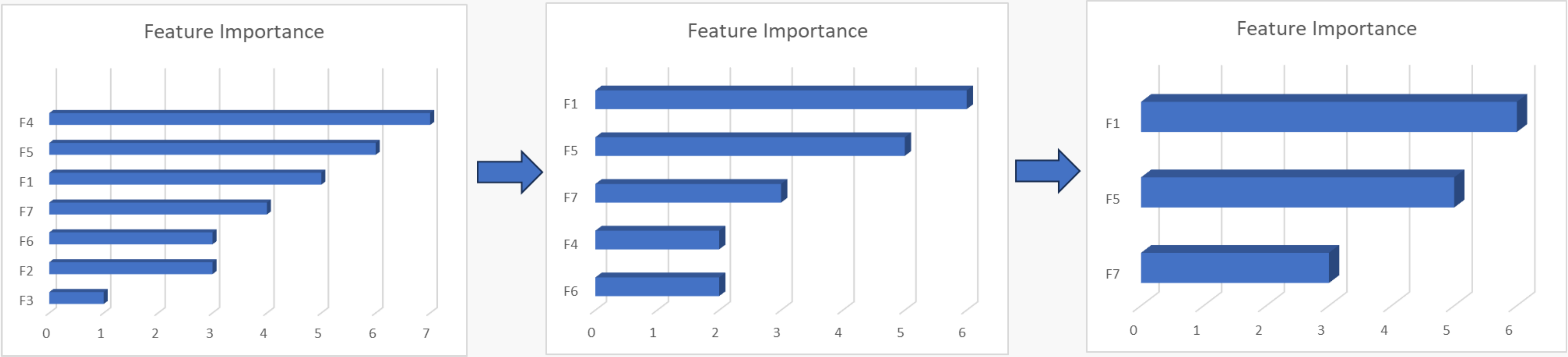
RFE (Gregorutti et al., 2017) is designed to select an efficient feature set amongst highly correlated features.

Use a random forest to predict forecast accuracy using features with $W_m > 0$:

Recursion:

(1) Rank features on predictive importance

(2) Eliminate least important feature(s)

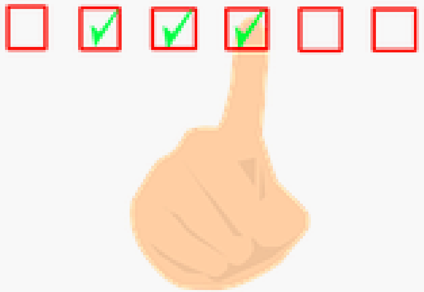


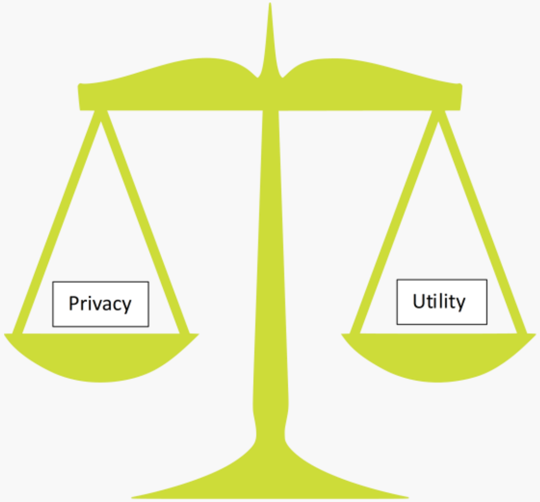
Methodological advantages of k-nearest time series + (k-nTS+)

- Selects an efficient set of features from a larger set of potentially correlated features.

- Specifically incorporates forecast accuracy into the data protection process (key theme of my research: the usefulness of protected data matters!)

- Flexibility! Users can specify:
  - Forecast horizon(s)
  - Forecasting model(s)
  - Accuracy metric(s)
  - Original set of time series features





An Application to Privacy

Can we use features to swap a private value of a time series with a randomized value from another time series?

Early M forecasting competitions required anonymity so forecasters didn't cheat!

M3 Monthly Micro

- Protect the identity of competition time series
- Contains features representative of real-world data (Spiliotis et al., 2020)

Privacy: what is the probability of identifying a time series?

Identification disclosure risk (Nin & Torra, 2006, 2009): average proportion of J time series which are correctly identified across S simulated privacy attacks.

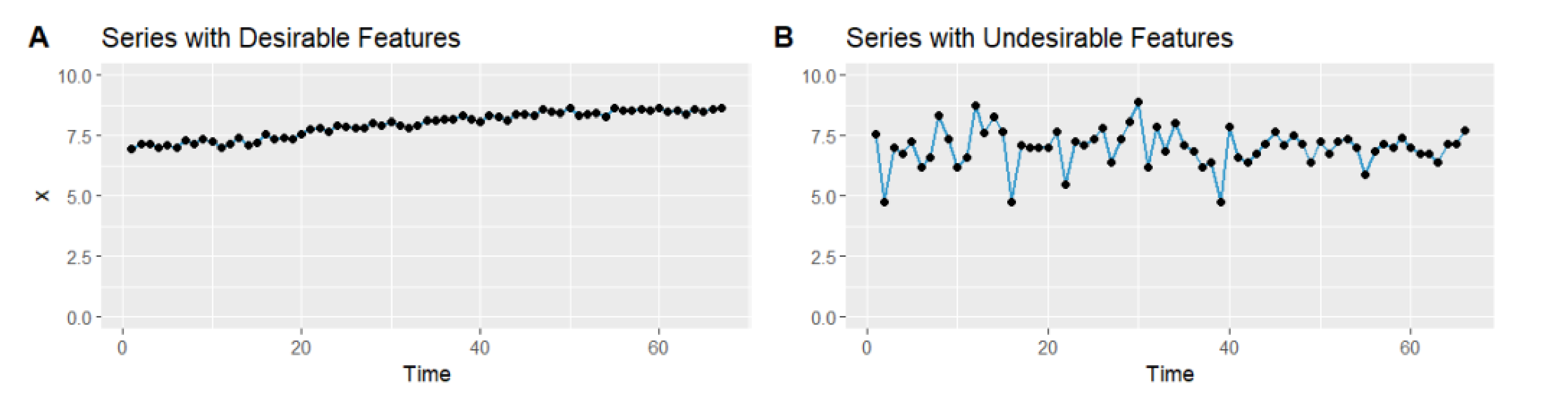
$$\bar{P} = \frac{1}{J \times S} \sum_{s=1}^{S} \sum_{i=1}^{J} I\left(\hat{M}_i^s = j^*\right)$$

$\hat{M}_i^s$ is the adversary's prediction of the identity of the ith time series.

$\bar{P} = 100\%$ when all time series are identified

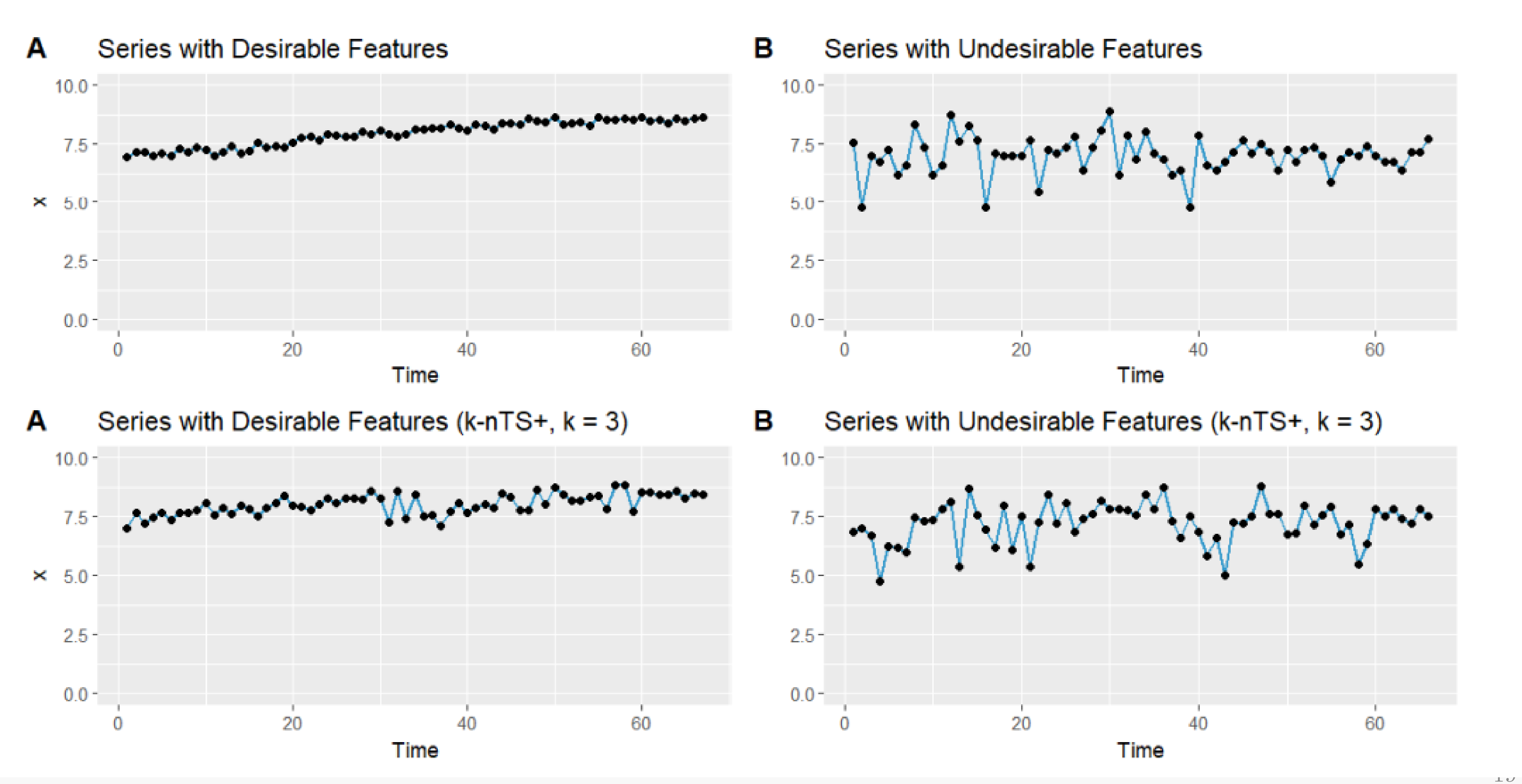
$\bar{P} = \frac{1}{J}$ when an adversary randomly guesses

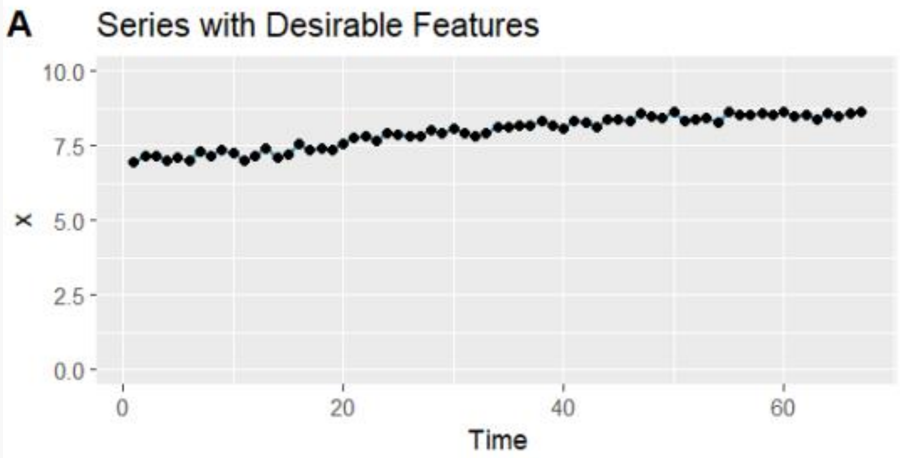
Illustration: desirable (A) vs. undesirable (B) time series

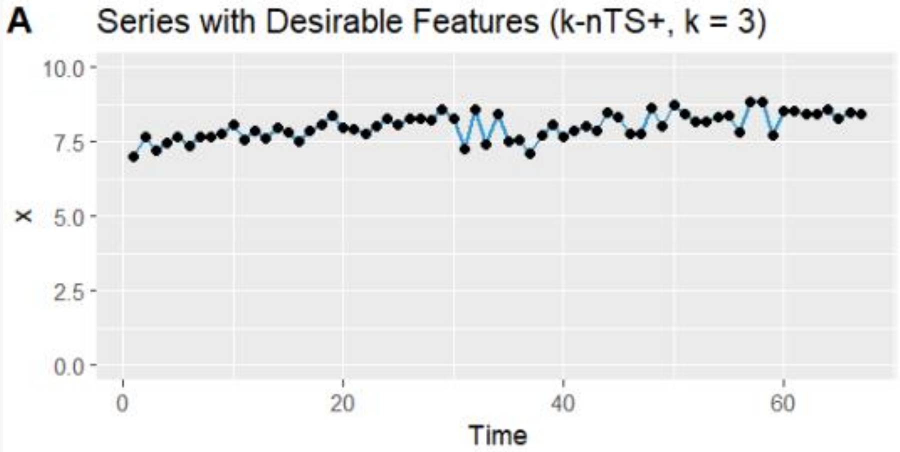


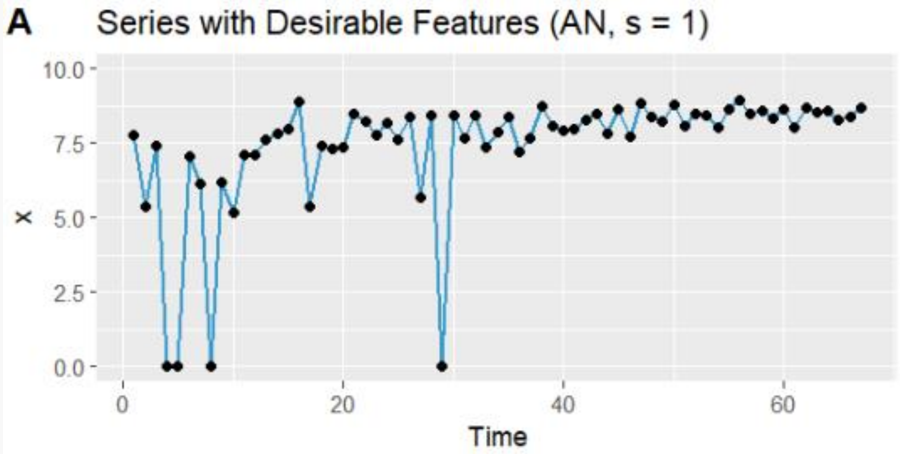
	Spectral Entropy	*Hurst*	*Skewness*	*Kurtosis*	*Error ACF*	*Trend*	*Seasonality*	*Mean*	*Variance*	*Spike*	*Max Variance Shift*	*Max Level Shift*
Series A	0.07	0.99	-0.41	-1.24	-0.09	0.97	0.16	7.96	0.29	0.0000	0.05	0.57
Series B	1	0.5	-0.57	1.16	-0.19	0.12	0.23	7.01	0.65	0.0001	1.1	0.7

k-nTS+ maintains “accuracy improving” features



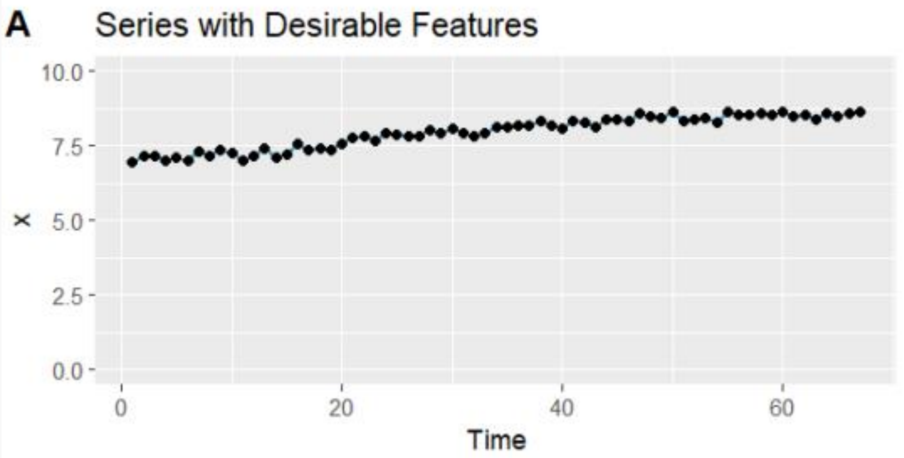


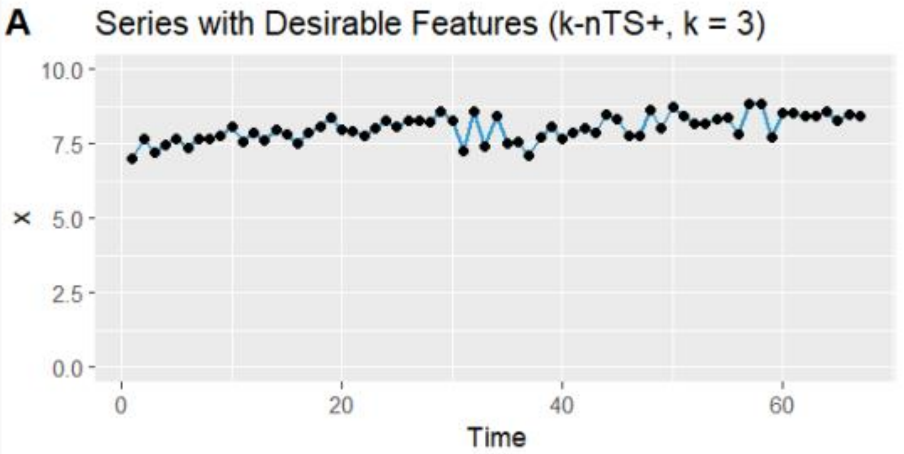


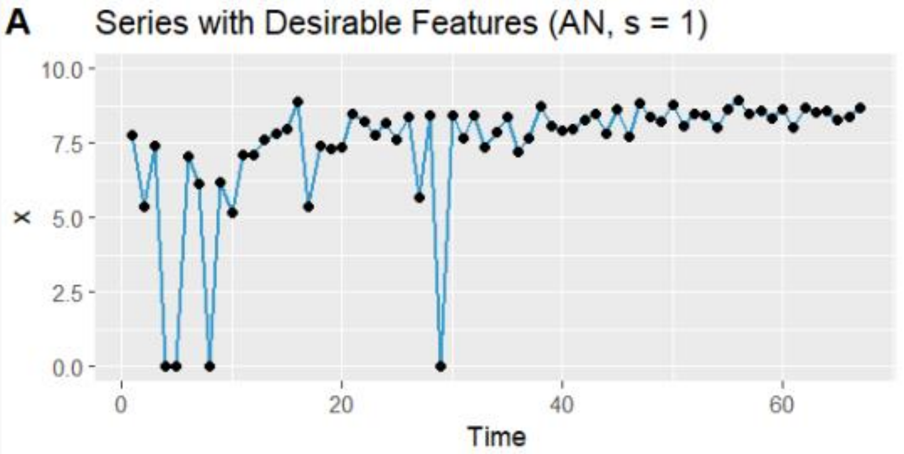


Feature	Desirable Time Series (left Fig. 7)		
	Original	k-nTS+ (k=3)	AN (s=1)
Spectral Entropy	0.07	0.89	0.92
Hurst	0.99	0.81	0.76
Skewness	-0.41	-0.18	-2.74
Kurtosis	-1.24	-0.74	6.99
Error ACF	-0.09	-0.22	-0.20
Trend	0.97	0.58	0.49
Seasonality	0.16	0.25	0.39
Mean	7.96	8.02	7.41
Variance	0.29	0.19	4.27
Spike	0.0000	0.0000	0.0037
Max Variance Shift	0.05	0.24	9.37
Max Level Shift	0.57	0.51	2.77

k-nTS+ (and AN) change features with (-) improvement weights





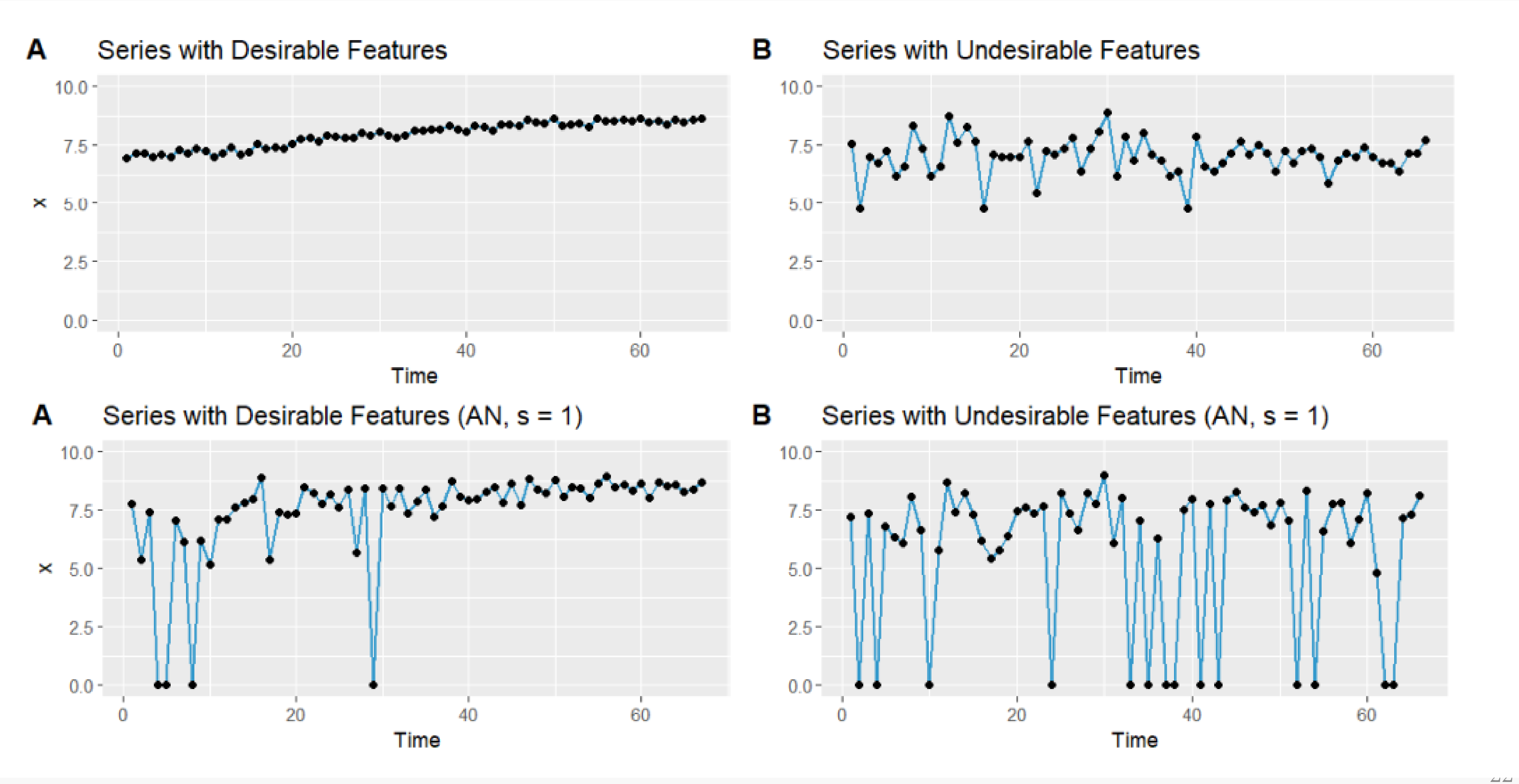


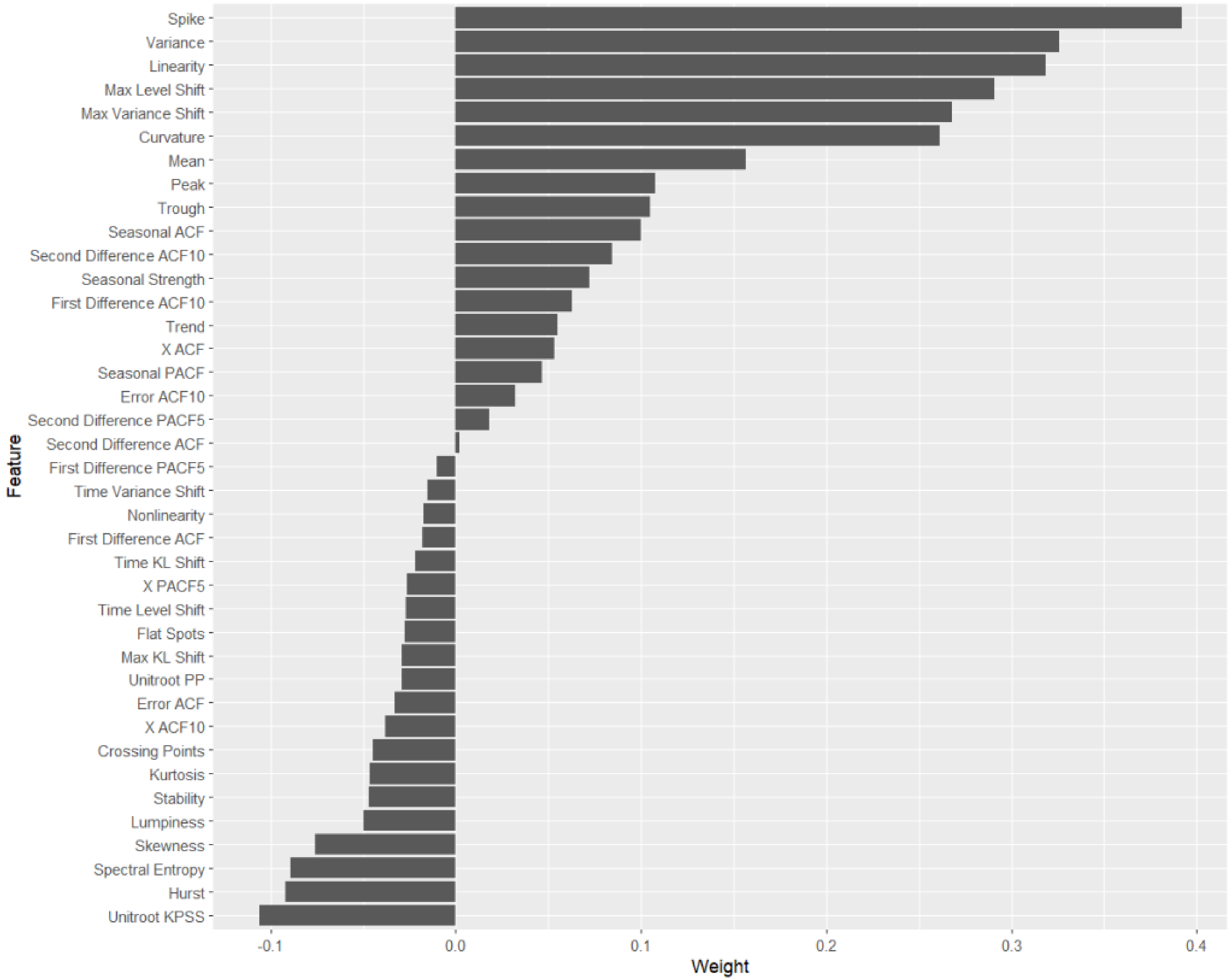
Feature	**Desirable Time Series (left Fig. 7)**		
	Original	**k-nTS+ (k=3)**	**AN (s=1)**
Spectral Entropy	0.07	0.89	0.92
Hurst	0.99	0.81	0.76
Skewness	-0.41	-0.18	-2.74
Kurtosis	-1.24	-0.74	6.99
Error ACF	-0.09	-0.22	-0.20
Trend	0.97	0.58	0.49
Seasonality	0.16	0.25	0.39
Mean	7.96	8.02	7.41
Variance	0.29	0.19	4.27
Spike	0.0000	0.0000	0.0037
Max Variance Shift	0.05	0.24	9.37
Max Level Shift	0.57	0.51	2.77

k-nTS+ maintains features with (+) improvement weights

AN destroys features with (+) improvement weights

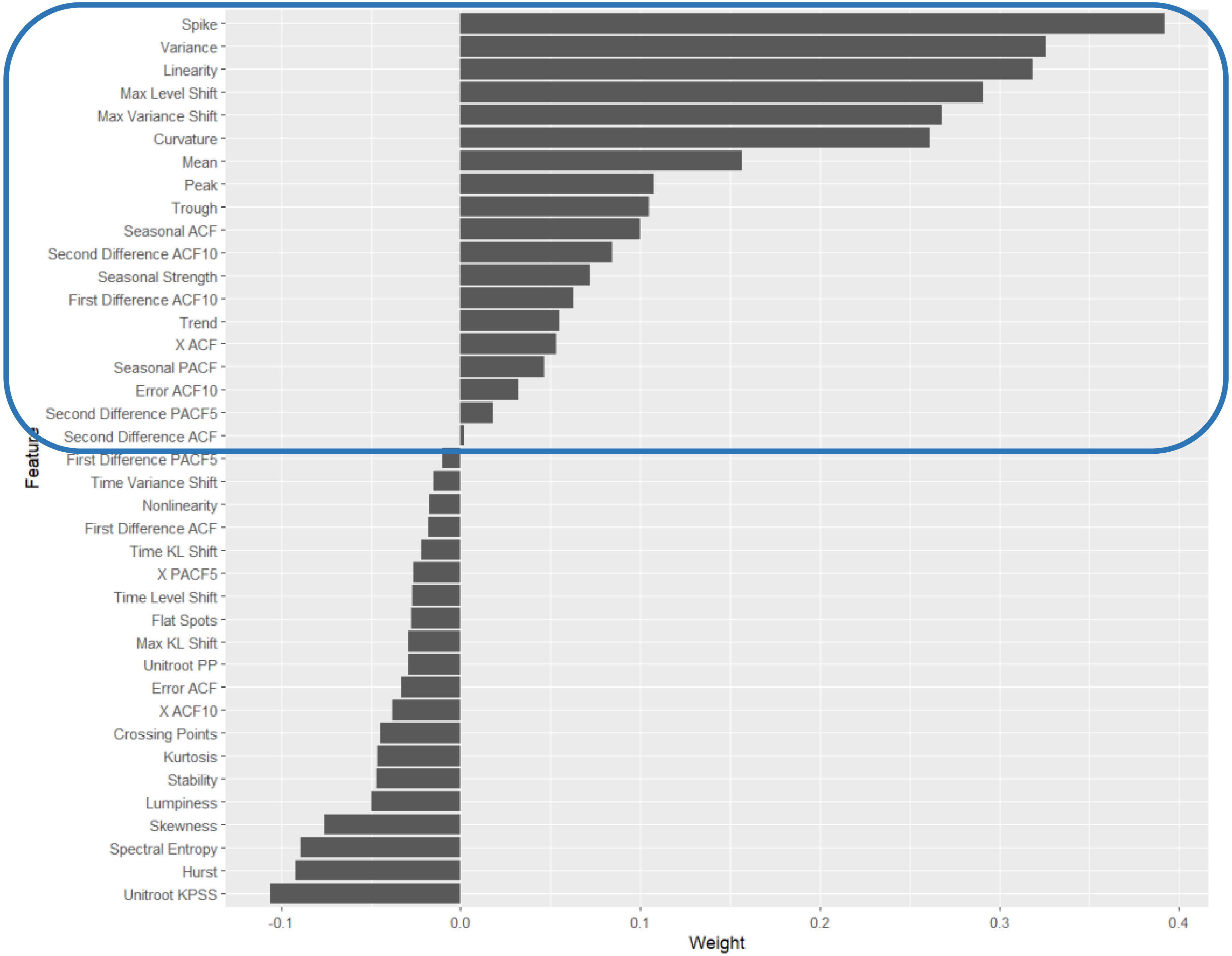
Competitor methods destroy accuracy improving features





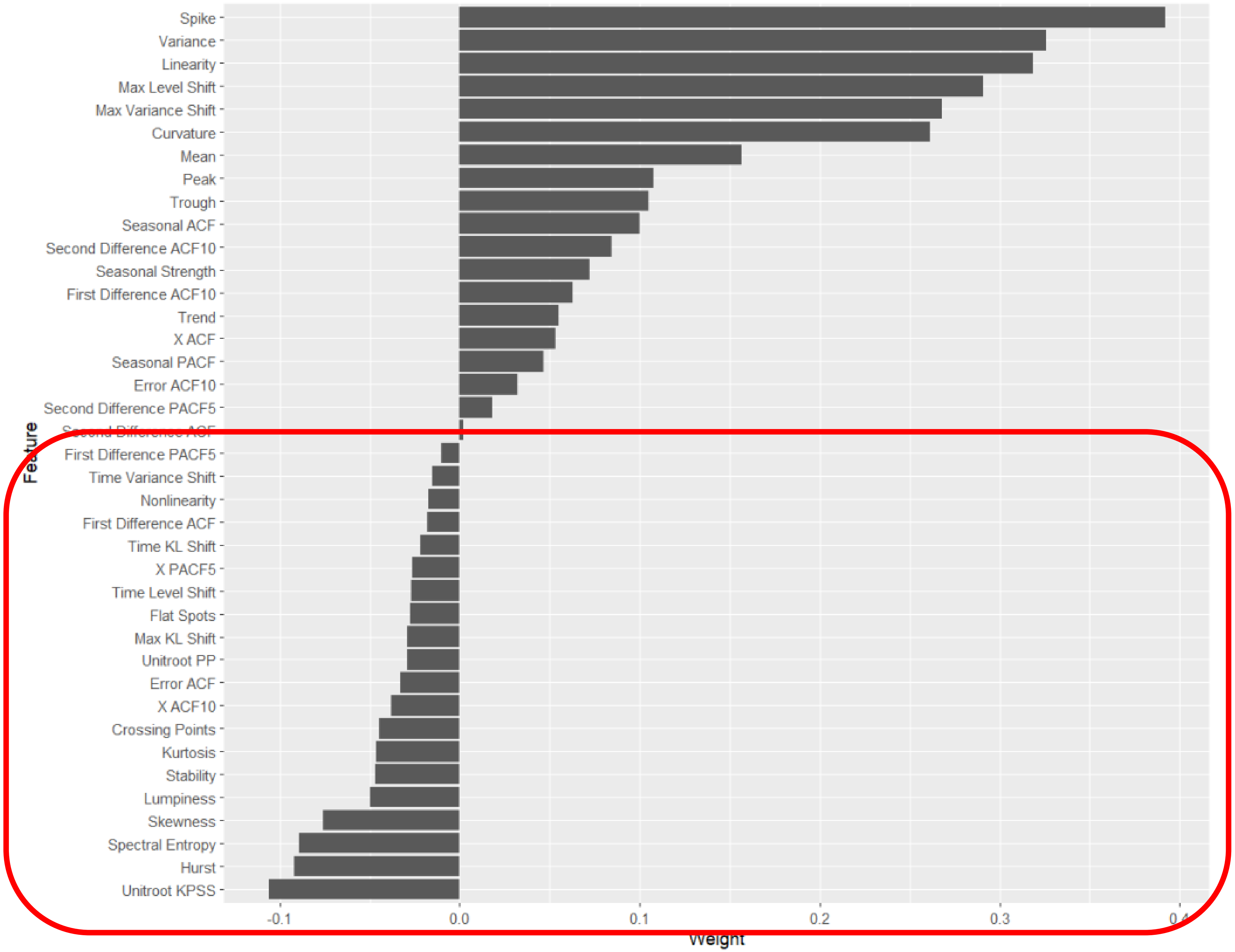
Results for all time series

Improvement weights ranked from high (+) to low (-)



Improvement weights (+)
Spike
Variance
Linearity
Max level shift
Max variance shift

Improvement weights ranked from high (+) to low (-)

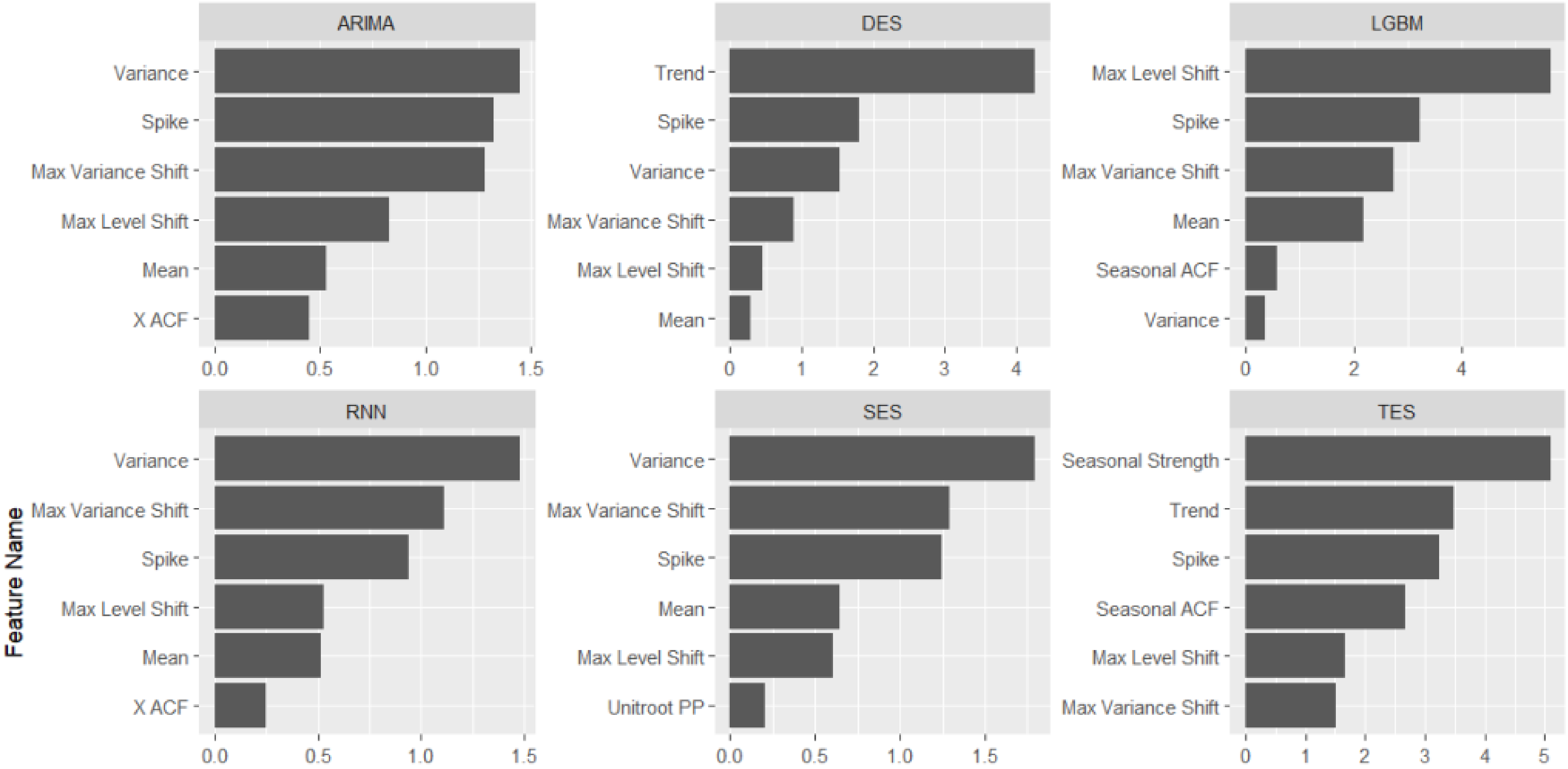


<u>Improvement weights (+)</u>
Spike
Variance
Linearity
Max level shift
Max variance shift

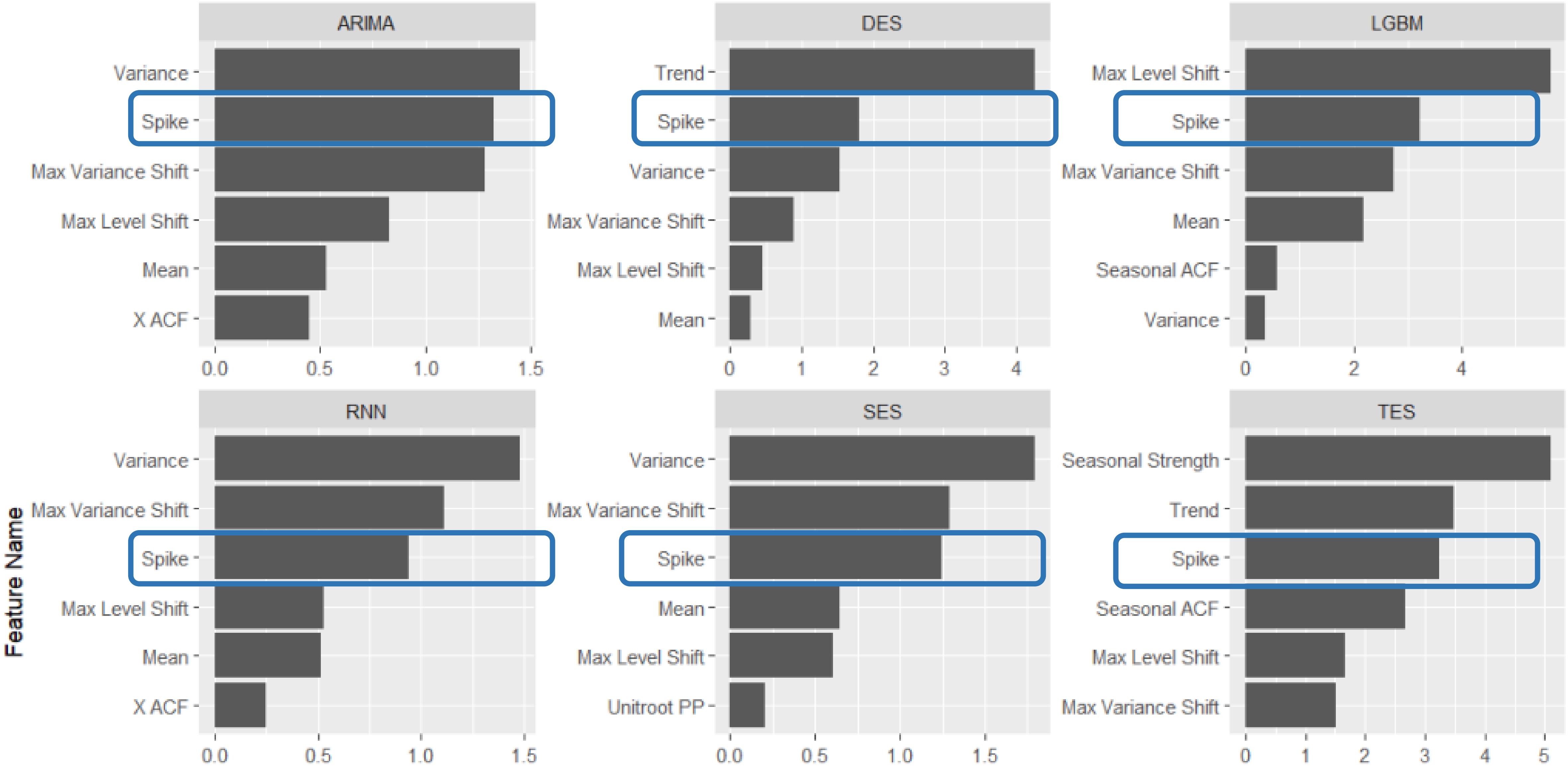
Improvement weights ranked from high (+) to low (-)

<u>Improvement weights (-)</u>
Unitroot KPSS
Hurst
Spectral entropy
Skewness
Lumpiness

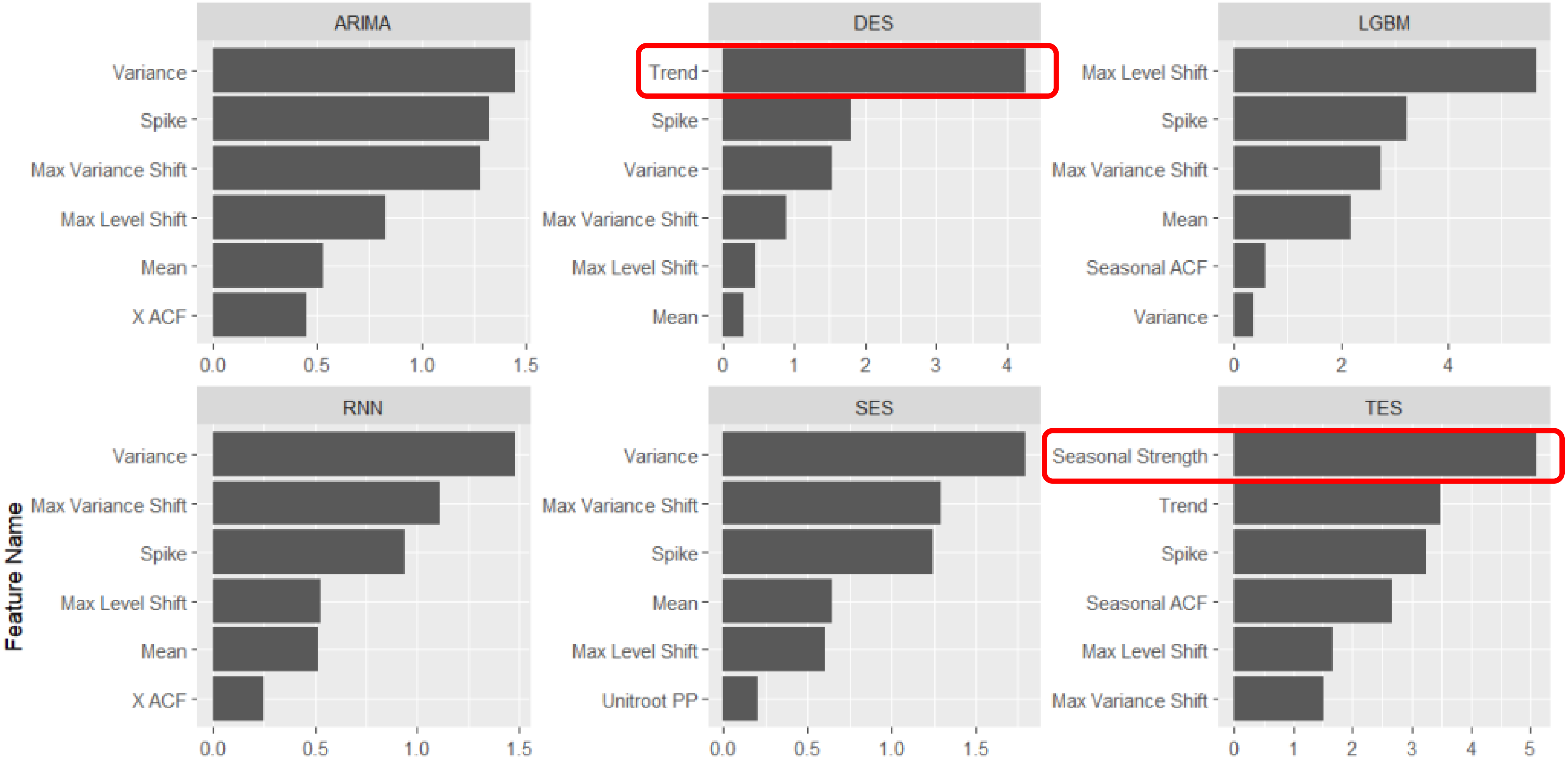
Top six “accuracy improving” features across forecasting models

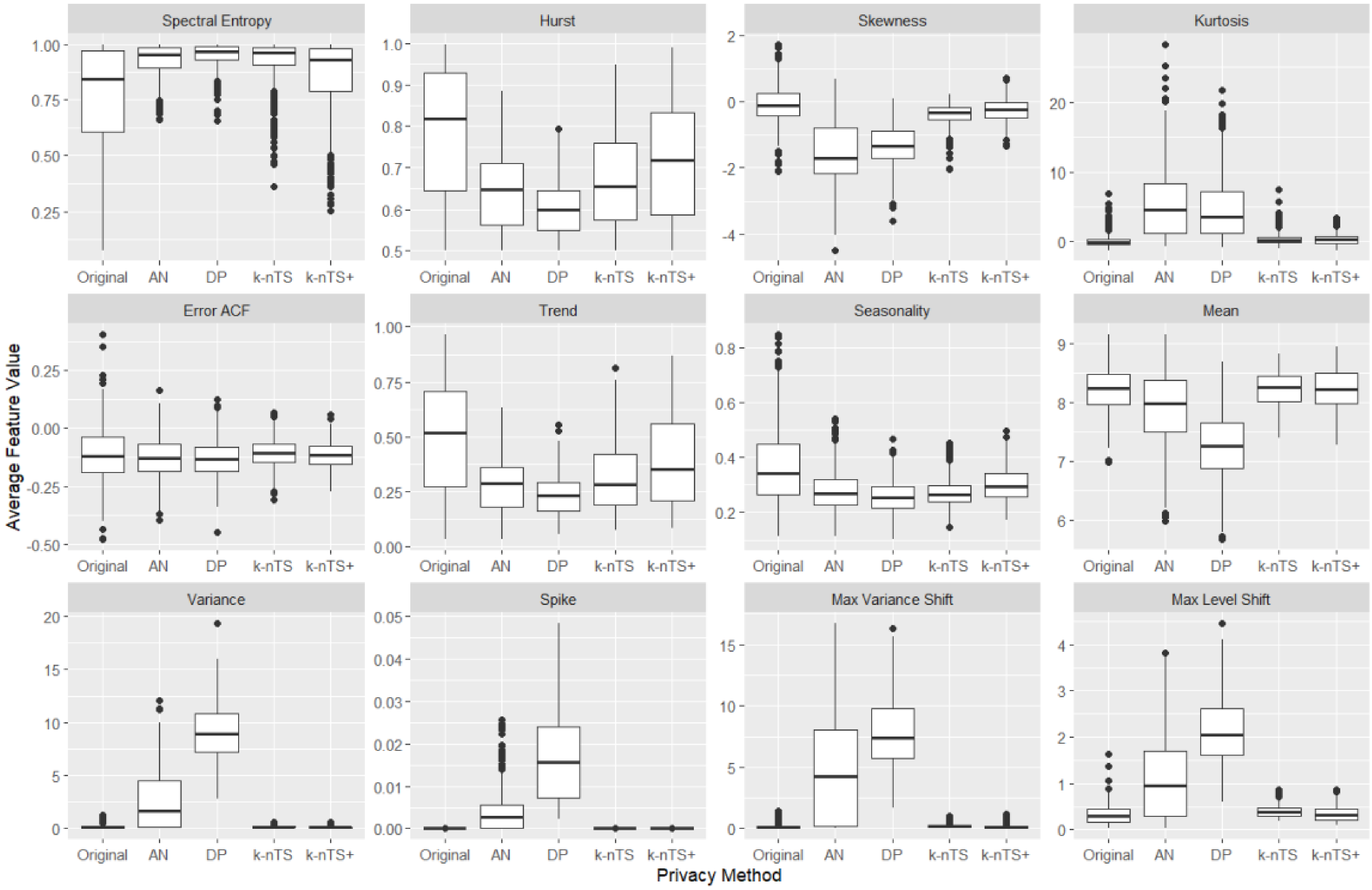


Top six “accuracy improving” features across forecasting models

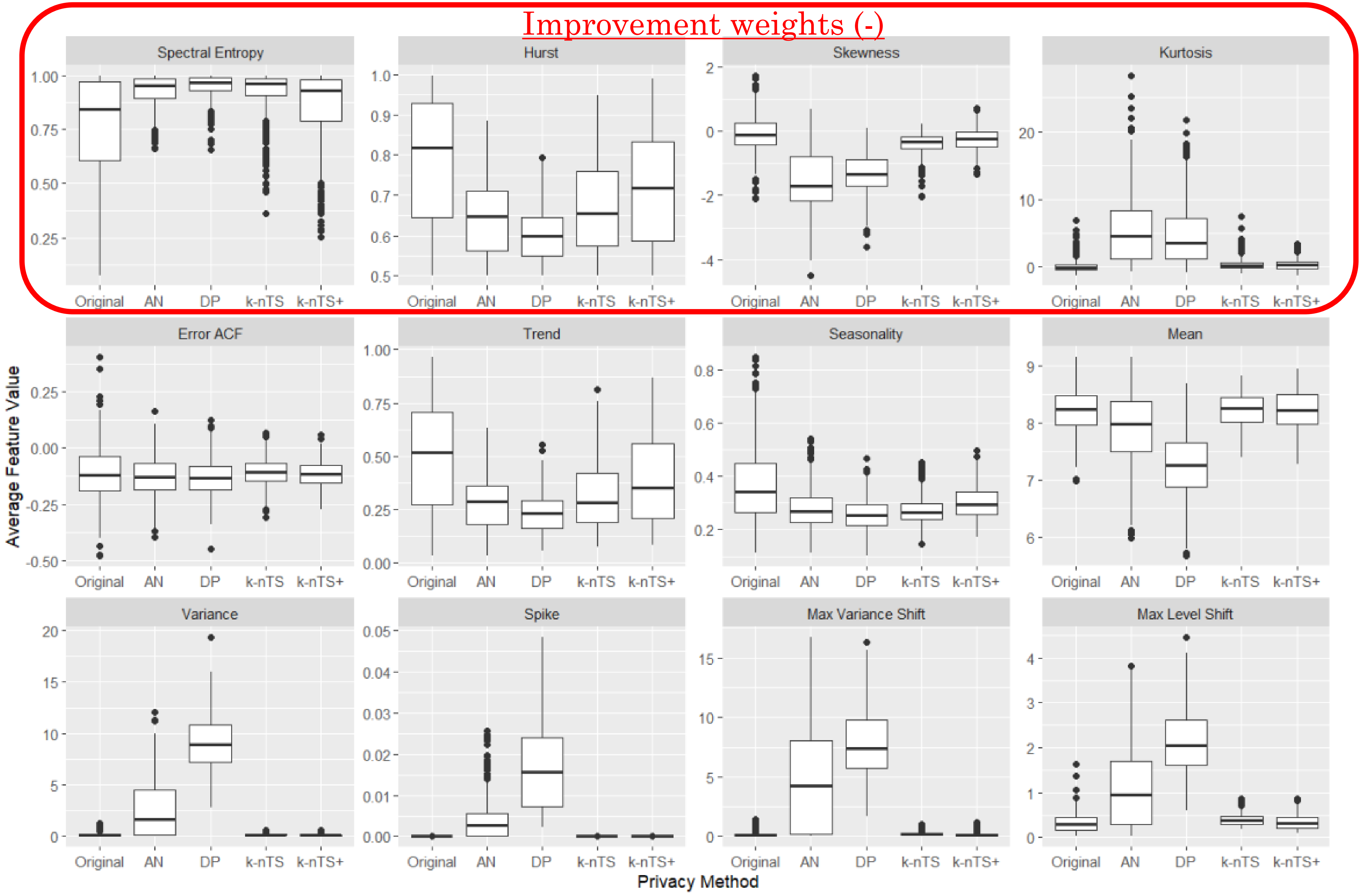


Some features show up in forecasting models with a mathematical relationship to them





Distributions of features across 474 time series

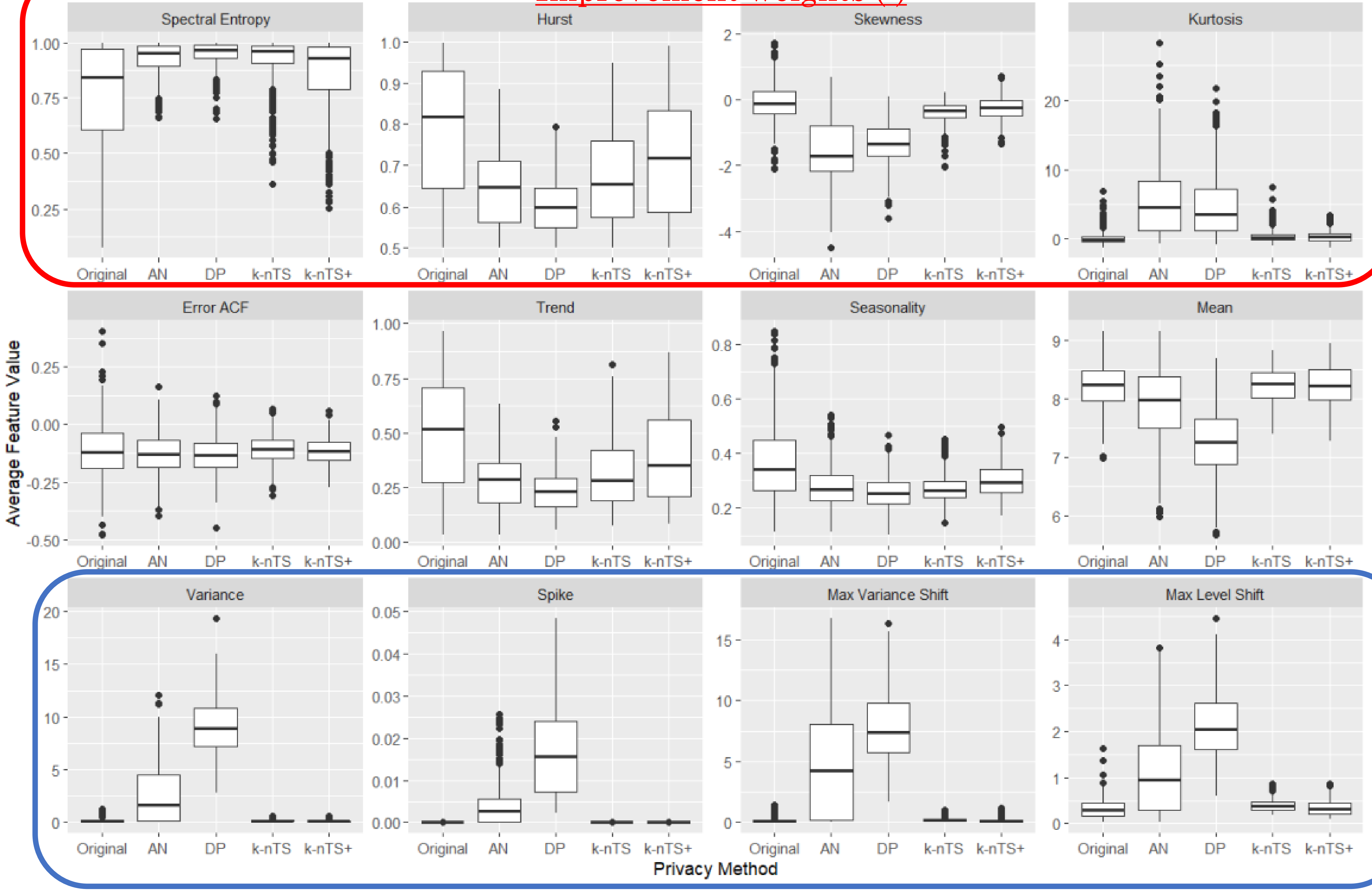


Distributions of features across 474 time series

Proposed method (k-nTS+) maintains feature distributions better than others

Proposed method without feedback loop (k-nTS) degrades slightly

Improvement weights (-)



Improvement weights (+)

Distributions of features across 474 time series

Proposed method (k-nTS+) maintains feature distributions better than others

Proposed method without feedback loop (k-nTS) degrades slightly

Differential privacy destroys feature distributions

k-nTS+: The Best Balance Between Forecast Accuracy and Privacy

Privacy Method	Parameter Value	Privacy (Identification Disclosure Risk)	Accuracy (MAE)
Original Data	**-**	100.0%	685.71 (0.0%)
***k*-nTS+**	**7**	3.5%	822.3 (+19.9%)
	3	3.3%	781.0 (+13.9%)
***k*-nTS (features selected from literature)**	**7**	2.1%	987.0 (+43.9%)
	3	2.1%	956.9 (+39.6%)
Differential Privacy	**1.0**	1.9%	3310.3 (+382.8%)
	4.6	13.6%	1401.0 (+104.3%)
	10	49.0%	899.4 (+31.2%)

Proposed method achieves a 13.9% forecast accuracy loss with a 3.3% average identification probability

Differential privacy at non-private levels of ε (4.6 or 10) have high identification probabilities with unusable forecast accuracy

Conclusion

- Features that are most predictive of forecast accuracy are not necessarily most useful for swapping time series values
- Produced protected time series with only 13.9% forecast accuracy loss and 3.3% re-identification risk
- Organizations can create protected time series data that preserves forecast accuracy and time series features

Conclusion

- Judgmental adjustments can improve forecast accuracy, but privacy adjustments degrade forecast accuracy.
- k-nTS+ can be adapted to replace specific sensitive or missing values, or preserve features for other use cases (*e.g.*, classification, price elasticity estimation, what else?)
- Efficient machine learning-based feature selection is applicable to any time series feature selection problem (continuous and binary targets acceptable)

Limitations

- Only considered one privacy attack: identification disclosure
  - What about attribute disclosure?
    - Are time series features sensitive?
  - Differential privacy + time series features?

- Hierarchical time series
  - Key application: product purchases have been shown to have privacy issues (Li, Schneider, Gupta, Yu 2022)
  - Are aggregate time series as sensitive as granular series?
  - Can we add constraints on swapping granular series values to maintain aggregate values?

Thank you!

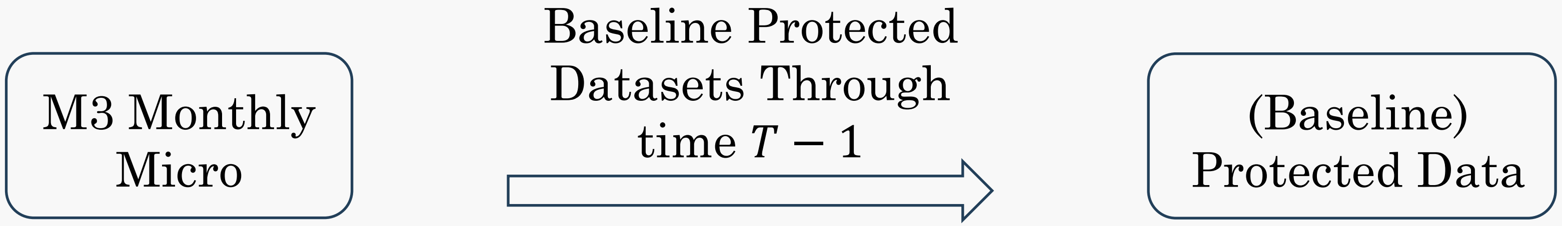
Contact: mjs624@drexel.edu

University of Moratuwa
TRANSPORT MANAGEMENT AND LOGISTICS ENGINEERING (TMLE)

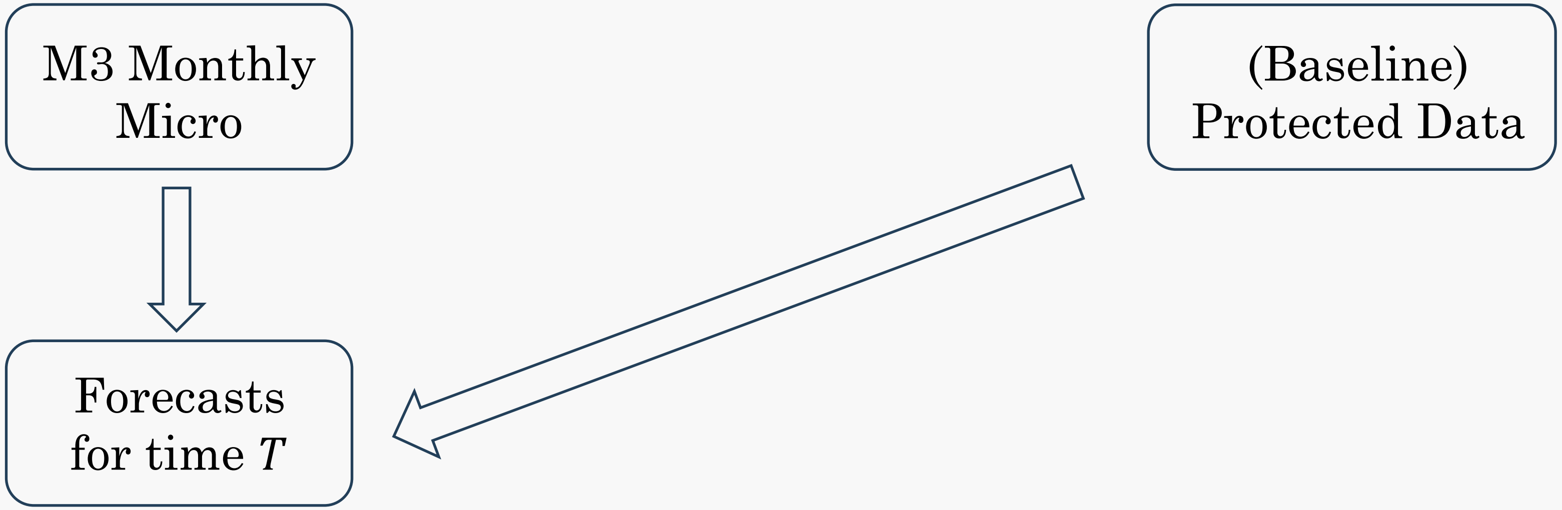
DREXEL UNIVERSITY
LeBow
College of Business

Appendix

Generate protected data with benchmarks



Produce forecasts for simple and complex models

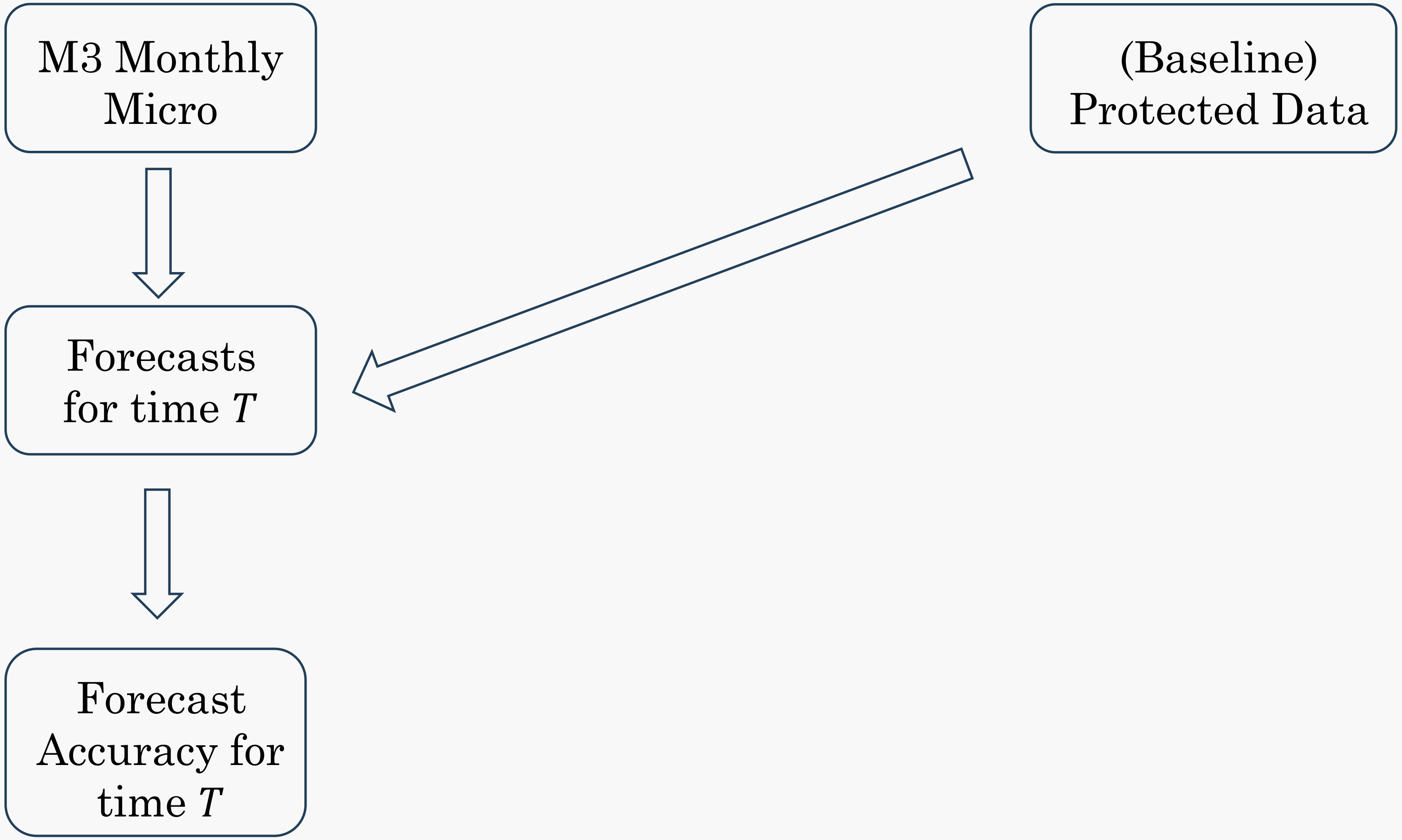


Models

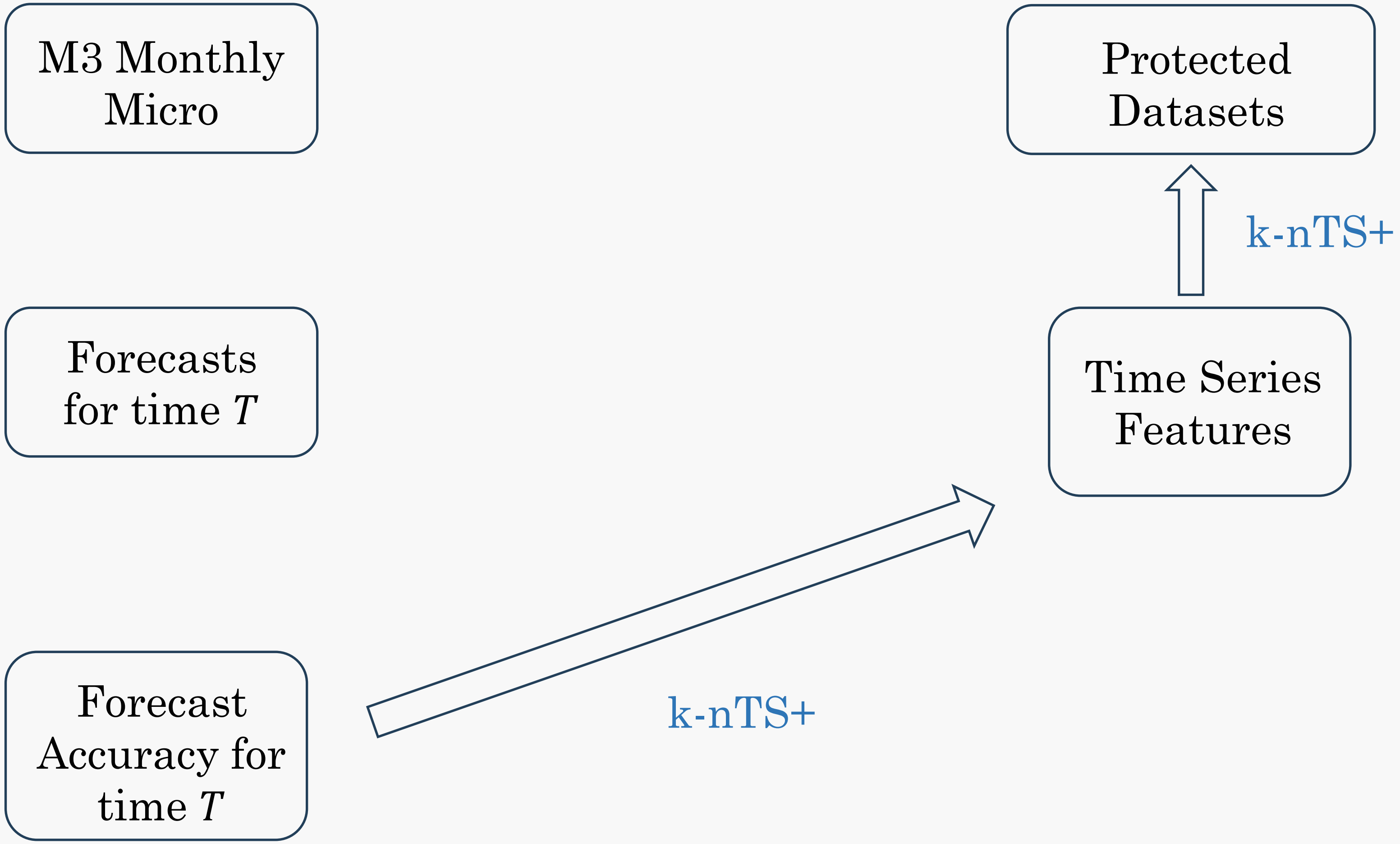
- SES, DES, TES
- Auto-ARIMA
- VAR
- LGBM
- RNN (LSTM)

Increasing complexity

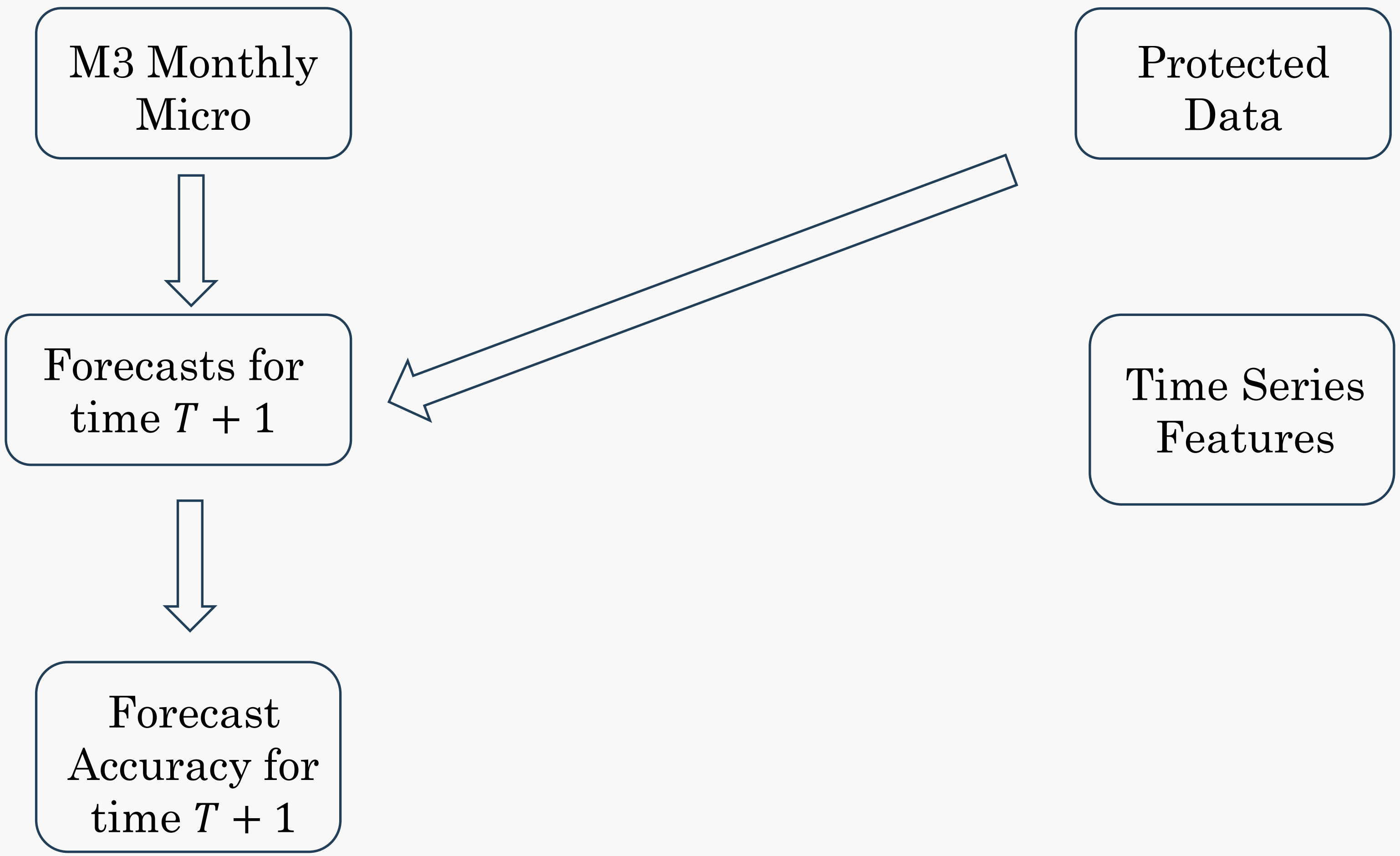
Measure accuracy to see variation in time series features



Machine learning feedback loop to inform data protection for k-nTS+



Forecast for T+1 and repeat process…



Adjusting forecasts can improve forecast accuracy…

Judgmental Forecast adjustments (Petropoulos et al., 2022)

- Directly change a *system* or *model* forecast (does not affect data)
- Incorporate intuition/experience, special events, confidential info
- Improve forecast accuracy by 5-10% on average (Davydenko & Fildes, 2013; Khosrowabadi et al., 2022)

But privacy adjustments are not the same as judgmental adjustments!

Large negative adjustments produce larger accuracy improvements (Fildes et al., 2009; Davydenko & Fildes, 2013)

AvgRelAE (and percentage of adjustments that improved accuracy) by magnitude and direction

		Direction		
		Positive	Negative	Total
Magnitude	Large	1.35 (40.5%)	1.47 (30.4%)	**1.41 (35.6%)**
	Medium	1.12 (44.4%)	1.17 (41.9%)	**1.14 (43.2%)**
	Small	0.99 (49.1%)	1.06 (46.8%)	**1.03 (47.9%)**
	Total	**1.14 (44.6%)**	**1.21 (40.2%)**	**1.17 (42.5%)**

On average, forecast accuracy worsened for nearly every combination of magnitude, direction, and coefficient of variation

Privacy adjustments had better forecast accuracy when the adjustments were small or positive, or when the coefficient of variation of the original series was large.

Features affect adjustments and forecast accuracy

- 73% of M3 monthly micro time series have negative slopes!
- Positive adjustments may dampen forecasts, and negative adjustments may overestimate the impact of trend (Hyndman & Athanasopoulos, 2021)

AvgRelAE (and percentage of adjustments that improved accuracy) by Slope and Direction.

		Direction		
		Positive	Negative	**Total**
Slope	Positive	1.13 (42.9%)	1.14 (41.6%)	**1.14 (42.2%)**
	Negative	1.14 (45.0%)	1.24 (39.6%)	**1.18 (42.6%)**
	Total	**1.14 (44.6%)**	**1.21 (40.2%)**	**1.17 (42.5%)**

Baseline Data Protection Methods Add Random Noise to Time Series Values

Differential Privacy:

For time series $x_j = (A_{j,1}, \dots, A_{j,T})$:

$$P_{j,t} = A_{j,t} + r$$

$$r \sim Laplace\left(0, \frac{\Delta f_1}{\epsilon}\right)$$

$$\Delta f_1 = max \left\| x_i - x_j \right\|_1$$

$\Uparrow$ ϵ, Privacy $\Downarrow$

$\epsilon \leq 1$ (Dwork, 2011)

Baseline Data Protection Methods Add Random Noise to Time Series Values

Additive Noise:

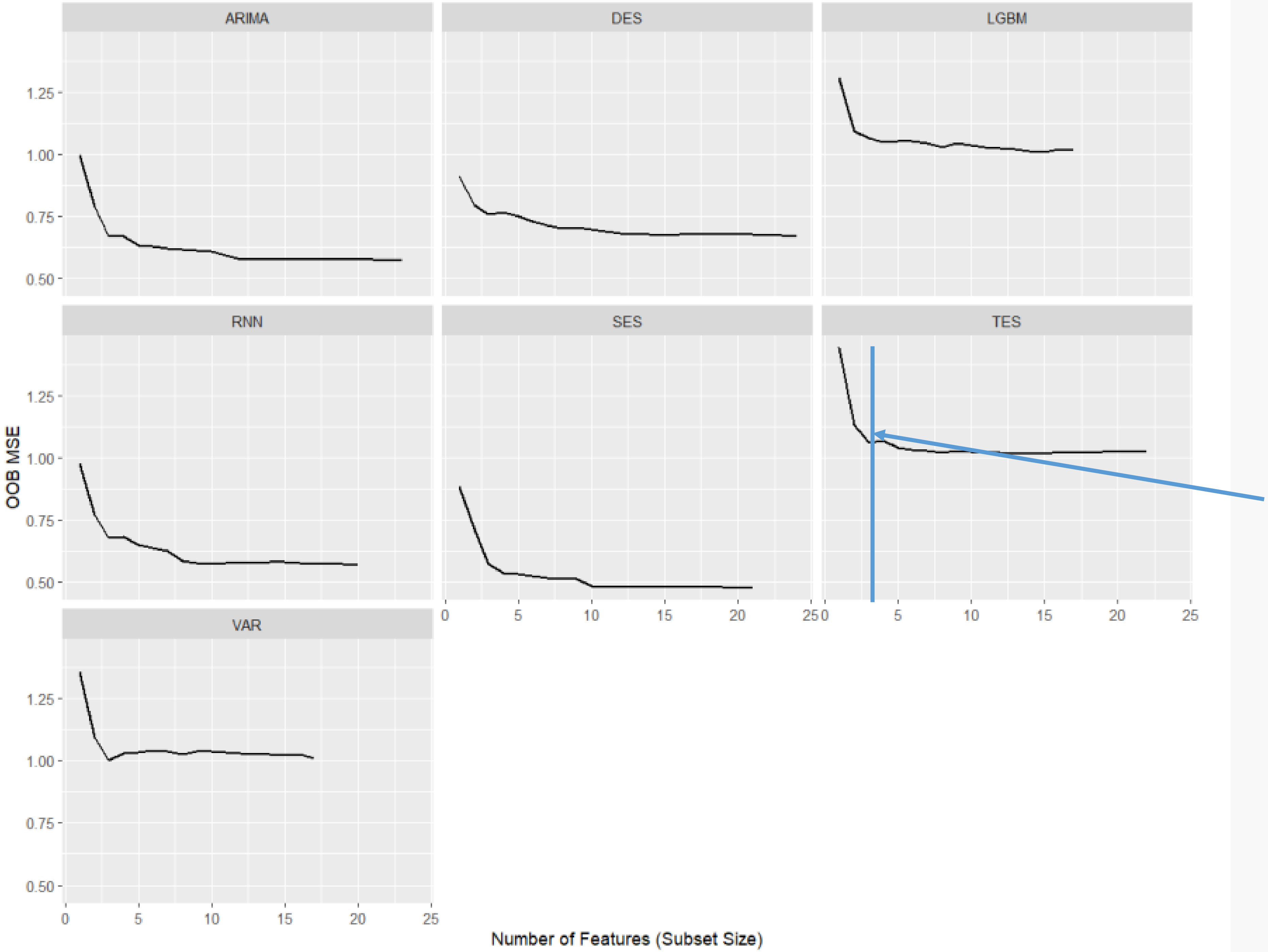
For time series $x_j = \left(A_{j,1}, \ldots, A_{j,T}\right)$:

$$P_{j,t} = A_{j,t} + r$$

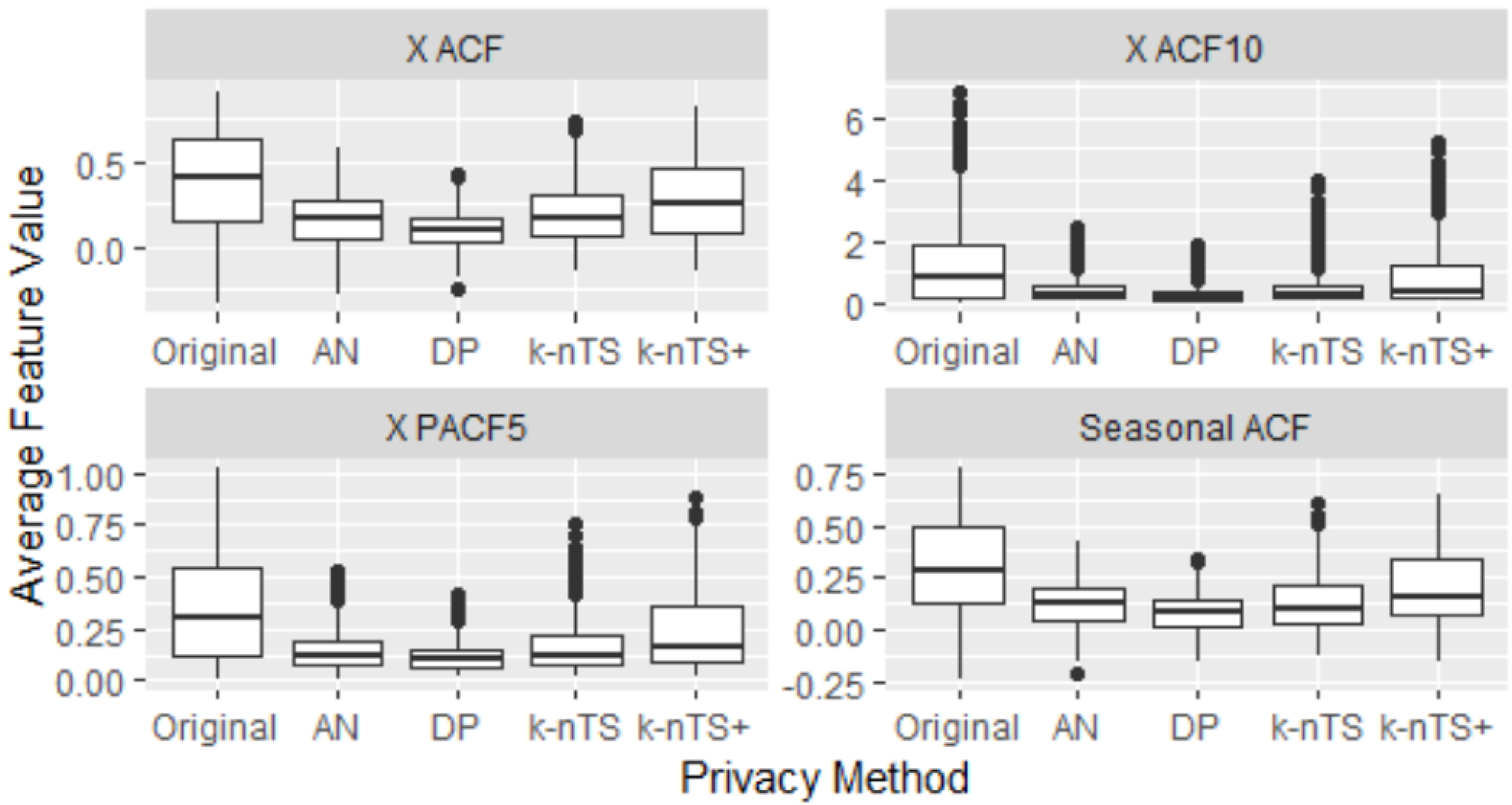
$$r \sim Normal(0, \sigma^2)$$

$$\sigma^2 = s\sigma_{x_j}^2$$

$\uparrow$ s, Privacy $\uparrow$



Only a few features are needed to accurately predict forecast accuracy!



Swapped series maintain some autocorrelation… Which *may* limit increases in spectral entropy.

	MAE Ranks		Standard Deviation of Forecast Error Ranks	
Model	Original	Protected	Original	Protected
TES	1 (637.90)	1 (731.30)	2 (859.30)	4 (920.57)
Auto-ARIMA	2 (646.07)	4 (764.83)	1 (834.78)	1 (897.67)
RNN	3 (665.38)	5 (783.15)	5 (883.86)	5 (966.35)
DES	4 (680.54)	2 (743.68)	3 (866.35)	2 (901.22)
SES	5 (686.71)	3 (752.08)	4 (867.13)	3 (914.20)
LGBM	6 (709.48)	6 (809.00)	7 (919.67)	6 (982.35)
VAR	7 (773.90)	7 (883.07)	6 (892.62)	7 (998.08)

k-nTS+ swapping tends to maintain the ranks of the best (and worst) performing models